International Journal of Engineering and Technology Innovation, vol. 14, no. 2, 2024, pp. 115-136

Design of Deep Learning Acoustic Sonar Receiver with Temporal/ Spatial Underwater Channel Feature Extraction Capability

Chih-Ta Yen[1,*], Un-Hung Chen[2]

[1]Department of Electrical Engineering, National Taiwan Ocean University, Keelung, Taiwan, ROC

[2]Department of Electrical Engineering, National Formosa University, Yunlin, Taiwan, ROC

Received 06 November 2023; received in revised form 25 December 2023; accepted 26 December 2023

DOI: https://doi.org/10.46604/ijeti.2023.13057

Abstract

In this study, deep learning network technology is employed to solve the problem of rapid changes in underwater channels. The modulation techniques employed are frequency-shift keying (FSK) and the BELLHOP module of MATLAB; they are used to create water with multipath, Doppler shifts, and additive Gaussian white noise such that underwater acoustic receiving signals simulating the actual ocean environment can be obtained. The southwest coastal area of Taiwan is simulated in the manuscript. The results reveal that optimizing the environment by using the virtual time reversal mirror (VTRM) technique can generally mitigate the bit error rates (BERs) of the deep learning network's model receiver and traditional demodulation receiver. Lastly, seven deep learning networks are deployed to demodulate the FSK signals, and these approaches are compared with traditional demodulation techniques to determine the deep learning network techniques that are most suitable for marine environments.

Keywords: underwater acoustic, deep learning, frequency-shift keying (FSK), feature extraction, virtual time reversal mirror (VTRM)

1 Introduction

In recent years, underwater wireless communication technology has been widely developed because of its numerous civil and military applications, such as ocean exploration, national defense, and marine commerce. Undeniably, underwater is considered one of the most complex communication environments due to numerous problems such as strong multipath effects, Doppler frequency shifts, and high attenuation. MATLAB's BELLHOP module is often used to establish an environment for simulating underwater acoustic (UWA) communication [1]. This module is an open-source beam/ray tracing model that can be used to evaluate acoustic pressure fields.

To effectively use BELLHOP to simulate an actual environment, Morozs et al. [2] employed the Virtual Timeseries Experiment module. This module helped BELLHOP establish a time-varying channel impulse response. They also utilized the world ocean simulation system of the UWA network to generate the environment for a specific geographical region. Most of the relevant modules and kits are constructed based on mathematical assumptions and approximations rather than using real underwater communication data. In the real ocean, the distribution of underwater sound velocities also varies with the season. Jiang et al. [3] used BELLHOP to construct sound velocity distribution maps for the four seasons (i.e., spring, summer, autumn, and winter) and three sea surface models for analysis. Based on the results, changes in the sea surface have little influence on the multipath effects, whereas the seasonal environment does affect the average transport loss at various depths.

* Corresponding author. E-mail address: chihtayen@gmail.com

English language proofreader: Chih-Wei Chang

Acoustic communication is an approach commonly used for underwater communication. To optimize underwater system design to meet the expected requirements, researchers are gradually turning to unconventional methods such as machine learning and deep learning, to analyze challenging underwater environments. Onasami et al. [4] used a deep neural network and long short-term memory (LSTM) to restore real underwater data and the signal received by a simulated UWA channel and to reduce the damage exerted by the channel environment on the transmission signal. Meanwhile, in LSTM, the mean absolute percentage error reached 3.14% [4]. Li et al. [5] reported that the measurement of underwater communication in a non-test field is challenging owing to the complex underwater channels, the variety of transmission signals, and the amount of measurement data. Thus, their team used the generative adversarial network to mitigate noise in signals and employed a convolutional neural network to distinguish noise from the true signal captured in an underwater communication environment.

Additionally, migration learning is deployed to construct a migration learning model of a generative adversarial network. The model was used to generate data from other environments to overcome the problem of insufficient data for the target waters [5]. To improve the performance of traditional systems concerning a low noise ratio and multipath effects, Liu et al. [6] proposed a deep-learning-based cyclic shift keying spread spectrum (CSK-SS) UWA communication system and demodulated the received signals by using neural network models constructed using LSTM and bidirectional LSTM (BiLSTM). Numerical simulations and real data unveiled that the deep-learning-based CSK-SS UWA communication system was more reliable than the traditional system, and its bit error rate (BER) was as low as 10^{-3} in an environment with a signal-to-noise ratio (SNR) of -8 dB [6].

The acoustic time reversal mirror was first proposed by Jackson and Dowling [7] and successfully tested at sea by Kuperman et al. [8]. The basic principle behind this technology is that a signal is sent by the transmitter and received by the receiver, and the receiver then retransmits the signal to the transmitter in a reverse manner, causing constructive interference at the transmitter. This process has also been employed for simulated data in which case the mirror is called a virtual time reversal mirror (VTRM). Despite the reducibility of the influence of multipath signals on a VTRM, the ability of the VTRM to focus is weak if the reversed signal does not include important paths. A VTRM also introduces additional noise and thus needs to be optimized [9]. The other technology of sonar image noise cancellation is proposed by James et al. [10]. The data-adaptive methods handle the mixed noise in sonar images, which consists of the additive Gaussian noise and the multiplicative speckle effect. A patch-based denoising method is applied in two phases to remove both types of noises.

The acoustic technology employed in underwater communication is an old and mature technology. Given its maturity, however, making decisive breakthroughs in the current situation is difficult, and the accuracy of transmissions can be improved only through the development of novel modulation and coding techniques. This study attempts to achieve the effect of signal demodulation by using deep learning networks and compares this demodulation method with traditional demodulation methods.

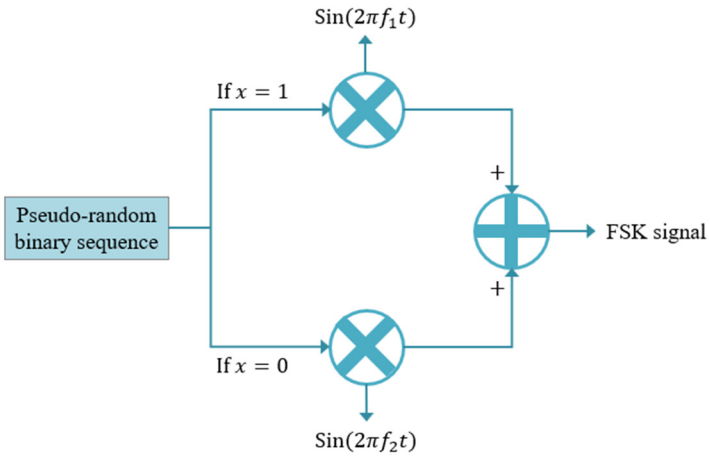
2. Methods of Frequency-Shift Keying (FSK) Modulation and Acoustic Signal Simulation

This research mainly uses FSK modulation as the modulation method for UWA transmission. To complete FSK modulation, both the establishment of traditional FSK modulation and the installation of a signal to an analog environment are indisputably the sine qua nones.

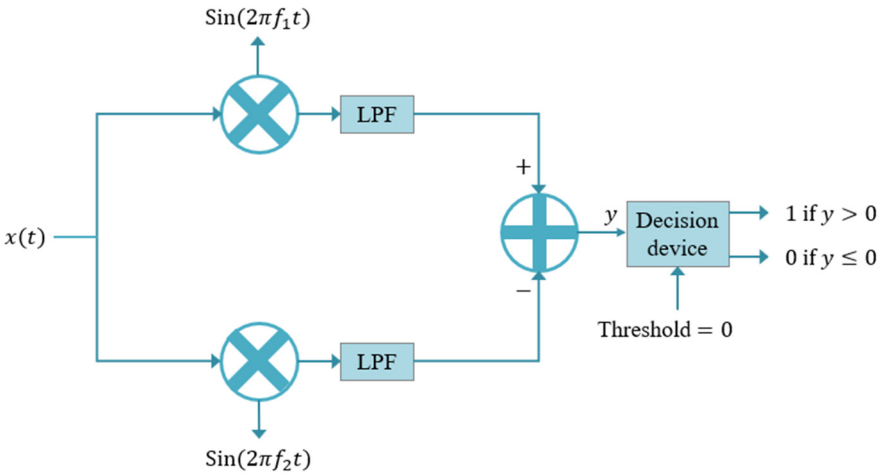
2.1. Establishment of FSK modulation

FSK is a modulation method that uses frequencies different from that of the carrier signal. The principle of FSK is to set a fixed frequency signal as the carrier wave when the transmission signal is 0. When the transmission signal is 1, another fixed-frequency signal is set as the carrier. Therefore, the carrier signals of different frequencies can be used to distinguish between the transmission signals 0 and 1. The FSK demodulation method considered in the paper uses the coherent demodulation approach. Coherent demodulation is essentially dual parallel amplitude modulation (AM) demodulation. When the frequency

of the input signal approaches the frequency of the input signal of the multiplier, the output is greater than zero, and the outputs of the upper path and lower path are compared to determine whether the transmitted signal is 1 or 0 [11]. Fig. 1 presents a schematic of FSK modulation. Fig. 1(a) depicts the FSK modulation signal architecture, and Fig. 1(b) delineates the FSK demodulation signal architecture.



(a) FSK modulation transmitter



(b) FSK coherent demodulation receiver

Fig. 1 Traditional modulation methods used in this study

The following equation is employed to determine the transmission signal after binary frequency shift keying (BFSK) modulation:

$$s(t) = \sin(2\pi f_c t) \tag{1}$$

where f_c is the carrier frequency and changes depending on whether the transmission signal renders 0 or 1.

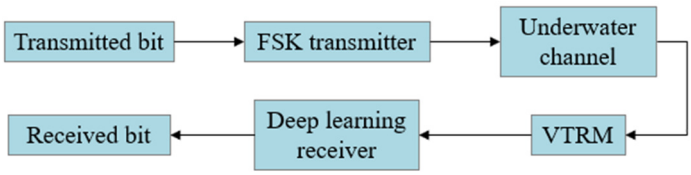


Fig. 2 The block diagram of the proposed deep learning underwater communication system

The block diagram of the proposed deep learning communication system is illustrated in Fig. 2. The transmitted bit first passed to the FSK transmitter and then through the underwater channel. After the transmitted signal is destroyed by the underwater channel, the VTRM technology is employed to suppress the multipath effect. Finally, the communication performance is enhanced by the proposed deep learning receiver method.

2.2. Establishment of the underwater simulation environment

Sound transmission underwater is mainly affected by multipath effects, environmental noise, and transmission attenuation. To create a realistic simulation environment that approximates the real environment, this study uses the BELLHOP beam/ray tracing module based on the Acoustic Toolbox [12] and the Arlpy suite of Python to create an underwater transmission environment and simulate the acoustic ray diagram in that environment. The simulated channel model is constructed using the method proposed by Liu et al. [6]. The underwater channel receiving model of UWA is shown in the following formula, in which $s(t)$ is the transmission signal and $r(t)$ is the received signal. The transmission signal is convolved with the impulse response of the simulated environment. Lastly, additive white Gaussian noise (AWGN) is added to obtain the received signal.

The signal received in the underwater channel environment is expressed as follows:

$$r(t) = s(t) \otimes h(t) + n(t) \tag{2}$$

where $r(t)$ is the received signal of FSK, $h(t)$ is the impulse response of the underwater channel, and $n(t)$ is the AWGN.

2.2.1. Establishment of the multipath effects

The formation of a multipath in the marine environment is mainly affected by two factors: the reflection of sound from the sea surface, the seabed, or any object; and the refraction of sound in the water, which is related to the speed of sound underwater. Sound reflections create reverberations, resulting in reflection phase and amplitude changes. Sound refractions are affected by the sound velocity distribution, temperature, salinity, and pressure. In this study, the impulse response under certain environmental conditions is obtained by simulating the topographic map and sound velocity distribution map of an actual environment to establish an acoustic ray diagram. The transmission signal formed by simulating the underwater channel is calculated from the impulse response.

As mentioned, the impulse response of the acoustic module is mainly affected by reflection and refraction in the channel environment. These characteristics determine the number of main transmission paths and the relative strength and delay of each path. Infinite multipaths generated notwithstanding, considerable energy is lost through multiple reflections, and these signals are discarded owing to the insufficiency of strength. As a result, the main multipath data are retained.

The UWA transmission environment is severely affected by multipaths, and each path has a different delay time and attenuation strength. The underwater impulse response can be expressed as follows:

$$h(t,\pi) = \sum_{p} \alpha_p(t) \delta\left[\tau - \tau_p(t)\right] \tag{3}$$

where α_p and τ_p are the attenuation and delay time of path p.

2.2.2. Doppler shift

Underwater transmission channels have large Doppler shifts and produce different values in different transmission bands; thus, Doppler shift effects are widely considered to be more difficult to manage than multipath effects. The underwater transmission environment has severe time variability, which also causes the channel attenuation to change chronologically [13]. In this study, the received signal is established under fixed movement by the method described by Gong et al. [14], and the offset value is added to the carrier of the transmitted signal through the following formula to obtain the received signals caused by the Doppler phenomenon under fixed displacement:

$$r(t) = \sum_{p} A_p \sin(2\pi(f_c + f_D)t) + n(t) \tag{4}$$

where A_p is the amplitude, f_D is the Doppler shift frequency, and f_c is the carrier wave frequency.

3. Methods of Optimizing Transmission Channels

Underwater communications suffer from multipath interference problems of varying intensity in different sea areas. Therefore, the multipath interference can be suppressed to a minimum before using deep learning technology, the trained deep learning models have the best generalization ability. The following section introduces exponential sine sweep and time reversal signal processing technologies to reduce the problem of multiple paths of underwater communication in different sea areas.

3.1. Exponential sine sweep (ESS)

The ESS is a technique for evaluating impulse response and an improvement on the sine frequency sweep proposed by Farina in 2000 [15]. Generally, the instantaneous frequency of a sine frequency sweep signal changes linearly manner, whereas that of an ESS signal changes in a nonlinear manner; the instantaneous frequency changes exponentially with time. The ESS signal is defined as follows:

$$x(t) = \sin\left(\frac{2\pi f_1 T}{\ln\left|f_2/f_1\right|} \times e^{t/T \ln\left|f_2/f_1\right|} - 1\right) \tag{5}$$

where f_1 and f_2 are the initial and end frequencies of the set frequency sweep, respectively, and T is the transmission time of the frequency sweep.

Fig. 3 presents a schematic of an exponential frequency sweep. The frequency set in this schematic is 1-20 Hz, the transmission time is 2 s, and the sampling rate is 1 MHz. The top and bottom graphs exhibit the instantaneous frequency and ESS signal, respectively, at different time points. The actual pulse response can be obtained when the exponential sweeping frequency is convolved with the time-reversed transmitted signal after obtaining the received signal through the transmission channel [16]. Fig. 4 shows the steps to obtaining the pulse response using the ESS frequency.

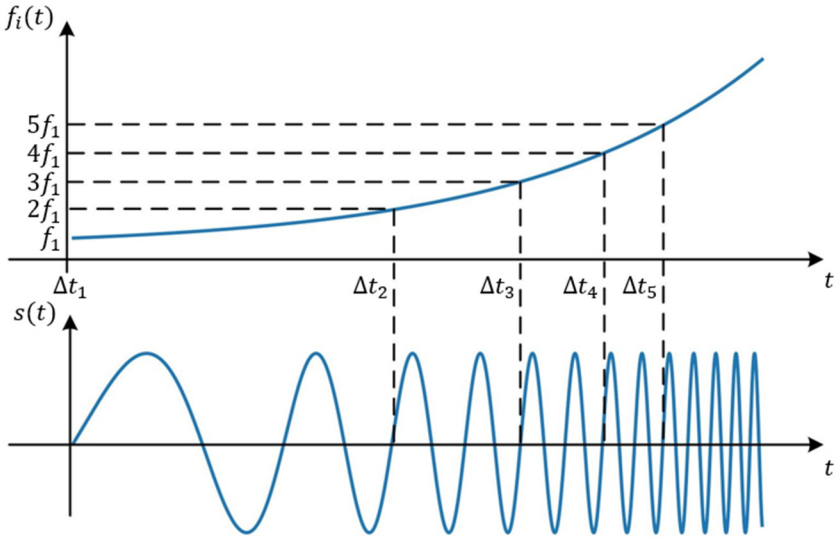


Fig. 3 Schematic of the exponential sine frequency sweep time, wave travel, and frequency

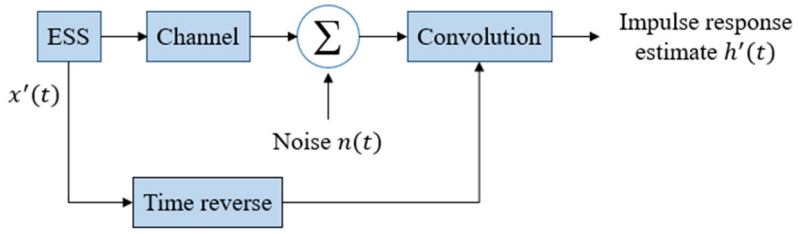


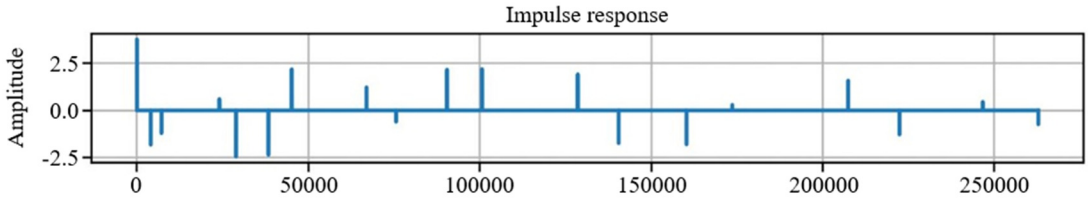
Fig. 4 The block diagram of impulse response evaluation using an exponential sine frequency sweep

3.2. VTRM

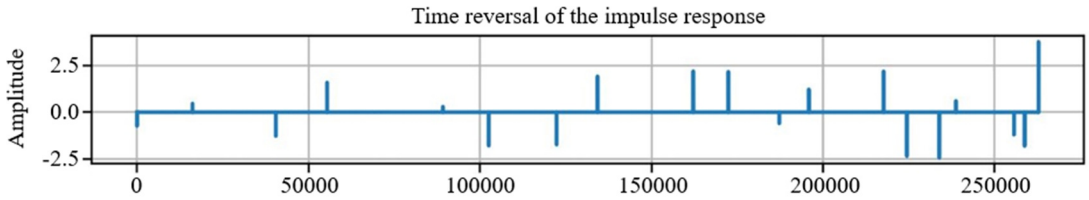
Time-reversal signal processing technology is a technology that focuses on the received waveform. The time reversal mirror concentrates multipaths with different time delays based on the principle of time reversal to optimize the pulse response channel. However, the use of the time reversal mirroring technique first requires the pulse response of the transmission channel. In this study, an ESS technique is used to obtain the evaluated impulse response and optimize the transmission channel through the formula below. Fig. 5 shows a schematic of the VTRM, from which the effect of the original pulse response and the pulse response on the channel after VTRM processing can be compared.

$$r(t) = s(t) \otimes h(t) + h'(-t) + n(t) \otimes h'(-t) \tag{6}$$

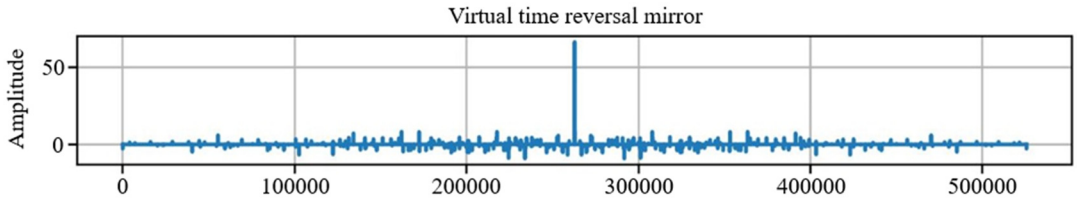
where $r(t)$ is a received signal after VTRM, $s(t)$ is the transmitted signal, $h(t)$ and $h'(-t)$ are the channel response and estimated channel response, respectively.



(a) Impulse response



(b) Inverted impulse response



(c) VTRM Signal

Fig. 5 Schematic of the VTRM processing

4. Introduction and Analysis of Deep Learning Networks

Deep learning technology has improved signal processing in underwater communication, enhancing resistance to noise and distortion during transmission. How to design a deep learning model for underwater communication is the most important issue in the study? The following section introduces the architecture and characteristics of these networks based on spatial features, temporal features, and spatiotemporal feature extraction.

4.1. LSTM

LSTM is an improvement of the recurrent neural network (RNN), which is mainly used for the classification, processing, and prediction of time series. The important data in time series data may fall within the front and back data, and LSTM has a strong ability to learn long-term dependent input data. Given its characteristics, LSTM surpasses the traditional RNN for time-related data. The basic network architecture of LSTM is illustrated in Fig. 6.

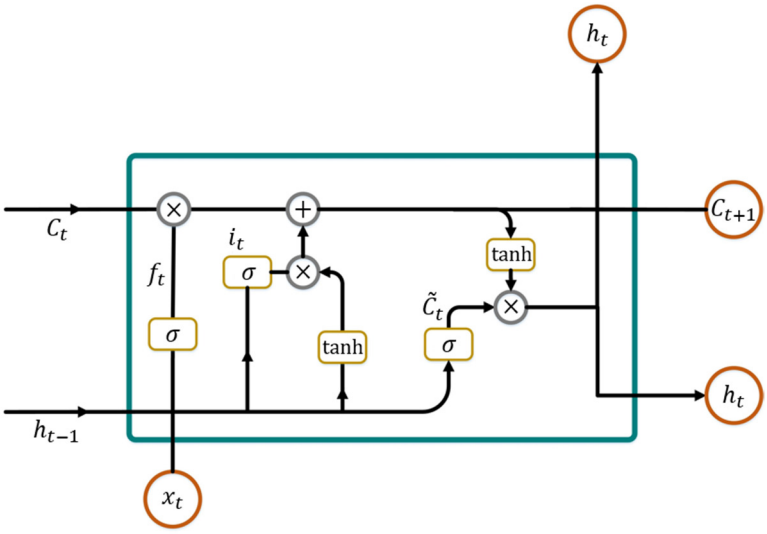


Fig. 6 Basic network architecture of LSTM

An LSTM network comprises a forget gate, memory unit, input gate, and output gate. The forget gate controls the degree to which the data in the long-term memory need to be forgotten, whereas the memory unit records the calculated value at this stage, adds it to the long-term memory, and passes it on for usage by the next unit. The input gate calculates whether the current input and the newly generated memory unit should be added to the long-term memory. The output gate controls whether the value calculated at this stage should be output; if the output is absent, the output of this layer will be regarded as zero. The calculation methods of the aforementioned LSTM steps can be expressed by:

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \tag{7}$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \tag{8}$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \tag{9}$$

$$\tilde{C}_t = \tanh(x_t U^g + h_{t-1} W^g) \tag{10}$$

$$C_t = \sigma(f_t \times C_{t-1} + i_t \times \tilde{C}_t) \tag{11}$$

$$h_t = \tanh(C_t) \times o_t \tag{12}$$

where i_t is the input gate; f_t is the forget gate; o_t he output gate; $\tilde{C}_t$ is the current memory unit; C_t is the memory unit that is passed to the next stage; h_t is the input value; x_t is the current input data; h_{t-1} is the output value of the previous stage; U^i and W^i are the input values when calculating the input gate and weight of the output value of the previous stage, respectively; U^f and W^f are the weight of the input value and the output value of the previous stage in the calculation of the forgetting gate, respectively; U^o and W^o are the input value in the calculation of the input gate and the weight of the output value of the previous stage, respectively; U^g and W^g are the input value of the current memory unit and the weight of the output value of the previous stage, respectively; and C_{t-1} is the value of the memory unit in the previous layer.

Fig. 7 shows the overall architecture of the LSTM network used in this study. The number of neurons in the two LSTM layers is set to 256 and 512, respectively, and batch normalization and random discard are used to improve the speed of network training and reduce the likelihood of overfitting. The parameter “dropout” is set to 0.5. The fully connected layer consists of three layers containing 512, 256, and 1 neuron, respectively. The final activation function is a sigmoid function, and the signals are classified using a threshold value, which is set to 0.5.

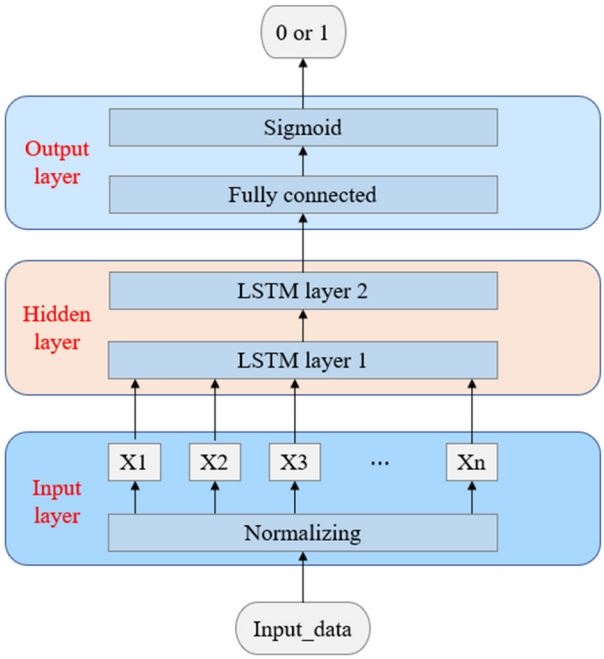


Fig. 7 LSTM network model

4.2. *Gated recurrent unit* (*GRU*)

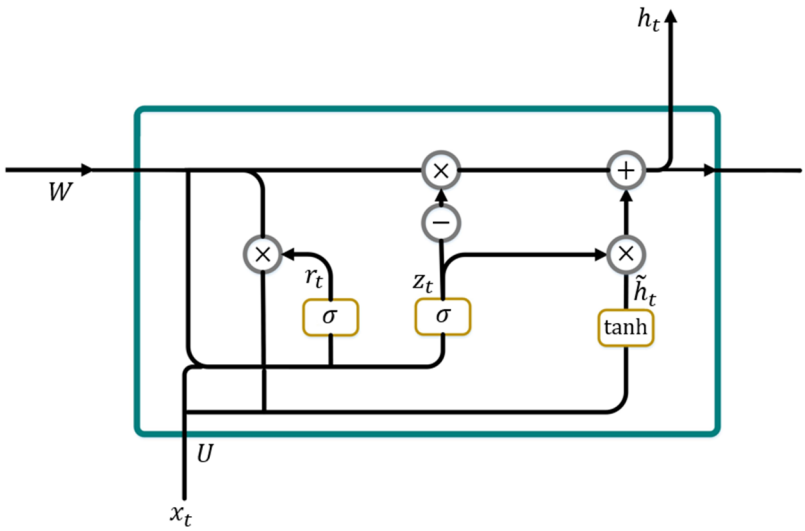


Fig. 8 Basic GRU model

The GRU is an improvement on the LSTM model. Although LSTM is suitable for processing time series data, a tremendous amount of computing time is required. In GRU, the forgetting gate and input gate of LSTM are replaced with a reset gate, and the memory unit and output gate are combined into an update gate to speed up the execution speed and reduce memory consumption. Furthermore, it also creates a nuance of the calculation in LSTM. The basic GRU model is illustrated in Fig. 8, and the calculation method is expressed in:

$$z_t = \sigma(x_t U^z + h_{t-1} W^z) \tag{13}$$

$$r_t = \sigma(x_t U^r + h_{t-1} W^r) \tag{14}$$

$$\tilde{h}_t = \tanh(x_t U^r + r_t h_{t-1} W^r) \tag{15}$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \tag{16}$$

where z_t is the update gate; r_t is the reset gate; $\tilde{h}_t$ is the current hidden value; h_t is the current output data; h_{t-1} is the output data of the previous layer; x_t is the current input value; U^z and W^z are the input value for the calculation of the update gate and the weight of the output value of the previous stage, respectively; and U^r and W^r are the input value for the calculation of the update gate and the weight of the output value of the previous stage, respectively.

Fig. 9 illustrates the overall network architecture of the GRU model used in this study. The number of neurons in the two GRU layers is 256 and 512, respectively. Batch normalization and random dropout are used to improve the speed of network training and reduce the likelihood of overfitting. The parameter "dropout" is set to 0.5, and the fully connected layer consists of a three-layer network, with the three layers containing 512, 256, and merely 1 neuron, respectively. The final activation function used is a sigmoid function, and the signal is sorted using a threshold value, which is set to 0.5.

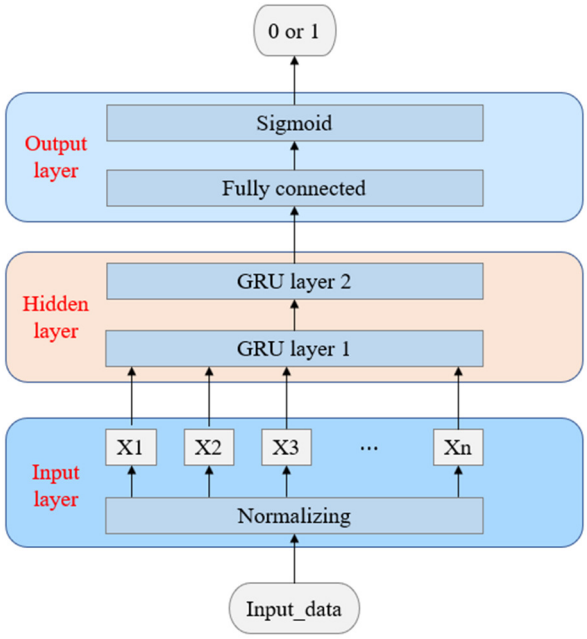


Fig. 9 GRU network model

4.3. BiLSTM and BiGRU

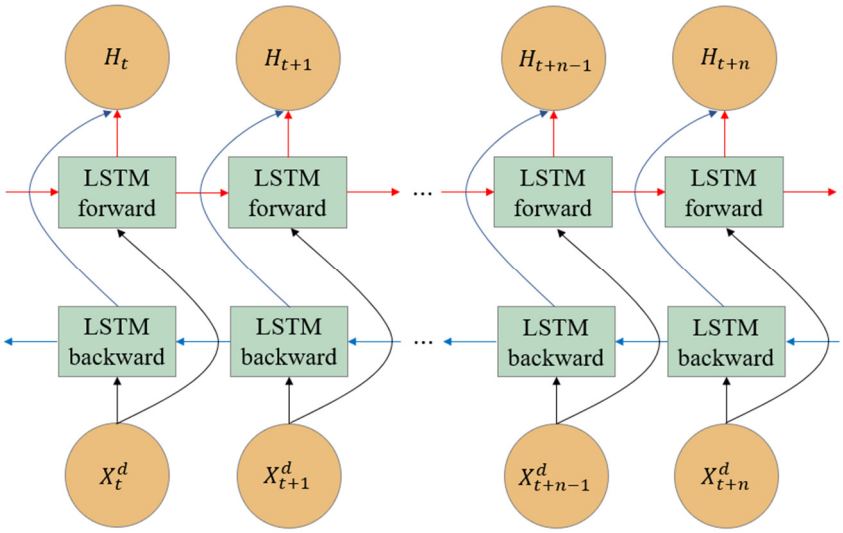


Fig. 10 Basic BiLSTM architecture

Although LSTM and GRU are often used to deal with time series problems, the limitation of only obtaining past data and information because the structure is palpable; they ignore future information and fail to learn all-time series information, which leads to relatively low prediction effectiveness. A bidirectional network consists of two unidirectional networks stacked one

above another, one of which is used for forward transmission and the other for backward transmission. The input data for the prediction are the current bit data, past time features, and future time features; the current time features are calculated and transmitted to the forward and backward channels of the two-way network. Finally, the adjusted hidden states of the forward pass and the backward pass are taken as the output. In this way, more features are obtained, thereby improving prediction performance.

Fig. 10 illustrates the basic structure of the BiLSTM, where X_t^d to X_{t+n}^d represent the input data of LSTM at different times, n is determined based on the batch size, and H_t and are the hidden states of the forward and backward pass, respectively. The architecture of BiGRU is the same as that of BiLSTM except that the LSTM is changed to GRU.

Fig. 11 shows the overall architecture of the BiLSTM used in this study. The number of neurons in the two BiLSTM layers is 256 and 512, respectively. Batch normalization and random dropout are used to improve the speed of network training and reduce the likelihood of overfitting. The parameter "dropout" is set to 0.5, and the fully connected layer consists of a two-layer network containing 128 and 1 sigmoid. The final activation function used is a sigmoid function, and the signal is sorted using a threshold value, which is set to 0.5.

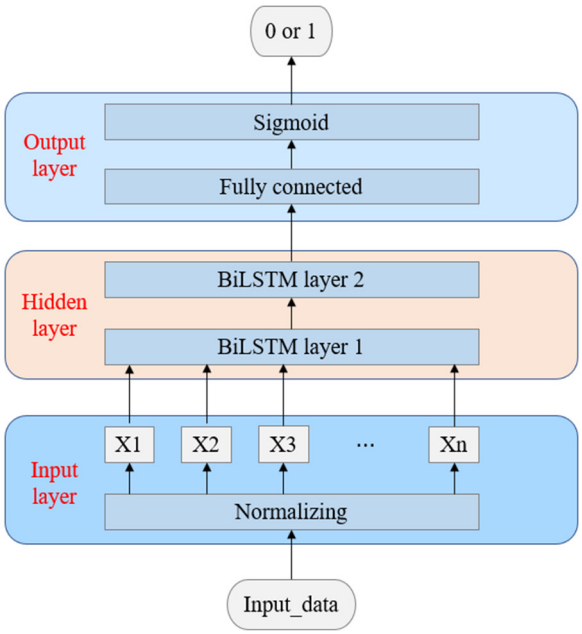


Fig. 11 BiLSTM network model

4.4. CNN-LSTM and CNN-GRU

CNN-LSTM, as the name implies, is a network model that combines CNN and LSTM. The working principle is to use a CNN to process input data and use LSTM as a classifier. CNN-LSTM thus employs a CNN to capture short-term features and LSTM to capture long-term features [17]. Fig. 12 shows the overall CNN-LSTM network architecture employed in this study.

The number of neurons in the two one-dimensional convolutional layers (Con1D layers) is 126 and 512, respectively, and the convolution kernel size is set to 3. Max pooling, batch normalization, and random dropout are used to enhance the speed of network training and reduce the likelihood of overfitting. The parameter "dropout" is set to 0.5, the number of neurons in the LSTM layers (LSTM layers) is 128 and 512, respectively, and batch normalization and random dropout are used in conjunction. The last fully connected layer comprises a network containing 1 neuron. A sigmoid activation function is deployed, and the signal is classified using a threshold, which is set to 0.5. The model architecture of CNN-GRU is the same as that of CNN-LSTM except that LSTM is changed to GRU.

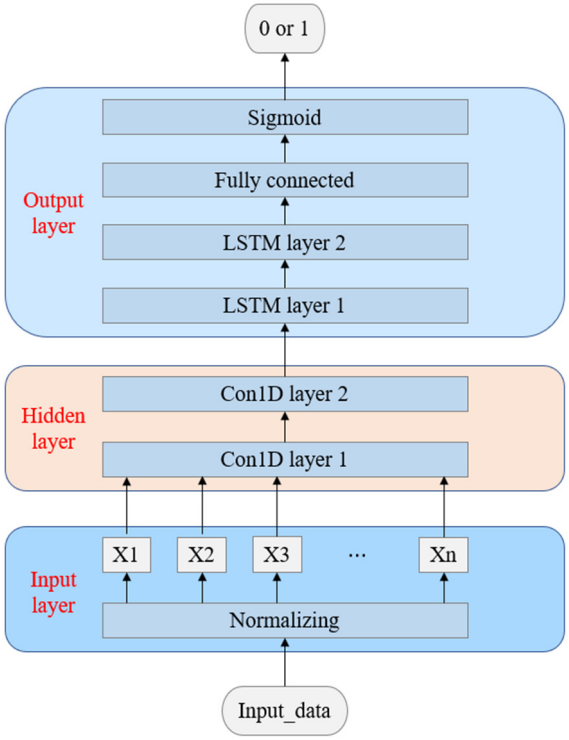


Fig. 12 CNN-LSTM network model

4.5. *Stacked BiLSTM-CNN*

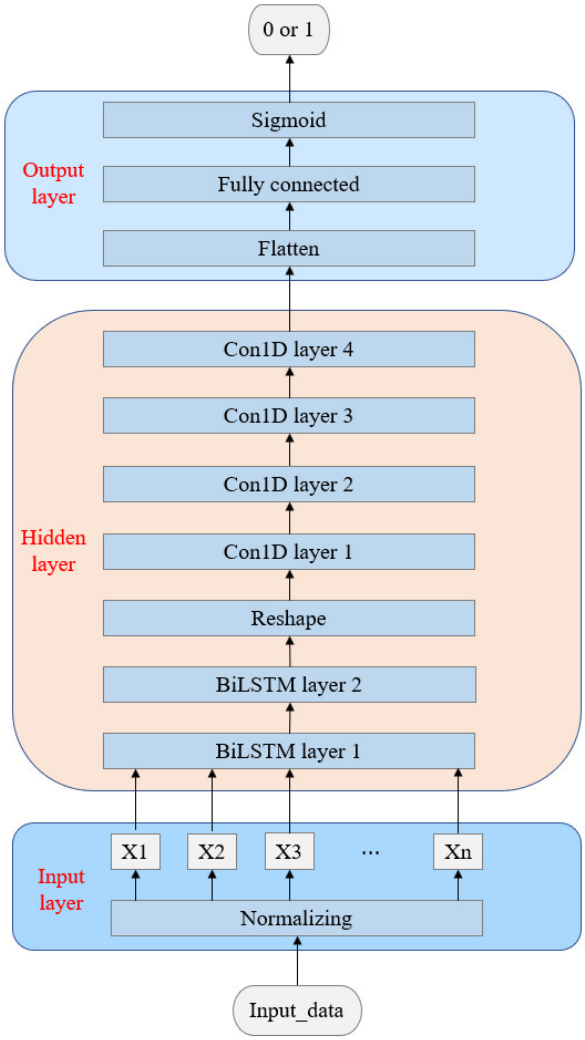


Fig. 13 Stacked BiLSTM-CNN network model

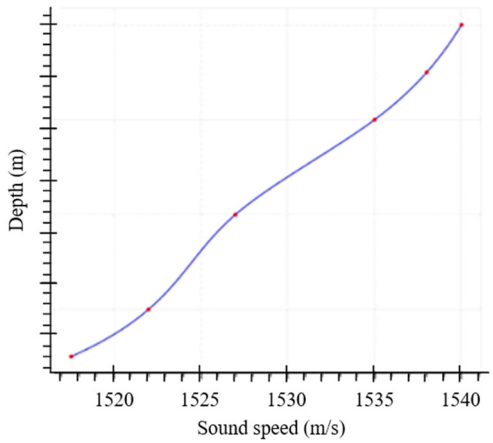
This network proposed by Utebayeva et al. [18] consists of two BiLSTM layers to handle the sequence prediction problem of the input data. Additionally, it employs a CNN layer to extract features and a fully connected layer as the output layer for prediction. Fig. 13 illustrates the overall network architecture of the Stacked BiLSTM-CNN used in this study. The number of neurons in the two BiLSTM layers is 256 and 512, respectively. Batch normalization and random dropout are used. The parameter "dropout" is set to 0.5, and the number of neurons in the four one-dimensional convolutional layers (Con1D layers) is 16, 32, 64, and 128, respectively. The size of the convolutional kernel is set to 3. Max pooling and batch normalization are used between the one-dimensional convolutional layers to improve the network's convergence speed. Owing to the limited input format of Con1D layers, a reshape layer is used to convert the features extracted by the BiLSTM layer into the format accepted by Con1D. Finally, a flat layer and a fully connected layer are used for classification. The fully connected layer comprises a network containing 1 neuron. The activation function used is a sigmoid function, and the signal is classified by a threshold, which is set to 0.5.

5. Experimental Methods and System Configurations

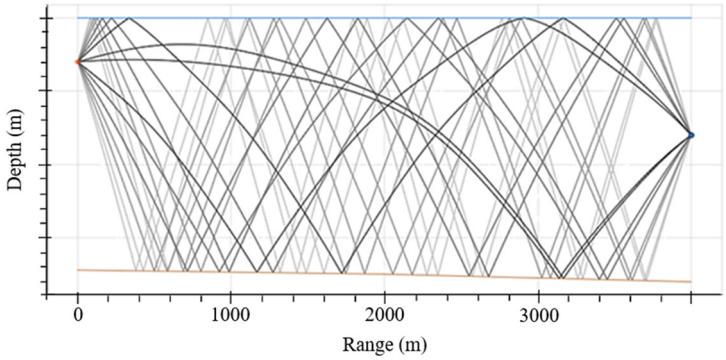
Underwater communication channels are a relatively difficult transmission medium due to the variability of link quality concerning location and applications in different sea areas. Before deploying underwater communications, the performance of underwater communication systems should be predicted based on the sound frequencies transmitted underwater. The following sections introduce the establishment of a simulation environment.

5.1. Establishment of the simulation environment

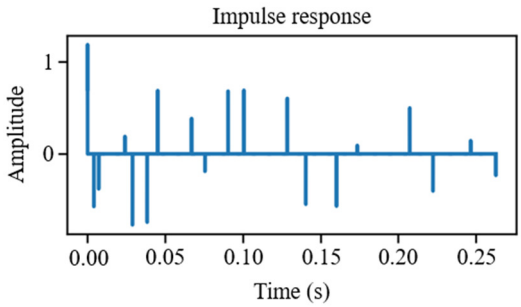
This study simulates sound transmission models in sea areas. Concerning the environment, the environmental parameters of the southwestern sea area of Taiwan stand at a depth of 170 m, which are obtained from the experimental data of literature [19-20] and are used to simulate the ocean model of actual sea areas and research the network training demodulator.



(a) Sound velocity distribution diagrams



(b) Acoustic ray diagram



(c) Impulse response

Fig. 14 Schematic of the ocean simulation environment

Fig. 14 presents the sound velocity map, acoustic ray map, and impulse response map of the sea environment. In the acoustic maps, the blue line is the sea surface, the yellow line is the seabed, the yellow point is the transmitting end, the blue point is the receiving end, and the black line is the sound line received by the receiving end.

From the acoustic ray diagram, the transmission route of the sound can be determined, and from the depth of the sound map, the intensity of the energy when the receiver receives the signal can be discovered. The simulations indicate great differences between the shallow sea and the outer sea, influenced by the topography and sea surface. As shown in the impulse response presented in Fig. 14(c), in addition to the main transmission path, other paths do not suffer much reflection-related attenuation; this makes the interference of multiple paths in the sea environment more severe than that in the less multipath environment.

Experimental parameter settings are closely related to underwater channel characteristics. To enable comparison with the reference, an underwater communication system using the same parameters is simulated in the reference. Other simulation parameters are shown in Table 1, where the source intensity, intermediate frequency, and acquisition frequency are set in this study by using FSK modulation.

Table 1 Simulation parameter configuration

Environment model	Environment
Transmitter depth (m)	30
Receiver depth (m)	80
Receiver distance (m)	4000
Launch azimuth (angle)	50
Sound source intensity (dB)	51-80
Intermediate frequency (Hz)	32 k
Sampling frequency (Hz)	1 M

5.2. Database establishment and preprocessing technology

When the underwater environment is established, the received signal after FSK modulation is obtainable. The received signal is subsequently provided for a deep learning network after suppressing multipath interference through VTRM processing technology. The following sections introduce how to establish the received data and VTRM preprocessing technology.

5.2.1. Establishment of received data

In this study, 100 sets of data are established in the form of random binary sequences, and each set contains a binary sequence of 1000 bits. Through FSK modulation and the simulation environment outlined in Section 5.1, the FSK signal received by the simulated sea area is obtained. Having considered that the intensity of noise in actual ocean measurement environments has a fixed value within a short period when no abnormal events occur, this study uses Python to establish a fixed intensity AWGN signal and adds it to the signals transmitted at various transmission strengths to create a dataset for different SNR environments. The range of the established data SNR is -14 to 14 dB. The calculation equations for SNR, as provided in Louza and Jesus [21], are discretized as follows:

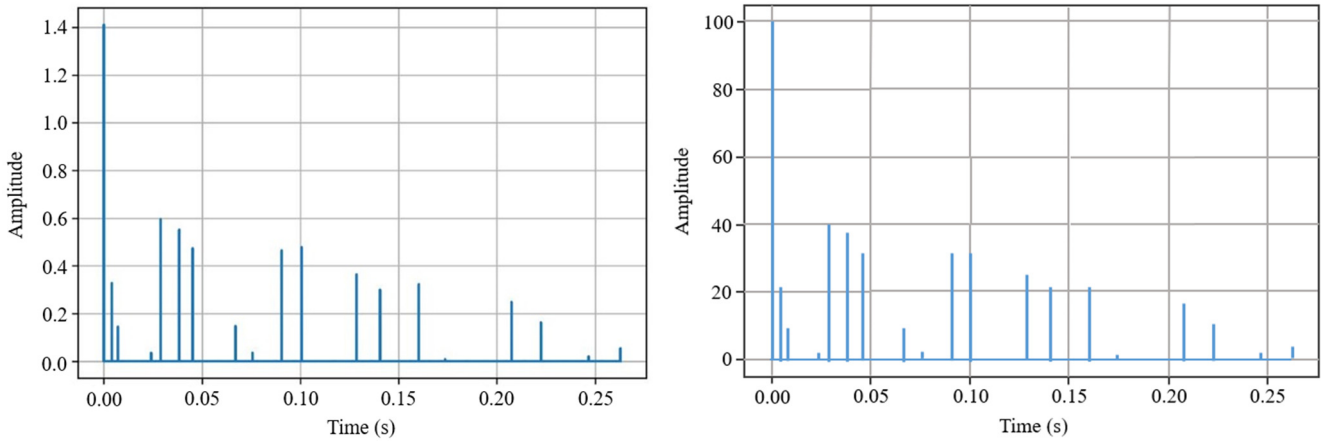
$$w(k) = S_r(k) - y(k) \tag{17}$$

$$SNR = \frac{\sum_{k=0}^{N} y(k)}{\sum_{k=0}^{N} w(k)} \tag{18}$$

where $S_r(k)$ is the received signal with noise, $y(k)$ is the received signal without noise, $w(k)$ is the noise, and N is the transmitted bit period.

5.2.2. Preprocessing of VTRM

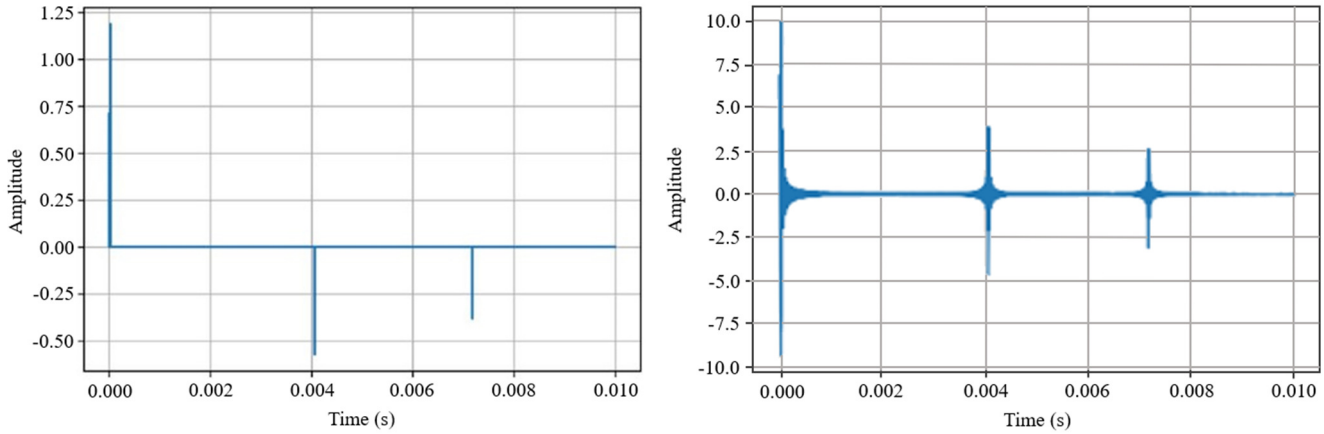
In addition to FSK, ESS can be applied to the transmission signal. Specifically, this method can be used to predict the impulse response in various environments, and the evaluated pulse response can be used to perform VTRM processing on the FSK received signal to optimize the multipath interference in the transmission channel. The evaluation of the ESS impulse response is conducted following the methodology outlined by Santoso et al. [16], which involves performing convolution operations on the received signal of the ESS transmitted through the channel and the time-reversed signal of the ESS transmission signal.



(a) Actual PDP (b) ESS-evaluated PDP

Fig. 15 Comparison of PDP plots

First, the ESS-evaluated impulse response and actual impulse response are converted into a power delay profile (PDP) to determine the presence of any difference in the delay and signal strength of the actual and evaluated impulse responses. Fig. 15 reveals that the ratio of the amplitude intensity of the power-delay curve of the evaluated impulse response is consistent with the time delay, which means that the evaluated impulse response has identified the impulse response. However, the expression method is still different. Comparing the amplitude of the evaluated pulse response with that of the actual pulse response, Fig. 16 evinces that the actual pulse response is presented in the form of a pulse, whereas the evaluated pulse response is presented in the form of a carrier wave. Therefore, the next step is to change the representation of the pulse response and extract the important pulse features from the carrier wave.



(a) Actual impulse response (b) ESS-evaluated impulse responses

Fig. 16 Comparison of the first 10,000 impulse responses

The main pulse signal must be preserved, and the redundant signal added due to the carrier waveform must be removed. Fig. 17 shows the pulse signal expressed by the evaluated pulse response in the time interval from 6.7 to 7.6 ms. This signal is a pulse signal expressed in the form of a carrier wave if it is converted into the impulse form of the actual impulse response. Only the strongest signal in the evaluated impulse response is retained, and it creates an impulse response expressed in the form of a pulse signal.

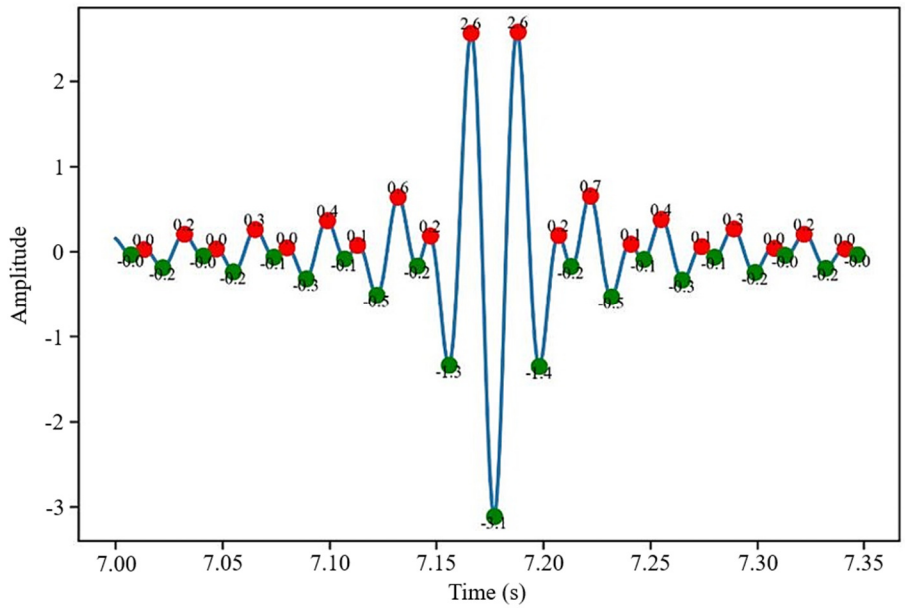


Fig. 17 Schematic of the evaluated pulse response between 6.7 and 7.6 ms

However, a problem subsequently emanates when identifying the pulse response by using the peak approach, which means that the strength of the pulse signal affects the strength of the additional carrier wave. When the signal strength is greater than anticipated, the additional signal will be stronger than the other pulse signals, which affects the accuracy of the pulse response. As illustrated in Fig. 18, this study performs envelope processing on the pulse signal and then takes the peak value to reduce the interference of an additional carrier wave.

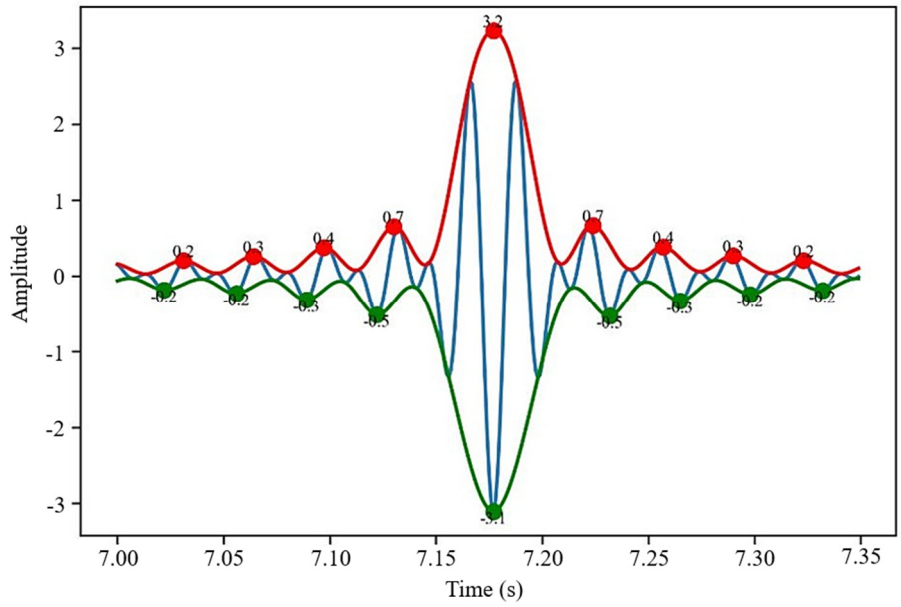
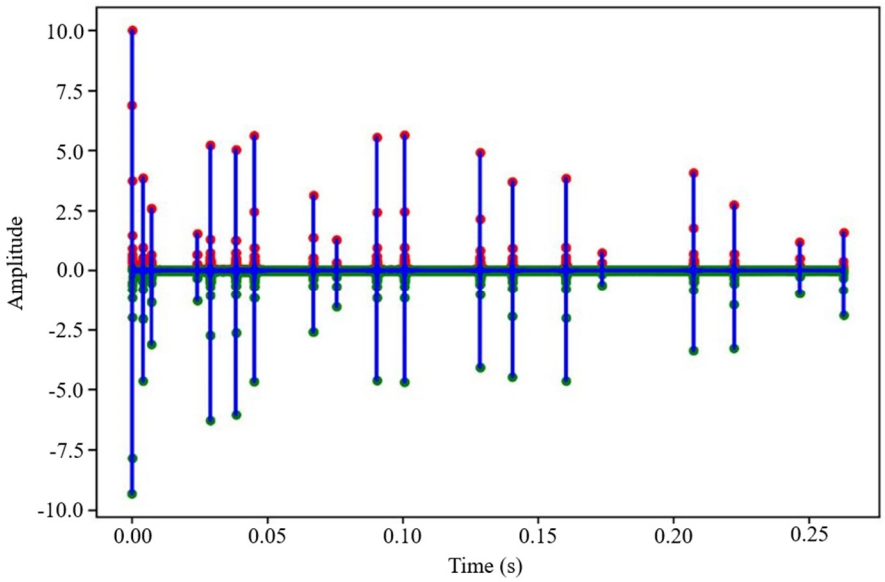
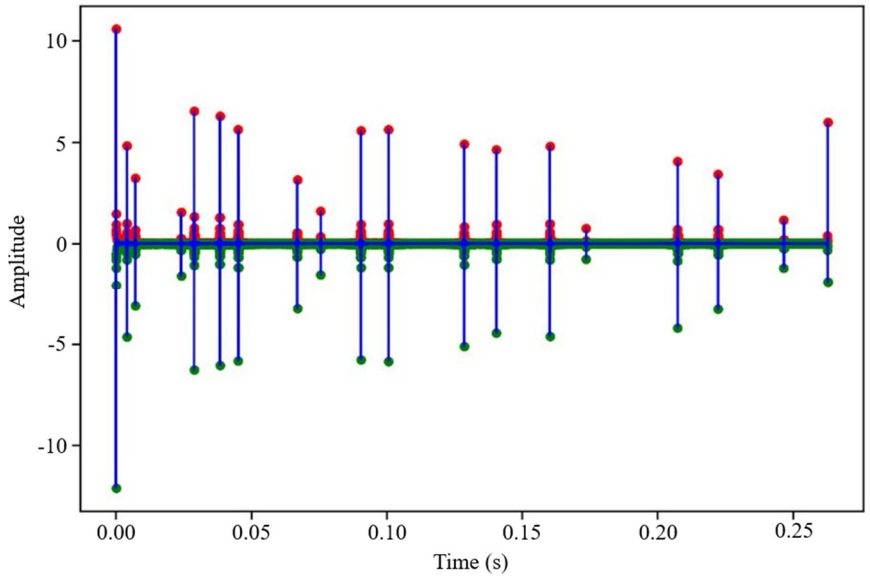


Fig. 18 Schematic of envelope processing

The comparison chart displayed in Fig. 19 indicates that envelope processing effectively reduces the influence of the additional carrier wave, but the peak value marked by the envelope line is inconsistent with the evaluated peak value; thus, the envelope line is used to confirm the position of the peak value. In this case, because of multipaths, the evaluated pulse signal at every signal generated is reduced to the main wave marked by the upper and lower envelopes and numerous additional carrier waves with little energy. This study establishes an impulse response in the form of an impulse in the following manner: the first 25 signals with higher intensities in the upper and lower envelopes are compared, the time in the time signals in which the upper and lower envelopes have similar intensities is recorded, and the positivity or negativity of the pulse response and the delay time are determined by the signal intensity of the pulse response at that time, with evaluation using the original ESS.



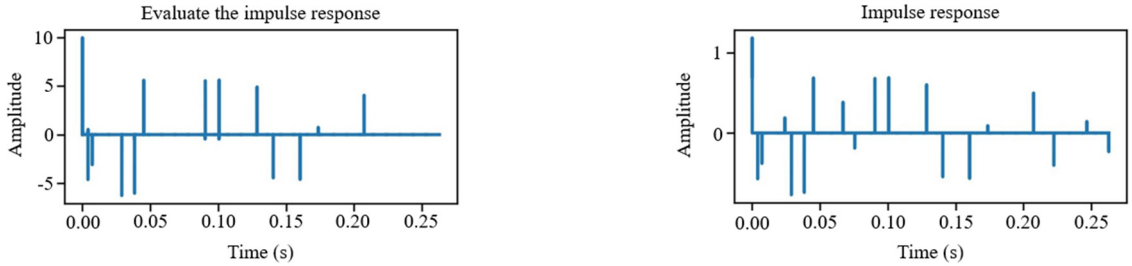
(a) Unprocessed



(b) Envelope processing

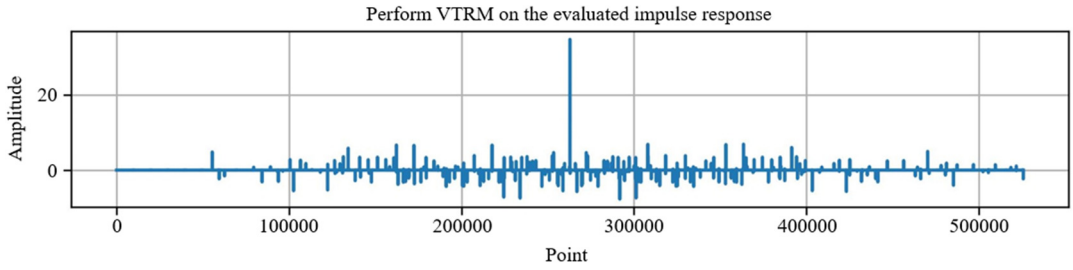
Fig. 19 Comparison of the peak values of the marker evaluation of pulse response

Fig. 20 shows VTRM comparison diagrams of the processed ESS-evaluated impulse response and the actual impulse response in the simulated sea area. The figure reveals that although the evaluated pulse response is still slightly different from the actual response, the evaluated pulse response after VTRM processing is almost identically presented to the actual pulse response after VTRM processing. Therefore, this study performs VTRM processing on the received signals by using the evaluated pulse response to restore the pulse signal. Additionally, Santoso et al. [16] noted that it is normal for an evaluated impulse response to differ slightly from the actual impulse response in terms of magnitude, where either can be greater than the other.

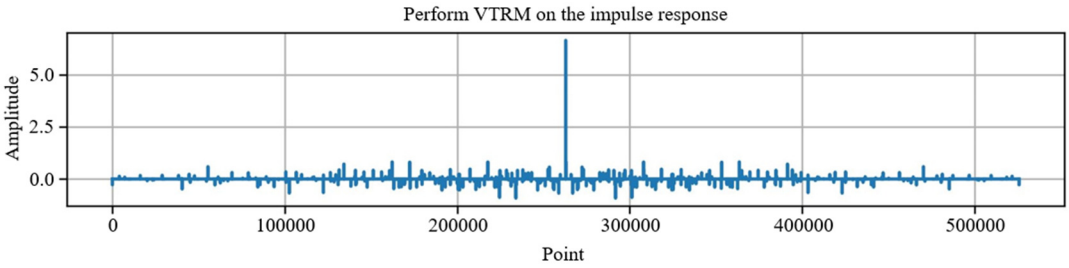


(a) Actual pulse response (b) Impulse response evaluated with ESS

Fig. 20 Comparison between the evaluated impulse response and the actual impulse response before and after VTRM processing



(c) VTRM performed on the evaluated impulse response



(d) VTRM performed on the actual impulse response

Fig. 20 Comparison between the evaluated impulse response and the actual impulse response before and after VTRM processing (continued)

Lastly, subjecting the simulated received signal to VTRM processing can remove the delay caused by the VTRM. The magnitude of the delay can be determined from the results of the VTRM based on the evaluated pulse response of each environment, and a bandpass filter can then be used to remove excess noise. The upper and lower cutoff frequencies are set at 34 ! 500 and 31 ! 500 Hz, respectively; the filtered data are converted into 3D data for network training.

The input data of the network are an $NX \times NT \times NS$ tensor, where NT is the number of time steps set as the window size, which defines the number of input variables used to predict the sequent time step, while NX is the number of data entries. Table 2 shows the SNR and (0, 1) distribution of the verification datasets with different SNRs formed by changing the emission intensity for the three simulation environments. This study uses the data of B_env1 as the training data set. Overall, 80%, 10%, and 10% of the data are employed as the training set, test set, and verification set, respectively, to train the different network models. The final trained models are applied to the validation data set, and the BERs at different SNRs are calculated, enabling comparison with the traditional demodulation techniques.

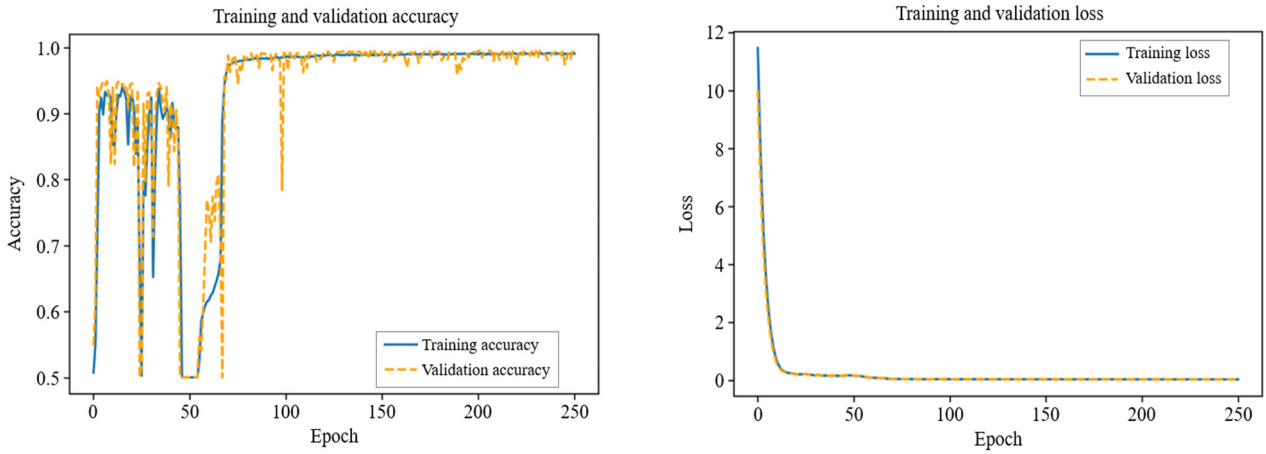
Table 2 SNR and (0, 1) distribution of different environment data sets

SNR (dB)	Classify bit "1" and bit "0"		
	0	1	Total
-14	49951	50049	100000
-11	49897	50103	100000
-8	49898	50102	100000
-6	49854	50146	100000
-3	50061	49939	100000
0	50088	49912	100000
2	49936	50064	100000
5	49945	50055	100000
8	50194	49806	100000
11	49874	50126	100000
14	50115	49885	100000

5.3. Analysis and discussion of network experiment results

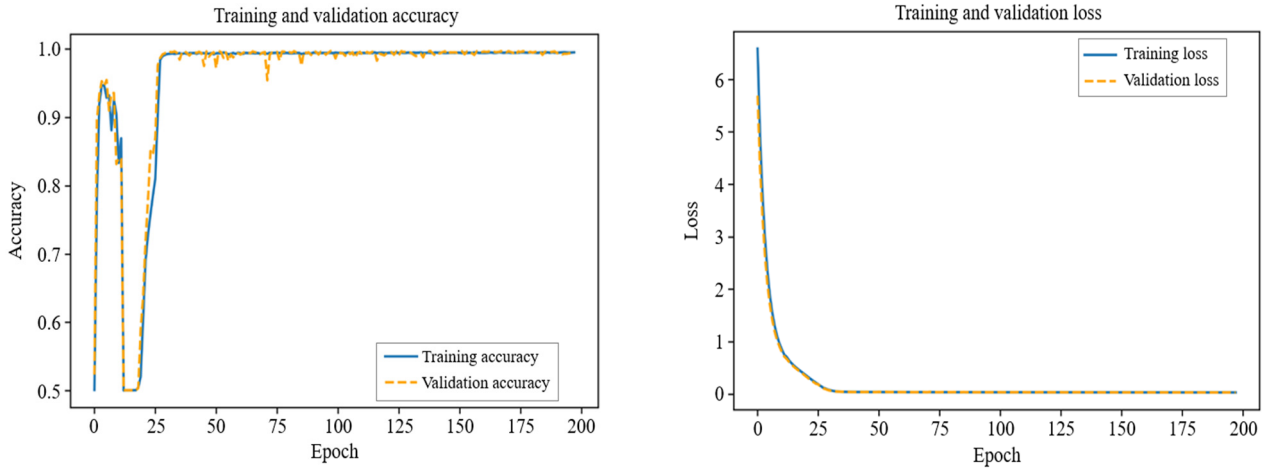
In this study, the BER is used to compare various deep learning model receivers with traditional FSK demodulation receivers. This study uses the data set of SNR of -14 dB for training, and reason is that a model trained on an environment with severe noise can be applied to data from environments with less noise.

5.3.1. Convergence curve and accuracy during the training of different networks



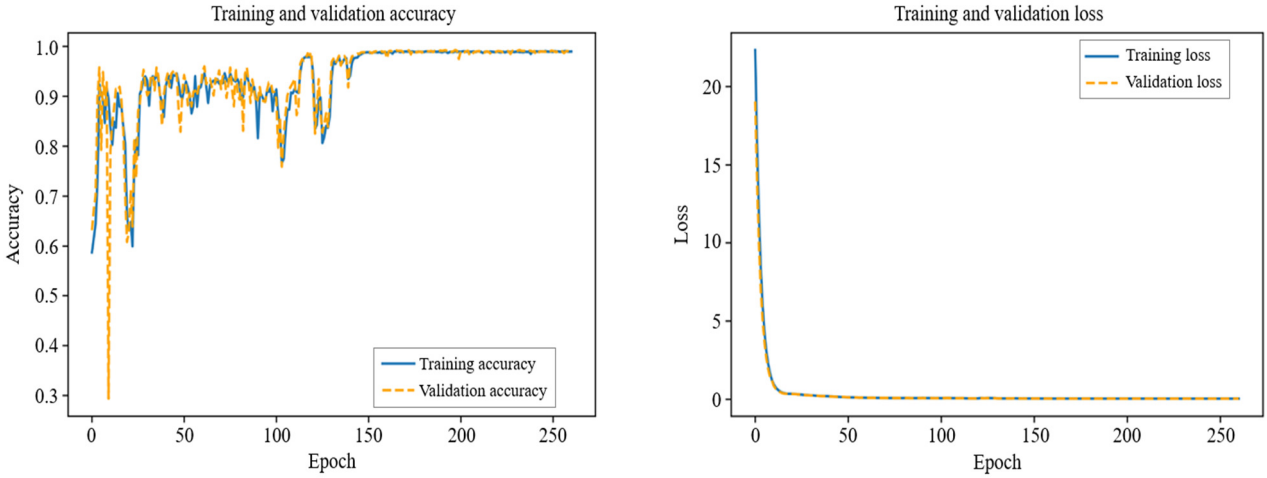
(a) Accuracy (b) Loss

Fig. 21 GRU training validation curve for the training data



(a) Accuracy (b) Loss

Fig. 22 LSTM training validation curve for the training data: (a) and (b)

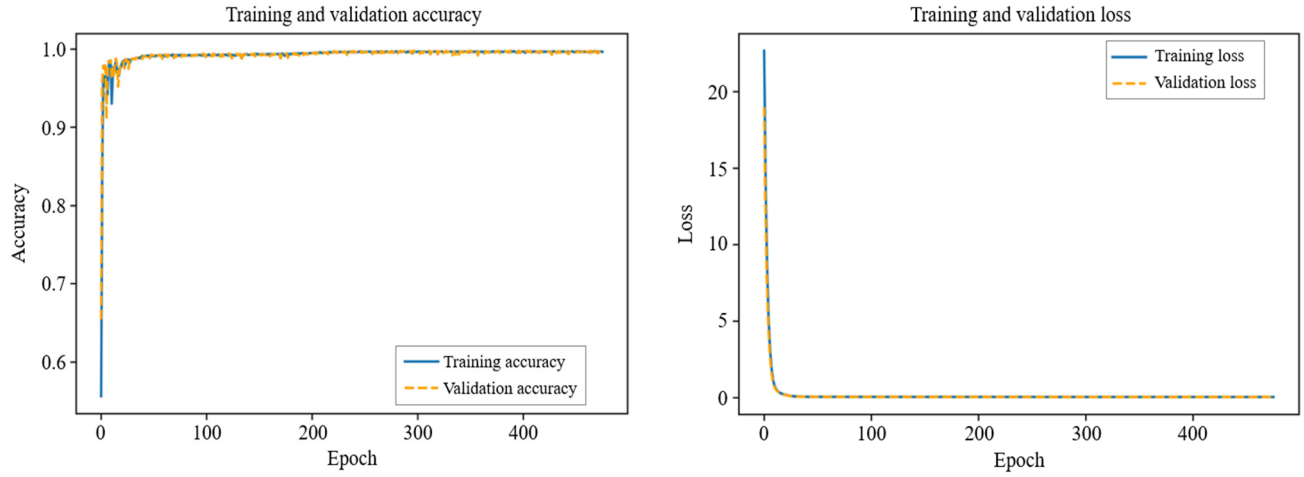


(a) Accuracy (b) Loss

Fig. 23 BiGRU training validation curve for the training data

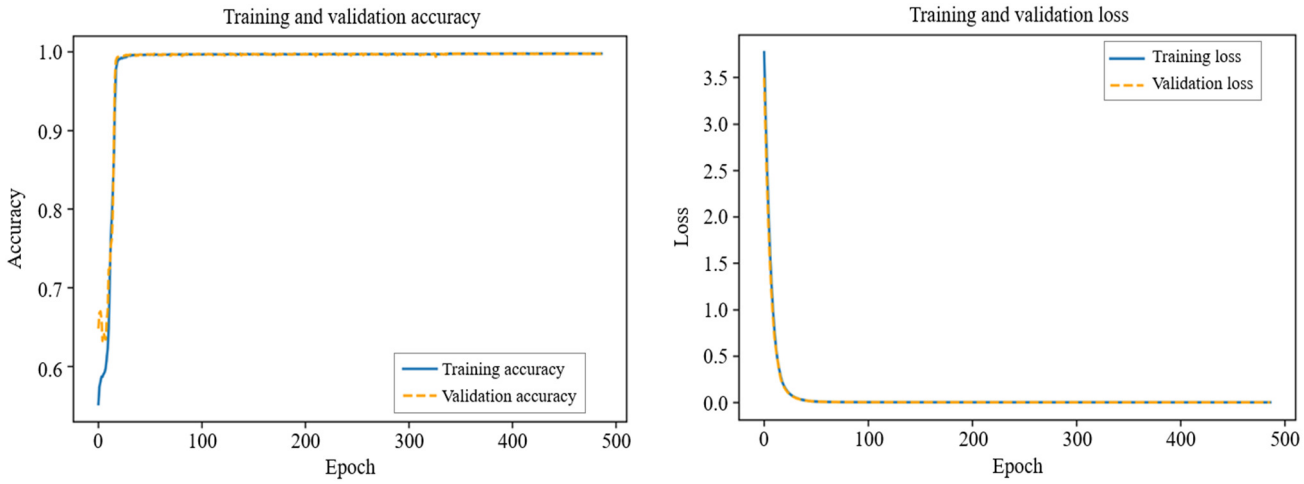
Figs. 21-27 display the accuracy and loss curves of the seven networks during network training. The networks all converge stably except for the GRU and BiGRU networks. Although the curves for these networks are convergent, their accuracy is unstable. This is because the GRU and BiGRU losses always converge to around 0.2 during network training, whereas those

of other networks converge to approximately 0.03. This difference is very serious in FSK demodulation. The accuracy and convergence curve of CNN-GRU reveals a major decrease in the convergence curve around the 24th iteration. The change in the convergence curve is a drop in the loss from 0.2 to 0.03, which leads to an increase in the accuracy to higher than 0.99 simultaneously and stabilization until the early stop technology ends the network training.



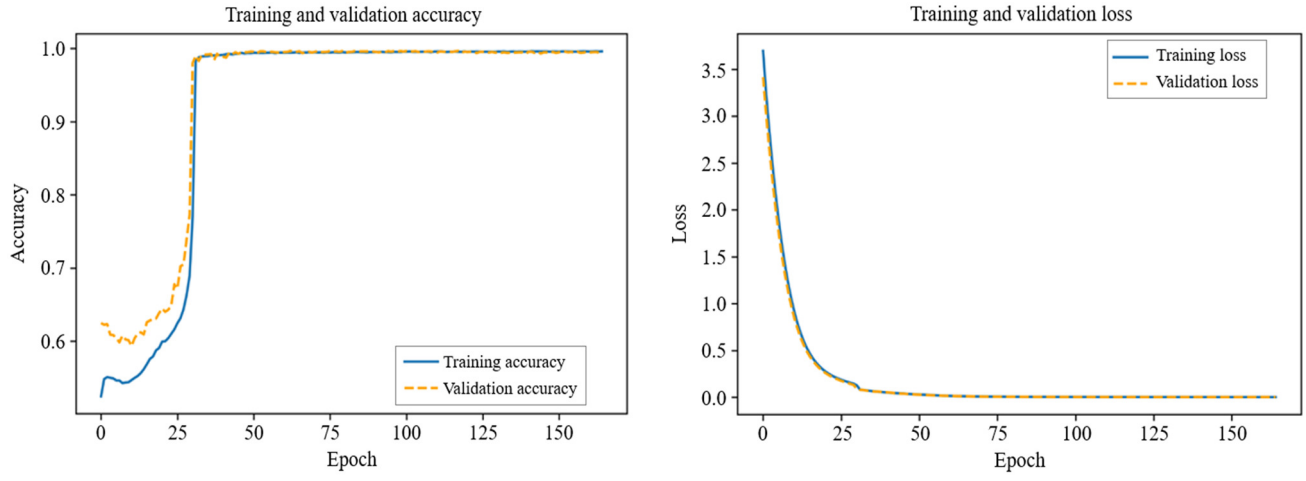
(a) Accuracy (b) Loss

Fig. 24 BiLSTM training validation curve for the training data



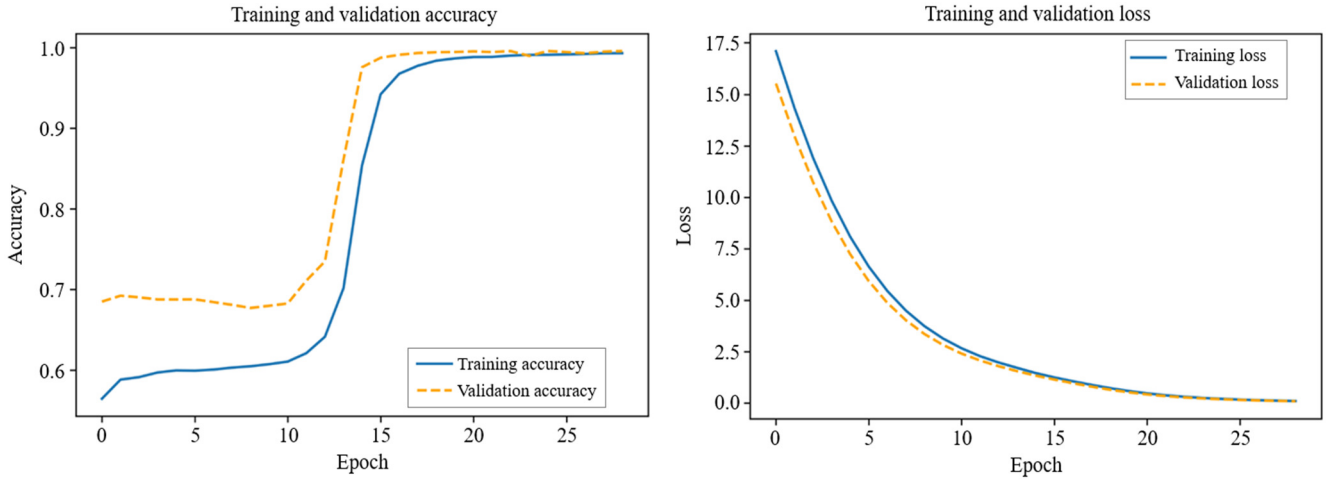
(a) Accuracy (b) Loss

Fig. 25 CNN-LSTM training validation curve for the training data



(a) Accuracy (b) Loss

Fig. 26 CNN-GRU training validation curve for the training data



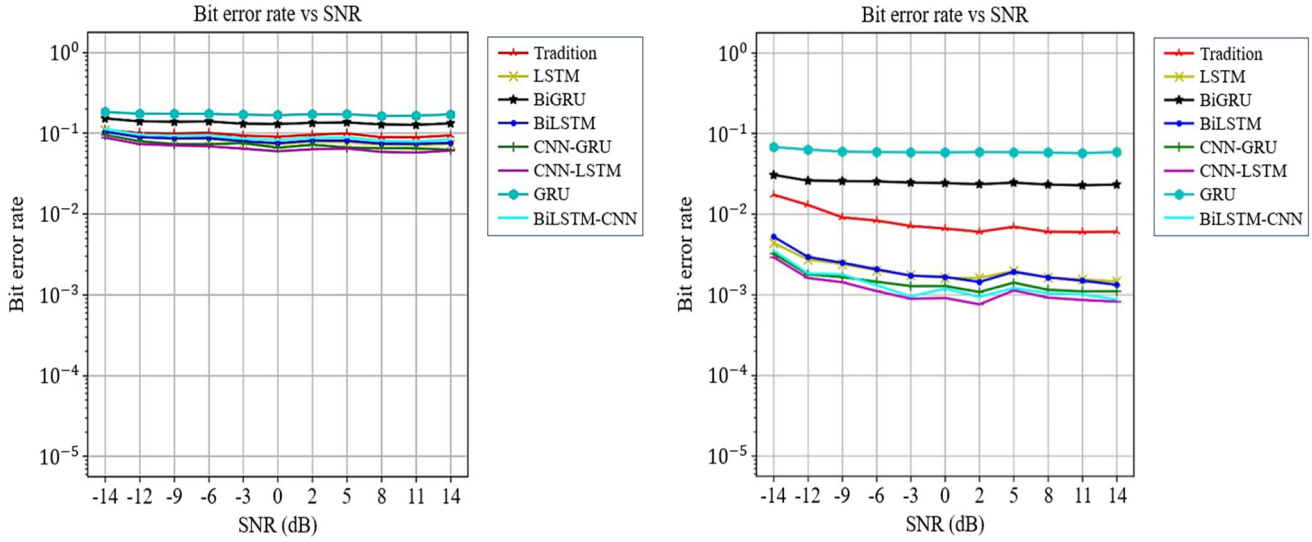
(a) Accuracy (b) Loss

Fig. 27 Stacked BiLSTM-CNN training validation curve for the training data

The reason for the inability of GRU and BiGRU to converge is that this study is conducted by demodulating the network with time-domain signals and the features contained in the time-domain features are too numerous and complicated, incurring an inability of GRU to identify effective time-series features from the LSTM-simplified network. During the network training, the loss stops at approximately 0.2; thus, this study uses the CNN for feature extraction and then uses GRU to stabilize the convergence to lower than 0.03.

5.3.2. Comparison of various network training in different SNR environments

This section compares the differences between using VTRM-optimized channels and non-VTRM-optimized channels in various underwater environments. This study compares the sea area with an SNR of -14 to 14 dB. Fig. 28 compares the noise interference in underwater environment, i.e., the southwest coast, and the moments when the VTRM was not and was used to optimize the environment, respectively.



(a) Without VTRM processing (b) With VTRM processing

Fig. 28 Network model BER curves for the test set under various numbers of noise

It confirms that the channel is greatly improved after VTRM processing and the BER is considerably reduced in all noise environments due to a strong influence emerging from the multipath. The BER performance using VTRM with different SNRs and network models is shown in Table 3. Owing to the proximity of both BER curves, the detailed values of the BER curves can be obtained from the test data when using the VTRM-optimized channel and the non-VTRM-optimized channel. The BER

does not decrease steadily in the case of an increase in SNR due to the unclear effect of noise. The BER data and SNR values obtained in this study are random binary transmission bit data, hence the data transmitted under different SNRs and environments are not the same, causing the randomness of the BER to not decrease steadily when the noise is not substantial.

Table 3 BER of different SNRs and network models using VTRM

Model	SNR (dB)										
	14	11	8	5	2	0	-3	-6	-8	-11	-14
LSTM	0.00146	0.00155	0.00163	0.00197	0.00161	0.00162	0.00174	0.002	0.0024	0.00273	0.00439
BiLSTM	0.0133	0.00149	0.00164	0.00192	0.00144	0.00166	0.00172	0.00207	0.00249	0.00294	0.00524
CNN-GRU	0.0011	0.0011	0.00115	0.00141	0.00108	0.00128	0.00128	0.00145	0.00165	0.00179	0.00322
CNN-LSTM	0.00082	0.00086	0.00092	0.00113	0.00076	0.00091	0.00089	0.00111	0.00143	0.00161	0.00292
GRU	0.05874	0.05688	0.05797	0.05841	0.05886	0.05807	0.05843	0.05878	0.05947	0.06304	0.06773
BiGRU	0.02316	0.02276	0.02315	0.02453	0.02343	0.02422	0.02457	0.02541	0.02569	0.02594	0.03065
Stacked BiLSTM-CNN	0.00087	0.00101	0.00104	0.00121	0.00094	0.00119	0.00095	0.00131	0.00178	0.00184	0.00354
Tradition (FSK)	0.00448	0.00474	0.00454	0.0054	0.0051	0.00466	0.00506	0.00543	0.00582	0.0057	0.00731

6. Conclusion

The sea area shows that multipaths have a substantial impact on received signals. Because of the strong influence of the multipaths, the BER of the traditional FSK receiver is approximately 0.08 notwithstanding the noise content. The accuracy of deep learning can be significantly better than that of traditional demodulation techniques if the deep learning model converges steadily. According to the present validation based on three sea areas, networks with CNN-retrieved features—such as CNN-GRU, CNN-LSTM, and Stacked BiLSTM-CNN—have an advantage regarding the recognition of values.

In other words, whereas LSTM and GRU are well suited for processing temporal features, models trained using a CNN to capture important spatial features are more robust when applied to the complex FSK modulation signals of UWA communication time-domain features. Such a result facilitates the application of network models to other received data processed by VTRM but not trained. The BER performance of purposed CNN-LSTM with VTRM has a superior performance of 0.00082 than traditional demodulation with VTRM of 0.00448 when the SNR stands at 14 dB. The results of the paper demonstrate that the proposed deep learning method significantly enhances the performance of conventional underwater sonar systems. In the future, transfer learning techniques and domain adaptation methods can use knowledge from related fields and enhance the generalization ability of deep learning models in underwater communication.

Acknowledgment

Thanks for supporting the National Science and Technology Council NSTC (grant no. NSTC 111-2634-F-019-001 and NSTC 112-2221-E-019-023) and the National Taiwan Ocean University. Moreover, thanks to editor's kind coordination and the reviewers for constructive suggestions.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] M. B. Porter, The BELLHOP Manual and User's Guide: Preliminary Draft, Heat, Light, and Sound Research, Inc. Technical Report 260, January 31, 2011.

[2] N. Morozs, W. Gorma, B. T. Henson, L. Shen, P. D. Mitchell, and Y. V. Zakharov, "Channel Modeling for Underwater Acoustic Network Simulation," IEEE Access, vol. 8, pp. 136151-136175, 2020.

[3] R. Jiang, S. Cao, C. Xue, and L. Tang, "Modeling and Analyzing of Underwater Acoustic Channels with Curvilinear Boundaries in Shallow Ocean," IEEE International Conference on Signal Processing, Communications and Computing, pp. 1-6, October 2017.

[4] O. Onasami, D. Adesina, and L. Qian, "Underwater Acoustic Communication Channel Modeling Using Deep Learning," Proceedings of the 15th International Conference on Underwater Networks & Systems, pp. 1-8, November 2021.

[5] Y. Li, B. Wang, G. Shao, S. Shao, and X. Pei, "Blind Detection of Underwater Acoustic Communication Signals Based on Deep Learning," IEEE Access, vol. 8, pp. 204114-204131, 2020.

[6] Y. Liu, F. Zhou, G. Qiao, Y. Zhao, G. Yang, X. Liu, et al., "Deep Learning-Based Cyclic Shift Keying Spread Spectrum Underwater Acoustic Communication," Journal of Marine Science and Engineering, vol. 9, no. 11, article no. 1252, November 2021.

[7] D. R. Jackson and D. R. Dowling, "Phase Conjugation in Underwater Acoustics," The Journal of the Acoustical Society of America, vol. 89, no. 1, pp. 171-181, January 1991.

[8] W. A. Kuperman, W. S. Hodgkiss, H. C. Song, T. Akal, C. Ferla, and D. R. Jackson, "Phase Conjugation in the Ocean: Experimental Demonstration of an Acoustic Time-Reversal Mirror," The Journal of the Acoustical Society of America, vol. 103, no. 1, pp. 25-40, January 1998.

[9] A. J. Silva and S. M. Jesus, "Underwater Communications Using Virtual Time Reversal in a Variable Geometry Channel," OCEANS '02 MTS/IEEE, vol. 4, pp. 2416-2421, October 2002.

[10] R. James, H. Appukuttan, and L. A. Joseph, "Mixed Noise Removal by Processing of Patches," Proceedings of Engineering and Technology Innovation, vol. 17, pp. 32-41, January 2021.

[11] M. Li, H. Zhong, and M. Li, "Neural Network Demodulator for Frequency Shift Keying," 2008 International Conference on Computer Science and Software Engineering, vol. 4, pp. 843-846, December 2008.

[12] I. Masmitja, D. Corregidor, J. M. López, E. Martinez, J. Navarro, and S. Gomariz, "Miniaturised Bidirectional Acoustic Tag to Enhance Marine Animal Tracking Studies," IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1-6, May 2021.

[13] F. Qu, Z. Wang, L. Yang, and Z. Wu, "A Journey Toward Modeling and Resolving Doppler in Underwater Acoustic Communications," IEEE Communications Magazine, vol. 54, no. 2, pp. 49-55, February 2016.

[14] Z. Gong, C. Li, and F. Jiang, "Analysis of the Underwater Multi-Path Reflections on Doppler Shift Estimation," IEEE Wireless Communications Letters, vol. 9, no. 10, pp. 1758-1762, October 2020.

[15] A. Farina, "Simultaneous Measurement of Impulse Response and Distortion with a Swept-Sine Technique," 108th AES Convention, article no. 5093, February 2000.

[16] T. B. Santoso, E. Widjiati, Wirawan, and G. Hendrantoro, "The Evaluation of Probe Signals for Impulse Response Measurements in Shallow Water Environment," IEEE Transactions on Instrumentation and Measurement, vol. 65, no. 6, pp. 1292-1299, June 2016.

[17] L. D. Ding, S. L. Wang, and W. Zhang, "Modulation Classification of Underwater Acoustic Communication Signals Based on Deep Learning," OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), pp. 1-4, May 2018.

[18] D. Utebayeva, M. Alduraibi, L. Ilipbayeva, and Y. Temirgaliyev, "Stacked BiLSTM - CNN for Multiple Label UAV Sound Classification," Fourth IEEE International Conference on Robotic Computing (IRC), pp. 470-474, November 2020.

[19] R. Jiang, S. Cao, C. Xue, and L. Tang, "Modeling and Analyzing of Underwater Acoustic Channels with Curvilinear Boundaries in Shallow Ocean," IEEE International Conference on Signal Processing, Communications and Computing, pp. 1-6, October 2017.

[20] W. Y. Chen, "Simulation and Analysis of Sonar Performances: The Study on Estimation Around Taiwan Waters," M.S. thesis, Institute of Undersea Technology, National Sun Yat-sen University, Kaohsiung, Taiwan, 2013.

[21] F. B. Louza and S. M. Jesus, "The Effects of Upwelling over Low SNR Communications in Shallow Water," OCEANS 2021 San Diego – Porto, pp. 1-6, September 2021.