



OPEN ACCESS

EDITED BY

Xuebo Zhang,
Northwest Normal University, China

REVIEWED BY

Xin Qing,
Harbin Engineering University, China
Jiahua Zhu,
National University of Defense Technology,
China

*CORRESPONDENCE

Tingting Lv
✉ tingtinglu@ouc.edu.cn

†These authors have contributed equally to this work

RECEIVED 01 November 2023

ACCEPTED 14 December 2023

PUBLISHED 11 January 2024

CITATION

Zhou M, Zhang H, Lv T, Huang W, Duan Y and Gao Y (2024) A simplified decision feedback Chebyshev function link neural network with intelligent initialization for underwater acoustic channel equalization.
Front. Mar. Sci. 10:1331635.
doi: 10.3389/fmars.2023.1331635

COPYRIGHT

© 2024 Zhou, Zhang, Lv, Huang, Duan and Gao. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

A simplified decision feedback Chebyshev function link neural network with intelligent initialization for underwater acoustic channel equalization

Manli Zhou^{1†}, Hao Zhang^{1,2†}, Tingting Lv^{1*}, Wei Huang¹, Yingying Duan¹ and Yong Gao¹

¹Department of Electronic Engineering, Ocean University of China, Qingdao, China, ²Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada

Introduction: In shallow-water environments, the reliability of underwater communication links is often compromised by significant multipath effects. Some equalization techniques such as decision feedback equalizer, and deep neural network equalizer suffer from slow convergence and high computational complexity.

Methods: To address this challenge, this paper proposes a simplified decision feedback Chebyshev function link neural network equalizer (SDF-CFLNNE). The structure of the SDF-CFLNNE employs Chebyshev polynomial function expansion modules to directly and non-linearly transform the input signals into the output layer, without the inclusion of hidden layers. Additionally, it feeds the decision signal back to the input layer rather than the function expansion module, which significantly reduces computational complexity. Considering that, in the training phase of neural networks, the random initialization of weights and biases can substantially impact the training process and the ultimate performance of the network, this paper proposes a chaotic sparrow search algorithm combining the osprey optimization algorithm and Cauchy mutation (OCCSSA) to optimize the initial weights and thresholds of the proposed equalizer. The OCCSSA utilizes the Piecewise chaotic population initialization and combines the exploration strategy of the osprey with the Cauchy mutation strategy to enhance both global and local search capabilities.

Results: Simulations were conducted using underwater multipath signals generated by the Bellhop Acoustic Toolbox. The results demonstrate that the performance of the SDF-CFLNNE initialized by OCCSSA surpasses that of CFLNNE-based and traditional nonlinear equalizers, with a notable improvement of 2–6 dB in terms of signal-to-noise ratio at a bit error rate (BER) of 10⁻⁴ and a reduced mean square error (MSE). Furthermore, the effectiveness of the proposed equalizer was validated using the lake experimental data, demonstrating lower BER and MSE with improved stability.

Discussion: This underscores the promise of employing the SDFCFLNNE initialized by OCCSSA as a promising solution to enhance the robustness of underwater communication in challenging environments.

KEYWORDS

decision feedback equalizer, Chebyshev function link artificial neural network, sparrow search algorithm, osprey optimization algorithm, chaotic mapping, Cauchy mutation

1 Introduction

The shallow water acoustic environment is complex and changeable, which often exhibits intricate signal multipath effects and Doppler frequency shifts (Stojanovic and Preisig, 2009; Huang et al., 2018). The multipath propagation of underwater acoustic (UWA) signals originates from the effects of acoustic boundaries (such as reflection from the water surface and seabed), refraction caused by the non-uniformly distributed dissound speed in the water, as well as scattering from particles. The complex multipath results in significant signal time spreading, thereby causing severe intersymbol interference. In typical shallow water acoustic communications, intersymbol interference may span over hundreds of symbols. Consequently, at the receiver end, it is essential for the channel equalization to possess strong adaptive channel tracking capabilities (Song et al., 2006; Wang et al., 2021). This poses a significant challenge for reliable and efficient UWA communication (Zhang et al., 2018).

To combat intersymbol interference caused by time-varying multipath propagation, extensive research has been conducted on various channel equalization techniques. Single-carrier schemes and time-domain equalization techniques offer high spectral efficiency and robustness, albeit at the cost of high receiver complexity (Stojanovic and Preisig, 2009; Zhang et al., 2018). The proposed adaptive step-size least mean square performs well for many channel types, but for certain complex non-stationary UWA channels, the rapid tracking capability of recursive least square is essential (Freitag et al., 1997). To achieve reliable coherent communication over UWA channels, a receiver was designed which combines the recursive least square algorithm with a second-order digital phase-locked loop for carrier synchronization and performs fractionally spaced decision feedback equalization of the received signals. The parameters of this receiver are adaptively adjusted (Stojanovic et al., 1994). An adaptive nonlinearity (piecewise linear) was introduced into the channel equalization algorithm and its effectiveness was demonstrated through highly realistic experiments conducted on real-field data as well as accurate simulations of UWA channels (Kari et al., 2017). In recent years, in order to alleviate propagation errors, expedite convergence speed, and further enhance receiver performance, there has been growing research on adaptive turbo equalization (He et al., 2019; Xi et al.,

2019; Qin et al., 2020). Considering the sparsity inherent in UWA channels, sparse matrices have been utilized to construct sparse equalizers, aiming to achieve faster convergence and lower error rates (Xi et al., 2020; Wang et al., 2021; Wang et al., 2021). Additionally, the equalization challenges in an impulsive interference single-carrier modulation system based on a parameterized model are addressed, and a two-step equalization algorithm is proposed (Ge et al., 2022). The robust equalization for single-carrier underwater acoustic communication in sparse impulsive interference environment was proposed (Wei et al., 2023). This algorithm is based on the framework of variational Bayesian inference and possesses the unique capability of simultaneously accounting for the sparsity inherent in the channel and impulse interference. At the same time, several waveform design (Zhu et al., 2023) and enhanced receiver schemes (Zhang et al., 2021; Liu et al., 2023) were proposed to further address inter-symbol interference and multipath propagation issues. However, the complex multipath effect of the UWA channels contributes to the slow convergence rate and extensive computational requirements of traditional equalization algorithms. As a result, there is substantial room for improvement in UWA communication systems.

In recent years, machine learning techniques have garnered attention across various fields. Particularly, deep learning (DL) technology holds tremendous potential for addressing non-parametric problems such as object detection and recognition (Tsai et al., 2013), speech recognition (Zhang and Wang, 2016), target tracking (Milan et al., 2017), wireless communication (Wang et al., 2017; Ma et al., 2018; van Heteren, 2022; Mishra et al., 2023). In order to reduce the computational costs of traditional equalizers, machine learning-based equalizers have been introduced to mitigate intersymbol interference. Channel equalization can be viewed as a classification problem, where the equalizer is designed as a decision device with the motivation to classify the transmitted signals as accurately as possible (Zhang and Yang, 2020). Gibson et al. introduced an adaptive equalizer employing a neural network architecture based on multilayer perceptrons (MLP) to counter intersymbol interference on linear channels with Gaussian white noise (Gibson et al., 1989). Chang et al. proposed a neural network-based decision feedback equalizer (DFE) that obviates the need for time-consuming complex-valued backpropagation training algorithms (Chang and Wang, 1995). Gao et al. demonstrated

that in underwater digital communication scenarios, their proposed blind equalizer achieves faster convergence speed and smaller mean square error (MSE) compared to original MLP-based equalizers that require training data (Gao et al., 2009). Zhang et al. proposed a DL-based time-varying UWA channel single-carrier communication receiver to adapt to the dynamic characteristics of UWA channels. The receiver operates in an alternating mode between online training and testing (Zhang et al., 2019b). Zhang et al. introduced a DL-based UWA communication orthogonal frequency-division multiplexing receiver. A stack of convolutional layers with skip connections effectively extracts meaningful features from the received signal and reconstructs the original transmitted symbols (Zhang et al., 2022). Radial basis function (RBF) neural networks have garnered the attention of many researchers due to their simple structure and high learning efficiency, and have been utilized for addressing channel equalization issues (Lee and Sankar, 2007; Guha and Patra, 2009; Ning et al., 2009).

However, with a higher channel order, a greater number of RBF centers are required, ultimately resulting in an excessive computational burden. To overcome these drawbacks of MLP and RBF, another novel single-layer neural network, known as the Functional Link Neural Network (FLNN), was proposed by Paul. Due to the non-linear processing of signals in the FLNN, it can generate arbitrarily complex decision regions (Patra et al., 1999). This network features a simple structure with only input and output layers, and the hidden layer is entirely replaced by non-linear mappings. These mappings are introduced through the expansion of input patterns using trigonometric polynomials and other basis functions like Gaussian polynomials, orthogonal polynomials, Legendre polynomials, and Chebyshev polynomials (Burse et al., 2010). The FLNN increases the dimensionality of the input signal space by a set of linearly independent non-linear functions, thus reducing computational load and allowing for straightforward hardware implementation (Patra et al., 2008; Zhang and Yang, 2020). Moreover, research indicates that non-linear equalizers based on FLNN outperform MLP, RBF, and PPN equalizers in terms of MSE, convergence rate, bit error rate (BER), and computational complexity (Patra et al., 1999). Lee et al. introduced a Chebyshev Neural Network for static function approximation, which is more computationally efficient than trigonometric polynomials when expanding the input space for extended static function approximation and non-linear dynamic system identification (Lee and Jeng, 1998). Patra et al. have employed Chebyshev Functional Link Neural Networks (CFLNN) for channel equalization of four quadrature amplitude modulation signals (Patra and Kot, 2002; Patra et al., 2005). Hussain combined traditional DFE with FLNN, proposing a Decision Feedback Functional Link Neural Network Equalizer (DFFLNN) (Hussain et al., 1997). Building upon this, they introduced a Chebyshev orthogonal polynomial cascaded FLNN for non-linear channel equalization (Zhao and Zhang, 2008) and an adaptive DFE based on the combination of the FIR and FLNN (Zhao et al., 2011). Moreover, Convolutional Neural Network (He et al., 2023), Recurrent Neural Networks (Kechriotis et al., 1994; Chagra et al.,

2005; Xiao et al., 2008; Zhao et al., 2010; Li et al., 2021; Qiao et al., 2022), Fuzzy Neural Networks (Heng et al., 2006; Chang and Ho, 2009; Chang and Ho, 2011), Extreme Learning Machines (Yang et al., 2018; Liu et al., 2019), Wavelet Neural Networks (Xiao and Dong, 2015), Support Vector Machines (Zhang et al., 2019a), other neural network models and Deep Reinforcement Learning (He and Tao, 2023) have been employed for channel equalization.

Swarm intelligence optimization algorithms are a class of bio-inspired algorithms inspired by the behavioral patterns of certain social organisms in the natural world. The central idea is to conduct both global and local searches within a solution space to find optimal solutions. These algorithms provide a new approach to solving complex problems without centralized control or a global model. In recent years, new swarm intelligence optimization algorithms have continuously emerged. Scholars have drawn inspiration from the behavior of various animals such as ants, wolves, birds, moths, whales, sparrows, and more to propose a series of swarm intelligence optimization algorithms, including the Particle Swarm Optimization (PSO) algorithm (Kennedy and Eberhart, 1995), the Grey Wolf Optimization (GWO) algorithm (Mirjalili et al., 2014), the Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016), the Bald Eagle Search (BES) algorithm (Alsattar et al., 2020), the Sparrow Search Algorithm (SSA) (Xue and Shen, 2020), the Cooperation Search Algorithm (CSA) (Feng et al., 2021), artificial gorilla troops optimizer (GTO) (Abdollahzadeh et al., 2021), white shark optimizer (WSO) (Braik et al., 2022), dung beetle optimizer (DBO) (Xue and Shen, 2023) and Osprey Optimization Algorithm (OOA) (Dehghani and Trojovský, 2023). The Sparrow Search Algorithm (SSA) was first introduced by Xue et al. in 2020 (Xue and Shen, 2020). In comparison to other algorithms, SSA offers several advantages, including fast convergence, strong optimization capabilities, and a wider range of application scenarios. As a result, SSA has garnered the attention of researchers from various fields. However, SSA does have limitations in terms of initial population quality, search capabilities, and population diversity. To address these issues, the Improved Sparrow Search Algorithm (ISSA) was proposed (Song et al., 2020). ISSA introduces non-linear decay in the position updates of producers, which facilitates the exploration and utilization of the search space. ISSA incorporates a mutation strategy to update the positions of scavengers with lower energy, combining chaotic search with local development by higher-energy scavengers. This enhances diversity and prevents falling into local optima. At the same time, the Tent mapping is used to initialize the population. Then, for the producers, an adaptive weight strategy is combined with the Levy flight mechanism, making the fusion search approach more comprehensive and flexible. Finally, in the scavenger stage, a variable spiral search strategy is employed to provide a more detailed search scope (Ouyang et al., 2021).

Traditional network equalizers suffer from problems such as large steady-state errors, slow convergence, susceptibility to local minima during the search process, and the curse of dimensionality. Moreover, in the training phase of neural networks, the random initialization of weights and biases can substantially impact the

training process and the ultimate performance of the network. In contrast, swarm intelligence optimization algorithms exhibit strong convergence and high precision advantages in the optimization process of practical problems. Therefore, they have become popular research topics in the field of equalizer optimization methods. A modified constant modulus algorithm digital channel equalizer learning algorithm based on PSO is proposed by Sahu (Sahu and Majumder, 2021). The particle swarm algorithm is employed as the training algorithm, resulting in a shorter convergence time and better performance compared to traditional LMS algorithms. This equalizer avoids introducing any phase ambiguity and does not get trapped in local optima. A novel training strategy using the Fuzzy Firefly Algorithm is proposed for channel equalization (Mohapatra et al., 2022). By employing an appropriate network topology and parameters, the suggested training system exhibits enhanced exploration and exploitation capabilities, as well as the ability to address local minima issues. An enhanced Grasshopper Optimization Algorithm (GOA) is proposed for nonlinear wireless communication channel equalization (Ingle and Jatoh, 2023). By combining Levy flights and greedy selection operators with the basic GOA, the diversity of the swarm is increased. Simulation results on four nonlinear channels demonstrate the exploration and exploitation capabilities of the improved Grasshopper Optimization Algorithm in terms of MSE and BER performance. An effective equalizer based on artificial neural networks is proposed by Shwetha (Shwetha et al., 2023). The Battle Royale Optimization method, as introduced, is utilized to train the weights of the neural network. The effectiveness of this approach is demonstrated through the evaluation of performance metrics such as MSE, mean squared residual error, and BER.

In shallow water acoustic propagation, there often exists severe multipath effects. Traditional equalization techniques may require hundreds of taps, greatly increasing system complexity. While the DFFLNNE (Hussain et al., 1997) outperforms FLNNE and traditional DFE, but it increases the

dimensionality of the input layer, raising the complexity of the network structure. Simultaneously, during the training phase of the network, random initialization of weights and biases can affect the neural network’s training process and final performance. Improper initialization can lead to problems such as gradient vanishing or exploding, causing training to be infeasible or overly slow. To enhance communication reliability without increasing system complexity, this paper proposes a simplified decision feedback Chebyshev functional link neural network equalizer (SDF-CFLNNE) initialized with swarm intelligence optimization algorithms. The papers contributions can be summarized as follows.

1. To address the issue of unreliability in underwater communication links caused by significant strong multipath effects in shallow-water environments, we propose a simplified decision feedback Chebyshev function link neural network equalizer.
2. To optimize the initial weights and thresholds of the proposed equalizer, We propose a Chaotic Sparrow Search Algorithm combining osprey optimization algorithm and Cauchy mutation. This approach mitigates the instability resulting from random weight initialization in the network equalizer.

The rest of this paper is organized as follows. In Section 2, a novel simplified decision feedback Chebyshev functional link neural network equalizer is proposed to address the unreliability of communication due to multipath effects. In Section 3, a chaotic sparrow search algorithm combining osprey optimization algorithm and Cauchy mutation is proposed for intelligent optimization of network weight and bias initialization. We validate the method through simulation and lake experimental data processing in Section 4. Finally, conclusions are given in Section 5.

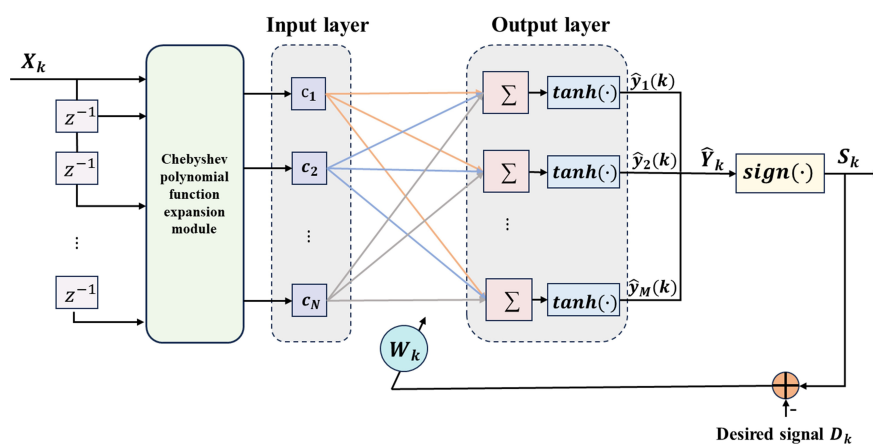


FIGURE 1 The structural diagram of the CFLNNE.

2 A novel simplified decision feedback Chebyshev functional link neural network equalizer

2.1 Decision feedback Chebyshev functional link neural network equalizer

To overcome intersymbol interference caused by multipath effects, a channel equalizer is embedded in the receiver to restore the transmitted signal. The FLNNE (Patra et al., 1999) has no hidden layers and is composed solely of a function extension module and a single-layer perceptron. This composition enables the generation of complex decision regions through the creation of nonlinear decision boundaries. In contrast to the linear weighting of input patterns generated by linear connections in MLP, the function expansion module enhances the dimensionality of input patterns by applying a set of linearly independent functions to elements or the entire pattern itself, thus enhancing its representation in high-dimensional space. Moreover, due to its single-layer structure, this FLNN structure exhibits lower computational complexity and faster convergence speed compared to other traditional neural networks. As widely recognized, utilizing the optimal approximation theory, Chebyshev orthogonal polynomials possess a robust capability for nonlinear approximation (Patra et al., 2005). The function expansion module in this context is composed of Chebyshev polynomials and their outer products, serving to simulate nonlinear channels, to construct the Chebyshev Functional Link Neural Network Equalizer (CFLNNE). The Figure 1 illustrates the structure of the CFLNNE.

Chebyshev polynomials are a set of orthogonal polynomials defined as solutions to the Chebyshev differential equation, denoted as $T_n(x)$. Chebyshev polynomials are computationally more tractable compared to trigonometric polynomials. The first several Chebyshev polynomials are given by $T_0(x) = 1$, $T_1(x) = x$,

and $T_2(x) = 2x^2 - 1$. When the input signal is $\mathbf{X}_k = [x_1(k), x_2(k), \dots, x_M(k)]^T$, the higher-order Chebyshev polynomials for $-1 < x < 1$ can be generated using the following recursion formula Equation 1:

$$\begin{aligned} c_1(\mathbf{X}_k) &= T_0(\mathbf{X}_k) = 1, \\ c_2(\mathbf{X}_k) &= T_1(\mathbf{X}_k) = \mathbf{X}_k, \\ &\vdots \\ c_{n+2}(\mathbf{X}_k) &= T_{n+1}(\mathbf{X}_k) = 2\mathbf{X}_k T_n(\mathbf{X}_k) - T_{n-1}(\mathbf{X}_k) \end{aligned} \tag{1}$$

In CFLNNE, the input signal denoted as \mathbf{X}_k is expanded into N linearly independent functions using Chebyshev polynomials, and can be represented as $\mathbf{C}_k = [c_1(\mathbf{X}_k) c_2(\mathbf{X}_k) \dots c_N(\mathbf{X}_k)]^T$.

Through forward propagation, the j -th neuron of the output layer can be represented as Equations 2 and 3:

$$u_j(k) = \sum_{i=1}^N w_{ji}(k) c_i(X_k) + b_j(k) \quad i = 1, 2, \dots, N; j = 1, 2, \dots, M. \tag{2}$$

$$\begin{aligned} \hat{y}_j(k) &= f(u_j(k)) = f\left(\sum_{i=1}^N w_{ji}(k) c_i(X_k) + b_j(k)\right) \quad i = 1, 2, \dots, N; j \\ &= 1, 2, \dots, M. \end{aligned} \tag{3}$$

where w represents the weight coefficients from the input layer to the output layer, and b represents the bias of the output layer. The nonlinear activation function here is $f(\cdot) = \tanh(\cdot)$, and its derivative is denoted as $f'(\cdot)$.

The output signal after decision device can be represented as Equation 4:

$$s(k) = \text{sign}(\hat{y}(k)) = \begin{cases} -1 & \text{if } \hat{y}(k) < 0 \\ 1 & \text{if } \hat{y}(k) \geq 0 \end{cases} \tag{4}$$

Taking advantage of the traditional decision feedback equalizer's ability to mitigate inter-symbol interference introduced by the preceding information symbol, Hussain et al. integrated the DFE with a FLNN,

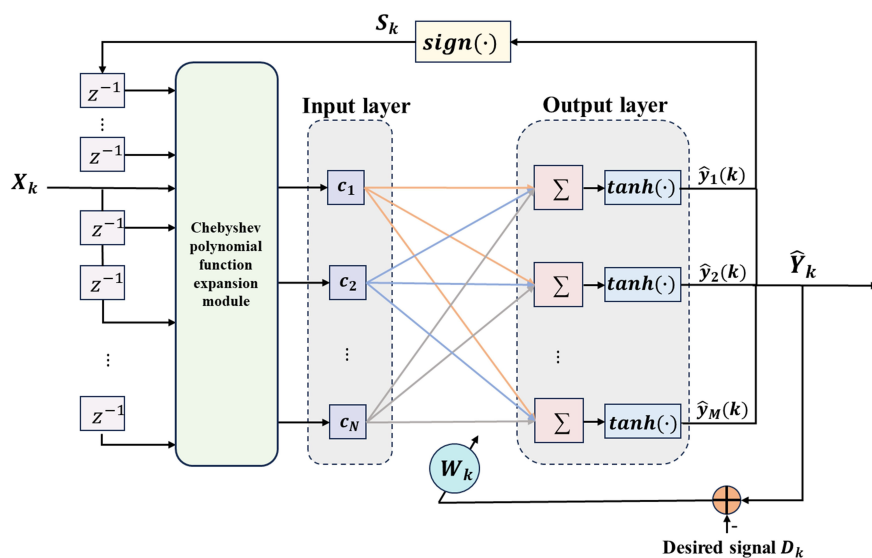


FIGURE 2 The structural diagram of the DF-CFLNNE.

creating the decision feedback functional link neural network (Hussain et al., 1997). To further enhance the nonlinear approximation capabilities of the function link module, the DFE is combined with CFLNNE to form a decision feedback Chebyshev functional link neural network (DF-CFLNNE), as illustrated in Figure 2.

The input signal $\mathbf{X}_k = [x_1(k), x_2(k), \dots, x_M(k)]^T$ and feedback signal $\mathbf{S}_k = [s_1(k), \dots, s_{N_2}(k)]^T$ from the decision device are jointly used as the input signal $\mathbf{Z}_k = [\mathbf{X}_k, \mathbf{S}_k]$ for the DF-CFLNNE, where N_2 represents the order of the feedback delay path. The key distinction from CFLNNE is that CFLNNE takes only \mathbf{X}_k as its input signal, without the feedback signals from the decision device. Subsequently, the input signal \mathbf{Z}_k of DF-CFLNNE is expanded into N linearly independent functions using Chebyshev polynomials, denoted as $\mathbf{C}_k = [c_1(\mathbf{Z}_k), c_2(\mathbf{Z}_k), \dots, c_N(\mathbf{Z}_k)]^T$, where \mathbf{C}_k serves as the input to the network's input layer.

Through forward propagation, the j -th neuron of the output layer can be represented as Equations 5 and 6:

$$u_j(k) = \sum_{i=1}^N w_{ji}(k)c_i(\mathbf{Z}_k) + b_j(k) \quad i = 1, 2, \dots, N; j = 1, 2, \dots, M. \quad (5)$$

$$\hat{y}_j(k) = f(u_j(k)) = f\left(\sum_{i=1}^N w_{ji}(k)c_i(\mathbf{Z}_k) + b_j(k)\right) \quad i = 1, 2, \dots, N; j = 1, 2, \dots, M. \quad (6)$$

For convenience, these values of functions can be represented in matrix form as Equation 7:

$$\hat{\mathbf{Y}}_k = f(\mathbf{W}_k \mathbf{C}_k + \mathbf{B}_k) \quad (7)$$

where \mathbf{W}_k is an $M \times N$ dimensional matrix, i.e., $\mathbf{W}_k = [w_{j1}, w_{j2}, \dots, w_{jN}]$. \mathbf{B}_k is an $M \times 1$ dimensional matrix, i.e., $\mathbf{B}_k = [b_1, b_2, \dots, b_M]$. The

output of the entire network can be represented in matrix form as $\hat{\mathbf{Y}}_k = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_M]^T$.

We use the MSE as the loss function, which can be represented as Equation 8:

$$\mathbf{J}_k = \sum (\mathbf{D}_k - \hat{\mathbf{Y}}_k)^2 \quad (8)$$

where \mathbf{D}_k represents the desired output sequence at time instant k .

The backpropagation algorithm is employed here to train the DF-CFLNNE. The training process is expressed as follows Equations 9 and 10:

$$\begin{aligned} \frac{d\mathbf{J}_k}{d\mathbf{W}_k} &= \frac{\partial \mathbf{J}_k}{\partial \hat{\mathbf{Y}}_k} \cdot \frac{\partial \hat{\mathbf{Y}}_k}{\partial \mathbf{U}_k} \cdot \frac{\partial \mathbf{U}_k}{\partial \mathbf{W}_k} \\ &= 2(\mathbf{D}_k - \hat{\mathbf{Y}}_k) \cdot f'(\mathbf{U}_k) \cdot \mathbf{C}_k \\ &= 2(\mathbf{D}_k - \hat{\mathbf{Y}}_k) \cdot (1 - \hat{\mathbf{Y}}_k^2) \cdot \mathbf{C}_k \end{aligned} \quad (9)$$

$$\begin{aligned} \frac{d\mathbf{J}_k}{d\mathbf{B}_k} &= \frac{\partial \mathbf{J}_k}{\partial \hat{\mathbf{Y}}_k} \cdot \frac{\partial \hat{\mathbf{Y}}_k}{\partial \mathbf{U}_k} \cdot \frac{\partial \mathbf{U}_k}{\partial \mathbf{B}_k} \\ &= 2(\mathbf{D}_k - \hat{\mathbf{Y}}_k) \cdot f'(\mathbf{U}_k) \\ &= 2(\mathbf{D}_k - \hat{\mathbf{Y}}_k) \cdot (1 - \hat{\mathbf{Y}}_k^2) \end{aligned} \quad (10)$$

According to the gradient descent algorithm, there will be Equations 11 and 12:

$$\mathbf{W}_k = \mathbf{W}_k - \mu \frac{d\mathbf{J}_k}{d\mathbf{W}_k} \quad (11)$$

$$\mathbf{B}_k = \mathbf{B}_k - \mu \frac{d\mathbf{J}_k}{d\mathbf{B}_k} \quad (12)$$

where the parameter μ denotes the learning factor.

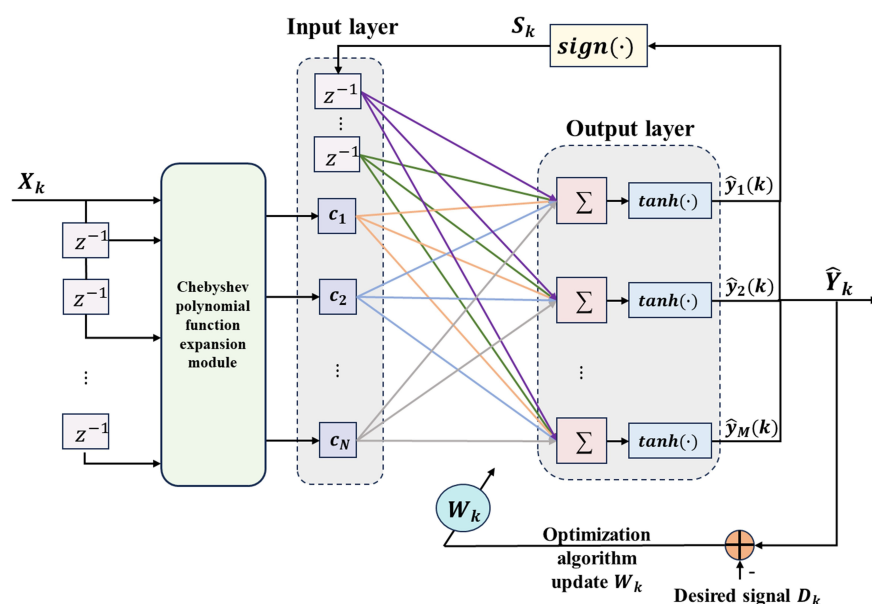


FIGURE 3 The structural diagram of the SDF-CFLNNE.

2.2 Simplified decision feedback Chebyshev functional link neural network equalizer

DF-CFLNNE increases the system’s performance at the cost of increased complexity. In order to reduce hardware costs without compromising system performance, a simplified DF-CFLNNE (SDF-CFLNNE) structure is proposed as illustrated in Figure 3.

In the SDF-CFLNNE structure, the post-decision output signal is directly fed back to the input layer of the neural network, rather than being used as an input signal to the network, and it no longer goes through the function expansion module. Namely, the input layer signal is composed of the Chebyshev polynomial function expansion of the received signal, denoted as $\mathbf{C}_k = [c_1(\mathbf{X}_k), c_2(\mathbf{X}_k), \dots, c_N(\mathbf{X}_k)]^T$, and the feedback signal after decision, denoted as $\mathbf{S}_k = [s_1(k), \dots, s_{N_2}(k)]^T$, which can be represented as $\mathbf{G}_k = [c_1(\mathbf{X}_k), c_2(\mathbf{X}_k), \dots, c_N(\mathbf{X}_k), s_1(k), \dots, s_{N_2}(k)]^T = [g_1(k), \dots, g_P(k)]^T, P = N + N_2$. It can be observed from Figures 2, 3 that the number of input signals in SDF-CFLNNE is fewer compared to DF-CFLNNE. Consequently, the number of signals after the function expansion module for SDF-CFLNNE is significantly reduced compared to DF-CFLNNE. This streamlined system structure enhances computational efficiency.

Through forward propagation, the j -th neuron of the output layer can be represented as Equations 13 and 14:

$$u_j(k) = \sum_{i=1}^N w_{ji}(k)g_i(k) + b_j(k) \quad i = 1, 2, \dots, P; j = 1, 2, \dots, M. \quad (13)$$

$$\begin{aligned} \hat{y}_j(k) &= f(u_j(k)) = f\left(\sum_{i=1}^N w_{ji}(k)g_i(k) + b_j(k)\right) \quad i = 1, 2, \dots, P; j \\ &= 1, 2, \dots, M. \end{aligned} \quad (14)$$

where w represents the weight coefficients from the input layer to the output layer, and b represents the bias of the output layer. The nonlinear activation function here is $f(\cdot) = \tanh(\cdot)$.

For convenience, these values of functions can be represented in matrix form as Equation 15:

$$\hat{\mathbf{Y}}_k = f(\mathbf{W}_k \mathbf{G}_k + \mathbf{B}_k) \quad (15)$$

where \mathbf{W}_k is an $M \times P$ dimensional matrix, i.e., $\mathbf{W}_k = [w_{j1}, w_{j2}, \dots, w_{jP}]$. \mathbf{B}_k is an $M \times 1$ dimensional matrix, i.e., $\mathbf{B}_k = [b_1, b_2, \dots, b_M]$. The output of the entire network can be represented in matrix form as $\hat{\mathbf{Y}}_k = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_M]^T$. We still adopt the MSE, as given in Equation 8, as the loss function.

The BP algorithm is employed here to train the SDF-CFLNN. According to Equations 9 and 10, the training process is expressed as follows Equations 16 and 17:

$$\begin{aligned} \mathbf{W}_{k+1} &= \mathbf{W}_k - \mu \frac{dJ_k}{d\mathbf{W}_k} \\ &= \mathbf{W}_k - \mu \frac{\partial J_k}{\partial \hat{\mathbf{Y}}_k} \cdot \frac{\partial \hat{\mathbf{Y}}_k}{\partial \mathbf{U}_k} \cdot \frac{\partial \mathbf{U}_k}{\partial \mathbf{W}_k} \\ &= \mathbf{W}_k - 2\mu(\mathbf{D}_k - \hat{\mathbf{Y}}_k) \cdot f'(\mathbf{U}_k) \cdot \mathbf{G}_k \\ &= \mathbf{W}_k - 2\mu(\mathbf{D}_k - \hat{\mathbf{Y}}_k) \cdot (1 - \hat{\mathbf{Y}}_k^2) \cdot \mathbf{G}_k \end{aligned} \quad (16)$$

$$\begin{aligned} \mathbf{B}_{k+1} &= \mathbf{B}_k - \mu \frac{dJ_k}{d\mathbf{B}_k} \\ &= \mathbf{B}_k - \mu \frac{\partial J_k}{\partial \hat{\mathbf{Y}}_k} \cdot \frac{\partial \hat{\mathbf{Y}}_k}{\partial \mathbf{U}_k} \cdot \frac{\partial \mathbf{U}_k}{\partial \mathbf{B}_k} \\ &= \mathbf{B}_k - 2\mu(\mathbf{D}_k - \hat{\mathbf{Y}}_k) \cdot f'(\mathbf{U}_k) \\ &= \mathbf{B}_k - 2\mu(\mathbf{D}_k - \hat{\mathbf{Y}}_k) \cdot (1 - \hat{\mathbf{Y}}_k^2) \end{aligned} \quad (17)$$

where the parameter μ denotes the learning factor.

SDF-CFLNNE directly inputs the decision feedback signal into the network’s input layer instead of the function expansion module, reducing the number of neurons in the input layer. In this way, we can obtain the improvement of system performance from the feedback signal without increasing the number of neurons in the input layer of the network. It reduces system complexity, enhances computational efficiency, and accelerates convergence speed.

3 A novel chaotic sparrow search algorithm combining osprey optimization algorithm and Cauchy mutation

In this section, a OCCSSA is proposed to solve the impact of the random initialization of network weights on the convergence of the training process and network performance. This algorithm utilizes chaotic mapping for random population initialization and combines the osprey optimization algorithm with the Cauchy mutation criterion to update the positions in the SSA. The use of the osprey optimization algorithm.

(OOA) in the initial phase provides a global exploration strategy, where a random attack on one of the food sources helps mitigate the SSA’s over-reliance on the previous generation’s sparrow positions for updates. In the second phase, Cauchy mutation is applied to perturb individuals in the sparrow positions, thereby expanding the search scope of the SSA and enhancing its ability to escape local optima.

3.1 Preliminaries

We briefly introduce the basic framework of SSA, chaotic mapping, OOA, Cauchy mutation, and some basic concepts.

3.1.1 Sparrow search algorithm

Sparrows are typically gregarious birds. Captive house sparrows come in two different types, referred to as “producers” and “scroungers” (Barnard and Sibly, 1981). The producers actively search for sources of food, while the scroungers obtain food through the producers. Additionally, these birds are typically capable of flexibly employing behavioral strategies and switching between producing and scrounging (Liker and Barta, 2002). It can be said that, in order to find food, the sparrows often utilize both producer and scrounger strategies simultaneously (Barnard and Sibly, 1981; Xue and Shen, 2020).

Assuming there are N sparrows in a d -dimensional search space, the position of each sparrow can be represented by the following matrix Equation 18:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & \dots & x_{1,d} \\ x_{2,1} & x_{2,2} & \dots & \dots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & \dots & \dots & x_{N,d} \end{bmatrix} \quad (18)$$

The positions of sparrows in the search space are randomly initialized using Equation 19.

$$x_{i,j} = lb_j + r_{i,j} \cdot (ub_j - lb_j) \quad (19)$$

where $x_{i,j}$ represents the position of the i -th sparrow in the j -th dimension. $r_{i,j}$ is a random number in the interval $[0,1]$. lb_j and ub_j are the lower and upper bounds of the j -th dimension of the problem variables, respectively.

The fitness values of all sparrows can be represented by the following vector Equation 20:

$$\mathbf{F}_X = \begin{bmatrix} f([x_{1,1} \ x_{1,2} \ \dots \ x_{1,d}]) \\ f([x_{2,1} \ x_{2,2} \ \dots \ x_{2,d}]) \\ \vdots \\ f([x_{N,1} \ x_{N,2} \ \dots \ x_{N,d}]) \end{bmatrix} \quad (20)$$

The first stage is the exploration phase. The producers with better fitness values are given preference when it comes to acquiring food during the search process. Additionally, because producers take on the responsibility of food searching and guiding the entire population's movement, they have a wider search area compared to the scroungers. Moreover, when a sparrow detects a predator, it initiates an alarm by chirping. If the alarm value surpasses a predefined safety threshold, producers must lead all the scroungers to a safe zone. Throughout each iteration, the positions of producers are updated as follows Equation 21:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot \text{iter}_{\max}}\right) & \text{if } R_2 < ST \\ X_{i,j}^t + Q \cdot L & \text{if } R_2 \geq ST \end{cases} \quad (21)$$

where t represents the current iteration number, $j = 1, 2, \dots, d$. $X_{i,j}^{t+1}$ represents the value of the j -th dimension for the i -th sparrow at the t -th iteration. iter_{\max} signifies the maximum number of iterations. $\alpha \in (0,1]$ is a random number. Q is a random number following a normal distribution. L is a matrix of size $1 \times d$ in which every element is equal to 1. R_2 (where $R_2 \in [0,1]$) represents the alarm value. ST (where $ST \in [0.5,1]$) stands for the safety threshold.

If $R_2 < ST$, it signifies an absence of predators in the vicinity, prompting the producers to transition into an expansive search mode. However, when $R_2 \geq ST$, it signifies that certain sparrows have detected predators, necessitating a swift relocation of all sparrows to alternative safe areas.

The second phase is the development phase. Scroungers follow the producers who can offer the best food to search for nourishment. Meanwhile, some scroungers may continuously monitor the producers, and if they notice a producer has found

good food, they immediately leave their current location to compete for the food. If they succeed, they can acquire the producer's food immediately. The position update formula for the scroungers is as follows Equation 22:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{P}\right) & \text{if } i > N/2 \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \quad (22)$$

where X_p represents the optimal position occupied by the producer. X_{worst} represents the current global worst location. A is a $1 \times d$ matrix where each element is randomly set as 1 or -1 , and $A^+ = A^T(AA^T)^{-1}$. If $i > N/2$, it indicates that the i -th scrounger, with the worst fitness value, is highly likely to be in a starved state.

When they sense danger, sparrows located at the edge of the flock quickly move towards a safe area, while those in the middle of the flock move randomly to get closer to others. We assume that the sparrows aware of the danger constitute between 10% and 20% of the total population. The initial positions of these sparrows are randomly generated within the entire population and can be expressed using the following formula Equation 23:

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t + \beta \cdot |X_{i,j}^t - X_{\text{best}}^t| & \text{if } f_i > f_g \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{\text{worst}}^t|}{(f_i - f_w) + \epsilon}\right) & \text{if } f_i = f_g \end{cases} \quad (23)$$

where X_{best} is the current global optimal location. β , as a step size control parameter, is a random number following a normal distribution with a mean of 0 and a variance of 1. $K \in [-1,1]$ is a random number and denotes the direction in which the sparrow moves and is also the step size control coefficient. ϵ is a small constant to avoid division by zero errors. Here, f_i represents the fitness value of the current sparrow, and f_g and f_w are the current global best and worst fitness values, respectively.

If $f_i > f_g$, this signifies that the sparrow is positioned at the group's periphery. X_{best} denotes the location of the population center and is considered safe. If $f_i = f_g$, it implies that sparrows in the middle of the group have sensed danger and must approach the others.

3.1.2 Chaotic mapping

A chaotic matrix is a typical source of "ordered chaos," exhibiting unique characteristics of randomness and state transitivity. Under certain "rules," chaotic sequences traverse all different states within a defined range. Chaotic sequences generally possess several key features, including nonlinearity, sensitivity to initial conditions, transitivity, randomness, strange attractors (chaotic attractors), global stability and local instability, and long-term unpredictability.

In the context of intelligent optimization algorithms, random initialization of the population is often achieved using a uniform distribution. Compared to standard random search based on conventional probability distributions, the use of chaotic mappings in intelligent optimization algorithms can help populations escape local minima and enable faster iterative searches.

3.1.3 Osprey optimization algorithm

The OOA was proposed by Mohammad Dehghani and Pavel Trojovský in 2023(Dehghani and Trojovský, 2023), simulating the hunting behavior of ospreys.

The first phase is the exploration phase, involving the locating and capturing of fish. Ospreys, with their powerful vision, are formidable predators capable of spotting fish beneath the water’s surface. Once they’ve pinpointed a fish’s location, they dive underwater to attack and capture it. The initial stage of population update in OOA draws inspiration from this natural osprey behavior. Modeling how ospreys hunt fish results in substantial alterations to the ospreys’ positions within the search space. This, in turn, enhances OOA’s ability to explore and locate optimal regions while avoiding local optima.

Let’s assume there are N ospreys in a d -dimensional search space. For each osprey, the positions of other ospreys in the search space that have a better objective function value are considered underwater fish. The set of fish for each osprey is specified using Equation 24.

$$FP_i = \{X_k \mid k \in \{1, 2, \dots, N\} \wedge F_k < F_i\} \cup \{X_{best}\} \quad (24)$$

where FP_i is the set of fish positions for the i -th osprey and X_{best} is the best candidate solution.

The osprey employs a random process to detect the location of one of these fishes, and it initiates an attack. Through modeling the osprey’s movement as it approaches the fish, a new position is computed for the osprey by Equation 25 and Equation 26.

$$x_{ij}^{P1} = x_{ij} + r_{ij} \cdot (SF_{ij} - I_{ij} \cdot x_{ij}) \quad (25)$$

$$x_{ij}^{P1} = \begin{cases} x_{ij}^{P1}, lb_j \leq x_{ij}^{P1} \leq ub_j \\ lb_j, x_{ij}^{P1} < lb_j \\ ub_j, x_{ij}^{P1} > ub_j \end{cases} \quad (26)$$

This new position, if it results in a better objective function value, replaces the osprey’s previous position by Equation 27.

$$X_i = \begin{cases} X_i^{P1}, F_i^{P1} < F_i \\ X_i, \text{ else} \end{cases} \quad (27)$$

where x_{ij}^{P1} is the new position of the i -th osprey in the j -th dimension in the first phase, F_i^{P1} is its fitness value, and SF_{ij} is the fish chosen by the i -th osprey in the j -th dimension. r_{ij} is a random number within the range $[1,2]$, and I_{ij} is a random number chosen from the set $\{1,2\}$.

The second phase is known as the development stage. After successfully capturing a fish, the osprey relocates it to a secure and suitable spot for consumption. This modeling, involving the relocation of the fish, introduces minor adjustments to the osprey’s positions within the search space. Consequently, it enhances OOA’s capability for exploiting the local search and converging towards improved solutions around the identified ones.

In the OOA design, the emulation of osprey behavior involves initially determining a new random position for each individual in the population, akin to a “fish-eating spot.” This calculation is based

on Equation 28. Subsequently, if this new position results in an improved objective function value, it is employed to replace the previous position of the respective osprey according to Equation 29.

$$x_{ij}^{P2} = x_{ij} + \frac{lb_j + r \cdot (ub_j - lb_j)}{t} \quad (28)$$

$$x_{ij}^{P2} = \begin{cases} x_{ij}^{P2}, lb_j \leq x_{ij}^{P2} \leq ub_j \\ lb_j, x_{ij}^{P2} < lb_j \\ ub_j, x_{ij}^{P2} > ub_j \end{cases} \quad (29)$$

This new position, if it results in a better objective function value, replaces the osprey’s previous position by Equation 30.

$$X_i = \begin{cases} X_i^{P2}, F_i^{P2} < F_i \\ X_i, \text{ else} \end{cases} \quad (30)$$

where x_{ij}^{P2} is the new position of the i -th osprey in the j -th dimension in the second phase, F_i^{P2} is its fitness value, and SF_{ij} is the fish chosen by the i -th osprey in the j -th dimension. r is a random number within the range of $[1,2]$, t represents the current iteration count, and T is the maximum number of iterations.

3.1.4 Cauchy mutation

The Cauchy mutation is derived from the Cauchy distribution. The probability density function of the one-dimensional Cauchy distribution is given by Equation 31:

$$f(x) = \frac{1}{\pi} \cdot \frac{a}{a + x^2} \quad (31)$$

here, when $a = 1$, it is the standard Cauchy distribution.

The Cauchy distribution is similar to the standard normal distribution. It is a continuous probability distribution that has smaller values near the origin, is more elongated towards the ends, and approaches zero at a slower rate. Therefore, compared to the normal distribution, it can introduce larger disturbances. By utilizing Cauchy mutation for perturbing individuals in the sparrow position updates, the SSA’s search scope is expanded, leading to an improved ability to escape local optima.

3.2 Chaotic sparrow search algorithm combining osprey optimization algorithm and Cauchy mutation

Traditional SSA employs a random initialization method for the population, which can lead to premature convergence and slower convergence speed. To address this, this paper adopts a chaotic population initialization approach. This ensures randomness in the population while enhancing the algorithm’s convergence performance and diversifying the population. This helps prevent algorithm stagnation caused by a homogenous population.

The positions of sparrows in the search space are initialized using Piecewise chaotic mapping, as shown in Equation 32.

$$x_{i,j} = lb_j + chaos_{i,j} \cdot (ub_j - lb_j) \tag{32}$$

Where $chaos_{i,j}$ represents the chaotic mapping.

The first phase is the exploration phase. For producers, the Equation 25 of OOA’s global exploration strategy in the first phase replaces the original producer position update Equation 21 of the SSA. OOA aims to address the SSA’s overreliance on the update method based on the positions of the previous generation of sparrows.

The update method for the positions of producers in the sparrow algorithm is determined based on the simulation of the osprey’s movement toward fish. For each sparrow, the locations of other sparrows in the search space with superior fitness values are considered as food. Equation 33 is utilized to determine the set of superior food chosen by each sparrow.

$$FP_i = \{X_k | k \in \{1, 2, \dots, N\} \wedge F_k < F_i\} \cup \{X_{best}\} \tag{33}$$

where FP_i represents the food collection for the i -th sparrow, and X_{best} is the position of the best sparrow.

The sparrows randomly detect the position of one of the foods and go hunting. During each iteration, the positions of the producers are updated according to Equation 34. If the updated position is better, the sparrow’s previous position is replaced.

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t + r_{i,j} \cdot (SF_{i,j} - I_{i,j} \cdot X_{i,j}^t) & \text{if } R_2 < ST \\ X_{i,j}^t + Q \cdot L & \text{if } R_2 \geq ST \end{cases} \tag{34}$$

where $SF_{i,j}$ represents the food chosen by the i -th sparrow in the j -th dimension.

The second phase is the development stage. Scroungers often focus their search around the best discoverers. Food competition can also occur during this period, where a scrounger tries to become the producer. To prevent the algorithm from getting trapped in

local optima, a Cauchy mutation strategy is introduced into the equation for updating the scroungers. The updated scrounger position equation, replacing the original SSA’s scrounger position update equation, is as follows:

$$X_{i,j}^{t+1} = X_{best}(t) + cauchy(0, 1) \otimes X_{best}(t) \tag{35}$$

where X_{best} represents the current global best position. $cauchy(0,1)$ is the standard Cauchy distribution function, and \otimes denotes multiplication.

The sparrows that sense danger still undergo updates according to Equation 23.

The pseudocode for OCCSSA, which we have proposed, is presented in Algorithm 1.

4 Simulation and lake experiments results

4.1 Simulation validation and comparison

4.1.1 Comparison of the neural network equalizers

We used Bellhop Acoustic Toolbox to generate a time-varying UWA channel model to evaluate equalizers (Zhou et al., 2022). The parameters for the time-varying Bellhop channel simulator are listed in Table 1. We conducted simulations in three different underwater acoustic channel environments, with the different transmitter depth and seabed medium. The sound speed profile of 200 m is shown in Figure 4. The acoustic transmission loss obtained using the acoustic toolbox is shown in the Figures 5A, C, E. The maximum transmission loss is approximately 70 dB. The channel impulse response plot from the sound source to the receiving point is shown in the Figures 5B, D, F.

TABLE 1 Bellhop simulation parameters setup.

Parameter	CH1	CH2	CH3
Modulation type	QPSK	QPSK	QPSK
Sound source frequency	10 kHz	10 kHz	10 kHz
Sound pressure level	195 dB	195 dB	195 dB
Angle of sound wave emitted by transmitter point	-20° ~ + 20°	-20° ~ + 20°	-20° ~ + 20°
Sea water depth	200 m	200 m	200 m
Transmitter depth	80 m	80 m	150 m
Receiver depth	40 m	40 m	40 m
Distance from receiving point to transmitter point	3 km	3 km	3 km
P-wave speed of sound in bottom	1511.96 m/s	1511.96 m/s	1511.96 m/s
Density at the cutoff depth of the seabed medium	1.421 g/cm ³	2.034 g/cm ³	1.421 g/cm ³
Attenuation coefficient of the seabed medium	0.078 dB/wavelength	0.479 dB/wavelength	0.078 dB/wavelength

Input: Max_iter : maximum iteration; PD : the number of producers; SD : the number of sparrows who perceive the danger; R_2 : the alarm value; n : the population size.

Output: X_{best} : the current global best position.

1: **Initialize:** Using Equation 32 to initialize the population

2: **while** $iter < Max_iter$ **do**

3: Sort the fitness values to find the current best and worst individuals.

4: $R_2 = rand(1)$

5: **for** $i = 1 : PD$ **do**

6: Find a better food location using Equation 33.

7: Producers randomly select a food and update their positions using Equation 34.

8: **end for**

9: Find the optimal population and record as X_{best} .

10: **for** $i = (PD + 1) : n$ **do**

11: Scroungers use the Cauchy mutation strategy to update their positions using Equation 35.

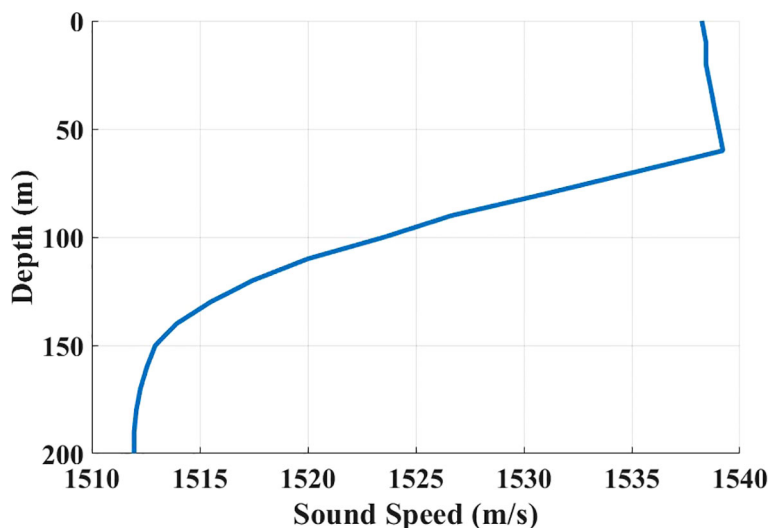


FIGURE 4 The sound speed profile of 200 m.

```

12: end for
13: for i = 1: SD do
14: Sparrows partially aware of danger update their
    positions using Equation 23.
15: end for
16: Obtain the global best value  $X_{best}$ .
17: iter = iter + 1
18: end while
19: return  $X_{best}$ 
    
```

Algorithm 1. The pseudocode of OCCSSA.

We will compare equalizers based on MLP (Gibson et al., 1989), CFLNN (Patra et al., 2005), DFCFLNN (Zhao and Zhang, 2008) and traditional DFE with phase-locked loop (PLL) (Stojanovic et al., 1994).

CFLNN has 10 feedforward taps, and its input signal is represented as $\mathbf{X}_k = [r_{k,i}, r_{k,q}, r_{k+1,i}, r_{k+1,q}, \dots, r_{k+9,i}, r_{k+9,q}]^T$. Here, r_k , i represents the I-path of the k-th signal, and $r_{k,q}$ represents the Qpath of the k-th signal. The input signal is transformed into 121 dimensions through a six-order Chebyshev transformation. Therefore, the input layer has 121 nodes, and the output layer has 2 nodes. In DF-CFLNN, the tap coefficient of feedback is 2, and the input signal consists of $\mathbf{X}_k = [r_{k,i}, r_{k,q}, r_{k+1,i}, r_{k+1,q}, \dots, r_{k+9,i}, r_{k+9,q}]^T$ and the feedback signal $\mathbf{S}_k = [s_{k,i}, s_{k,q}, s_{k-1,i}, s_{k-1,q}]$ from the decision device. The input signal is transformed into 97 dimensions through a four order Chebyshev transformation. Therefore, the input layer has 97 nodes, and the output layer has 2 nodes. In SDF-CFLNN, the tap coefficient of feedback is 2, and the input signal $\mathbf{X}_k = [r_{k,i}, r_{k,q}, r_{k+1,i}, r_{k+1,q}, \dots, r_{k+9,i}, r_{k+9,q}]^T$ is transformed into 41 dimensions through a two-order Chebyshev transformation. The input layer has 45 nodes, including 41 nodes for input signals and 4 nodes for feedback signals. The parameters for each equalizer are as shown in Table 2.

To enhance the reliability, we conducted 30 independent trials on each equalizer. In our setup, we use 10,000 QPSK signals as input data, with 80% serving as training data and 20% as testing data. In

each trial, the maximum iteration count was set to 1000. Since this paper focuses on the learning and equalization capabilities of neural networks, in our simulations, we assumed perfect time sequence recovery.

The simulated BER graph is shown in the Figures 6A, C, E. It can be observed that our proposed SDFCFLNNE exhibits the best BER performance in two different underwater acoustic environments, followed by DF-CFLNNE, traditional DFE-PLL, and CFLNNE in descending order. In complex underwater environments, the MLP equalizer performs poorly and is the least effective. The MSE iteration curve at SNR=10 dB is shown in the Figures 6B, D, F. SDF-CFLNNE converges the fastest, with minimal initial oscillations, and exhibits smooth and stable convergence. It also has the smallest MSE value when reaching a steady state. In CH1, SDF-CFLNNE reaches convergence in about 20 iterations, while DFCFLNNE and CFLNNE reach convergence around 300 iterations with minor oscillations. MLP achieves basic convergence in approximately 70 iterations but experiences significant oscillations. In CH2, SDFCFLNNE reaches convergence in about 40 iterations, while DF-CFLNNE, CFLNNE, and MLP all converge around 300 iterations with minor oscillations.

4.1.2 Comparison of the swarm intelligence optimization algorithms

4.1.2.1 Benchmark test functions

Benchmark test functions are typically utilized to evaluate the performance of optimization algorithms. We utilized the CEC2005 benchmark test functions as provided in Table 3 (Suganthan et al., 2005) to assess the applicability and effectiveness of the proposed OCCSSA algorithm.

4.1.2.2 Comparison of chaotic mapping methods

In order to select the more effective chaotic mapping, we initialized the population of OCCSSA using ten different chaotic

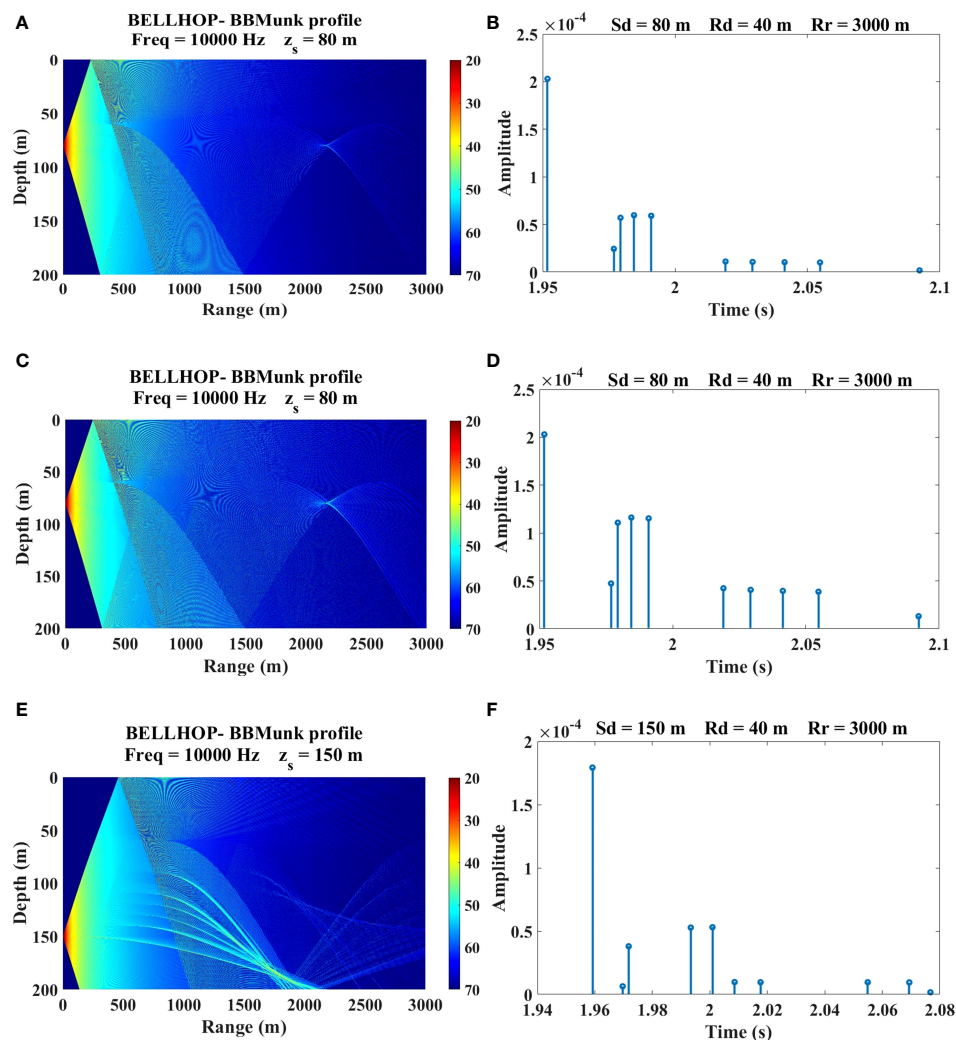


FIGURE 5 The channel information. (A) The transmission loss at CH1. (B) the impulse response at CH1. (C) The transmission loss at CH2. (D) the impulse response at CH2. (E) The transmission loss at CH3. (F) the impulse response at CH3.

mappings, namely Tent, Logistic, Cubic, Chebyshev, Piecewise, Sinusoidal, Sine, ICMIC, Circle, and Bernoulli.

The experimental test environment is as follows: 12th Gen Intel (R) Core(TM) i7-12700H CPU with a base frequency of 2.70 GHz and 16.0 GB of RAM. The operating system used is Windows 11,

and the integrated development environment (IDE) is Matlab 2021a.

To increase the credibility of the algorithm, we conducted 30 independent trials on each test function. The maximum iteration count was set to 500, and the population size was 50. The

TABLE 2 Network parameter configuration.

Parameter	DFE-PLL	MLP	CFLNN	DF-CFLNN	SDF-CFLNN
Tap coefficient of feedforward	100	10	10	10	10
Tap coefficient of feedback	50	-	2	2	2
Order of Chebyshev polynomials	-	-	6	4	2
Number of nodes in the input layer	-	20	121	97	45
Number of nodes in the hidden layer	-	41	-	-	-
Number of nodes in the output layer	-	4	4	4	4
Step size	-	0.005	0.025	0.025	0.025

TABLE 3 Details of the CEC2005 benchmark suite.

Classification	Function	Dim	[Xmin, Xmax]	Fmin
Unimodal benchmark functions	$F_1(x) = \sum_{i=1}^n x_i^2$	30	[100,100]	0
	$F_2(x) = \sum_{i=1}^n x_i^2 + \prod_{i=1}^n x_i $	30	[100,100]	0
	$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	30	[100,100]	0
	$F_4(x) = \max_i \{ x_i \}, 1 \leq i \leq n$	30	[100,100]	0
	$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[30,30]	0
	$F_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[100,100]	0
	$F_7(x) = \sum_{i=1}^n ix_i^4 + random[0, 1)$	30	[1.28,1.28]	0
Multimodal benchmark functions	$F_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[500,500]	418.9829 × dim
	$F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[5.12,5.12]	0
	$F_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{2} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	[32,32]	0
	$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[600,600]	0
	$F_{12}(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^n (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i)] + (y_n - 1)^2 \right\}$ $y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	[50,50]	0
	$F_{13} = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_1 + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\}$ $+ \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	[50,50]	0
Fixed-Dimension multimodal benchmark functions	$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	[65,65]	1
	$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_1^2 + b_1 x_2)}{b_1^2 + b_1 x_3 + x_4} \right]^2$	4	[5,5]	0.00030
	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[5,5]	1.0316
	$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[5,5]	0.398
	$F_{18}(x) = [1 + (x_1 + x_2 + x_3)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
	$F_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^4 a_{ij}(x_j - p_{ij})^2\right)$	3	[0,1]	3.86

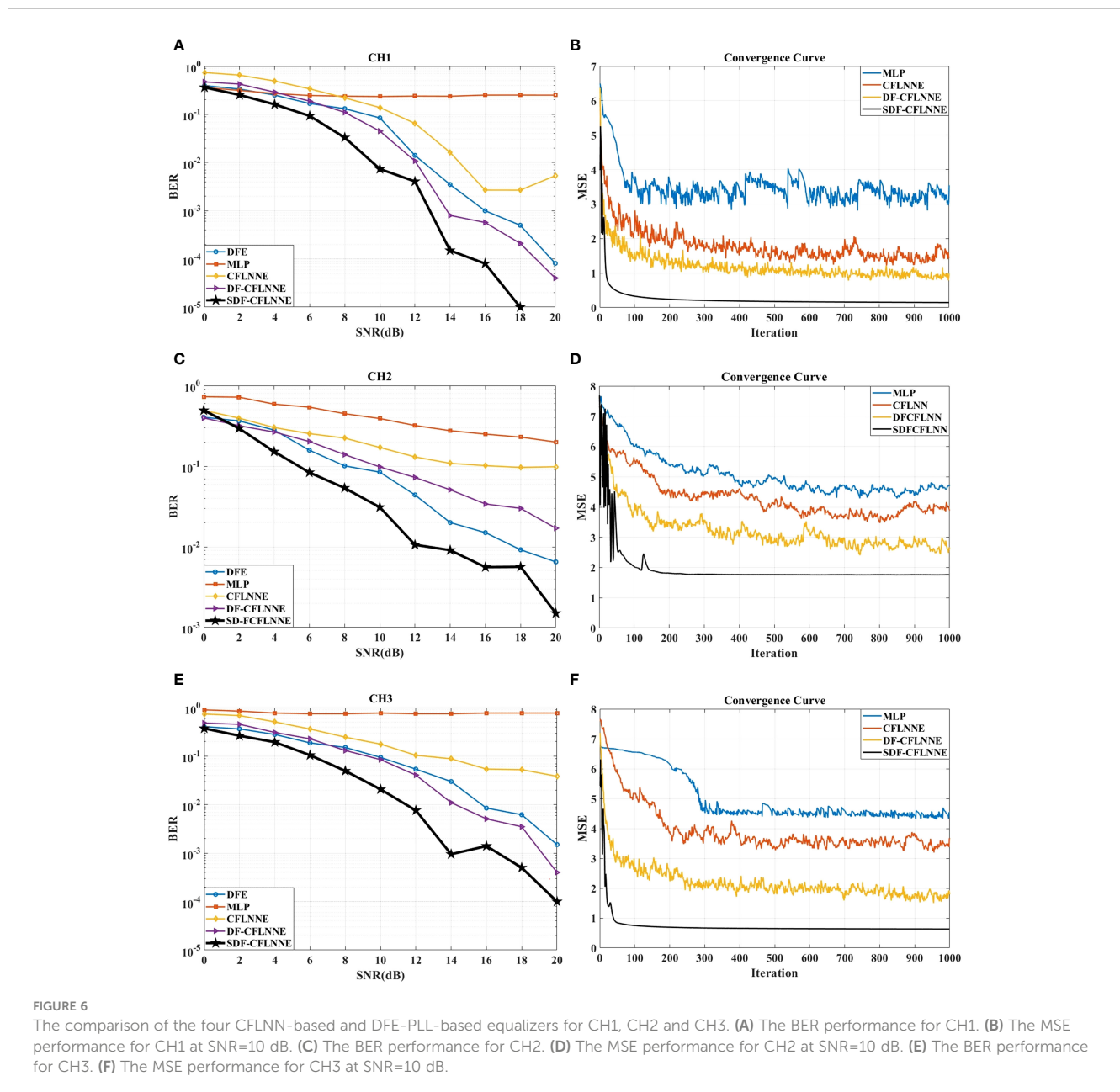
(Continued)

TABLE 3 Continued

Classification	Function	Dim	[Xmin, Xmax]	Fmin
	$F_{20}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right)$	6	[0,1]	3.32
	$F_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	10.1532

experimental results indicate that overall, the Piecewise chaotic mapping exhibited superior convergence speed and accuracy. We selected the results for F15 and F21, where the convergence effects are more pronounced, for illustration, as shown in Figure 7. Therefore, we chose the Piecewise mapping as the method for random population initialization. The expression for the Piecewise mapping is as shown in Equation 36.

$$x(t+1) = \begin{cases} \frac{x(t)}{p}, & 0 \leq x(t) < p \\ \frac{x(t)-p}{0.5-p}, & p \leq x(t) < 0.5 \\ \frac{1-p-x(t)}{0.5-p}, & 0.5 \leq x(t) < 1-p \\ \frac{1-x(t)}{p}, & 1-p \leq x(t) < 1 \end{cases} \quad (36)$$



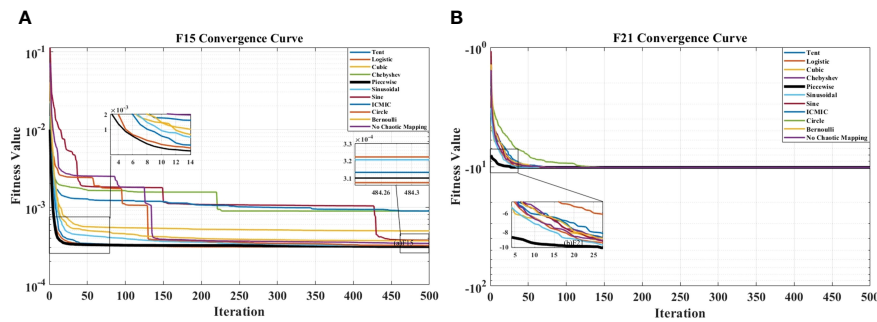


FIGURE 7 The convergence curves of initializing the OCCSSA population with ten different chaotic mappings: (A) F15. (B) F21.

where $p = 0.4$, $x(1) = rand$.

4.1.2.3 Comparison with the improved SSA-type optimization algorithms

Precision and convergence speed are important indicators for measuring the quality of an algorithm. In order to better validate the effectiveness of the proposed algorithm, this section assesses performance metrics, including convergence precision, using CEC2005 benchmark test functions. This section reproduces SSA (Xue and Shen, 2020), ISSA (Song et al., 2020), and the Adaptive Spiral Flying Sparrow Search Algorithm (ASFSSA) (Ouyang et al., 2021) to compare their performance against the proposed optimization algorithm.

To enhance the algorithm’s reliability, we conducted 30 independent trials on each test function. In each trial, the maximum iteration count was set to 500, and the population size was 50. The safety threshold was set to 0.8, and the number of producers and sparrows sensing danger was both set to 20% of the population size. Tables 4–6 provide the optimization indicators for each algorithm during the optimization process, including the average objective function value, standard deviation, median, best value, average runtime, and algorithm ranking. Under the same standard test functions, the average represents the convergence accuracy of the algorithm, while the standard deviation reflects its stability. The algorithm ranking in this paper is based on both the average and standard deviation, where smaller values indicate better

TABLE 4 The comparative data for the improved SSA-type optimization algorithms on the CEC2005 multi-modal functions.

Function Name	Algorithm Name	Best Value	Worst Value	Average Value	StandDP	Median Value	Average Running Time	Rank
F8	SSA	-3952.9522	-2788.2072	-3355.182	300.0432	-3338.7384	0.28386	4
	ASFSSA	-4189.8289	-3081.3567	-3601.1772	276.0404	-3594.6031	0.65211	2
	ISSA	-4189.8289	-2641.0282	-3190.4933	322.1179	-3151.9709	0.56114	3
	OCCSSA	-4189.8289	-4189.8289	-4189.8289	1.8807e-12	-4189.8289	0.36753	1
F9	SSA	0	0	0	0	0	0.2154	4
	ASFSSA	0	0	0	0	0	0.57664	1
	ISSA	0	0	0	0	0	0.54976	3
	OCCSSA	0	0	0	0	0	0.36489	2
F10	SSA	4.4409e-16	4.4409e-16	4.4409e-16	0	4.4409e-16	0.23346	4
	ASFSSA	4.4409e-16	4.4409e-16	4.4409e-16	0	4.4409e-16	0.61848	1
	ISSA	4.4409e-16	4.4409e-16	4.4409e-16	0	4.4409e-16	0.54772	3
	OCCSSA	4.4409e-16	4.4409e-16	4.4409e-16	0	4.4409e-16	0.37073	2
F11	SSA	0	0	0	0	0	0.36387	4
	ASFSSA	0	0	0	0	0	0.66729	1
	ISSA	0	0	0	0	0	0.6481	3
	OCCSSA	0	0	0	0	0	0.453	2

(Continued)

TABLE 4 Continued

Function Name	Algorithm Name	Best Value	Worst Value	Average Value	StandDP	Median Value	Average Running Time	Rank
F12	SSA	4.7116e-32	1.2762e-28	7.7739e-30	2.3715e-29	3.8534e-31	0.43226	2
	ASFSSA	4.7864e-31	9.1663e-27	7.9713e-28	2.0578e-27	4.4766e-29	1.1384	4
	ISSA	4.7116e-32	4.9052e-32	4.7649e-32	6.1095e-34	4.7237e-32	1.0566	1
	OCCSSA	5.2199e-32	2.0845e-27	1.102e-28	3.8796e-28	3.7765e-30	0.76728	3
F13	SSA	1.473e-32	1.4984e-26	9.0349e-28	2.7677e-27	3.5795e-29	0.66066	3
	ASFSSA	2.0893e-32	8.6921e-27	7.6682e-28	2.13e-27	2.6357e-29	1.1247	2
	ISSA	1.3128e-30	0.010987	0.0018312	0.0041648	4.867e-28	1.0698	4
	OCCSSA	1.3498e-32	9.19e-28	3.9444e-29	1.6717e-28	1.3786e-30	0.79072	1
F14	SSA	0.998	12.6705	5.868	5.5657	2.9821	1.8161	4
	ASFSSA	0.998	10.7632	3.9379	3.9286	2.9821	2.4586	3
	ISSA	0.998	12.6705	1.7179	2.1848	0.998	1.9249	2
	OCCSSA	0.998	0.998	0.998	2.9156e-16	0.998	1.5888	1
F15	SSA	3.0749e-4	3.5402e-4	3.0904e-4	8.4967e-06	3.0749e-4	0.079858	2
	ASFSSA	3.0749e-4	0.0012232	4.3684e-4	2.5835e-4	3.0749e-4	0.19687	3
	ISSA	3.0749e-4	0.0015941	6.61e-4	4.7977e-4	3.764e-4	0.13209	4
	OCCSSA	3.0749e-4	3.075e-4	3.0749e-4	3.599e-09	3.0749e-4	0.1641	1
F16	SSA	-1.0316	-1.0316	-1.0316	6.3208e-16	-1.0316	0.083546	3
	ASFSSA	-1.0316	-1.0316	-1.0316	5.2964e-16	-1.0316	0.5866	2
	ISSA	-1.0316	-1.0316	-1.0316	6.5843e-16	-1.0316	0.31525	4
	OCCSSA	-1.0316	-1.0316	-1.0316	4.9651e-16	-1.0316	0.38597	1
F17	SSA	0.39789	0.39789	0.39789	0	0.39789	0.2027	2
	ASFSSA	0.39789	0.39789	0.39789	0	0.39789	0.53944	3
	ISSA	0.39789	0.39789	0.39789	0	0.39789	0.29137	1
	OCCSSA	0.39789	0.39789	0.39789	0	0.39789	0.41833	4
F18	SSA	3	30	3.9	4.9295	3	0.22676	2
	ASFSSA	3	3	3	1.2176e-15	3	0.61142	1
	ISSA	3	30	3.9	4.9295	3	0.32566	3
	OCCSSA	3	30	7.5	10.2343	3	0.41765	4
F19	SSA	-3.8628	-3.8628	-3.8628	2.5391e-15	-3.8628	0.28393	3
	ASFSSA	-3.8628	-3.8628	-3.8628	2.3557e-15	-3.8628	0.70925	2
	ISSA	-3.8628	-3.8549	-3.8625	0.001439	-3.8628	0.45951	4
	OCCSSA	-3.8628	-3.8628	-3.8628	2.1787e-15	-3.8628	0.504	1
F20	SSA	-3.322	-3.2031	-3.2507	0.059241	-3.2031	0.27128	3
	ASFSSA	-3.322	-3.2031	-3.2744	0.059241	-3.322	0.68092	2
	ISSA	-3.322	-3.2031	-3.2784	0.058273	-3.322	0.48302	4
	OCCSSA	-3.322	-3.2031	-3.3141	0.030164	-3.322	0.47212	1
F21	SSA	-10.1532	-5.0552	-9.6434	1.5555	-10.1532	0.26862	2
	ASFSSA	-10.1532	-5.0552	-8.4682	2.3059	-10.0281	0.55033	4

(Continued)

TABLE 4 Continued

Function Name	Algorithm Name	Best Value	Worst Value	Average Value	StandDP	Median Value	Average Running Time	Rank
	ISSA	-10.1532	-5.0552	-8.6174	2.372	-10.1511	0.33206	3
	OCCSSA	-10.1532	-10.1532	-10.1532	5.1842e-15	-10.1532	0.48869	1

TABLE 5 The comparative data for the swarm intelligence optimization algorithms on the CEC2005 multimodal functions.

Function Name	Algorithm Name	Best Value	Worst Value	Average Value	StandDP	Median Value	Average Running Time	Rank
F8	PSO	-8.7193e+3	-3.9441e+3	-6.6761e+3	1.0476e+3	-6.7351e+3	0.097544	8
	GWO	-7.9899e+3	-5.0501e+3	-6.4564e+3	6.2055e+3	-6.5501e+3	0.21649	9
	WOA	-1.2568e+4	8.5968e+3	1.1133e+4	1.4966e+3	-1.1671e+4	0.085122	4
	BES	-9.8392e+3	-4.8897e+3	-6.8080e+3	1.5443e+3	-6.3557e+3	1.7086	7
	CSA	-1.2567e+4	-5.7760e+3	-8.0776e+3	2.6827e+3	-6.3615e+3	0.082586	6
	GTO	-1.2569e+4	-1.2569e+4	-1.2569e+4	6.8563e-07	-1.2569e+4	0.35746	1
	WSO	-4.4925e+3	-3.0767e+3	-3.6360e+3	3.6998e+3	-3.5819e+3	0.068351	10
	DBO	-1.2352 + 4	-6.8907e+3	-9.3852e+3	1.3641e+3	-9.3650e+3	0.076335	5
	OOA	-1.2569e+4	-9.0163e+3	-1.1266e+4	1.7415e+3	-1.2569e+4	0.17756	3
	OCCSSA	-1.2569e+4	-1.2451e+4	-1.2569e+4	21.6229	-1.2569e+4	0.49263	2
F9	PSO	101.3281	218.3835	152.4089	26.5922	151.7649	0.075587	9
	GWO	0	8.9196	1.8651	2.6026	4.2633e-13	0.18007	6
	WOA	0	0	0	0	0	0.051678	4
	BES	0	0	0	0	0	1.3967	5
	CSA	7.9697	47.8554	27.9259	11.4001	25.4951	0.054526	8
	GTO	0	0	0	0	0	0.29075	3
	WSO	15.7193	146.5208	39.0134	27.0612	29.6164	0.068292	10
	DBO	0	25.8689	2.1348	5.4652	0	0.061137	7
	OOA	0	0	0	0	0	0.093629	2
	OCCSSA	0	0	0	0	0	0.41125	1
F10	PSO	1.318	3.306	2.3647	0.46286	2.3992	0.077702	9
	GWO	3.9524e-14	6.4393e-14	4.4142e-14	5.8377e-15	4.3077e-14	0.17897	7
	WOA	4.4409e-16	7.5495e-15	3.8784e-15	2.5523e-15	3.9968e-15	0.055302	6
	BES	4.4409e-16	4.4409e-16	4.4409e-16	0	4.4409e-16	1.4385	1
	CSA	19.9668	19.9668	19.9668	4.9591e-12	19.9668	0.057118	8
	GTO	4.4409e-16	4.4409e-16	4.4409e-16	0	4.4409e-16	0.29324	4
	WSO	2.9679	5.4777	4.0622	0.65712	4.0067	0.072419	10
	OOA	4.4409e-16	4.4409e-16	4.4409e-16	0	4.4409e-16	0.10506	3
	DBO	4.4409e-16	3.9968e-15	5.6251e-16	6.4863e-16	4.4409e-16	0.06724	5
	OCCSSA	4.4409e-16	4.4409e-16	4.4409e-16	0	4.4409e-16	0.42693	2
F11	PSO	0.033408	0.16374	0.093846	0.033652	0.091152	0.13796	8
	GWO	0	0.027649	0.0036764	0.0076973	0	0.23287	6

(Continued)

TABLE 5 Continued

Function Name	Algorithm Name	Best Value	Worst Value	Average Value	StandDP	Median Value	Average Running Time	Rank
	WOA	0	0.13137	0.0043788	0.023984	0	0.10538	7
	BES	0	0	0	0	0	1.0244	4
	CSA	1.0139	1.7255	1.1649	0.14593	1.1152	0.10955	9
	GTO	0	0	0	0	0	0.3944	3
	WSO	1.1382	2.3146	1.5244	0.25651	1.4776	0.12313	10
	DBO	0	0	0	0	0	0.072711	5
	OOA	0	0	0	0	0	0.19774	2
	OCCSSA	0	0	0	0	0	0.51511	1
F12	PSO	0.0030888	0.12305	0.037425	0.030208	0.028614	0.4018	8
	GWO	0.013497	0.048772	0.02788	0.0086555	0.02709	0.52616	7
	WOA	0.0013778	0.11719	0.0092898	0.020855	0.0041805	0.40458	6
	BES	1.5705e-32	3.0953e-32	1.669e-32	3.7477e-33	1.5705e-32	2.6525	1
	CSA	1.8485	11.4071	3.9982	1.784	3.6541	0.39943	10
	GTO	2.024e-13	7.5911e-09	7.1513e-10	1.5397e-09	1.1799e-10	0.98566	4
	WSO	0.19453	2.8473	1.1124	0.63026	0.94623	0.40955	9
	DBO	1.0548e-11	1.2499e-07	4.8707e-09	2.2709e-08	2.9129e-10	0.26447	5
	OOA	1.5705e-32	1.0317e-11	4.11e-13	1.8928e-12	1.5705e-32	0.84577	3
	OCCSSA	1.5705e-32	1.9101e-27	1.1224e-28	3.9124e-28	4.7087e-32	1.0582	2
F13	PSO	0.11726	0.69786	0.35562	0.17123	0.33052	0.41634	7
	GWO	0.098496	0.73036	0.38326	0.17023	0.38972	0.5469	6
	WOA	0.047162	0.69122	0.21665	0.17243	0.16514	0.40542	5
	BES	0.097371	2.9661	2.6729	0.79559	2.9661	2.6043	8
	CSA	2.3803	71.6642	33.0108	23.2431	37.6155	0.41566	10
	GTO	1.1212e-12	0.010987	0.0011045	0.0033508	4.046e-09	0.99464	3
	WSO	3.5379	35.3039	15.231	8.2623	15.3338	0.41094	9
	DBO	1.7766e-09	0.29615	0.076464	0.078188	0.092316	0.26475	4
	OOA	1.3498e-32	7.8895e-10	2.6368e-11	1.4403e-10	1.733e-29	0.79109	2
	OCCSSA	1.3498e-32	1.6778e-26	1.047e-27	3.4861e-27	1.2296e-30	0.9622	1

algorithm performance and higher rankings. In cases where the average and standard deviation are the same, the comparison is based on the convergence speed in the convergence curves. When the convergence speeds are similar, the average runtime is considered. The optimal values, algorithms with a ranking of 1, and the shortest average running time among all compared algorithms are highlighted in bold.

The data results for the CEC2005 tests are presented in Table 4. Since the optimization of neural network weights involves multi-modal functions, we utilized the second-class functions (F8-F13) and third-class functions (F14-F21) from the CEC2005 benchmark test functions. In the case of CEC2005 multi-modal functions, despite these functions having multiple local optima, the proposed algorithm was able to successfully solve the

optimization problems. OCCSSA demonstrated the overall best performance, especially in F8, F13-F15, and F19-F21, where it achieved the best values for each indicator, ranking first. For F9-F11, ASFSSA performed the best, with OCCSSA ranking second. The runtime falls within a moderate range. From the convergence curve plots in Figure 8, it is evident that OCCSSA exhibited superior convergence speed and accuracy in F8, F14-F17, and F19-F21 compared to other algorithms. However, in the case of F18, OCCSSA performed poorly, with lower convergence accuracy than other algorithms. Overall, OCCSSA demonstrated good convergence speed and accuracy, as well as strong resistance to local optima in multi-modal functions. The introduction of multiple strategies significantly improved the algorithm's stability and search capabilities.

TABLE 6 The comparative data for the swarm intelligence optimization algorithms on the CEC2005 fixed dimension multi-modal functions.

Function Name	Algorithm Name	Best Value	Worst Value	Average Value	StandDP	Median Value	Average Running Time	Rank
F14	PSO	0.998	6.9033	1.8561	1.363	1.992	0.92824	7
	GWO	0.998	12.6705	4.4237	4.2869	2.9821	1.0034	9
	WOA	0.998	10.7632	2.2114	2.471	0.998	0.95681	8
	BES	0.998	12.6705	3.208	4.3568	0.998	4.1981	10
	CSA	0.998	1.992	1.0311	0.18148	0.998	0.96765	4
	GTO	0.998	0.998	0.998	0	0.998	2.0796	2
	WSO	0.998	0.998	0.998	2.4089e-13	0.998	0.91439	3
	DBO	0.998	2.9821	1.0641	0.36225	0.998	0.39187	5
	OOA	0.998	2.0263	1.1648	0.37943	0.998	1.982	6
	OCCSSA	0.998	0.998	0.998	2.9733e-16	0.998	1.7201	1
F15	PSO	5.5339e-4	1.0887e-3	8.9695e-4	1.466e-4	8.9653e-4	1.2332e-2	4
	GWO	3.075e-4	2.0363e-2	5.0635e-3	8.5883e-3	3.1016e-4	0.03277	10
	WOA	3.0888e-4	1.5269e-3	6.5179e-4	3.8321e-4	5.3985e-4	0.01804	9
	BES	3.0749e-4	1.2232e-3	3.3801e-4	1.671e-4	3.0749e-4	0.95058	5
	CSA	3.0749e-4	1.2232e-3	5.1769 e-4	3.5264e-4	3.0749e-4	0.035304	7
	GTO	3.0749e-4	1.2232e-3	3.6853e-4	2.3232e-4	3.0749e-4	0.2191	6
	WSO	3.0749e-4	3.0749e-4	3.0749 e-4	2.4691e-8	3.0749	0.03571	2
	DBO	3.0749e-4	1.4887e-3	7.3946e-4	3.5861e-4	7.8266e-4	0.060859	8
	OOA	3.0987e-4	8.2964e-4	4.4708e-4	1.3792e-4	3.870e-4	0.1038	3
	OCCSSA	3.0749e-4	3.0749e-4	3.0749e-4	2.8651e-08	3.0749e-4	0.38096	1
F16	PSO	-1.0316	-1.0316	-1.0316	5.1334e-16	-1.0316	0.020588	2
	GWO	-1.0316	-1.0316	-1.0316	1.5011e-08	-1.0316	0.055229	8
	WOA	-1.0316	-1.0316	-1.0316	5.1596e-10	-1.0316	0.045389	7
	BES	-1.0316	-1.0316	-1.0316	4.1946e-11	-1.0316	1.3678	6
	CSA	-1.0316	-1.0316	-1.0316	6.5195e-16	-1.0316	0.038116	5
	GTO	-1.0316	-1.0316	-1.0316	6.4539e-16	-1.0316	0.21813	4
	WSO	-1.0316	-1.0316	-1.0316	1.4226e-06	-1.0316	0.036043	9
	DBO	-1.0316	-1.0316	-1.0316	6.3877e-16	-1.0316	0.056909	3
	OOA	-1.0316	-1.0312	-1.0316	8.2782e-05	-1.0316	0.10267	10
	OCCSSA	-1.0316	-1.0316	-1.0316	4.8787e-16	-1.0316	0.40119	1
F17	PSO	0.39789	0.39789	0.39789	0	0.39789	0.0087385	4
	GWO	0.39789	0.3979	0.39789	1.8026e-06	0.39789	0.039837	6
	WOA	0.39789	0.3979	0.39789	3.0404e-06	0.39789	0.034851	7
	BES	0.39789	0.3979	0.39789	3.2532e-06	0.39789	1.3276	8
	CSA	0.39789	0.39789	0.39789	0	0.39789	0.027288	5
	GTO	0.39789	0.39789	0.39789	0	0.39789	0.20295	2
	WSO	0.39789	0.39793	0.39789	7.3998e-06	0.39789	0.027323	9
	DBO	0.39789	0.39789	0.39789	0	0.39789	0.049667	1

(Continued)

TABLE 6 Continued

Function Name	Algorithm Name	Best Value	Worst Value	Average Value	StandDP	Median Value	Average Running Time	Rank
	OOA	0.39789	0.40128	0.39801	6.1782e-04	0.39789	0.086509	10
	OCCSSA	0.39789	0.39789	0.39789	0	0.39789	0.39572	3
F18	PSO	3	3	3	4.7063e-15	3	0.0080774	2
	GWO	3	3.0001	3	1.6003e-05	3	0.038497	5
	WOA	0.0013778	0.11719	0.0092898	0.020855	0.0041805	0.40458	10
	BES	3	3	3	1.3143e-15	3	1.3119	6
	CSA	3	3	3	1.2315e-15	3	0.027287	7
	GTO	3	3	3	7.8233e-16	3	0.19681	1
	WSO	3	3	3	1.7992e-15	3	0.024549	4
	DBO	3	3	3	1.1186e-15	3	0.046369	8
	OOA	3	3.0674	3.0023	0.012296	3	0.079621	9
	OCCSSA	3	3	3	3.4586e-15	3	0.37656	3
	F19	PSO	-3.8628	-3.8628	-3.8628	2.3646e-15	-3.8628	0.024082
GWO		-3.8628	-3.8552	-3.8618	0.0022799	-3.8627	0.060494	8
WOA		-3.8628	-3.8504	-3.8609	0.0032281	-3.8622	0.052052	9
BES		-3.8628	-3.8628	-3.8628	2.7101e-15	-3.8628	1.4188	6
CSA		-3.8628	-3.8628	-3.8628	2.6543e-15	-3.8628	0.043274	4
GTO		-3.8628	-3.8628	-3.8628	2.6823e-15	-3.8628	0.22995	5
WSO		-3.8628	-3.8628	-3.8628	2.7101e-15	-3.8628	0.040928	7
DBO		-3.8628	-3.8628	-3.8628	2.6543e-15	-3.8628	0.059415	3
OOA		-3.8628	-3.679	-3.8038	0.046934	-3.8076	0.11594	10
OCCSSA		-3.8628	-3.8628	-3.8628	2.2629e-15	-3.8628	0.42646	1
F20	PSO	-3.322	-3.2031	-3.2705	0.059923	-3.322	0.030683	3
	GWO	-3.322	-3.1345	-3.2654	0.072981	3.322	0.073937	5
	WOA	-3.322	-2.8401	-3.2149	0.10919	-3.1884	0.047003	9
	BES	3.322	-3.2031	-3.2586	0.060328	-3.2031	1.443	6
	CSA	-3.322	-3.2014	-3.2585	0.060388	-3.2031	0.039499	7
	GTO	-3.322	-3.2031	-3.2705	0.059923	-3.322	0.24303	4
	WSO	-3.322	-3.2031	-3.318	0.021707	-3.322	0.044449	2
	DBO	-3.322	-3.0839	-3.2459	0.08448	-3.2625	0.061138	8
	OOA	-3.2121	-1.8054	-2.6624	0.29428	-2.6389	0.11226	10
	OCCSSA	-3.322	-3.322	-3.322	1.6739e-15	-3.322	0.4264	1
F21	PSO	-10.1532	-2.6305	-7.6382	3.0198	-10.1532	0.031031	10
	GWO	-10.153	-5.0552	-9.3055	1.9253	-10.1518	0.077393	5
	WOA	-10.1531	-5.0551	-8.9603	2.1912	-10.147	0.061021	7
	BES	-10.1532	-5.0552	-7.6028	2.5911	-7.5825	1.4532	8
	CSA	-10.1532	-5.1008	-9.1427	2.0555	-10.1532	0.050115	6
	GTO	-10.1532	-10.1532	-10.1532	6.3278e-15	-10.1532	0.24091	3

(Continued)

TABLE 6 Continued

Function Name	Algorithm Name	Best Value	Worst Value	Average Value	StandDP	Median Value	Average Running Time	Rank
	WSO	-10.1532	-10.1532	-10.1532	5.0459e-15	-10.1532	0.049681	1
	DBO	-10.1532	-2.6305	-6.7838	2.675	-5.1008	0.065969	9
	OOA	-10.1532	-10.1532	-10.1532	3.0155e-07	-10.1532	0.12924	4
	OCCSSA	-10.1532	-10.1532	-10.1532	5.3086e-15	-10.1532	0.41797	2

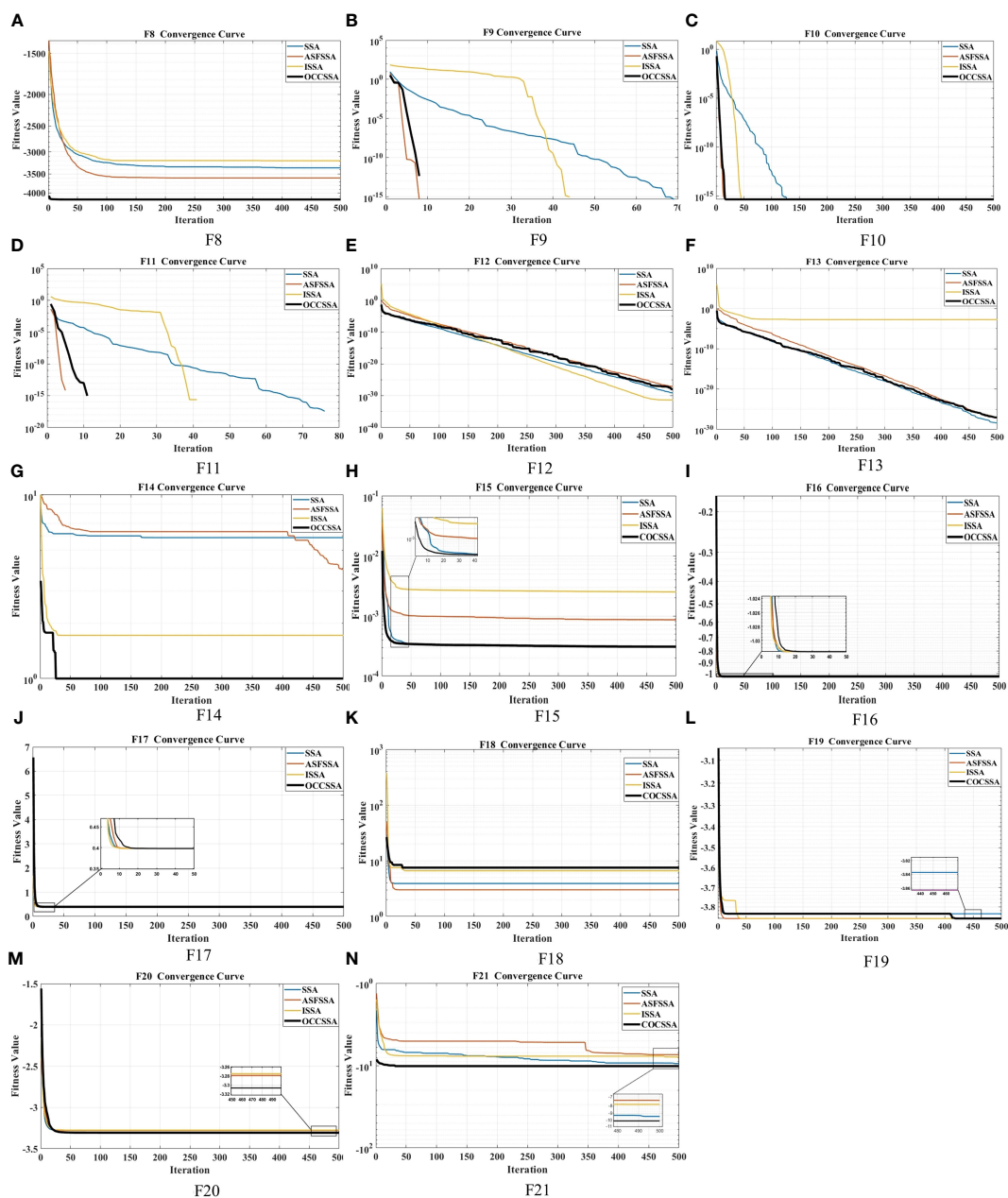


FIGURE 8 The convergence curve comparison graph of the improved SSA-type optimization algorithms on CEC2005 Multi-Modal functions: (A) F8. (B) F9. (C) F10. (D) F11. (E) F12. (F) F13. (G) F14. (H) F15. (I) F16. (J) F17. (K) F18. (L) F19. (M) F20. (N) F21.

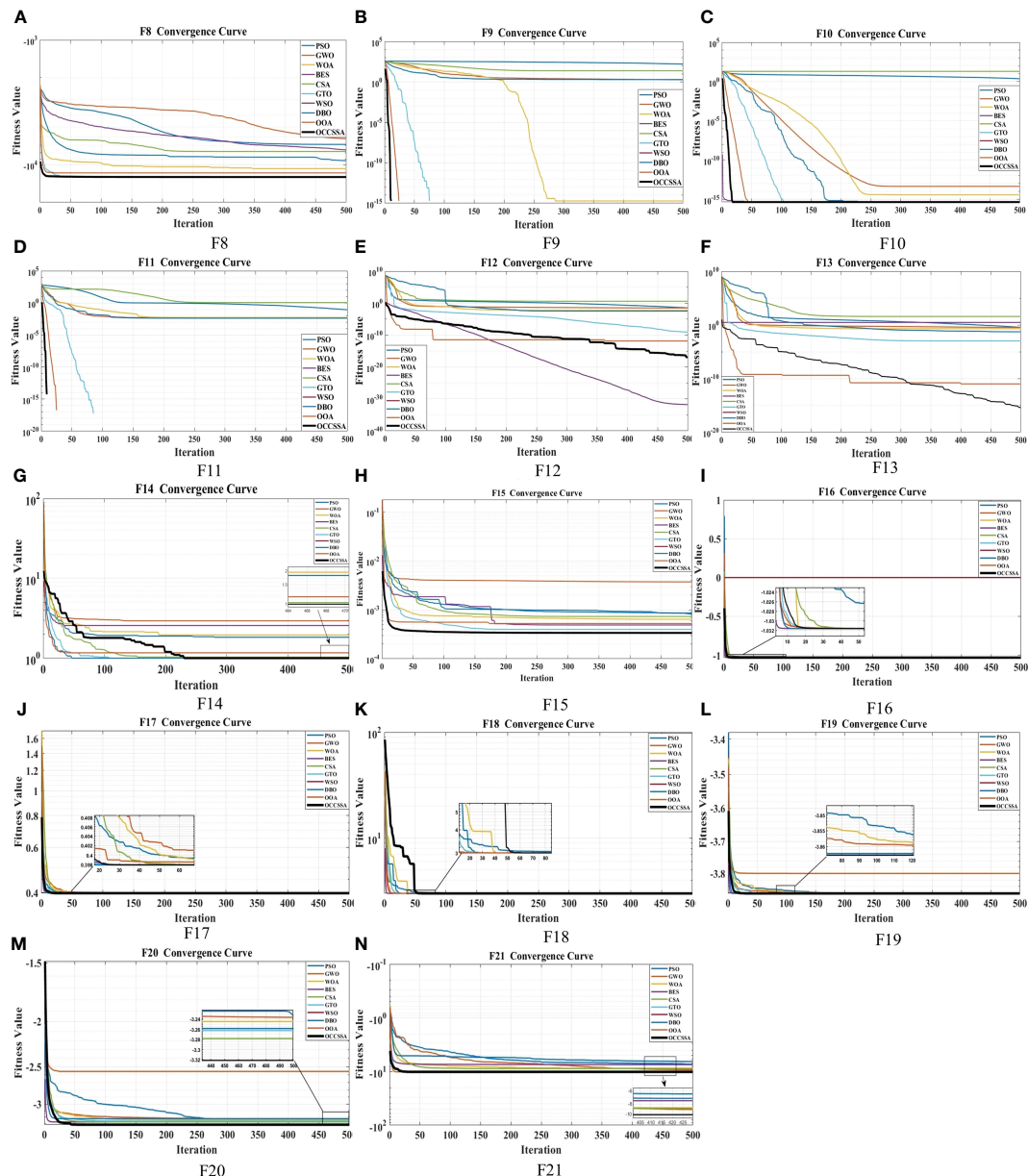


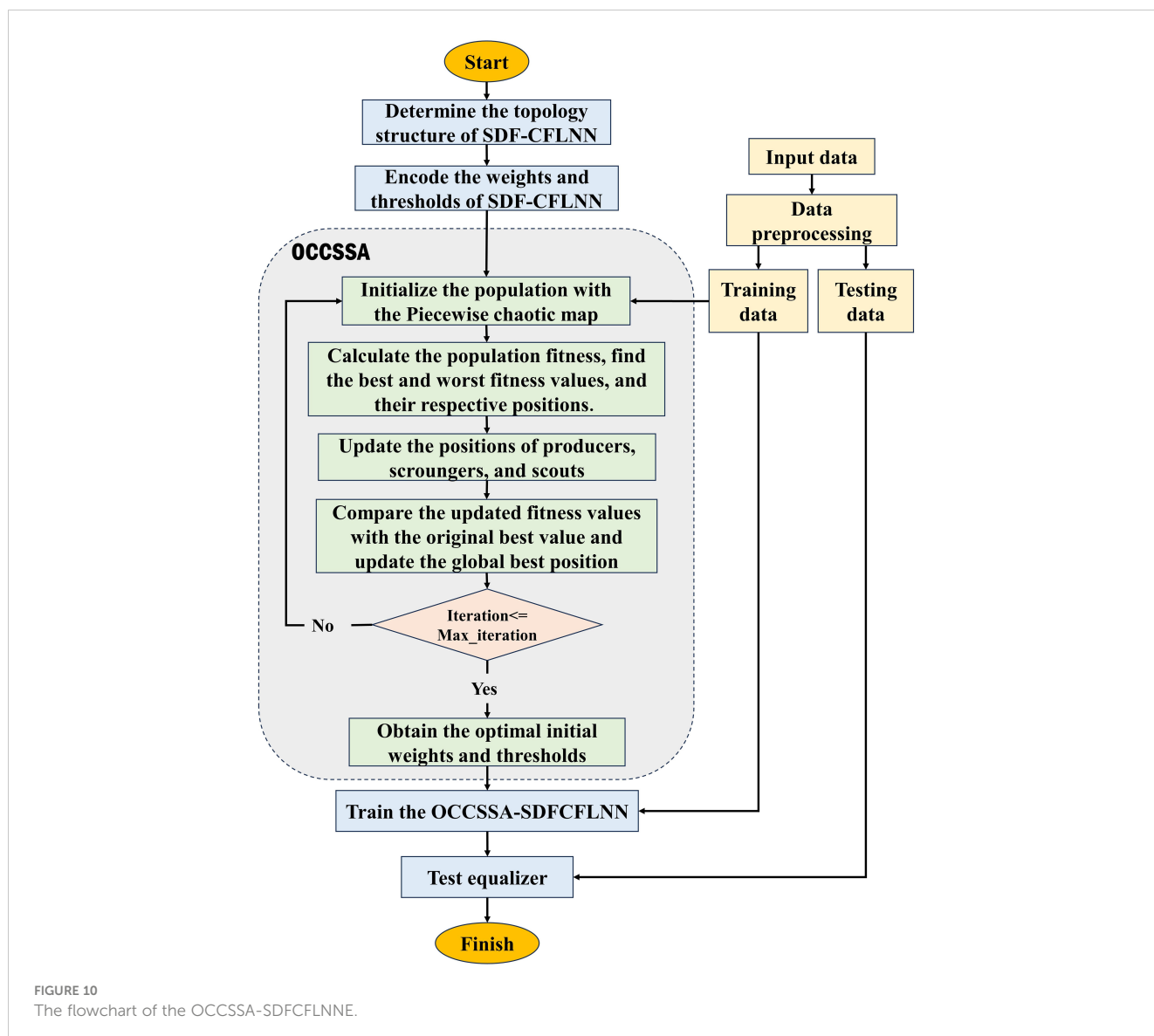
FIGURE 9 The convergence curve comparison graph of the swarm intelligence optimization algorithms on CEC2005 Multi-Modal functions: (A) F8. (B) F9. (C) F10. (D) F11. (E) F12. (F) F13. (G) F14. (H) F15. (I) F16. (J) F17. (K) F18. (L) F19. (M) F20. (N) F21.

4.1.2.4 Comparison with other swarm intelligence optimization algorithms

To further validate the effectiveness of the proposed algorithm, we compared OCCSSA with other recent swarm intelligence optimization algorithms. These include PSO (Kennedy and Eberhart, 1995), GWO (Mirjalili et al., 2014), WOA (Mirjalili and Lewis, 2016), BES (Alsattar et al., 2020), CSA (Feng et al., 2021), GTO (Abdollahzadeh et al., 2021), DBO (Braik et al., 2022), WSO (Xue and Shen, 2023), and OOA (Dehghani and Trojovský, 2023).

To enhance the credibility of the algorithm, we conducted 30 independent trials on each test function. In each trial, the maximum iteration count was set to 500, and the population size was 50.

The data results for CEC2005 tests are presented in Tables 5, 6. In CEC2005 multi-modal functions, despite the presence of multiple local optima, the proposed algorithm was able to successfully solve the optimization problems. OCCSSA demonstrated the best overall performance, especially in F8-F11, F13-F16, and F20, where it achieved the best values for each indicator, ranking first. For F12 and F21, OCCSSA ranked second, just behind BES and WSO, respectively. In the case of F17 and F18, OCCSSA ranked third. Due to its higher algorithm complexity, the runtime was in the middle to lower range. From the convergence curve plots in Figure 9, it is evident that OCCSSA exhibited overall better convergence speed and accuracy compared



to other algorithms. Overall, OCCSSA displayed strong resistance to local optima in multi-modal functions, and the introduction of multiple strategies significantly improved the algorithm's stability and search capabilities, with good convergence speed and accuracy.

4.1.3 Comparison of the random initialization and OCCSSA initialization of SDF-CFLNNE

During the training phase of a neural network, random initialization of weights and biases can significantly impact the training process and the final performance of the network. To address the issues related to improper initialization, such as gradient vanishing or exploding, and infeasible or slow convergence of the training process, the SDF-CFLNNE initialized intelligently with OCCSSA is proposed. The flowchart is as shown in the Figure 10.

First, determine the topology of SDF-CFLNN and encode its weights and thresholds. Then, input the encoded population into OCCSSA for initialization using the Piecewise chaotic map. Next,

calculate the fitness of the initial population and identify the best and worst population members. OCCSSA updates the positions of producers, scroungers, and scouts. The updated fitness is compared to the original best value, and the global best position is updated. When the maximum iteration is reached, obtain the best population as the initial weights and thresholds for training and testing the network.

We compared the SDFCFLNN equalizers with random initialization (R-SDFCFLNNE) and OCCSSA initialization (OCCSSA-SDFCFLNNE) in both CH1 and CH2 channel environments. To enhance the reliability of the algorithm, we conducted 10 independent experiments. In each experiment, the maximum iteration limit was set to 200, and SDF-CFLNNE still used the parameters from Table 2. For OCCSSA, we used a population size of 50, a safety threshold of 0.8, and the number of producers and the number of sparrows sensing danger were both set to 20% of the population size.

The BER performance of the R-SDFCFLNNE and OCCSSA-SDFCFLNNE are shown in Figures 11A, C, E. OCCSSA-

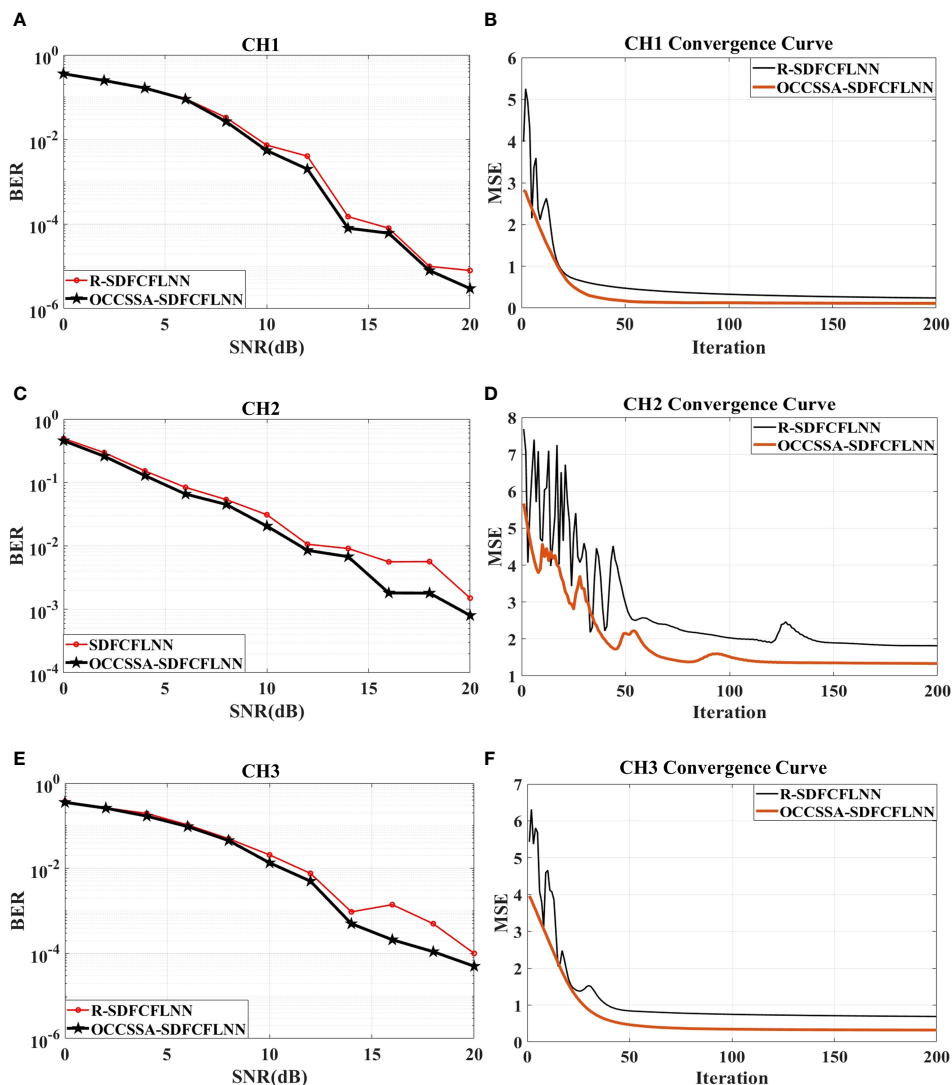


FIGURE 11
 The comparison of the R-SDFCFLNNE and OCCSSA-SDFCFLNNE for CH1, CH2 and CH3. (A) The BER performance for CH1. (B) The MSE performance for CH1 at SNR=10 dB. (C) The BER performance for CH2. (D) The MSE performance for CH2 at SNR=10 dB. (E) The BER performance for CH3. (F) The MSE performance for CH3 at SNR=10 dB.

SDFCFLNNE exhibits a slightly lower BER compared to R-SDFCFLNNE. Using CH1 as an example, as observed from Figures 6A, 11A, it is evident that at a BER of 10^{-3} , CFLNNE requires the SNR exceeding 20 dB, DF-CFLNNE requires the SNR of 16 dB, DFE requires the SNR of 13.8 dB, SDF-CFLNNE requires the SNR of 12.8 dB, and OCCSSA-SDFCFLNNE requires the SNR of 12.5 dB. OCCSSA-SDFCFLNNE demonstrates an improvement SNR of 0.2-8 dB compared to CFLNNE-based and traditional equalizers. At a BER of 10^{-4} , CFLNNE requires the SNR exceeding 20 dB, DFE requires the SNR of 19.7 dB, DF-CFLNNE requires the SNR of 18.8 dB, SDF-CFLNNE requires the SNR of 15.5 dB, and OCCSSA-SDFCFLNNE requires the SNR of 13.5 dB. OCCSSA-SDFCFLNNE outperforms CFLNNE-based and traditional equalizers, demonstrating an improvement SNR of 2-6 dB.

The MSE performance of the R-SDFCFLNNE and OCCSSA-SDFCFLNNE are shown in Figures 11B, D, F. Both of them converge at almost the same rate. However, R-SDFCFLNNE

exhibits minor oscillations in the early iterations, and the curve becomes smooth after convergence. In contrast, OCCSSA-SDFCFLNNE has an extremely smooth convergence curve, which is more stable. The MSE value of OCCSSA-SDFCFLNNE is smaller when it reaches a steady state, indicating more accurate signal recovery. When using OCCSSA initialization, it takes into account the specific characteristics and constraints of the communication channel to provide an optimal set of weight values for network initialization. This can lead to better initial conditions for the network, resulting in improved convergence and signal recovery.

4.2 Lake experiments and results

The analysis of lake experimental data has been presented in this part. On the day of the experiment, there was a slight surface fluctuation on the lake. Before conducting the lake experiments, the hydrophones

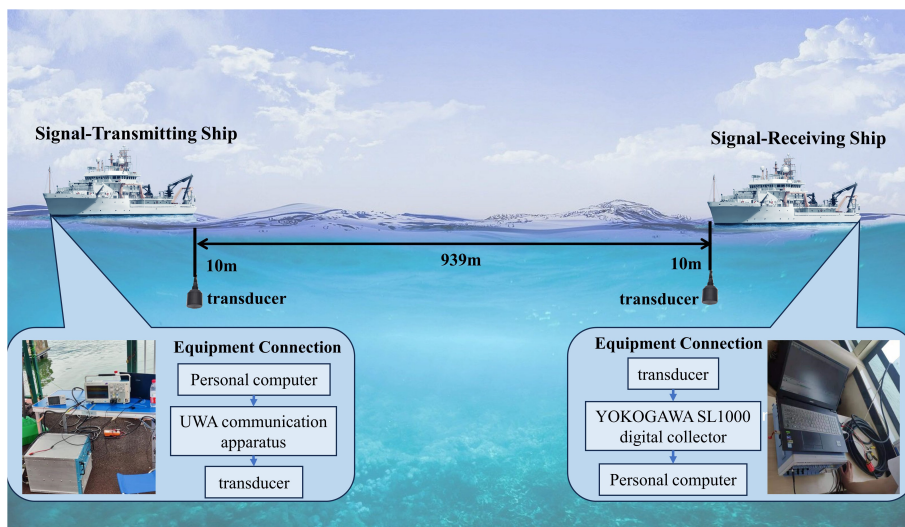


FIGURE 12 The layout diagram of transmitter and receiver.

	Doppler Estimation and Frame Synchronization Head	Training Sequences	Data Information	Frame Interval
Data I	1023 chips balanced Gold sequences	1500 chips	1000 chips	217 chips of data information
Data Q	1023 chips balanced Gold sequences	1500 chips	1000 chips	

Data Frame 1	Data Frame 2	...
--------------	--------------	-----

FIGURE 13 The frame format of the transmitting QPSK signal.

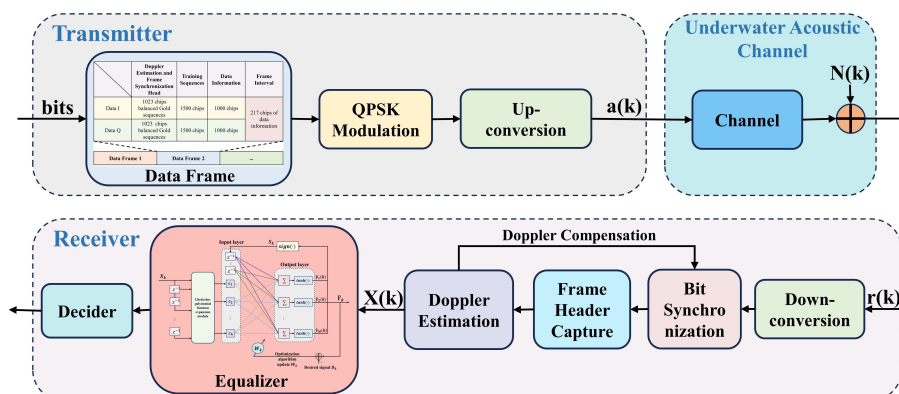


FIGURE 14 The signal processing flow.

TABLE 7 The BER comparison of every frame.

Frame	DFE-PLL	R-SDFCFLNNE	OCCSSA-SDFCFLNNE
1	0.1641	0.1222	0.0941
2	0.0962	0.0587	0.0425
3	0.1377	0.0959	0.0877
4	0.0854	0.0311	0.0195
5	0.1176	0.0734	0.0567

and other experimental equipment underwent meticulous calibration performed. Additionally, we carried out tasks such as assessing electrical connections and confirming the reliability of communication links within a controlled water tank environment. The equipment connection and layout for the transmitting ship and the receiving ship

are as shown in the Figure 12. The distance between the transmitting ship and the receiving ship is either 939m. Both the transmitting and receiving transducers are positioned 10 meters underwater.

The system employs QPSK modulation to facilitate data transmission. The signal frame format, as illustrated in Figure 13, has been specifically tailored for complex underwater acoustic conditions. Each data frame incorporates key elements such as Doppler estimation, frame synchronization headers, training sequences, data content, and frame intervals. This frame format is engineered to offer robust anti-Doppler capabilities and effectively mitigate cumulative timing errors. The signal undergoes amplification by a power amplifier and is then transmitted via a transducer. Simultaneously, multiple cycles of underwater acoustic signals are collected by the receivers using a digital collector linked to the transducer. The collected data are processed by DFE-PLL, R-SDFCFLNNE and OCCSSA-SDFCFLNNE. The signal processing flow at the receiving end is depicted in the Figure 14.

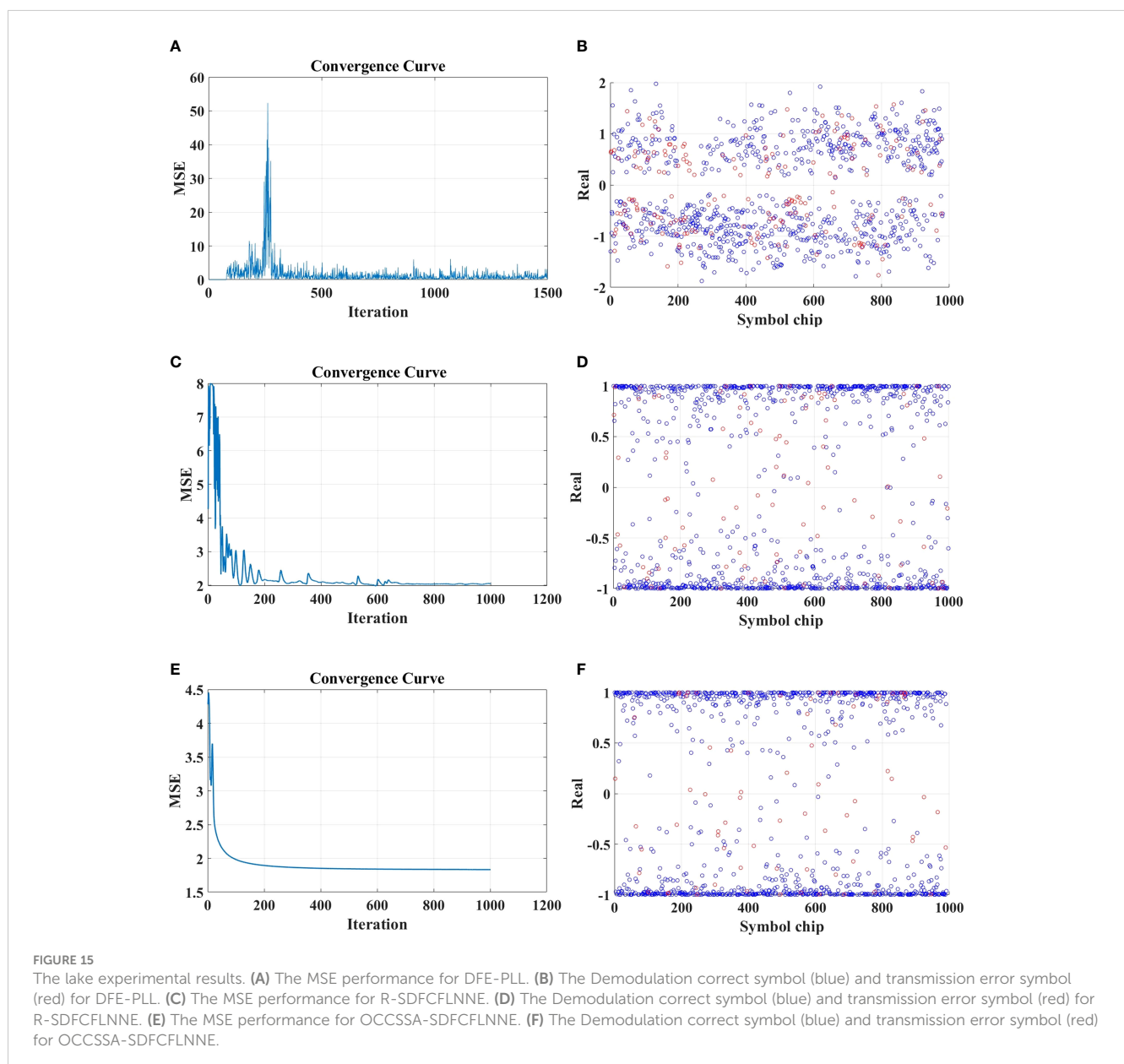


FIGURE 15 The lake experimental results. (A) The MSE performance for DFE-PLL. (B) The Demodulation correct symbol (blue) and transmission error symbol (red) for DFE-PLL. (C) The MSE performance for R-SDFCFLNNE. (D) The Demodulation correct symbol (blue) and transmission error symbol (red) for R-SDFCFLNNE. (E) The MSE performance for OCCSSA-SDFCFLNNE. (F) The Demodulation correct symbol (blue) and transmission error symbol (red) for OCCSSA-SDFCFLNNE.

The experimental results are shown in Table 7. We can observe that in the data processing of each frame, OCCSSA-SDFCFLNNE exhibits the lowest BER, followed by R-SDFCFLNNE, and DFE-PLL performing the least favorably. We've displayed the MSE convergence curves and the correct and incorrect symbol plots for the second frame of data. Figures 15A, C, E clearly demonstrate that the MSE curve in OCCSSA-SDFCFLNNE converges smoothly and reaches the lowest value, indicating a superior resistance to interference. While R-SDFCFLNNE exhibits minor initial fluctuations, DFE-PLL's MSE values fluctuate significantly during the initial phase. In Figures 15B, D, F, where blue represents correctly demodulated symbols and red represents incorrectly demodulated symbols, it's evident that OCCSSA-SDFCFLNNE has the fewest incorrect symbols and the best overall performance.

5 Conclusion

In the shallow-water environments, underwater communication links are susceptible to significant multipath effects. To address issues such as slow convergence and high system complexity in traditional channel equalizers, this paper proposes a simplified decision feedback Chebyshev function link neural network equalizer (SDF-CFLNNE). The SDF-CFLNNE structure's innovative approach employs Chebyshev polynomial function expansion modules, eliminating the need for hidden layers and enabling a direct, nonlinear transformation of input signals into the output layer. Additionally, it incorporates the feedback of decision signals into the input layer of the SDF-CFLNNE directly, instead of the function expansion module, which significantly reduces computational complexity. However, the effectiveness of neural networks crucially depends on the initial weights and biases, and random initialization can profoundly impact both the training process and the eventual performance of the network. To address this challenge, a novel chaotic sparrow search algorithm combining osprey optimization algorithm and Cauchy mutation (OCCSSA) is proposed. OCCSSA leverages a Piecewise chaotic population initialization strategy, combining the osprey's exploration tactics with the Cauchy mutation strategy to bolster global and local search capabilities. Simulation experiments, utilizing underwater multipath signals generated by the Bellhop Acoustic Toolbox, unequivocally demonstrate that the SDFCFLNNE initialized by OCCSSA outperforms both CFLNNE-based and traditional nonlinear equalizers. Notably, it achieves an impressive 2-6 dB improvement in SNR at a BER of 10^{-4} and exhibits a significantly reduced MSE. Furthermore, lake experimental data was employed to validate the effectiveness of the proposed equalizer. These results underscore the remarkable potential of the SDFCFLNNE initialized by OCCSSA as a compelling solution for significantly enhancing the reliability of underwater communication, particularly in the face of the challenges posed by complex underwater environments. This

research paves the way for more robust and efficient underwater communication systems, promising increased performance and greater accuracy in signal recovery.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

Author contributions

MZ: Conceptualization, Data curation, Methodology, Validation, Writing – original draft, Writing – review & editing. HZ: Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing. TL: Writing – review & editing, Funding acquisition, Project administration, Resources, Supervision. WH: Funding acquisition, Project administration, Resources, Supervision, Writing – review & editing. YD: Writing – review & editing, Data curation, Formal analysis, Investigation. YG: Data curation, Formal analysis, Investigation, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was financially supported by the National Natural Science Foundation of China (Grant No. 91938204 and 62271459), the Marine S and T fund of Shandong Province for Pilot National Laboratory for Marine Science and Technology (Qingdao) (No. 2018SDKJ0210).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Abdollahzadeh, B., Soleimani Gharehchopogh, F., and Mirjalili, S. (2021). Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intelligent Syst.* 36, 5887–5958. doi: 10.1002/int.22535
- Alsaltar, H. A., Zaidan, A., and Zaidan, B. (2020). Novel meta-heuristic bald eagle search optimisation algorithm. *Artif. Intell. Rev.* 53, 2237–2264. doi: 10.1007/s10462-019-09732-5
- Barnard, C. J., and Sibly, R. M. (1981). Producers and scroungers: a general model and its application to captive flocks of house sparrows. *Anim. Behav.* 29, 543–550. doi: 10.1016/S0003-3472(81)80117-0
- Braik, M., Hammouri, A., Atwan, J., Al-Betar, M. A., and Awadallah, M. A. (2022). White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems. *KnowledgeBased Syst.* 243, 108457. doi: 10.1016/j.knsys.2022.108457
- Burse, K., Yadav, R. N., and Shrivastava, S. (2010). Channel equalization using neural networks: A review. *IEEE Trans. systems man cybernetics Part C (Applications Reviews)* 40, 352–357. doi: 10.1109/TSMCC.2009.2038279
- Chagra, W., Bouani, F., Abdennour, R. B., Ksouri, M., and Favier, G. (2005). Equalization with decision delay estimation using recurrent neural networks. *Adv. Eng. Software* 36, 442–447. doi: 10.1016/j.advengsoft.2005.01.011
- Chang, Y.-J., and Ho, C.-L. (2009). “Decision feedback equalizers using self-constructing fuzzy neural networks,” in *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)*. 1483–1486 (IEEE).
- Chang, Y.-J., and Ho, C.-L. (2011). Scfn-based decision feedback equalization robust to frequency offset and phase noise. *Circuits Systems Signal Process.* 30, 929–940. doi: 10.1007/s00034-010-9258-5
- Chang, P.-R., and Wang, B.-C. (1995). Adaptive decision feedback equalization for digital satellite channels using multilayer neural networks. *IEEE J. selected areas Commun.* 13, 316–324. doi: 10.1109/49.345876
- Dehghani, M., and Trojovský, P. (2023). Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems. *Front. Mechanical Eng.* 8, 1126450. doi: 10.3389/fmech.2022.1126450
- Feng, Z.-k., Niu, W.-j., and Liu, S. (2021). Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems. *Appl. Soft Computing* 98, 106734. doi: 10.1016/j.asoc.2020.106734
- Freitag, L., Johnson, M., and Stojanovic, M. (1997). “Efficient equalizer update algorithms for acoustic communication channels of varying complexity,” in *Oceans’ 97. MTS/IEEE Conference Proceedings (Halifax, NS, Canada: IEEE)*, Vol. 1. 580–585.
- Gao, M., Guo, Y.-c., Liu, Z.-x., and Zhang, Y.-p. (2009). “Feed-forward neural network blind equalization algorithm based on super-exponential iterative,” in *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics (Hangzhou, China: IEEE)*, Vol. 1. 335–338.
- Ge, W., Wang, Z., Yin, J., and Han, X. (2022). Robust equalization for single-carrier underwater acoustic communications based on parameterized interference model. *IEEE Wireless Commun. Lett.* doi: 10.1109/LWC.2022.3223533
- Gibson, G. J., Siu, S., and Cowen, C. (1989). “Multilayer perceptron structures applied to adaptive equalizers for data communications,” in *International Conference on Acoustics, Speech, and Signal Processing (Glasgow, UK: IEEE)*, 1183–1186.
- Guha, D. R., and Patra, S. K. (2009). “Isi and burst noise interference minimization using wilcoxon generalized radial basis function equalizer,” in *2009 Fifth International Conference on MEMS NANO, and Smart Systems (Dubai, United Arab Emirates: IEEE)*, 89–92.
- He, C., Jing, L., Xi, R., Wang, H., Hua, F., Dang, Q., et al. (2019). Time-frequency domain turbo equalization for single-carrier underwater acoustic communications. *IEEE Access* 7, 73324–73335. doi: 10.1109/ACCESS.2019.2919757
- He, Y., and Tao, Y. (2023). “Deep reinforcement learning based cognitive equalization algorithm research in underwater communication,” in *2023 IEEE 3rd International Conference on Computer Communication and Artificial Intelligence (CCAI)* (Taiyuan, China: IEEE), 348–352.
- He, Q., Tao, J., Kong, X., Zhuang, Y., Jiang, M., and Qiao, Y. (2023). “Channel replay aided neural network equalizer for underwater acoustic communications,” in *OCEANS 2023-Limerick (Limerick, Ireland: IEEE)*, 1–5.
- Heng, S., Chen, W., Yin, H., Shipping, M., and Jizhang, Z. (2006). Decision feedback equalizer based on non-singleton fuzzy regular neural networks. *J. Syst. Eng. Electron.* 17, 896–900. doi: 10.1016/S1004-4132(07)60034-6
- Huang, J.-g., Wang, H., He, C.-b., Zhang, Q.-f., and Jing, L.-y. (2018). Underwater acoustic communication and the general performance evaluation criteria. *Front. Inf. Technol. Electronic Eng.* 19, 951–971. doi: 10.1631/FITEE.1700775
- Hussain, A., Soraghan, J. J., and Durrani, T. S. (1997). A new adaptive functional-link neural-network-based dfe for overcoming co-channel interference. *IEEE Trans. Commun.* 45, 1358–1362. doi: 10.1109/26.649741
- Ingle, K. K., and Jatoh, R. K. (2023). Non-linear channel equalization using modified grasshopper optimization algorithm. *Appl. Soft Computing* 110091. doi: 10.1016/j.asoc.2023.110091
- Kari, D., Vanli, N. D., and Kozat, S. S. (2017). Adaptive and efficient nonlinear channel equalization for underwater acoustic communication. *Phys. Communication* 24, 83–93. doi: 10.1016/j.phycom.2017.06.001
- Kechriotis, G., Zervas, E., and Manolakis, E. S. (1994). Using recurrent neural networks for adaptive communication channel equalization. *IEEE Trans. Neural Networks* 5, 267–278. doi: 10.1109/72.279190
- Kennedy, J., and Eberhart, R. (1995). “Particle swarm optimization,” in *Proceedings of ICNN’95 International conference on neural networks (WA, Australia: IEEE)*, Vol. 4. 1942–1948.
- Lee, T.-T., and Jeng, J.-T. (1998). The chebyshev-polynomials-based unified model neural networks for function approximation. *IEEE Trans. Systems Man Cybernetics Part B (Cybernetics)* 28, 925–935.
- Lee, J., and Sankar, R. (2007). Theoretical derivation of minimum mean square error of rbf based equalizer. *Signal Process.* 87, 1613–1625. doi: 10.1016/j.sigpro.2007.01.008
- Li, Y., Geng, T., Tian, R., and Gao, S. (2021). Machine-learning based equalizers for mitigating the interference in asynchronous mimo-owc systems. *J. Lightwave Technol.* 39, 2800–2808. doi: 10.1109/JLT.2021.3057396
- Liker, A., and Barta, Z. (2002). The effects of dominance on social foraging tactic use in house sparrows. *Behaviour*, 1061–1076. doi: 10.1163/15685390260337903
- Liu, W., Liu, J., Liu, T., Chen, H., and Wang, Y.-L. (2023). Detector design and performance analysis for target detection in subspace interference. *IEEE Signal Process. Lett.* doi: 10.1109/LSP.2023.3270080
- Liu, J., Mei, K., Zhang, X., Ma, D., and Wei, J. (2019). Online extreme learning machine-based channel estimation and equalization for ofdm systems. *IEEE Commun. Lett.* 23, 1276–1279. doi: 10.1109/LCOMM.2019.2916797
- Ma, X., Ye, H., and Li, Y. (2018). “Learning assisted estimation for time-varying channels,” in *2018 15th international symposium on wireless communication systems (ISWCS)* (Lisbon, Portugal: IEEE), 1–5.
- Milan, A., Rezaatoghfi, S. H., Dick, A., Reid, I., and Schindler, K. (2017). “Online multi-target tracking using recurrent neural networks,” in *Proceedings of the AAAI conference on Artificial Intelligence*, Vol. 31.
- Mirjalili, S., and Lewis, A. (2016). The whale optimization algorithm. *Adv. Eng. Software* 95, 51–67. doi: 10.1016/j.advengsoft.2016.01.008
- Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Adv. Eng. Software* 69, 46–61. doi: 10.1016/j.advengsoft.2013.12.001
- Mishra, J. P., Singh, K., and Chaudhary, H. (2023). “Recent advancement of ai technology for underwater acoustic communication,” in *AIP Conference Proceedings (AIP Publishing)*. (Jaipur, India: IEEE), Vol. 2752.
- Mohapatra, P. K., Rout, S. K., Bisoy, S. K., and Sain, M. (2022). Training strategy of fuzzy-firefly based ann in non-linear channel equalization. *IEEE Access* 10, 51229–51241. doi: 10.1109/ACCESS.2022.3174369
- Ning, X., Liu, Z., and Luo, Y. (2009). “Research on variable step-size blind equalization algorithm based on normalized rbf neural network in underwater acoustic communication,” in *Advances in Neural Networks-ISNN 2009: 6th International Symposium on Neural Networks, ISNN 2009 Wuhan, China, May 26-29, 2009 Proceedings, Part III 6* (Berlin Heidelberg: Springer), 1063–1070.
- Ouyang, C., Qiu, Y., and Zhu, D. (2021). Adaptive spiral flying sparrow search algorithm. *Sci. Programming* 2021, 1–16. doi: 10.1155/2021/6505253
- Patra, J. C., Chin, W. C., Meher, P. K., and Chakraborty, G. (2008). “Legendre-flann-based nonlinear channel equalization in wireless communication system,” in *2008 IEEE international conference on systems, man and cybernetics (Singapore: IEEE)*, 1826–1831.
- Patra, J. C., and Kot, A. C. (2002). Nonlinear dynamic system identification using chebyshev functional link artificial neural networks. *IEEE Trans. Systems Man Cybernetics Part B (Cybernetics)* 32, 505–511. doi: 10.1109/TSMCB.2002.1018769
- Patra, J. C., Pal, R. N., Baliarsingh, R., and Panda, G. (1999). Nonlinear channel equalization for qam signal constellation using artificial neural networks. *IEEE Trans. Systems Man Cybernetics Part B (Cybernetics)* 29, 262–271. doi: 10.1109/3477.752798
- Patra, J. C., Poh, W. B., Chaudhari, N. S., and Das, A. (2005). “Nonlinear channel equalization with qam signal using chebyshev artificial neural network,” in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005* (Montreal, QC, Canada: IEEE), Vol. 5. 3214–3219.
- Qiao, G., Liu, Y., Zhou, F., Zhao, Y., Mazhar, S., and Yang, G. (2022). Deep learning-based m-ary spread spectrum communication system in shallow water acoustic channel. *Appl. Acoustics* 192, 108742. doi: 10.1016/j.apacoust.2022.108742
- Qin, X., Qu, F., and Zheng, Y. R. (2020). Bayesian iterative channel estimation and turbo equalization for multiple-input-multiple-output underwater acoustic communications. *IEEE J. Oceanic Eng.* 46, 326–337. doi: 10.1109/OJEO.2019.2956299
- Sahu, J., and Majumder, S. (2021). “A particle swarm optimization based training algorithm for mcma blind adaptive equalizer,” in *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)* (Pune, India: IEEE), 462–465.
- Shwetha, N., Priyatham, M., and Gangadhar, N. (2023). Artificial neural network based channel equalization using battle royale optimization algorithm with different initialization strategies. *Multimedia Tools Appl.* 1–26. doi: 10.1007/s11042-023-16161-8

- Song, H. C., Hodgkiss, W., Kuperman, W., Stevenson, M., and Akal, T. (2006). Improvement of timereversal communications using adaptive channel equalizers. *IEEE J. Oceanic Eng.* 31, 487–496. doi: 10.1109/OE.2006.876139
- Song, W., Liu, S., Wang, X., and Wu, W. (2020). “An improved sparrow search algorithm,” in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. 537–543 (IEEE).
- Stojanovic, M., Catipovic, J. A., and Proakis, J. G. (1994). Phase-coherent digital communications for underwater acoustic channels. *IEEE J. oceanic Eng.* 19, 100–111. doi: 10.1109/48.289455
- Stojanovic, M., and Preisig, J. (2009). Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE Commun. magazine* 47, 84–89. doi: 10.1109/MCOM.2009.4752682
- Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., et al. (2005). Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization. *KanGAL Rep.* 2005005 2005.
- Tsai, C.-W., Lai, C.-F., Chiang, M.-C., and Yang, L. T. (2013). Data mining for internet of things: A survey. *IEEE Commun. Surveys Tutorials* 16, 77–97. doi: 10.1109/SURV.2013.103013.00206
- van Heteren, M. (2022). Link adaptation and equalization for underwater acoustic communication using machine learning.
- Wang, Z., Chen, F., Yu, H., and Shan, Z. (2021). Sparse decision feedback equalization for underwater acoustic channel based on minimum symbol error rate. *Int. J. Naval Architecture Ocean Eng.* 13, 617–627. doi: 10.1016/j.ijnaoe.2021.07.004
- Wang, T., Wen, C.-K., Wang, H., Gao, F., Jiang, T., and Jin, S. (2017). Deep learning for wireless physical layer: Opportunities and challenges. *China Commun.* 14, 92–111. doi: 10.1109/CC.2017.8233654
- Wei, G., Yizhen, J., Xiao, H., Xiao, Z., and Wentao, T. (2023). Robust equalization for single-carrier underwater acoustic communication in sparse impulsive interference environment. *Appl. Acoustics* 214, 109706. doi: 10.1016/j.apacoust.2023.109706
- Xi, J., Yan, S., Xu, L., and Hou, C. (2020). Sparsity-aware adaptive turbo equalization for underwater acoustic communications in the mariana trench. *IEEE J. Oceanic Eng.* 46, 338–351. doi: 10.1109/OE.2020.2982808
- Xi, J., Yan, S., Xu, L., Zhang, Z., and Zeng, D. (2019). Frequency–time domain turbo equalization for underwater acoustic communications. *IEEE J. Oceanic Eng.* 45, 665–679. doi: 10.1109/OE.2019.2891171
- Xiao, Y., and Dong, Y. (2015). Instantaneous gradient based dual mode wavelet neural network blind equalization for underwater acoustic channel. *Appl. Mathematics Inf. Sci.* 9, 1467. doi: 10.12785/amis/090341
- Xiao, Y., Dong, Y., and Li, Z. (2008). “Blind equalization in underwater acoustic communication by recurrent neural network with bias unit,” in *2008 7th World Congress on Intelligent Control and Automation* (Chongqing, China: IEEE), 2407–2410.
- Xue, J., and Shen, B. (2020). A novel swarm intelligence optimization approach: sparrow search algorithm. *Syst. Sci. control Eng.* 8, 22–34. doi: 10.1080/21642583.2019.1708830
- Xue, J., and Shen, B. (2023). Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomputing* 79, 7305–7336. doi: 10.1007/s11227-022-04959-6
- Yang, R., Yang, L., Zhang, J., Sun, C., Cong, W., and Zhu, S. (2018). “Blind equalization of qam signals via extreme learning machine,” in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)* (Xiamen, China: IEEE), 34–39.
- Zhang, Y., Li, C., Wang, H., Wang, J., Yang, F., and Meriaudeau, F. (2022). Deep learning aided ofdm receiver for underwater acoustic communications. *Appl. Acoustics* 187, 108515. doi: 10.1016/j.apacoust.2021.108515
- Zhang, Y., Li, J., Zakharov, Y. V., Li, J., Li, Y., Lin, C., et al. (2019b). Deep learning based single carrier communications over time-varying underwater acoustic channel. *IEEE Access* 7, 38420–38430. doi: 10.1109/ACCESS.2019.2906424
- Zhang, X.-L., and Wang, D. (2016). A deep ensemble learning method for monaural speech separation. *IEEE/ACM Trans. audio speech Lang. Process.* 24, 967–977. doi: 10.1109/TASLP.2016.2536478
- Zhang, X., Wu, H., Sun, H., and Ying, W. (2021). Multireceiver sas imagery based on monostatic conversion. *IEEE J. Selected Topics Appl. Earth Observations Remote Sens.* 14, 10835–10853. doi: 10.1109/JSTARS.2021.3121405
- Zhang, L., and Yang, L.-L. (2020). Machine learning for joint channel equalization and signal detection. *Mach. Learn. Future Wireless Commun.*, 213–241. doi: 10.1002/9781119562306.ch12
- Zhang, G., Yang, L., Chen, L., Zhao, B., Li, Y., and Wei, W. (2019a). “Blind equalization algorithm for underwater acoustic channel based on support vector regression,” in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (IHMSC) (Hangzhou, China: IEEE), Vol. 2. 163–166.
- Zhang, Y., Zakharov, Y. V., and Li, J. (2018). Soft-decision-driven sparse channel estimation and turbo equalization for mimo underwater acoustic communications. *IEEE Access* 6, 4955–4973. doi: 10.1109/ACCESS.2018.2794455
- Zhao, H., Zeng, X., Zhang, J., and Li, T. (2010). Nonlinear adaptive equalizer using a pipelined decision feedback recurrent neural network in communication systems. *IEEE Trans. Commun.* 58, 2193–2198. doi: 10.1109/TCOMM.2010.08.080612
- Zhao, H., Zeng, X., Zhang, X., Zhang, J., Liu, Y., and Wei, T. (2011). An adaptive decision feedback equalizer based on the combination of the fir and flnn. *Digital Signal Process.* 21, 679–689. doi: 10.1016/j.dsp.2011.05.004
- Zhao, H., and Zhang, J. (2008). Functional link neural network cascaded with chebyshev orthogonal polynomial for nonlinear channel equalization. *Signal Process.* 88, 1946–1957. doi: 10.1016/j.sigpro.2008.01.029
- Zhou, M., Zhang, H., Lv, T., Li, H., Xiang, D., Huang, S., et al. (2022). “Underwater acoustic channel modeling under different shallow seabed topography and sediment environment,” in *OCEANS 2022Chennai* (Chennai, India: IEEE), 1–7.
- Zhu, J., Song, Y., Jiang, N., Xie, Z., Fan, C., and Huang, X. (2023). Enhanced doppler resolution and sidelobe suppression performance for golay complementary waveforms. *Remote Sens.* 15, 2452. doi: 10.3390/rs15092452