*Article*

# Optimizing the Probabilistic Neural Network Model with the Improved Manta Ray Foraging Optimization Algorithm to Identify Pressure Fluctuation Signal Features

Xiyuan Liu [1], Liying Wang [1,2], Hongyan Yan [1,2,*], Qingjiao Cao [1,2], Luyao Zhang [1] and Weiguo Zhao [1,2,*]

1   School of Water Conservancy and Hydropower, Hebei University of Engineering, Handan 056038, China; jinle16@163.com (X.L.); wangliying@hebeu.edu.cn (L.W.); xianingziwei99@163.com (Q.C.); zly01221214@163.com (L.Z.)
2   Hebei Key Laboratory of Intelligent Water Conservancy, Hebei University of Engineering, Handan 056038, China
*   Correspondence: yanhongyan@hebeu.edu.cn (H.Y.); zhaoweiguo@hebeu.edu.cn (W.Z.)

**Abstract:** To improve the identification accuracy of pressure fluctuation signals in the draft tube of hydraulic turbines, this study proposes an improved manta ray foraging optimization (ITMRFO) algorithm to optimize the identification method of a probabilistic neural network (PNN). Specifically, first, discrete wavelet transform was used to extract features from vibration signals, and then, fuzzy c-means algorithm (FCM) clustering was used to automatically classify the collected information. In order to solve the local optimization problem of the manta ray foraging optimization (MRFO) algorithm, four optimization strategies were proposed. These included optimizing the initial population of the MRFO algorithm based on the elite opposition learning algorithm and using adaptive t distribution to replace its chain factor to optimize individual update strategies and other improvement strategies. The ITMRFO algorithm was compared with three algorithms on 23 test functions to verify its superiority. In order to improve the classification accuracy of the probabilistic neural network (PNN) affected by smoothing factors, an improved manta ray foraging optimization (ITMRFO) algorithm was used to optimize them. An ITMRFO-PNN model was established and compared with the PNN and MRFO-PNN models to evaluate their performance in identifying pressure fluctuation signals in turbine draft tubes. The evaluation indicators include confusion matrix, accuracy, precision, recall rate, F1-score, and accuracy and error rate. The experimental results confirm the correctness and effectiveness of the ITMRFO-PNN model, providing a solid theoretical foundation for identifying pressure fluctuation signals in hydraulic turbine draft tubes.

**Keywords:** hydraulic turbine; pressure fluctuation; manta ray foraging optimization algorithm; probabilistic neural network; signal identification

## 1. Introduction

The problem of vibration occurs during the operation of hydropower units on a regular basis and cannot be avoided [1]. The turbine, which is the most important piece of equipment in a hydropower plant, has a vibration mechanism that is more complicated than that of the other hydropower units. A large number of studies have shown that the operational status of the turbine can be analyzed by classifying and identifying the vibration signal characteristics [2,3] so as to determine whether the turbine needs to be inspected and repaired. Therefore, accurate identification of vibration signals has become particularly important.

The identification of hydraulic turbine vibration signals is a complex problem that involves multiple fields of knowledge. Feature extraction is the first step in identifying the vibration signals of hydraulic turbines. The authors in [4] put forward a method of pressure fluctuation signal analysis based on variational mode decomposition in view

of the non-stationarity of the pressure fluctuation signal in the draft tube of a hydraulic turbine. Pan et al. [5] suggested a new approach to analyze hydropower unit vibration data using local mean decomposition (LMD) and Wigner–Ville distribution. Lu et al. [6] used empirical node decomposition (EMD) and index energy to extract the dynamic characteristic information of the draft tube of a hydraulic turbine. Oguejiofor et al. [7] used discrete wavelet transform and principal component analysis to identify the power plant reactor coolant pumps' vibration signals. Four main component variables are enough to determine normal and pathological vibration signals. Shomaki et al. [8] proposed a new framework for bearing fault detection based on wavelet trans-form and discrete Fourier transform combined with deep learning. Lu et al. [9] suggested an upgraded Hilbert–Huang transform (HHT) approach with energy correlation fluctuation criterion for hydroelectric generating unit vibration signal feature extraction. Wang et al. [10] suggested a Hilbert–Huang transform (HHT)-based motor current signal analysis approach to obtain unstable pump operation features under cavitation situations. From this, it can be seen that there are multiple methods that can be used to extract signal features. This study used discrete wavelet transform for the feature extraction of vibration signals, which is an effective method that is widely used in the field of signal processing.

With the rapid development and widespread application of artificial intelligence, more and more research fields such as pattern identification and fault diagnosis are applying artificial intelligence to handle and solve problems. The deep learning method is used to classify and identify the signals of pressure changes in the draught tube of the Francis turbine. This is carried out to keep the turbine from breaking down or to find the problem quickly if the turbine does break down. Luo et al. [11] developed an adaptive Fisher-based deep convolutional neural network for rolling bearing fault identification. Lan et al. [12] optimized the smoothing factor of a probabilistic neural network (PNN) using the fruit fly optimization algorithm (FOA) and constructed an FOA-PNN model. The FOA-PNN model can predict the operating status of steam turbines in a short amount of time and monitor running faults in real time. Ma et al. [13] proposed an identification method based on singular value decomposition (SVD) and MPSO-SVM. Cao et al. [14] suggested an improved artificial bee colony optimization algorithm (IARO) with adaptive weight adjustment. IARO optimized support vector machine (SVM) to create an identification model to classify and identify vibration signals in different stages. Wang et al. [15] proposed a new convolutional neural network model, MIMTNet. This model has multidimensional signal inputs and multidimensional task outputs, which can improve the generalizability and accuracy of bearing fault diagnosis. Li et al. [16] proposed a one-time NAS fault diagnosis method that can use a supernet to include all candidate networks, train a supernet to evaluate the performance of candidate networks, and apply it to industrial faults. Ravikumar et al. [17] proposed a fault diagnosis model including multi-scale deep residual learning (MDRL-SLSTM) with stacked long short-term memory, which is used to process sequential data in the health prediction task of internal combustion engine transmission. Prosvirin et al. [18] suggested diagnosing rubbing problems of varied intensities using multivariate data and a multivariate one-dimensional convolutional neural network. Ji et al. [19] suggested a two-stage intelligent fault diagnosis system (order-tracking one-dimensional convolutional neural network, OT-1DCNN) to address the problem of fault identification under changing speed settings. Zhou et al. [20] proposed a method for the health diagnosis and early warning of electric motor bearings based on an integrated system of empirical mode decomposition, principal component analysis, and adaptive network fuzzy inference (EEMD-PCA-ANFIS).

Optimization algorithms possess excellent adaptability and global search capabilities [21], which can improve neural networks by improving the way they work and making them more accurate. Sánchez et al. [22] presented a new optimization technique for modular neural network (MNN) architecture based on granular computing and the firefly algorithm. The method was tested to verify its effectiveness and advantages in terms of face identification and compared with other methods. The results show that the

method is effective at optimally designing neural networks in pattern identification. Lan et al. [12] suggested optimizing probabilistic neural network (PNN) smoothing factors with the drosophila optimization algorithm (FOA). To categorize the operational condition of hydraulic turbine units, an FOA-PNN network model was constructed. The FOA-PNN has been shown to predict steam turbine operating conditions quickly and monitor operational issues in real time.

Based on the above research, the following conclusions can be drawn: extracting vibration signal features is the first step in signal identification; in the field of fault diagnosis, the accurate identification of vibration signal features is crucial; and optimization algorithms have emerged as a useful tool in neural network training, allowing for the development of more precise and efficient answers to many problems. High-precision signal feature identification technology can quickly and accurately locate fault points, and pressure fluctuation changes can reflect the water turbine's operating status. Therefore, this study mainly evaluated the feature identification method of the pressure fluctuation signal in the draft tube of a hydraulic turbine.

### 1.1. Motivation and Contribution

The safe and stable operation of hydraulic turbines is crucial for energy production and supply, and vibration signals are one of the important indicators of hydraulic turbine faults and problems. In order to better analyze and apply these signals, we used discrete wavelet transform for the feature extraction of vibration signals, which is an effective method that is widely used in the field of signal processing. By extracting the features, useful information in the vibration signal can be revealed, thereby achieving identification and analysis of the state of the hydraulic turbine.

In addition, in order to improve the accuracy and efficiency of signal classification and identification, optimization algorithms have played an important role. However, the manta ray foraging optimization (MRFO) algorithm often has problems such as being prone to falling into local optima. In order to improve the manta ray foraging optimization (MRFO) algorithm, we selected four improvement strategies. Firstly, by optimizing the initial population of the manta ray foraging optimization algorithm based on the elite-opposition-based learning algorithm, the algorithm's initial search ability was improved. Secondly, an adaptive t distribution was used to replace the chain factor and optimize individual update strategies and other improvement strategies to improve the algorithm's global search ability. These improvement strategies helped to improve the optimization performance of the manta ray foraging optimization algorithm. And by comparing the performance with particle swarm optimization (PSO), the sparrow search (SSA) algorithm, and the manta ray foraging optimization (MRFO) algorithm on 23 test functions, the superiority of the improved manta ray foraging optimization (ITMRFO) algorithm was verified.

Neural networks are commonly used for signal identification and pattern classification tasks. Probabilistic neural networks (PNNs) are particularly popular because they have a fast-learning speed and can be directly trained. When using probabilistic neural networks for classification, the smoothing factor is a key input parameter that affects the smoothness of decision boundaries. The selection of smoothing factors directly affects the performance and classification results of probabilistic neural networks. In order to optimize the smoothing factor of the probabilistic neural network, we adopted the improved manta ray foraging optimization (ITMRFO) algorithm for training. By optimizing the improved manta ray foraging optimization (ITMRFO) algorithm, we could determine the optimal smoothing factor, thereby improving the classification accuracy of probabilistic neural network (PNN) feature identification.

The main contributions of this study are highlighted as follows.

- Discrete wavelet transforms extracts pressure fluctuation signal features. The fuzzy c-means (FCM) clustering algorithm method, a partition-based clustering algorithm, classifies extracted features and automatically classifies vibration signal characteristics.

- To improve the manta ray foraging optimization (MRFO) algorithm, four improvement methodologies were chosen, including an elite-opposition-based learning algorithm and adaptive t distribution. The improved manta ray foraging optimization (ITMRFO) algorithm method was put to the test using 23 benchmark functions. The experimental findings demonstrate the good exploration and exploitation capabilities of the ITMRFO algorithm.
- Experiments were carried out on the vibration signal of the draft tube of a mixed flow turbine, and verification was achieved through the confusion matrix, accuracy, precision, recall, F1-score, and identification error rate of the training samples. The usefulness of the probabilistic neural network (PNN) identification method optimized by the improved manta ray foraging optimization (ITMRFO) algorithm was demonstrated. The results of the experiments show that the ITMRFO-PNN model is effective at identifying the vibration signs of the draft tube of the hydraulic turbine.

### 1.2. Paper Organization

The residual section of this study is as follows. Section 2 introduces the basic concepts of discrete wavelet transform, the fuzzy c-means (FCM) clustering algorithm, the manta ray foraging optimization (MRFO) algorithm, and the probabilistic neural network (PNN). The MRFO algorithm was improved, producing the ITMRFO algorithm. The improved manta ray foraging optimization (ITMRFO) algorithm was tested using 23 test functions, and the basic overview of the ITMRFO-PNN model was introduced. The identification results of the PNN model, the MRFO-PNN model, and the ITMRFO-PNN model were compared. The effectiveness and accuracy of the ITMRFO-PNN model were verified by comparing these three models through a confusion matrix, accuracy, precision, recall rate, and F1-score. Section 4 introduces the conclusions and future development directions.

### 2. Materials and Methods

#### 2.1. Discrete Wavelet Transform

In the 1980s, French scientists Grossman and Morlet proposed wavelet analysis for the analysis of seismic signals. Wavelet analysis is a method for analyzing non-stationary signals. The schematic diagram of wavelet analysis is shown in Figure 1.
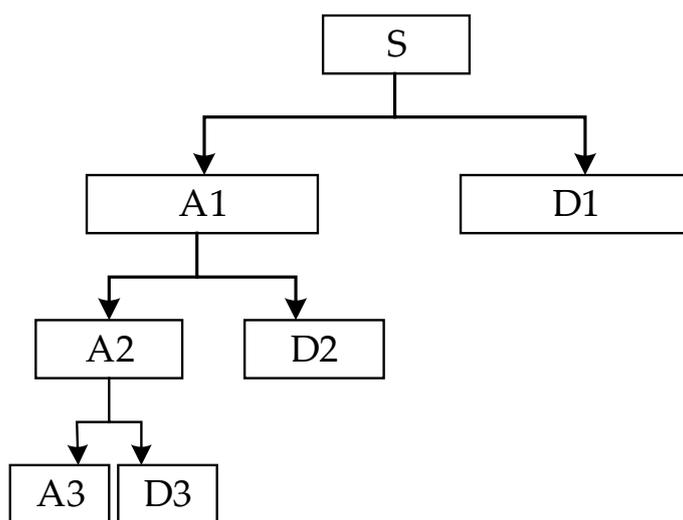


**Figure 1.** The schematic diagram of wavelet analysis.

Discrete wavelet transform (DWT) [23] is an important time-frequency analysis method, which can decompose nonlinear and non-stationary signals at multiple scales. Discrete wavelet transform decomposes the signal into a series of scale coefficients (approximate coefficients) and wavelet coefficients (detail coefficients). The wavelet inverse transform can

effectively extract useful information from the signal by selecting appropriate coefficients. The basis function of the discrete wavelet is:

$$WT_s(s,k) = \langle x(t), \psi_{s,k}(t) \rangle \tag{1}$$

*2.2. Fuzzy C-Means Clustering Algorithm*

Dumm proposed the fuzzy c-means clustering (FCM) [24] algorithm in 1974, and Bezdek improved and extended it in 1981. FCM clustering is to take a given set of samples $X = \{x_1, x_2 \ldots x_n\}$, divide this n-sample set into c classes, where $2 \leq c < n$, based on certain clustering principles and fuzzy criteria, and denote this c class by $Y = \{y_1, y_2 \ldots y_c\}$. The FCM algorithm has the following objective function:

$$J_{FCM}(U,Y) = \sum_{i=0}^{n} \sum_{j=1}^{c} u_{ij}^m \parallel x_i - y_j \parallel^2 (m \geq 0) \tag{2}$$

where c is the number of clusters, $n$ is the data sample, $\sum_{i=1}^{c} u_{ij} = 1, u_{ij} \in [0,1]$ is the weighting index ($m \in [0, \infty)$), and U is the input space X a fuzzy c-ization score. The Euclidean distance between the i-th sample data $x_i$ and the j-th clustering center $y_j$ is $\parallel x_i - y_j \parallel^2$.

The fuzzy c-means (FCM) clustering algorithm technique is frequently used [25,26] to determine the class of the sample points for the goal of automatically categorizing the sample data by obtaining the affiliation of each sample point to all class centers by optimizing the objective function. Therefore, the fuzzy c-means (FCM) clustering algorithm method is used to group the extracted pressure fluctuation eigenvectors.

*2.3. An Improved MRFO Algorithm*

2.3.1. MRFO Algorithm

In 2020, Zhao W. et al. [27] proposed a novel intelligent bionic population method called the manta ray foraging optimization algorithm.

The MRFO algorithm is a mathematical modeling of the foraging process of manta rays in the ocean, and it is based on the three foraging strategies of manta rays. Compared to intelligent heuristics such as particle swarm optimization and simulated annealing, the MRFO algorithm is characterized by high merit-seeking ability, fast convergence, and few parameters [28–30].

1.  Chain foraging

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + r\left(x_{best}^d(t) - x_i^d(t)\right) + \alpha\left(x_{best}^d - x_i^d(t)\right), i = 1 \\ x_i^d(t) + r\left(x_{i-1}^d(t) - x_i^d(t)\right) + \alpha\left(x_{best}^d - x_i^d(t)\right), i = 2,\ldots,N \end{cases} \tag{3}$$

$$\alpha = 2r\sqrt{|\log r|} \tag{4}$$

where $x_i^d(t)$ denotes the position of the tth generation, the ith individual in the dimension, $r$ denotes a random number uniformly distributed on $[0,1]$, $x_{best}^d(t)$ denotes the position of the best individual in the tth generation in the dth dimension, $N$ denotes the number of individuals, and $\alpha$ is the chain factor.

2.  Spiral foraging
    When $\frac{t}{T} > rand$

$$x_i^d(t+1) = \begin{cases} x_{best}^d(t) + r\left(x_{best}^d(t) - x_i^d(t)\right) + \beta\left(x_{best}^d - x_i^d(t)\right), i = 1 \\ x_{best}^d(t) + r\left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta\left(x_{best}^d - x_i^d(t)\right), i = 2,\ldots,N \end{cases} \tag{5}$$

$$\beta = 2e^{r_1 \frac{T-i+1}{T}} sin(2\pi r_1) \tag{6}$$

where $T$ is the total number of iterations, $r_1$ is a random number evenly distributed on the range $[0, 1]$, and $\beta$ is the spiral factor. When $\frac{t}{T} \leq rand$

$$x_i^d(t+1) = \begin{cases} x_{rand}^d(t) + r\left(x_{rand}^d(t) - x_i^d(t)\right) + \beta\left(x_{rand}^d - x_i^d(t)\right), i = 1 \\ x_{rand}^d(t) + r\left(x_{i-1}^d(t) - x_i^d(t)\right) + \beta\left(x_{rand}^d - x_i^d(t)\right), i = 2, \dots, N \end{cases} \quad (7)$$

$$x_{rand}^d = Lb^d + r\left(Ub^d - Lb^d\right) \quad (8)$$

where $x_{rand}^d(t)$ indicates the random position in generation tth and dimension dth. The upper and lower limits of a variable are denoted by $Ub^d$ and $Lb^d$.

3. The rolling foraging

$$x_i^d(t+1) = x_i^d(t) + S\left(r_2 x_{best}^d - r_3 x_i^d(t)\right), \ i = 1, 2, \dots, N \quad (9)$$

$$S = 2 \quad (10)$$

where $S$ is the rollover factor, and the random integers $r_2$ and $r_3$ are both equally distributed on the range $[0, 1]$.

The MRFO algorithm's rapid convergence and prematurity lead to poor population diversity, making it easy to slip into local optimal solutions [31]. To increase population variety and jump out of the local optimum, the initial population of the MRFO algorithm, chain foraging strategy, and spiral foraging strategy are all modified.

### 2.3.2. Elite Opposition-Based Learning

The population intelligent optimization algorithm's performance is greatly influenced by the quality of the starting population, and a high-quality starting population increases the likelihood of finding the best overall solution. Tizhoosh [32] proposed the opposition-based learning strategy (OBL) in 2005, which has been shown to increase the probability of finding a globally optimum solution by almost 50% [33] and has been applied to various optimization algorithms [34–36]. The main idea behind elite-opposition-based learning is to use the practical information of elite individuals to create opposite individuals in the area where the current individuals are. The opposite individuals are then placed in competition with the current individuals, and the best individuals are chosen to be the next generation.

The elite opposition-based learning strategy is used in the MRFO initialization phase to construct the opposite population of the current population. In order to create a new population, the original population is mixed with the opposing population. The new population is ranked according to population quality. In order to ensure the quality of the selected population, the first 50% of the new population is selected as the initial solution.

### 2.3.3. Adaptive T-Distribution Strategy

T-distributions, also known as student distributions, contain parametric degrees of freedom. The shape of the curve of the t-distribution is related to the size. The lesser the degree of freedom, the flatter the t-distribution curve, the lower the center of the curve, and the bigger the tail on both sides [37,38]. The probability density is as follows:

$$p(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\,\Gamma\left(\frac{n}{2}\right)} \cdot \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}} \quad (11)$$

where $\Gamma\left(\frac{n+1}{2}\right) = \int_0^{+\infty} x^{\frac{k+1}{2}-1} e^{-x} dx$ is the Euler integral of the second kind, $\Gamma()$ denotes the gamma function. $n$ represents the degrees of freedom parameter for the t-distribution. $x$ is the value at which the probability density function is evaluated.

When $t(n \to \infty) \to N(0, 1)$, both deviations can often be disregarded for $n \geq 30$. However, when $t(n = 1) = C(0, 1)$, the Cauchy distribution is $C(0, 1)$ and the Gaussian

distribution is $N(0,1)$. In other words, the two boundary special case distributions of the t-distribution are the standard Gaussian distribution and the Cauchy distribution. The relationship between the densities of the Gaussian, Cauchy, and t distributions is shown in Figure 2. The tail curve of the Cauchy distribution has a long and flat form, while the tail curve of the normal distribution has a short and steep form. And the Cauchy variation has a greater likelihood of producing next-generation points far from the parent than the Gaussian variation.
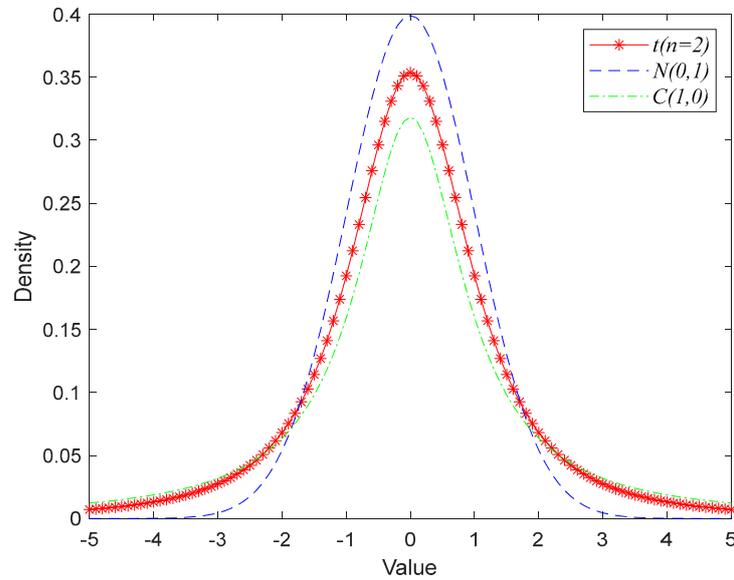


**Figure 2.** Probability density plot of Cauchy distribution, Gaussian distribution, and T-distribution.

In the chain predation process, manta rays form a predation chain from head to tail. The moving direction and step length of the next position of individual manta rays are jointly determined by the current optimum solution and the part of the previous individual. T-distribution is used to replace the chain factor $\alpha$ in the chain foraging place of the MRFO algorithm.

In the chain foraging and spiral foraging of the MRFO algorithm, $r$ is stochastic and unstable, so the part of the equation that multiplies with $r$ is removed. The improved algorithm is named ITMRFO.

2.3.4. ITMFRO Algorithm

1. Chain foraging

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + t\left(x_{best}^d - x_i^d(t)\right), i = 1 \\ x_i^d(t) + t\left(x_{best}^d - x_i^d(t)\right), i = 2, \ldots, N \end{cases} \tag{12}$$

where $x_i^d(t)$ denotes the position of the tth generation, the ith individual in the dimension, $x_{best}^d(t)$ denotes the position of the best individual in the tth generation in the dth dimension, $N$ denotes the number of individuals, and $t$ is the t-distribution.

2. Spiral foraging
   When $\frac{t}{T} > rand$

$$x_i^d(t+1) = \begin{cases} x_{best}^d(t) + \beta\left(x_{best}^d - x_i^d(t)\right), i = 1 \\ x_{best}^d(t) + \beta\left(x_{best}^d - x_i^d(t)\right), i = 2, \ldots, N \end{cases} \tag{13}$$

$$\beta = 2e^{r_1 \frac{T-i+1}{T}} sin(2\pi r_1) \tag{14}$$

where $T$ is the total number of iterations, $r_1$ is a random number evenly distributed on the range $[0, 1]$, and $\beta$ is the spiral factor. When $\frac{t}{T} \leq rand$

$$x_i^d(t+1) = \begin{cases} x_k^d(t) + \beta\left(x_{rand}^d - x_i^d(t)\right), i = 1 \\ x_k^d(t) + \beta\left(x_{rand}^d - x_i^d(t)\right), i = 2, \ldots, N \end{cases} \tag{15}$$

$$x_{rand}^d = Lb^d + r\left(Ub^d - Lb^d\right) \tag{16}$$

where $x_{rand}^d(t)$ indicates the random position in generation tth and dimension dth and $x_k^d(t)$ indicates the top 50% of the population.

3. The rolling foraging

$$x_i^d(t+1) = x_i^d(t) + S\left(r_2 x_{best}^d - r_3 x_i^d(t)\right), \ i = 1, 2, \ldots, N \tag{17}$$

$$S = 2 \tag{18}$$

where $S$ is the rollover factor and the random integers $r_2$ and $r_3$ are both equally distributed on the range $[0, 1]$.

### 2.3.5. Algorithm Comparison Validation

There were 23 chosen benchmark test functions [39]. Experimental comparisons were made between numerical simulations of four intelligence algorithms, including the sparrow search algorithm (SSA), the particle swarm optimization (PSO) algorithm, the manta ray foraging optimization (MRFO) algorithm, and the improved manta ray foraging optimization (ITMRFO) algorithm. Based on the optimum value, the worst value, the mean value, and the standard deviation, these four algorithms were assessed.

For this experiment, a maximum of 500 iterations were allowed, and the initial population size for each of the four algorithms was set at 30. The setting parameters of these four algorithms can be seen in Table 1. Table 2, Table 3, and Table 4 show the unimodal test function, multimodal test function, and fixed-dimension multimodal test function, respectively.

**Table 1.** Algorithm parameter.

| Algorithms | Main Parameters |
|---|---|
| SSA | Early warning value $ST = 0.6$, the weight of finder $ST = 0.7$, the weight of joiner $ST = 0.3$, and Sparrows aware of danger weight $ST = 0.2$ |
| PSO | $\omega = 0.9, c_1 = c_2 = 2$ |
| MRFO | $S = 2$ |
| ITMRFO | $S = 2$ |

**Table 2.** Unimodal test functions.

| Function | D | Range |
|---|---|---|
| $F_1(X) = \sum_{i=1}^{n} x_i^2$ | 30 | $[-100, 100]^D$ |
| $F_2(X) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$ | 30 | $[-10, 10]^D$ |
| $F_3(X) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_j\right)^2$ | 30 | $[-100, 100]^D$ |
| $F_4(X) = max\{|x_i|, 1 \leq i \leq n\}$ | 30 | $[-100, 100]^D$ |
| $F_5(X) = \sum_{i=1}^{n} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$ | 30 | $[-30, 30]^D$ |
| $F_6(X) = \sum_{i=1}^{n-1} ([x_i + 0.5])^2$ | 30 | $[-100, 100]^D$ |
| $F_7(X) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$ | 30 | $[-1.28, 1.28]^D$ |

**Table 3.** Multimodal test functions.

| Function | D | Range |
|---|---|---|
| $F_8(X) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{|x_i|}\right)$ | 30 | $[-500, 500]^D$ |
| $F_9(X) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$ | 30 | $[-5.12, 5.12]^D$ |
| $F_{10}(X) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$ | 30 | $[-32, 32]^D$ |
| $F_{11}(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\frac{x_i}{\sqrt{i}} + 1$ | 30 | $[-600, 600]^D$ |
| $F_{12}(x) = \frac{\pi}{n}\{10sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_1-1)^2\left[1+10sin^2(\pi y_1+1)\right] + (y_n-1)^2\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4) + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ | 30 | $[-50, 50]^D$ |
| $F_{13}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{29}(x_i-1)^2 p[1+10sin^2(3\pi x_{i+1})] + (x_n-1)^2[1+sin^2(2\pi x_{30})]\} + \sum_{i=1}^{30} u(x_i, 5, 10, 4)$ | 30 | $[-50, 50]^D$ |

**Table 4.** Fixed-dimensional multimodal test functions.

| Function | D | Range |
|---|---|---|
| $F_{14}(X) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$ | 2 | $[-65.536, 65.536]^D$ |
| $F_{15}(X) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$ | 4 | $[-5, 5]^D$ |
| $F_{16}(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$ | 2 | $[-5, 5]^D$ |
| $F_{17}(X) = -\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$ | 2 | $[-5, 10] \times [0, 15]$ |
| $F_{18}(X) = [1 + (x_1+x_2+1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)^2] \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$ | 2 | $[-2, 2]^D$ |
| $F_{19}(X) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$ | 3 | $[0, 1]^D$ |
| $F_{20}(X) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$ | 6 | $[0, 1]^D$ |
| $F_{21}(X) = -\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$ | 4 | $[0, 10]^D$ |
| $F_{22}(X) = -\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$ | 4 | $[0, 10]^D$ |
| $F_{23}(X) = -\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T + c_i\right]^{-1}$ | 4 | $[0, 10]^D$ |

As can be seen in Table 5, the sparrow search algorithm (SSA) obtains the optimal global solutions for the optimum in functions F1–F5, and it obtains the optimal global solutions for the optimum, worst, mean, and standard deviation in function F5. The manta ray foraging optimization (MRFO) algorithm obtains globally optimal solutions in functions F2 and F4 for the standard deviation, and it also obtains globally optimal solutions in functions F1, F3, and F6 for the optimum, worst, mean, and standard deviation. The improved manta ray foraging optimization (ITMRFO) algorithm obtains globally optimal solutions in functions F1–F3 and F7 for the optimum, worst, mean, and standard deviation. And it obtains the optimal global solutions for the worst, mean, and standard deviation in function F4.

**Table 5.** Optimization results of the four intelligent optimization algorithms on unimodal test functions.

| Function | Value | SSA | PSO | MRFO | ITMRFO |
|---|---|---|---|---|---|
| **F1** | **Optimum** | **0** | $2.2590 \times 10$ | **0** | **0** |
| | **Worst** | $1.1305 \times 10^{42}$ | $6.5671 \times 10^{2}$ | **0** | **0** |
| | **Mean** | $3.7683 \times 10^{44}$ | $3.2095 \times 10^{2}$ | **0** | **0** |
| | **Standard** | $2.064 \times 10^{43}$ | $1.6547 \times 10^{2}$ | **0** | **0** |
| **F2** | **Optimum** | **0** | $4.096$ | $2.1756 \times 10$ | **0** |
| | **Worst** | $4.6943 \times 10^{-18}$ | $3.5087 \times 10$ | $8.5057 \times 10^{-202}$ | $\mathbf{1.1006 \times 10^{-289}}$ |
| | **Mean** | $1.5648 \times 10^{-19}$ | $1.6695 \times 10$ | $2.8356 \times 10^{-203}$ | $\mathbf{6.4115 \times 10^{-291}}$ |
| | **Standard** | $8.5706 \times 10^{-19}$ | $7.1839$ | **0** | **0** |
| **F3** | **Optimum** | **0** | $1.2559 \times 10^{3}$ | **0** | **0** |
| | **Worst** | $8.5409 \times 10^{-31}$ | $2.7754 \times 10^{4}$ | **0** | **0** |
| | **Mean** | $2.847 \times 10^{-32}$ | $7.8126 \times 10^{3}$ | **0** | **0** |
| | **Standard** | $1.5594 \times 10^{-31}$ | $5.8392 \times 10^{3}$ | **0** | **0** |
| **F4** | **Optimum** | **0** | $6.6225$ | $1.3387 \times 10^{-211}$ | $1.0118 \times 10^{-310}$ |
| | **Worst** | $1.2899 \times 10^{-40}$ | $1.4551 \times 10$ | $7.9795 \times 10^{-202}$ | $\mathbf{5.6387 \times 10^{-284}}$ |
| | **Mean** | $4.2997 \times 10^{-42}$ | $1.0199 \times 10$ | $6.1482 \times 10^{-203}$ | $\mathbf{1.8796 \times 10^{-285}}$ |
| | **Standard** | $2.355 \times 10$ | $2.3013$ | **0** | **0** |
| **F5** | **Optimum** | $\mathbf{5.5425 \times 10^{-6}}$ | $4.2109 \times 10^{2}$ | $2.1162 \times 10$ | $2.7003 \times 10$ |
| | **Worst** | $\mathbf{7.3554 \times 10^{-2}}$ | $3.9617 \times 10^{4}$ | $2.3790 \times 10$ | $2.8860 \times 10$ |
| | **Mean** | $\mathbf{2.0988 \times 10^{-2}}$ | $1.2007 \times 10^{4}$ | $2.2765 \times 10$ | $2.8152 \times 10$ |
| | **Standard** | $\mathbf{1.6765 \times 10^{-2}}$ | $8.7865 \times 10^{3}$ | $5.6801 \times 10^{-1}$ | $6.9318 \times 10^{-1}$ |
| **F6** | **Optimum** | $1.9537 \times 10^{-5}$ | $6.7162 \times 10$ | $\mathbf{4.7686 \times 10^{-12}}$ | $1.9852$ |
| | **Worst** | $2.5253 \times 10^{-4}$ | $6.3960 \times 10$ | $\mathbf{8.6445 \times 10^{-10}}$ | $3.8045$ |
| | **Mean** | $8.6289 \times 10^{-5}$ | $3.4391 \times 10^{2}$ | $\mathbf{9.3891 \times 10^{-11}}$ | $2.9016$ |
| | **Standard** | $5.0853 \times 10^{-5}$ | $1.2485 \times 10^{2}$ | $\mathbf{1.5542 \times 10^{-10}}$ | $4.32 \times 10^{-1}$ |
| **F7** | **Optimum** | $1.9284 \times 10^{-4}$ | $4.2776 \times 10^{-2}$ | $7.7868 \times 10^{-6}$ | $\mathbf{3.7422 \times 10^{-6}}$ |
| | **Worst** | $1.7391 \times 10^{-3}$ | $1.8851 \times 10$ | $4.6089 \times 10^{-4}$ | $\mathbf{3.5646 \times 10^{-4}}$ |
| | **Mean** | $6.4509 \times 10^{-4}$ | $1.928$ | $1.3824 \times 10^{-4}$ | $\mathbf{7.7893 \times 10^{-5}}$ |
| | **Standard** | $4.2451 \times 10^{-4}$ | $4.8185$ | $1.1841 \times 10^{-4}$ | $\mathbf{8.6124 \times 10^{-5}}$ |

As can be seen in Figure 3, the sparrow search algorithm (SSA) in function F5 has higher convergence accuracy than the other three algorithms. However, the manta ray foraging optimization (MRFO) algorithm and the improved manta ray foraging optimization (ITMRFO) algorithm converge faster and more consistently. Furthermore, in function F6, the manta ray foraging optimization (MRFO) algorithm converges faster than the three different algorithms, especially before 200 iterations. Compared to the three different optimization algorithms, the improved manta ray foraging optimization (ITMRFO) algorithm has a faster convergence speed and higher convergence accuracy for functions F1–F4 and F7.

From the above analysis, it can be seen that the convergence curve of particle swarm optimization (PSO) is at a standstill, the optimization is basically stopped, and the local optimization cannot be skipped. Among the four algorithms, the ability to jump out of the local optimum is the worst. The convergence curve of the sparrow search (SSA) algorithm is mainly stagnant, and only a small fraction can jump out of the local optimum. The ability of the sparrow search (SSA) algorithm to jump out of the local optimum is only stronger than that of particle swarm optimization (PSO). Although the manta ray foraging optimization (MRFO) algorithm has a strong ability to jump out of the local optimum, its convergence speed and accuracy are lower than the improved manta ray foraging optimization (ITMRFO) algorithm. Compared with the other three algorithms, the improved manta ray foraging optimization (ITMRFO) algorithm has the advantages

of superior performance, high accuracy, and fast convergence speed when dealing with unimodal test functions.
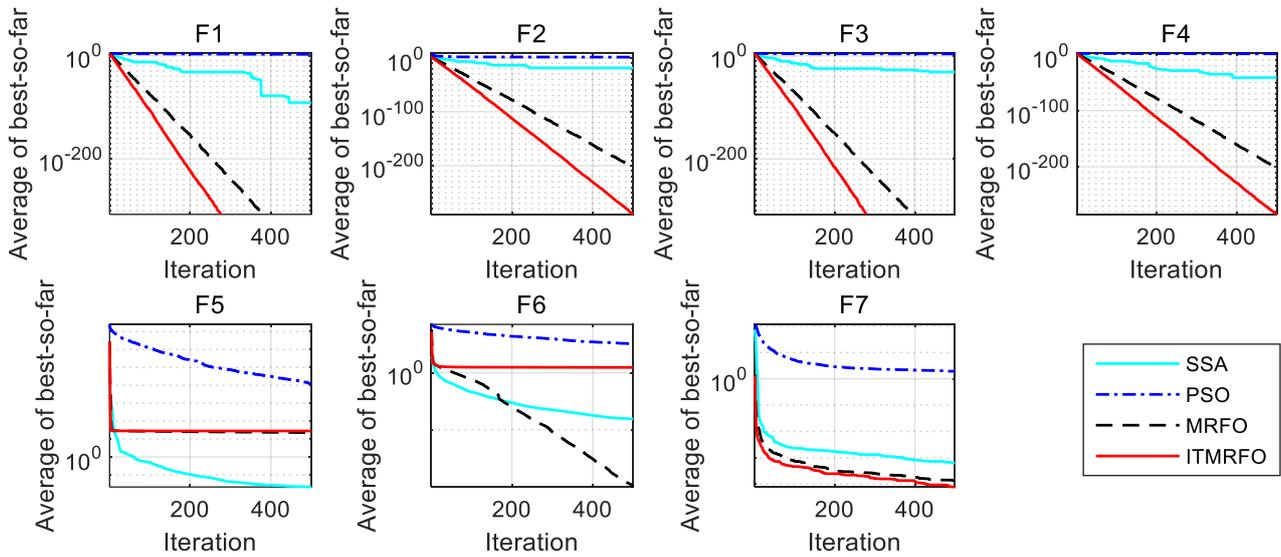


**Figure 3.** Convergence characteristics of the four algorithms on unimodal test functions.

As can be seen in Table 6, the sparrow search algorithm (SSA) obtains the glob-ally optimal solutions of the optimum, worst, mean, and standard deviation in functions F9–F11 and F13. The manta ray foraging optimization (MRFO) algorithm obtains the optimal global solution of the worst value and standard deviation in function F8 and the optimal global solution of the optimum, worst, mean, and standard deviation in functions F9–F12. The improved manta ray foraging optimization (ITMRFO) algorithm obtains the globally optimal solutions of the optimal value and the mean value in function F8; it also obtains the globally optimal solutions of the optimum, worst, mean, and standard deviation in functions F9–F11.

**Table 6.** Optimization results of the four intelligent optimization algorithms on multimodal test functions.

| Function | Value | SSA | PSO | MRFO | ITMRFO |
|---|---|---|---|---|---|
| **F8** | **Optimum** | $-1.2569 \times 10^4$ | $-9.8631 \times 10^3$ | $-9.3716 \times 10^3$ | $\mathbf{-6.9461 \times 10^{16}}$ |
| | **Worst** | $-5.2284 \times 10^3$ | $-5.4187 \times 10^3$ | $\mathbf{-6.9232 \times 10^3}$ | $-4.6265 \times 10^3$ |
| | **Mean** | $-8.8956 \times 10^3$ | $-7.5328 \times 10^3$ | $-8.3555 \times 10^3$ | $\mathbf{-2.3154 \times 10^{15}}$ |
| | **Standard** | $2.5607 \times 10^3$ | $1.0890 \times 10^3$ | $\mathbf{6.2578 \times 10^2}$ | $1.2682 \times 10$ |
| **F9** | **Optimum** | **0** | $1.5453 \times 10^2$ | **0** | **0** |
| | **Worst** | **0** | $2.5812 \times 10^5$ | **0** | **0** |
| | **Mean** | **0** | $1.9755 \times 10^5$ | **0** | **0** |
| | **Standard** | **0** | $3.0430 \times 10$ | **0** | **0** |
| **F10** | **Optimum** | $\mathbf{8.8818 \times 10^{-16}}$ | $4.3941$ | $\mathbf{8.8818 \times 10^{-16}}$ | $\mathbf{8.8818 \times 10^{-16}}$ |
| | **Worst** | $\mathbf{8.8818 \times 10^{-16}}$ | $7.4033$ | $\mathbf{8.8818 \times 10^{-16}}$ | $\mathbf{8.8818 \times 10^{-16}}$ |
| | **Mean** | $\mathbf{8.8818 \times 10^{-16}}$ | $5.8133$ | $\mathbf{8.8818 \times 10^{-16}}$ | $\mathbf{8.8818 \times 10^{-16}}$ |
| | **Standard** | **0** | $9.2016 \times 10^{-1}$ | **0** | **0** |
| **F11** | **Optimum** | **0** | $1.9237$ | **0** | **0** |
| | **Worst** | **0** | $7.8588$ | **0** | **0** |
| | **Mean** | **0** | $4.0544$ | **0** | **0** |
| | **Standard** | **0** | $1.5093$ | **0** | **0** |

**Table 6.** *Cont.*

| Function | Value | SSA | PSO | MRFO | ITMRFO |
|---|---|---|---|---|---|
| **F12** | **Optimum** | $2.9046 \times 10^{-6}$ | $7.563 \times 10^{-1}$ | $\mathbf{1.2186 \times 10^{-13}}$ | $1.1021 \times 10^{-1}$ |
| | **Worst** | $6.2883 \times 10^{-5}$ | $1.5219 \times 10$ | $\mathbf{8.7152 \times 10^{-12}}$ | $4.0626 \times 10^{-1}$ |
| | **Mean** | $2.1853 \times 10^{-5}$ | $6.6329$ | $\mathbf{2.7254 \times 10^{-12}}$ | $2.3181 \times 10^{-1}$ |
| | **Standard** | $1.622 \times 10^{-5}$ | $3.1326$ | $\mathbf{2.4554 \times 10^{-12}}$ | $6.2942 \times 10^{-2}$ |
| **F13** | **Optimum** | $\mathbf{4.5633 \times 10^{-6}}$ | $4.9391$ | $1.0987 \times 10^{-2}$ | $1.6503$ |
| | **Worst** | $\mathbf{4.693 \times 10^{-2}}$ | $3.9108 \times 10$ | $2.9661$ | $2.9842$ |
| | **Mean** | $\mathbf{5.1678 \times 10^{-3}}$ | $2.0094 \times 10$ | $2.6196$ | $2.7706$ |
| | **Standard** | $\mathbf{9.7266 \times 10^{-3}}$ | $8.9844$ | $9.1335 \times 10^{-1}$ | $4.1936 \times 10^{-1}$ |

As can be seen in Figure 4, the sparrow search algorithm (SSA) can jump out of the local optimum and converge more accurately than the other three algorithms in function F13. Still, the manta ray foraging optimization (MRFO) algorithm and improved manta ray foraging optimization (ITMRFO) algorithm can also jump out of the local optimum and are more stable than the SSA. The manta ray foraging optimization (MRFO) algorithm starts to converge faster than the other three algorithms before 200 iterations in function F12. Still, the improved manta ray foraging optimization (ITMRFO) algorithm can also jump out of the local optimum and is more stable than the sparrow search algorithm (SSA) and the manta ray foraging optimization (MRFO) algorithm. The improved manta ray foraging optimization (ITMRFO) algorithm converges fastest in functions F9–F11 and starts to converge faster and more accurately before 400 iterations in function F8.
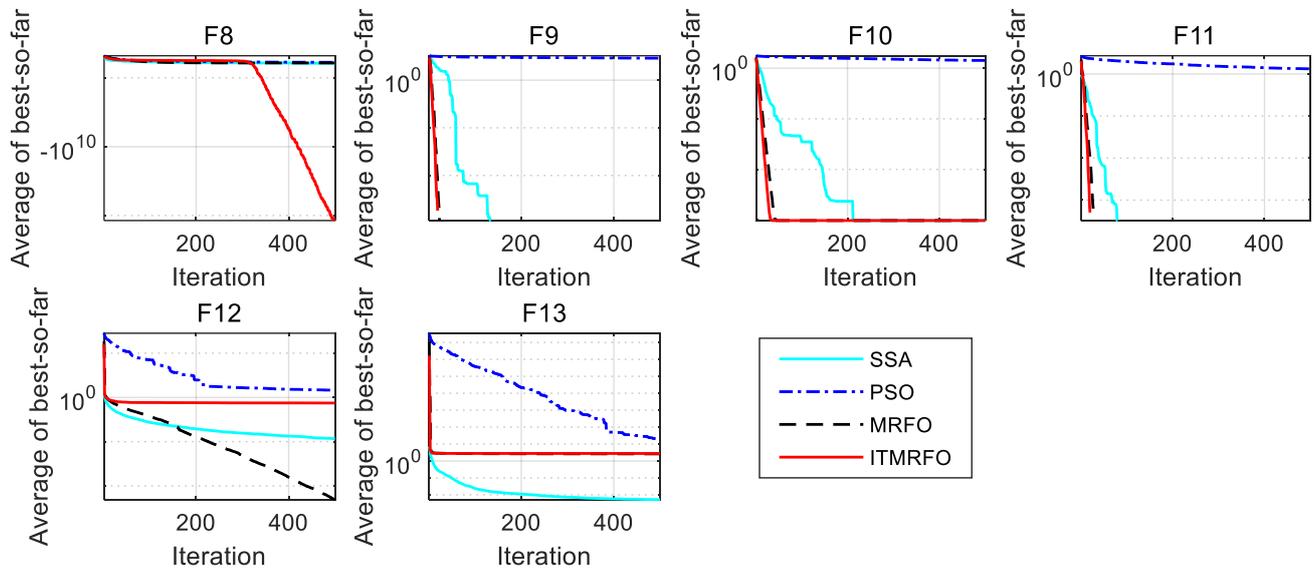


**Figure 4.** Convergence characteristics of the four algorithms on multimodal test functions.

From the above analysis, it can be seen that the convergence curve of particle swarm optimization (PSO) is mostly in a stagnant state, and the ability to jump out of the local optimum is the worst among the four algorithms. Although the sparrow search algorithm (SSA) has a strong ability to jump out of the local optimum, its convergence speed is slow. Although the manta ray foraging optimization (MRFO) algorithm has a strong ability to jump out of the local optimum, its convergence speed is lower than the improved manta ray foraging optimization (ITMRFO) algorithm. Therefore, the improved manta ray foraging optimization (ITMRFO) algorithm has the best performance, high convergence accuracy, and most of the fastest convergence speeds when dealing with multimodal test functions.

According to Table 7, the sparrow search algorithm (SSA) obtains globally optimal solutions for the optimum, worst, mean, and standard deviation in functions F21–F23.

The sparrow search algorithm (SSA) obtains globally optimal solutions for the optimum in functions F14–F15 and F17–F20. The sparrow search algorithm (SSA) obtains globally optimal solutions for the worst and mean in functions F15–F19. The sparrow search algorithm (SSA) obtains globally optimal solutions for the standard deviation in function F15. Particle swarm optimization (PSO) obtains globally optimal solutions for the optimum in functions F14, F16–F19, and F21–F23, and it also obtains globally optimal solutions for the worst and mean in functions F14 and F16–F17. The manta ray foraging optimization (MRFO) algorithm obtains globally optimal solutions for the optimum, worst, mean, and standard deviation in functions F14, F17, and F19. The manta ray foraging optimization (MRFO) algorithm obtains globally optimal solutions for the optimum in functions F16, F18, and F23. The manta ray foraging optimization (MRFO) algorithm obtains globally optimal solutions for the worst and mean in functions F16, F18, and F20. The manta ray foraging optimization (MRFO) algorithm obtains globally optimal solutions for the standard deviation in function F20. The improved manta ray foraging optimization (ITMRFO) algorithm obtains globally optimal solutions for the optimum, worst, mean, and standard deviation in functions F16–F18. The improved manta ray foraging optimization (ITMRFO) algorithm obtains globally optimal solutions for the optimum in functions F14–F15, F19, and F21–23. The improved manta ray foraging optimization (ITMRFO) algorithm obtains globally optimal solutions for the worst in function F19. The improved manta ray foraging optimization (ITMRFO) algorithm obtains globally optimal solutions for the worst in functions F19–F20. The improved manta ray foraging optimization (ITMRFO) algorithm obtains globally optimal solutions for the mean in functions F19–F20.

**Table 7.** Optimization results of the four optimization algorithms for fixed-dimensional multimodal test functions.

| Function | Value | SSA | PSO | MRFO | ITMRFO |
|---|---|---|---|---|---|
| **F14** | **Optimum** | $\mathbf{9.98 \times 10^{-1}}$ | $\mathbf{9.98 \times 10^{-1}}$ | $\mathbf{9.98 \times 10^{-1}}$ | $\mathbf{9.98 \times 10^{-1}}$ |
| | **Worst** | $1.2671 \times 10$ | $\mathbf{9.98 \times 10^{-1}}$ | $\mathbf{9.98 \times 10^{-1}}$ | $2.9821$ |
| | **Mean** | $1.02478 \times 10$ | $\mathbf{9.98 \times 10^{-1}}$ | $\mathbf{9.98 \times 10^{-1}}$ | $1.1964$ |
| | **Standard** | $4.299$ | $1.698 \times 10^{-10}$ | $\mathbf{5.8312 \times 10^{-17}}$ | $6.0541 \times 10^{-1}$ |
| **F15** | **Optimum** | $3.0849 \times 10^{-4}$ | $6.4758 \times 10^{-4}$ | $3.0749 \times 10^{-4}$ | $\mathbf{3.0749 \times 10^{-4}}$ |
| | **Worst** | $\mathbf{4.8808 \times 10^{-4}}$ | $2.973 \times 10^{-2}$ | $1.2232 \times 10^{-4}$ | $7.-04 \times 10^{-4}$ |
| | **Mean** | $\mathbf{3.3932 \times 10^{-4}}$ | $1.3029 \times 10^{-2}$ | $4.2958 \times 10^{-4}$ | $3.444 \times 10^{-4}$ |
| | **Standard** | $\mathbf{3.9562 \times 10^{-5}}$ | $1.0006 \times 10^{-2}$ | $3.166 \times 10^{-4}$ | $1.1039 \times 10^{-4}$ |
| **F16** | **Optimum** | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ |
| | **Worst** | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ |
| | **Mean** | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ | $\mathbf{-1.0316}$ |
| | **Standard** | $8.6536 \times 10^{-8}$ | $7.299 \times 10^{-5}$ | $6.5843 \times 10^{-16}$ | $\mathbf{6.1158 \times 10^{-16}}$ |
| **F17** | **Optimum** | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ |
| | **Worst** | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ |
| | **Mean** | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ | $\mathbf{3.9789 \times 10^{-1}}$ |
| | **Standard** | $4.9682 \times 10^{-7}$ | $1.6488 \times 10^{-5}$ | $\mathbf{0}$ | $\mathbf{0}$ |
| **F18** | **Optimum** | $\mathbf{3}$ | $\mathbf{3}$ | $\mathbf{3}$ | $\mathbf{3}$ |
| | **Worst** | $\mathbf{3}$ | $3.0017$ | $\mathbf{3}$ | $\mathbf{3}$ |
| | **Mean** | $\mathbf{3}$ | $3.0002$ | $\mathbf{3}$ | $\mathbf{3}$ |
| | **Standard** | $2.1119 \times 10^{-6}$ | $3.5805 \times 10^{-4}$ | $1.686 \times 10^{-15}$ | $\mathbf{1.3946 \times 10^{-15}}$ |
| **F19** | **Optimum** | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ |
| | **Worst** | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ |
| | **Mean** | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ | $\mathbf{-3.8628}$ |
| | **Standard** | $3.6072 \times 10^{-5}$ | $3.7109 \times 10^{-3}$ | $\mathbf{2.6543 \times 10^{-15}}$ | $1.4074 \times 10^{-13}$ |

**Table 7.** *Cont.*

| Function | Value | SSA | PSO | MRFO | ITMRFO |
|---|---|---|---|---|---|
| **F20** | **Optimum** | **−3.322** | −3.3215 | **−3.322** | −3.3219 |
| | **Worst** | −3.1149 | −2.7015 | **−3.2031** | −3.1144 |
| | **Mean** | −3.2519 | −3.1 | −3.2586 | **−3.2683** |
| | **Standard** | $7.7881 \times 10^{-2}$ | $1.9408 \times 10^{-1}$ | $\mathbf{6.0328 \times 10^{-2}}$ | $6.1883 \times 10^{-2}$ |
| **F21** | **Optimum** | $\mathbf{-1.0153 \times 10}$ | $\mathbf{-1.0153 \times 10}$ | $\mathbf{-1.0153 \times 10^{-1}}$ | $\mathbf{-1.0153 \times 10}$ |
| | **Worst** | $\mathbf{-1.0151 \times 10}$ | −2.6284 | −5.0552 | −5.0368 |
| | **Mean** | $\mathbf{-1.01525 \times 10}$ | −8.1955 | −7.9441 | −8.383 |
| | **Standard** | $\mathbf{5.597 \times 10^{-4}}$ | 2.6096 | 2.5694 | 2.4039 |
| **F22** | **Optimum** | $\mathbf{-1.0403 \times 10}$ | $\mathbf{-1.0403 \times 10}$ | $\mathbf{-1.0403 \times 10}$ | $\mathbf{-1.0403 \times 10}$ |
| | **Worst** | $\mathbf{-1.0402 \times 10}$ | −2.7615 | −3.7243 | −3.7077 |
| | **Mean** | $\mathbf{-1.0403 \times 10}$ | −9.6874 | −8.7629 | −8.2068 |
| | **Standard** | $\mathbf{2.4731 \times 10^{-4}}$ | 1.873 | 2.5592 | 2.6558 |
| **F23** | **Optimum** | $\mathbf{-1.0536 \times 10}$ | $\mathbf{-1.0536 \times 10}$ | $\mathbf{-1.0536 \times 10}$ | $\mathbf{-1.0536 \times 10}$ |
| | **Worst** | $\mathbf{-1.0535 \times 10}$ | $-1.0306 \times 10$ | −3.8354 | −5.0093 |
| | **Mean** | $\mathbf{-1.0536 \times 10}$ | $-1.0487 \times 10$ | −9.0512 | −9.3896 |
| | **Standard** | $\mathbf{2.5315 \times 10^{-4}}$ | $6.4829 \times 10^{-2}$ | 2.5151 | 2.0902 |

From Figures 5 and 6, it can be seen that the sparrow search algorithm (SSA) converges with higher accuracy on functions F21–F23 than the other three algorithms. The manta ray foraging optimization (MRFO) algorithm converges with higher accuracy on functions F14, F17, and F19. The improved manta ray foraging optimization (ITMRFO) algorithm converges faster and with higher accuracy on functions F15–F18 and F21. In addition, the ITMRFO algorithm converges faster and more consistently than the sparrow search algorithm (SSA).
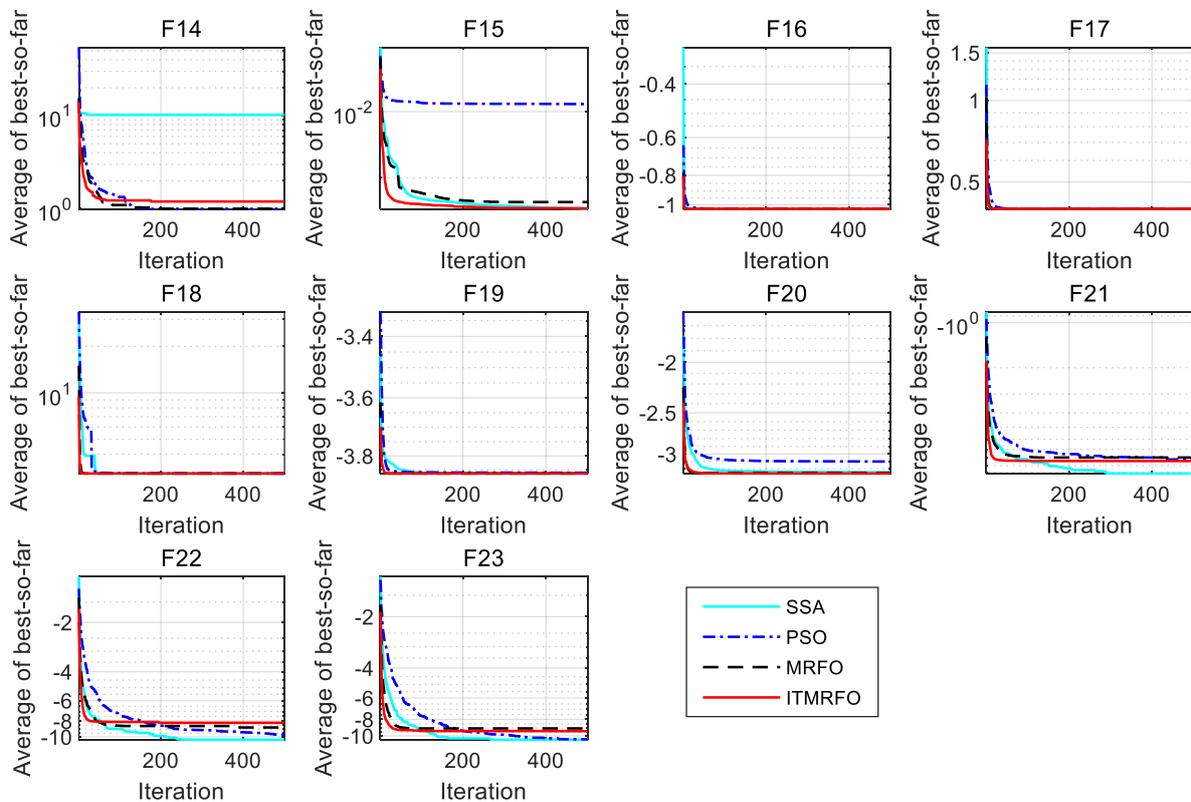


**Figure 5.** Convergence characteristics of the four algorithms on fixed-dimensional multimodal test functions.
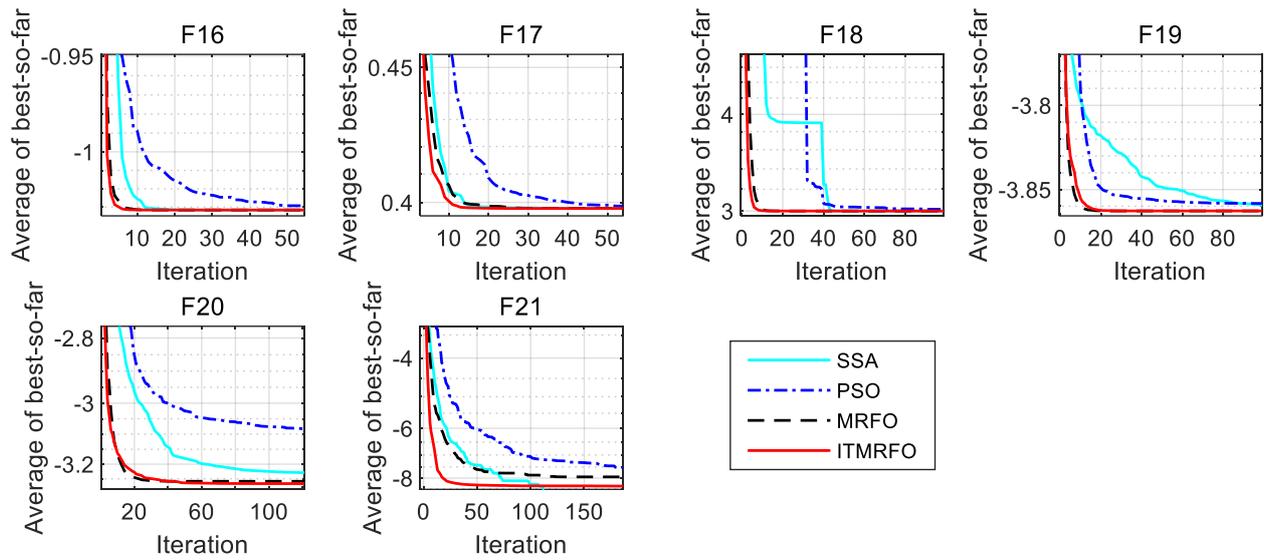
**Figure 6.** Partial enlargement of F16–F21.

The above analysis shows that all four intelligent optimization algorithms have better optimization results in fixed-dimensional multimodal test functions. Still, the improved manta ray foraging optimization (ITMRFO) algorithm has superior performance in dealing with fixed-dimensional multimodal test functions, with high convergence accuracy and mostly the fastest convergence speed.

Based on the optimization results of the four intelligent algorithms in 23 bench-mark test functions, it can be concluded that the improved manta ray foraging optimization (ITMRFO) algorithm is faster in terms of convergence speed and has higher stability and accuracy of convergence.

### 2.4. Probabilistic Neural Networks

The RBF [40] neural network, a feed-forward network based on the rule of Bayes, was developed by Dr. D.F. Speeht in 1989 [41]. The probabilistic neural network (PNN) is a variation of this network. Probabilistic neural networks (PNNs) are quick to learn, simple to train, and highly accurate in classifying objects. The global optimal solution probabilistic neural network (PNN) consists of the input layer, modal layer, accumulation layer, and output layer. The schematic diagram of the probabilistic neural network (PNN) is shown in Figure 7.
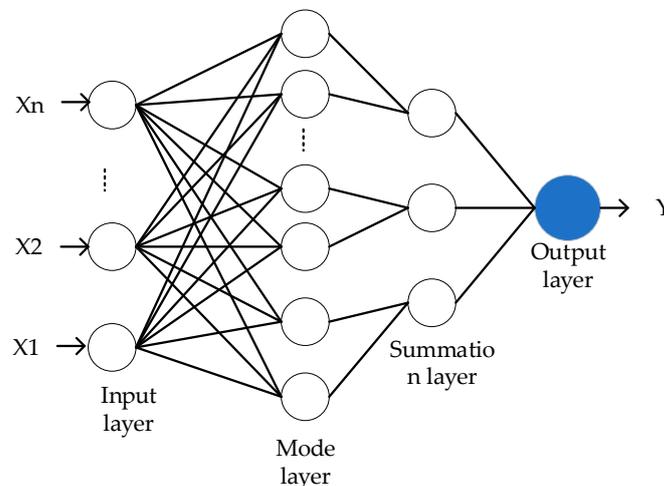


**Figure 7.** Basic structure of the PNN.

The input layer simply feeds eigenvectors into the neural network without performing any calculations.

The output of each pattern unit is determined as the pattern layer computes the matching relationship between each pattern in the training set using the eigenvectors from the input layer.

$$P(X, W_a) = exp\left[-\frac{(X - W_a)^T(X - W_a)}{2\delta^2}\right] \tag{19}$$

The variable $W_a$ is the identification sample, which also serves as the weight of the probabilistic neural network (PNN) connecting the input values and modalities; a is the number of training samples; and the smoothing factor, δ, significantly affects the performance of the probabilistic neural network (PNN).

The summation layer sums and averages the output of neurons of the same class in the pattern layer, which is expressed as:

$$f(x) = \frac{\sum_{a=1}^{N} P(X, W_a)}{N} \tag{20}$$

The output layer, which accepts the output of the summation layer, is made up of competing neurons. It performs a direct threshold discrimination, and eventually determines the category that corresponds to the characteristic line vector. It is written as:

$$Y = argmax f(x) \tag{21}$$

### 2.5. ITMRFO-Based Optimization of PNN Networks

Traditional neural networks need to be trained on a large amount of data due to their complex structure, which hinders their capacity to generalize and make accurate predictions. In contrast, probabilistic neural network (PNN) models use fewer parameters, do not require initial weight setting, and reduce human subjectivities affecting the model parameters and randomness in the network architecture.

The probabilistic neural network is a parallel method that uses the Bayesian minimum risk criterion to solve pattern classification problems [42,43]. A probabilistic neural network is "trained" using input training samples to establish the network's size, neuron centroids, connection thresholds, and weights. The threshold determines the width of the radial basis function. The larger the threshold, the greater the decay of the radial basis function as the input vector moves away from the weight vector, which affects the accuracy of the probabilistic neural network model. The probabilistic neural network (PNN) estimates the smoothing factor δ based on the minimum aver-age distance between samples. As a result, identical data points will bias the smoothing factor estimate, affecting the operating state and accuracy of the probabilistic neural network (PNN) model. The smoothing factor automatically adjusts the threshold of the pattern layer. Therefore, only the smoothing factor needs to be optimized.

The improved manta ray foraging optimization (ITMRFO) algorithm is used to optimize the smoothing factor of the probabilistic neural network (PNN). The ITMR-FO-PNN flow chart is shown in Figure 8.

The workflow for the ITMRFO-PNN model is as follows: (1) input the training sample data; (2) establish the population and specify the manta ray parameters (select the best population with the elite-opposition-based algorithm); (3) determine Rand < 0.5. If it holds, perform spiral foraging. Perform chain foraging if it does not hold; (4) calculate the fitness value and update the optimum position; (5) perform rollover foraging and update the position. (6) calculate the fitness value and update the optimum position; and (7) judge whether the end condition is satisfied; if so, output the optimum value; otherwise repeat steps 2–7.
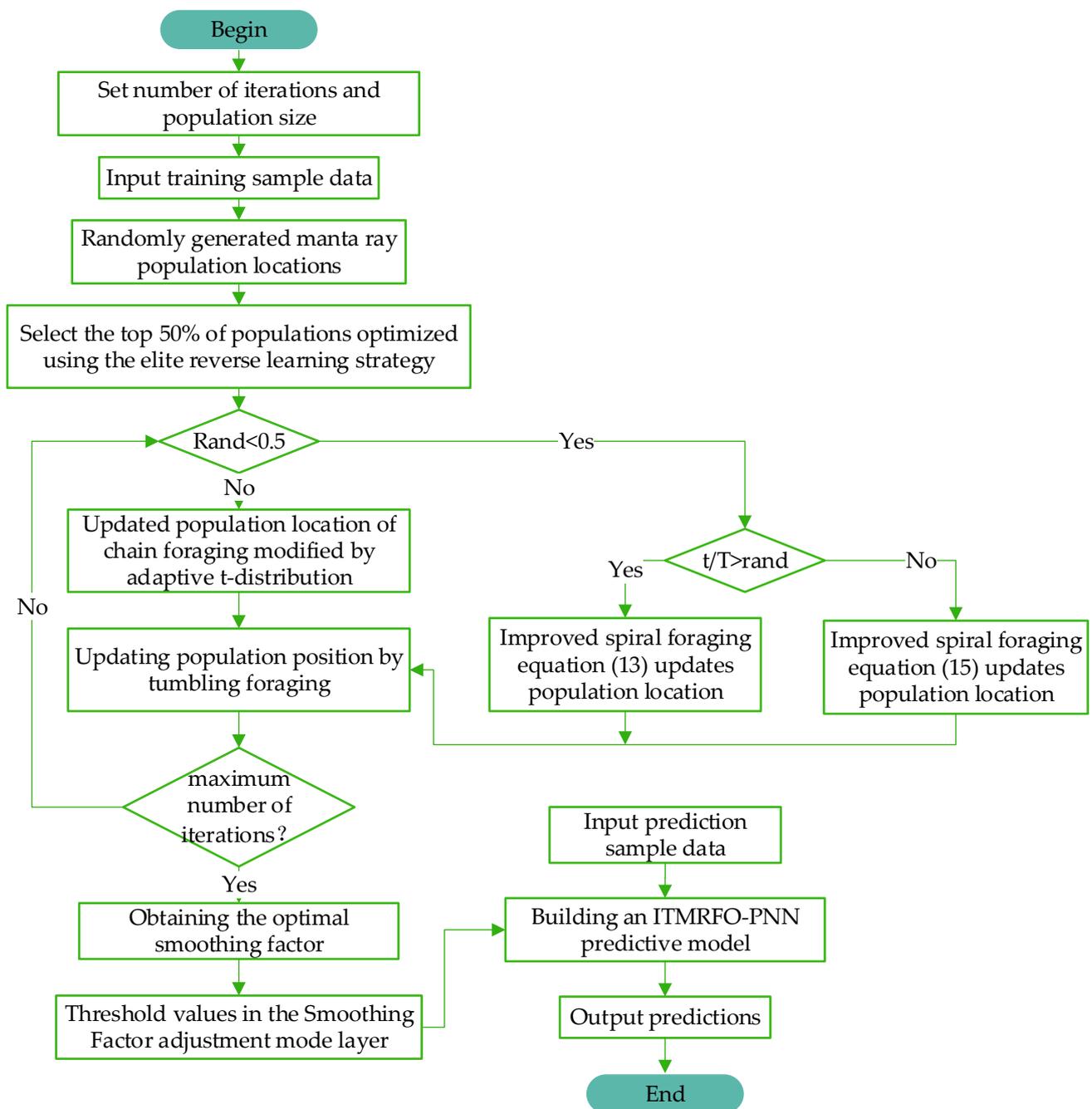
**Figure 8.** Basic flow chart of the ITMRFO-PNN.

*2.6. Model Structure*

The eigenvectors of draft tube pressure fluctuation are extracted by wavelet analysis, and the fuzzy c-means (FCM) clustering algorithm classifies the extracted eigenvectors. The probabilistic neural network (PNN) has the advantages of a simple structure, fast training speed, strong classification ability, etc. By training the vibration characteristic vector of draft tube pressure fluctuation, the probabilistic neural network (PNN) model is established to analyze the draft tube pressure fluctuation. The threshold and smoothing factor also impact how accurately the probabilistic neural network (PNN) can classify data. In this study, the smoothing factor of the probabilistic neural network (PNN) was optimized by the improved manta ray foraging optimization (ITMRFO) algorithm. The optimized smoothing factor automatically modifies the threshold. The ITMRFO-PNN network model

with more accurate classification results was established. The pressure fluctuation analysis model of the draft tube of the hydraulic turbine was established.

## 3. Experimental Process

### 3.1. Analysis of Experimental Data

The total installed capacity of the hydropower station considered is 1000 MW, and it is equipped with four mixed flow turbines with a single unit capacity of 250 MW. The rated head of the water turbine at the hydropower station is 305 m, and the rated speed is 333.3 r/min. In the hydropower station, the four turbines are of the same type and are in normal operation. This study classified and identified the characteristics of the pressure fluctuation signal of the draft tube of four normal running turbines. The sensor type was the INV9828 piezoelectric acceleration sensor, and the sampling frequency of the vibration signal was 1024 Hz.

The vibration signal acquisition process is shown below:

- Vibration measurement points were set up at the inlet and outlet of the draft tube of Units 1–4 of the hydropower station.
- The data measured at the inlet and outlet vibration points of the same unit were combined into a group of data. The pressure fluctuation data of four units in three different time periods was recorded during water pumping, and the pressure fluctuation data of four units in three different time periods were recorded during power generation.
- Finally, the pressure fluctuation data of the draft tube of these four units (24 sets of data in total) were analyzed.

The twenty-four groups of test data can be seen in Table 8. W1 represents the pumping time of Unit 1, and P1 indicates when Unit 1 generates power. The draft tube pressure fluctuation data for Unit 1 (number 11A) are shown in Table 9.

**Table 8.** The twenty-four groups of test data.

| Unit 1 | | Unit 2 | | Unit 3 | | Unit 4 | |
|---|---|---|---|---|---|---|---|
| **W1** | **P1** | **W2** | **P2** | **W3** | **P3** | **W4** | **P4** |
| 11A | 11B | 21A | 21B | 31A | 31B | 41A | 41B |
| 12A | 12B | 22A | 22B | 32A | 32B | 42A | 42B |
| 13A | 13B | 23A | 23B | 33A | 33B | 43A | 43B |

**Table 9.** Partial pressure fluctuation data of 11A.

| Serial Number | Import | Export |
|---|---|---|
| 1 | 0.561 | 0.601 |
| 2 | 0.561 | 0.601 |
| 3 | 0.561 | 0.601 |
| 4 | 0.561 | 0.601 |
| 5 | 0.561 | 0.601 |
| 6 | 0.561 | 0.601 |
| 7 | 0.561 | 0.601 |
| 8 | 0.561 | 0.601 |
| 9 | 0.561 | 0.601 |
| 10 | 0.561 | 0.601 |
| 11 | 0.561 | 0.601 |
| 12 | 0.561 | 0.601 |
| 13 | 0.561 | 0.601 |
| 14 | 0.561 | 0.601 |
| 15 | 0.561 | 0.601 |

*3.2. Processing of Hydraulic Turbine Vibrations*

There are 24 sets of data, each containing vibration data measured at two vibration measurement points. Firstly, we adopted a classic fixed threshold denoising method that is widely used in the field of signal processing. Compared with adaptive threshold methods, fixed threshold methods use predetermined threshold values to denoise signals, simplifying the process of threshold selection. At this stage, we used the Coif4 wavelet from the Coiflet wavelet series as the wavelet basis function. The Coif4 wavelet is a wavelet basis function with order 4 in the Coiflet series, which exhibits good characteristics in both the time and frequency domains. This combination can effectively denoise signals and capture their local features and detailed information, improving the accuracy and reliability of signal analysis.

The db5 wavelet basis function was used to decompose the signal, and then, the low frequency coefficients and high frequency coefficients in the wavelet decomposition structure were extracted. The db5 wavelet is the Daubechies wavelet series (dbN for short). The dbN function is a compactly supported orthonormal wavelet, which makes discrete wavelet analysis possible. Most dbN wavelets do not have symmetry. For orthogonal wavelet functions, the asymmetry is very obvious. Regularity increases with the increase in N. The stronger the normality is, the smoother the wavelet function is, the greater the vanishing moment is, and the wavelet coefficients will be suppressed. Therefore, the db5 wavelet was selected as the wavelet basis function of discrete wavelet transform in this experiment.

Vibration data were decomposed and reconstructed via one-dimensional discrete wavelet transform, and the low-frequency coefficients $ca1$–$ca3$ and high-frequency coefficients $ch1$–$ch3$ were extracted. Then, the data were processed via two-dimensional discrete wavelet transform, and their reconstruction coefficients $a1$, $v1$, and $d1$ were extracted.

Where $ca1$–$ca3$ refers to the high-frequency coefficients in the first to third layers of the one-dimensional wavelet decomposition and $ch1$–$ch3$ refers to the low-frequency coefficients in the first to third levels of the wavelet. The 2D reconstructed low-frequency coefficients, vertical directional components of the 2D reconstructed high-frequency coefficients, and diagonal directional components of the 2D reconstructed high-frequency coefficients were designated as $a1$, $v1$, and $d1$, respectively.

As can be seen in Figure 9, the wavelet analysis method was used to denoise, decompose, and reconstruct the pressure fluctuation signals at the inlet and outlet of the draft tube (data 11A) of Unit 1 when pumping water and obtain $ca1$–$ca3$, $ch1$–$ch3$, $a1$, $v1$, and $d1$.

As can be seen in Figure 10, the wavelet analysis method was used to denoise, decompose and reconstruct the pressure fluctuation signals at the inlet and outlet of the draft tube (data 11B) of Unit 1 during power generation and obtain $ca1$–$ca3$, $ch1$–$ch3$, $a1$, $v1$, and $d1$.

Fuzzy c-means clustering is one of the most widely used algorithms in fuzzy clustering. According to the similarity between data samples, the fuzzy c-means (FCM) clustering algorithm classifies the samples with high similarity into the same class by iteratively optimizing the objective function. The schematic diagram of the fuzzy c-means (FCM) clustering algorithm classification is shown in Figure 11.

Each set of data contains data from two vibration measuring points at the draft tube inlet and outlet (all measuring points measure the same amount of data in length). The maximum, minimum, sum of squares, and standard deviations of $ca1$–$ca3$, $ch1$–$ch3$, $a1$, $v1$, and $d1$ are calculated as characteristic vectors, and each group had $36 \times 2$ characteristic vectors. Since there are only 24 sets of data in total, the amount of data is too small. Therefore, after 100 repetitions of 24 groups of data, 2400 groups of data are finally obtained, forming a data matrix of $2400 \times 72$.
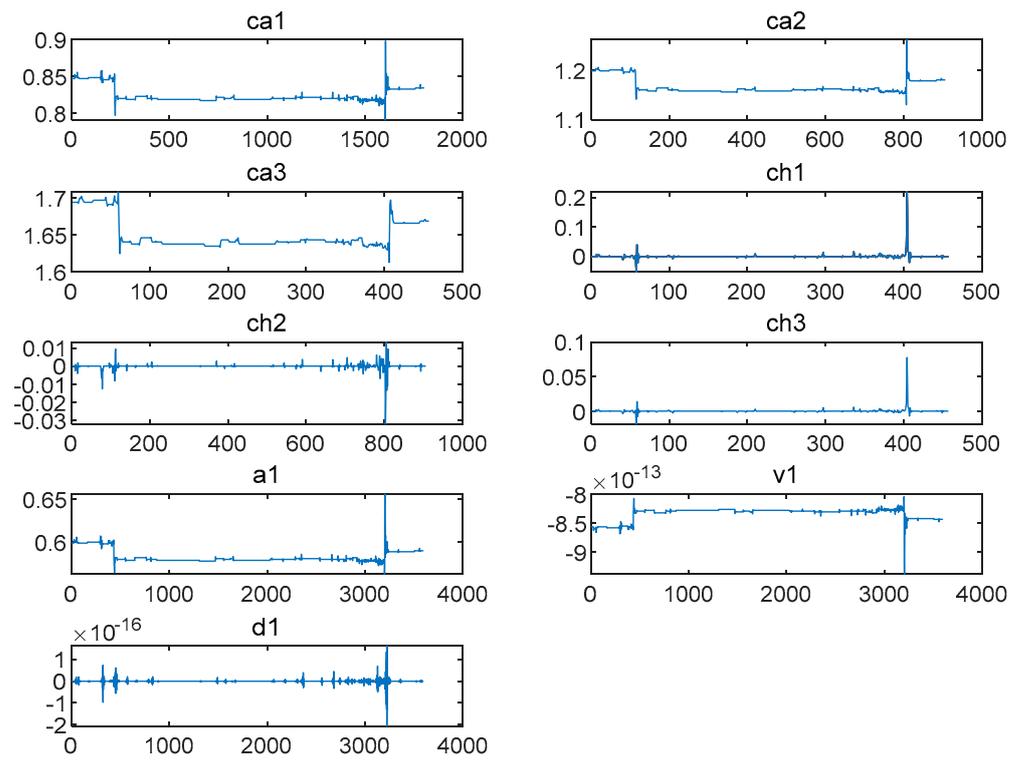
**Figure 9.** $ca1 - ca3, ch1 - ch3, a1, v1$, and $d1$ images extracted at the inlet during the pumping of Unit 1 (the data of 11A).
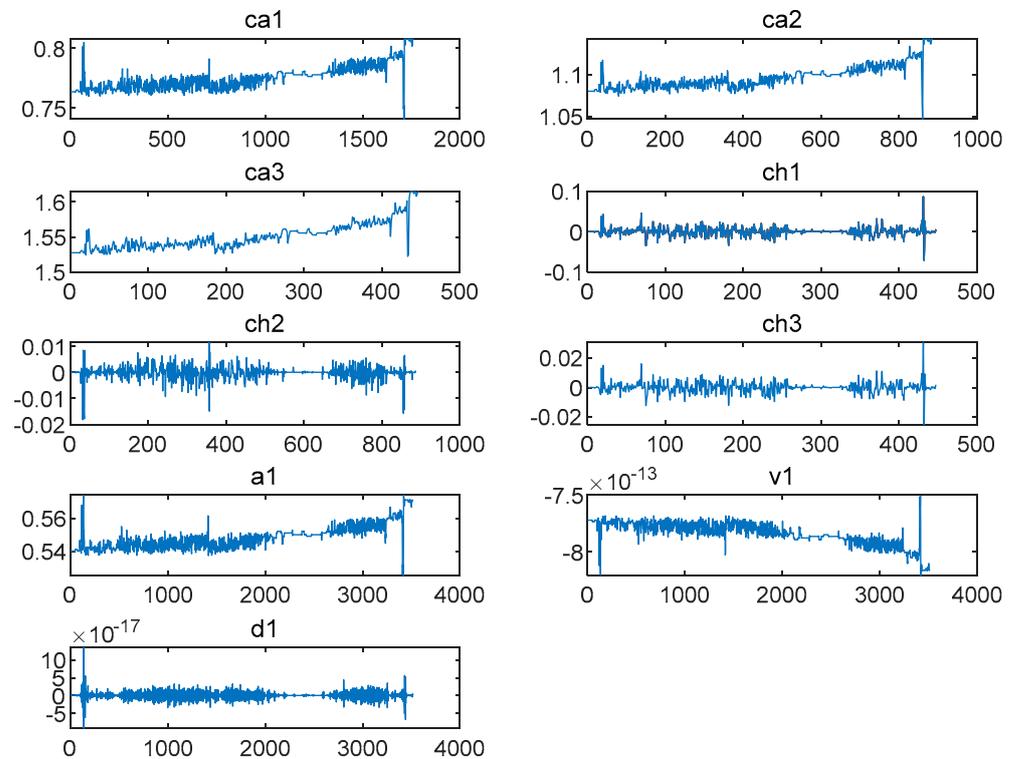


**Figure 10.** $ca1 - ca3, ch1 - ch3, a1, v1$, and $d1$ images extracted at the inlet during the power generation of Unit 1 (the data of 11B).
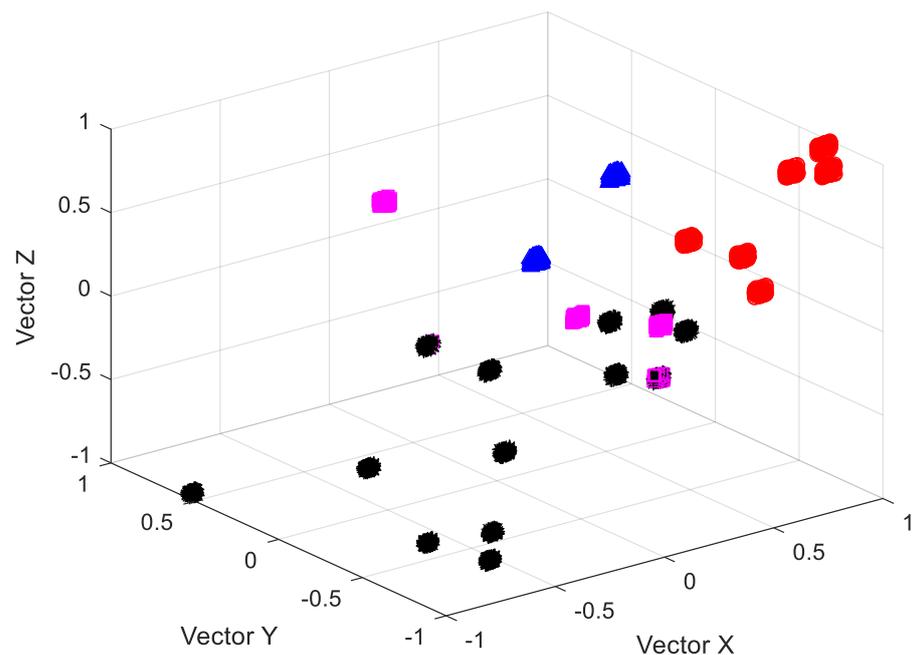
**Figure 11.** Classification map based on the fuzzy c-means clustering algorithm (FCM).

Then, the 2400 sets of data are classified as the characteristic vector matrix using the fuzzy c-means (FCM) clustering algorithm. This means using the fuzzy c-means (FCM) clustering algorithm for the initial clustering of pressure fluctuation characteristics of classification and set the number of clustering centers of fuzzy c-means (FCM) clustering algorithm to four, that is, 2400 samples were divided into four categories. The results of classification are as follows: there were 899 samples in the first category, 901 in the second category, 300 in the third category, and 300 in the fourth category.

The fuzzy c-means (FCM) clustering algorithm is used to carry out a random selection of fifty samples from each of the four classification results it processes, for a total of two hundred samples. The first 72 columns of each data sample are the eigenvectors of the sample. The seventy-third column is the output of the classification, represented by the numbers 1, 2, 3, and 4.

### 3.3. Analysis and Comparison of the PNN, the MRFO-PNN and the ITMRFO-PNN Model

The probabilistic neural network (PNN) model is suitable for classification problems. Its strong non-linear classification capability, the requirement to include the most informative characteristic sample of the eigenvectors, and other characteristics make it unique, which applies to the method suggested in Section 2.4.

MATLAB was used to carry out the simulation experiments. The probabilistic neural network (PNN) was used to analyze the 200 data samples selected in Section 3.2. The extracted eigenvectors were used as characteristic fusion input to the probabilistic neural network (PNN), while the output of the probabilistic neural network (PNN) was used to determine the classification of pressure fluctuation. A total of 80 were selected as test samples and 120 were selected as training input samples. Secondly, the probabilistic neural network (PNN) model created had 80 pattern layers, 4 output layers (corresponding to four classification categories), and 120 input layers. If the size of the smoothing factor is too small, the radial basis neurons in the probabilistic neural network will not be able to respond to all the intervals spanned by the input vectors. It will complicate the network calculation if it is too large. As a result, many numbers were manually selected throughout the calculating phase of this experiment to evaluate the categorization effect. The smoothing factor in the probabilistic neural network (PNN) model was selected as 0.9 because the network classification effect performed best when the SPEAD value was 0.9.

In Figure 12a–c, the horizontal coordinates represent the number of training samples, and the vertical coordinates represent the classification results. The 1–31 sample prediction model belongs to category 1, the 32–55 sample prediction model belongs to category 2, the 56–85 sample prediction model belongs to category 3, and the 86–120 sample prediction model belongs to category 4.
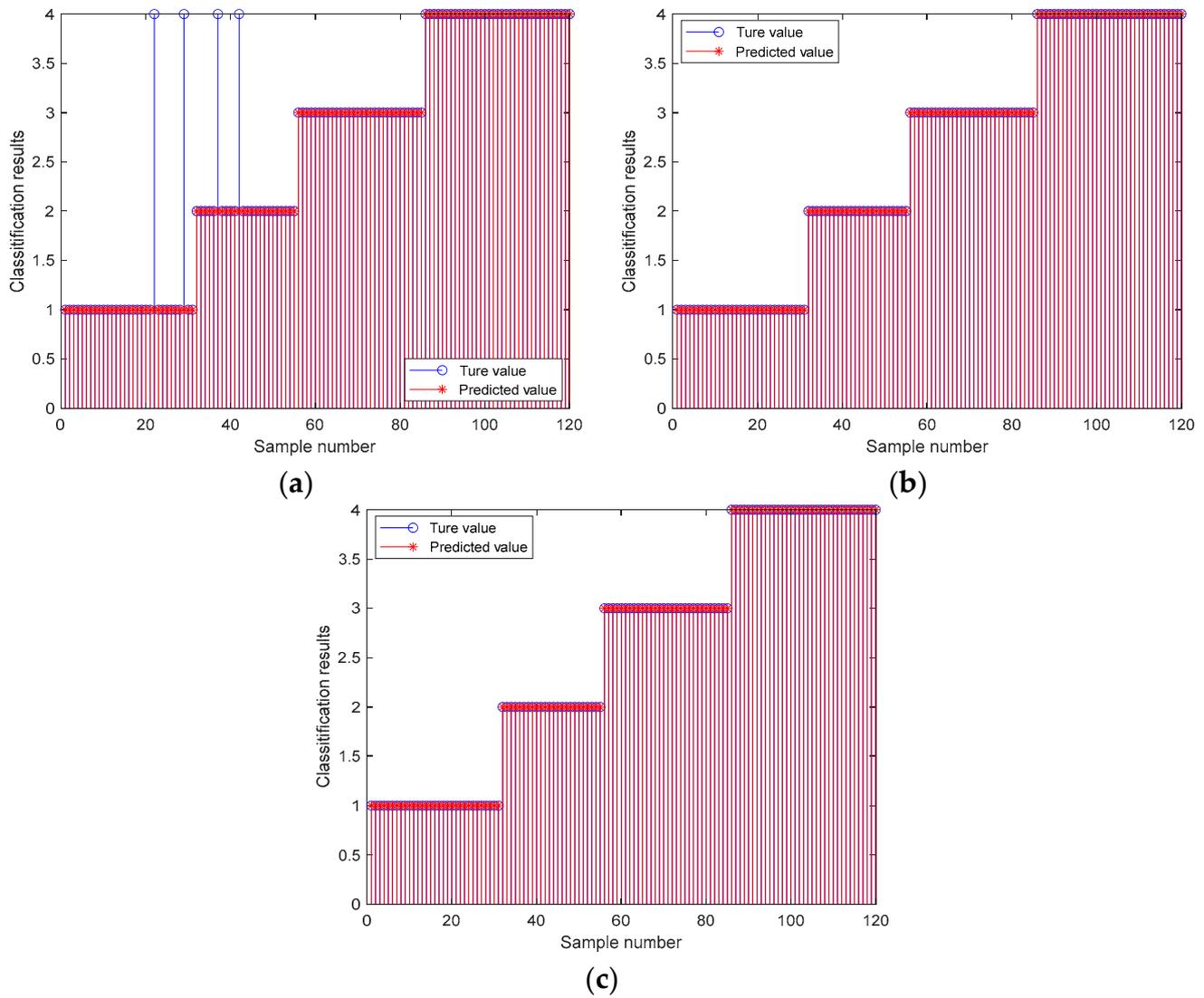


**Figure 12.** Training diagram of the PNN, MRFO-PNN, and ITMRFO-PNN. (**a**) PNN. (**b**) MRFO-PNN. (**c**) ITMRFO-PNN.

Figure 12a shows the results of probabilistic neural network (PNN) training, where the predicted values of the PNN training set are inconsistent with the true values; four samples were misclassified, and the training accuracy of the probabilistic neural network (PNN) was 96.67%. Figure 12b,c shows the training results for the two models: the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm (MRFO-PNN) and the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm (ITMRFO-PNN). With a training accuracy of 100% and accurate predictions, it can be seen that the predicted values of the training samples exactly match the true values.

In order to further test the extrapolation performance of the three models mentioned above, 120 training samples from the three models in Figure 12 were used to classify and predict the remaining 80 test samples. The results are shown in Figure 13. The prediction

patterns for samples 1–19 are category 1, samples 20–45 are category 2, samples 46–65 are category 3, and samples 66–80 are category 4. The horizontal coordinates indicate the test sample numbers, and the vertical coordinates indicate the classification results.
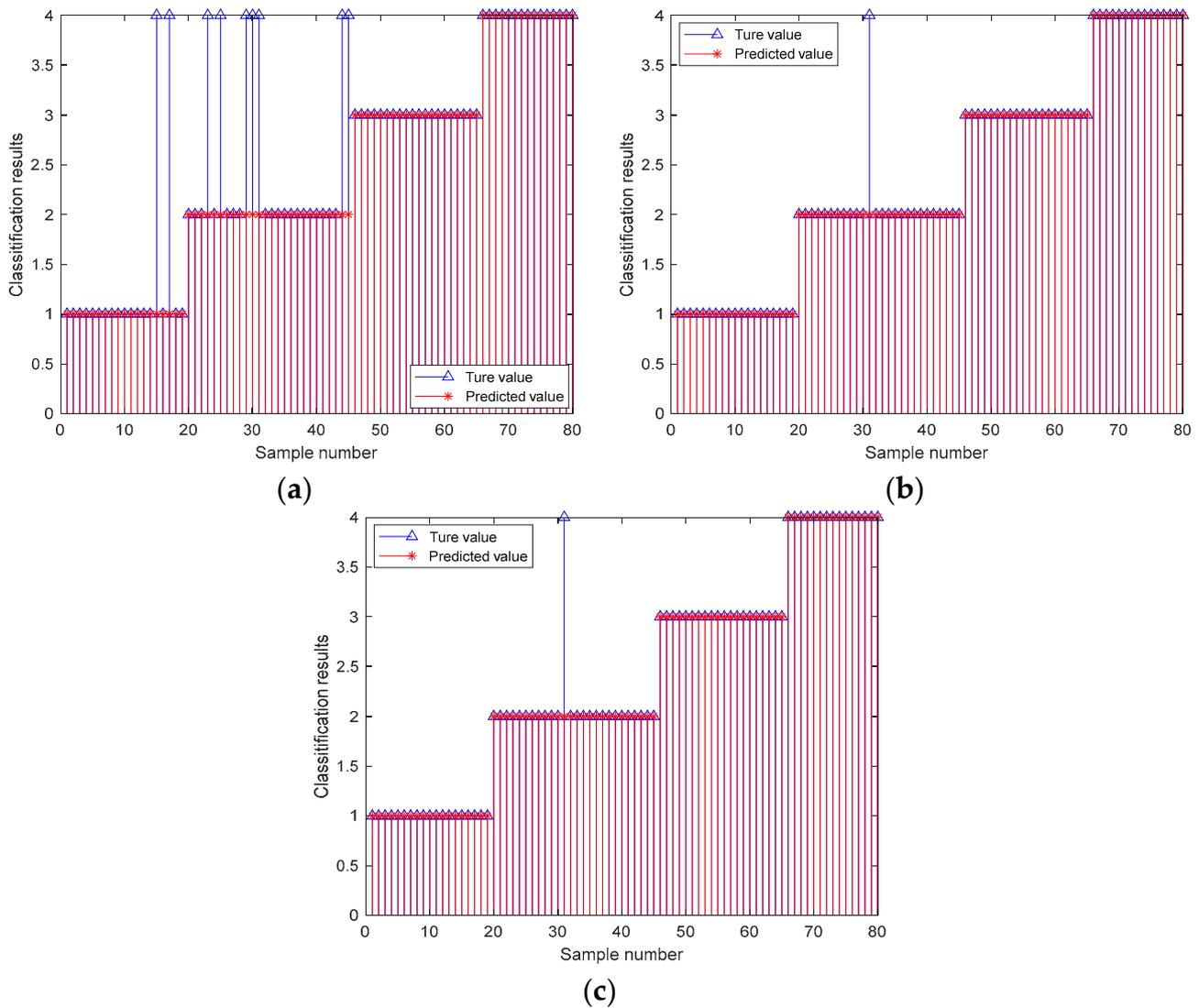


**Figure 13.** Prediction diagram of the PNN, MRFO-PNN, and ITMRFO-PNN. (**a**) PNN. (**b**) MRFO-PNN. (**c**) ITMRFO-PNN.

As can be seen in Figure 13a, the predicted values of the probabilistic neural network (PNN) test set were not consistent with the true values, with nine samples mis-classified, and the probabilistic neural network (PNN) test accuracy was 88.75%. From Figure 13b,c, it can be seen that the predicted values of the optimized probabilistic neural net-work model based on a manta ray foraging optimization algorithm (MRFO-PNN) model and optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) test set were inconsistent with the true values. Both had one sample misclassification, and both had a test accuracy of 96.67%, indicating high prediction accuracy. At this time, the smoothing factor for the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) was 0.31136, and the threshold was 2.674. The smoothing factor and threshold for the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) were 0.30993 and 3.3757, respectively.

Table 10 shows the results of identifying the pressure fluctuation characteristics of a draft tube based on a probabilistic neural network (PNN), an optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN), and an optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN). From Table 10, it can be concluded that the probabilistic neural network (PNN) model had two identification errors in category 1, seven in category 2, zero in category 3, and zero in category 4. The total identification rate of the model reached 88.75% (71/80). The optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) and optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) both had an identification error in the draft tube pressure fluctuation characteristic of category 2. The total identification rate of both models was 98.75% (79/80). Compared with the non-optimized probabilistic neural network (PNN) model, the identification accuracy of the draft tube pressure fluctuation of the turbine based on an optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) and an optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm (ITMRFO-PNN) was extremely high. It can be concluded that both the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm (MRFO-PNN) identification model and the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm (ITMRFO-PNN) identification model can effectively identify the draft tube pressure fluctuation of a turbine.

**Table 10.** Comparison of accuracy, precision, recall, and F1-score of the three models.

| Model | Category 1 | Category 2 | Category 3 | Category 4 | Accuracy % |
|---|---|---|---|---|---|
| PNN | 17/19 | 19/26 | 20/20 | 15/15 | 88.75 (71/80) |
| MRFO-PNN | 19/19 | 25/26 | 20/20 | 15/15 | 98.75 (79/80) |
| ITMRFO-PNN | 19/19 | 25/26 | 20/20 | 15/15 | 98.75 (79/80) |

*3.4. Performance Comparison of the Three Models*

We compared the results of our proposed optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) with the probabilistic neural network (PNN) model and the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) after conducting experiments. Firstly, we will discuss the first performance parameter, which is the confusion matrix.

The confusion matrix plot of the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) for the identification of draft tube pressure fluctuation signals is shown in Figures 14 and 15. The columns represent the actual labeled instances of the classes, while the rows represent the predicted instances of the actual classes. The correct class counts predicted by the model matrix are displayed in the diagonal positions, while the counts of the model's incorrect predictions are displayed outside the diagonal matrix.

Our proposed model, the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN), which is an optimized probabilistic neural network (PNN) model using the improved manta ray foraging optimization algorithm, achieved 100% accuracy in the training samples, as shown in Figure 14. In the testing samples, it achieved a prediction accuracy of 98.7%, as shown in Figure 15.

According to the results in Table 11, our proposed optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) exhibited more competitive results compared to the PNN model. The optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) and optimized probabilistic neural network model based on

an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) exhibited 100% in terms of accuracy, precision, recall, and F1-score on the training samples, surpassing the performance of the probabilistic neural network (PNN) model. Furthermore, on the testing samples, the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) and optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) achieved accuracy of 98.75%, precision of 98.44%, recall of 99.04%, and an impressive F1-score of 98.74% for the classification of pressure fluctuation signals. These results highlight the competitive edge of the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) and an optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) in accurately and effectively classifying pressure fluctuation signals.
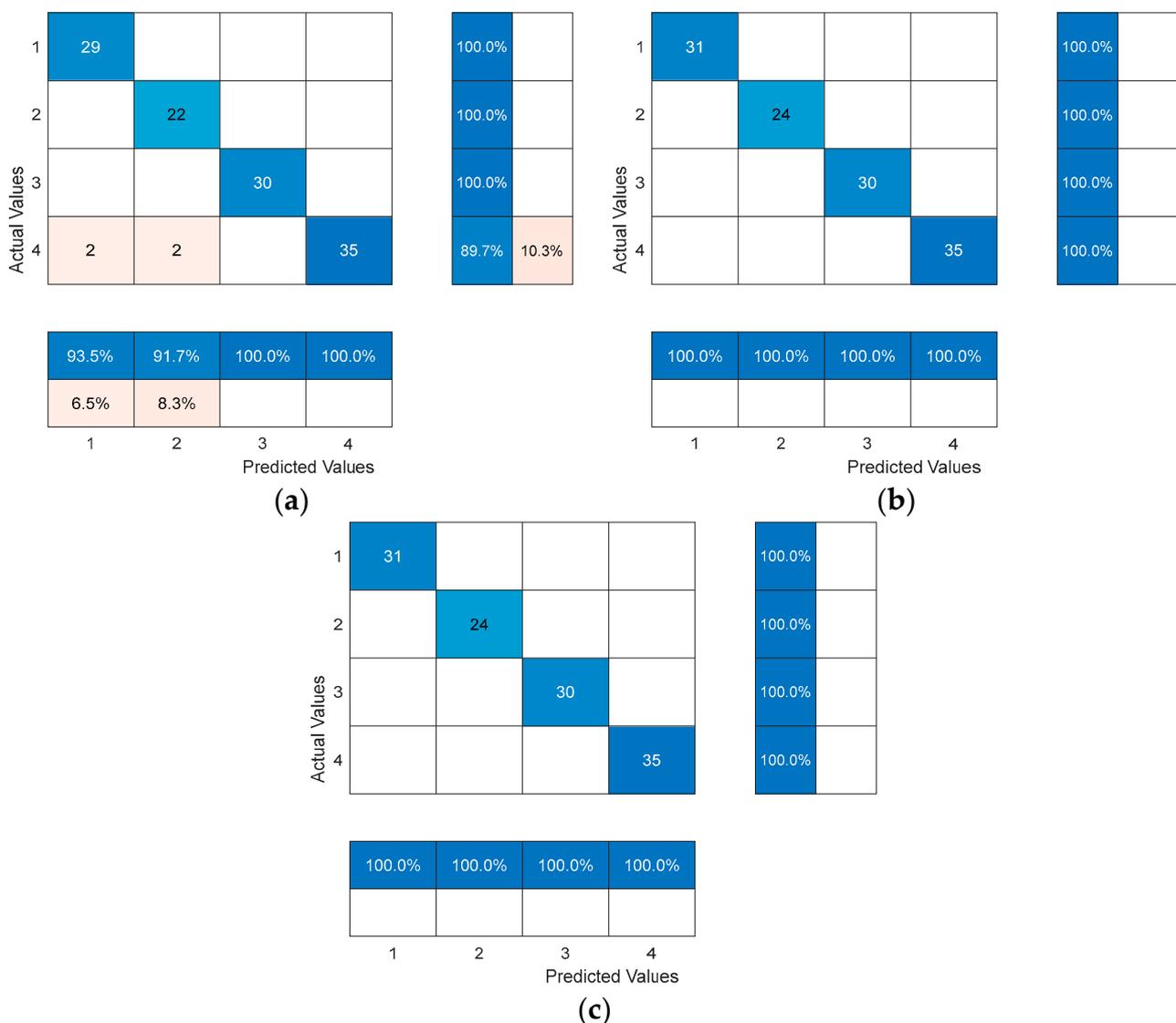


**Figure 14.** The confusion matrix of the PNN, MRFO-PNN, and ITMRFO-PNN models for the classification of pressure fluctuation signal training sets. (**a**) PNN. (**b**) MRFO-PNN. (**c**) ITMRFO-PNN.
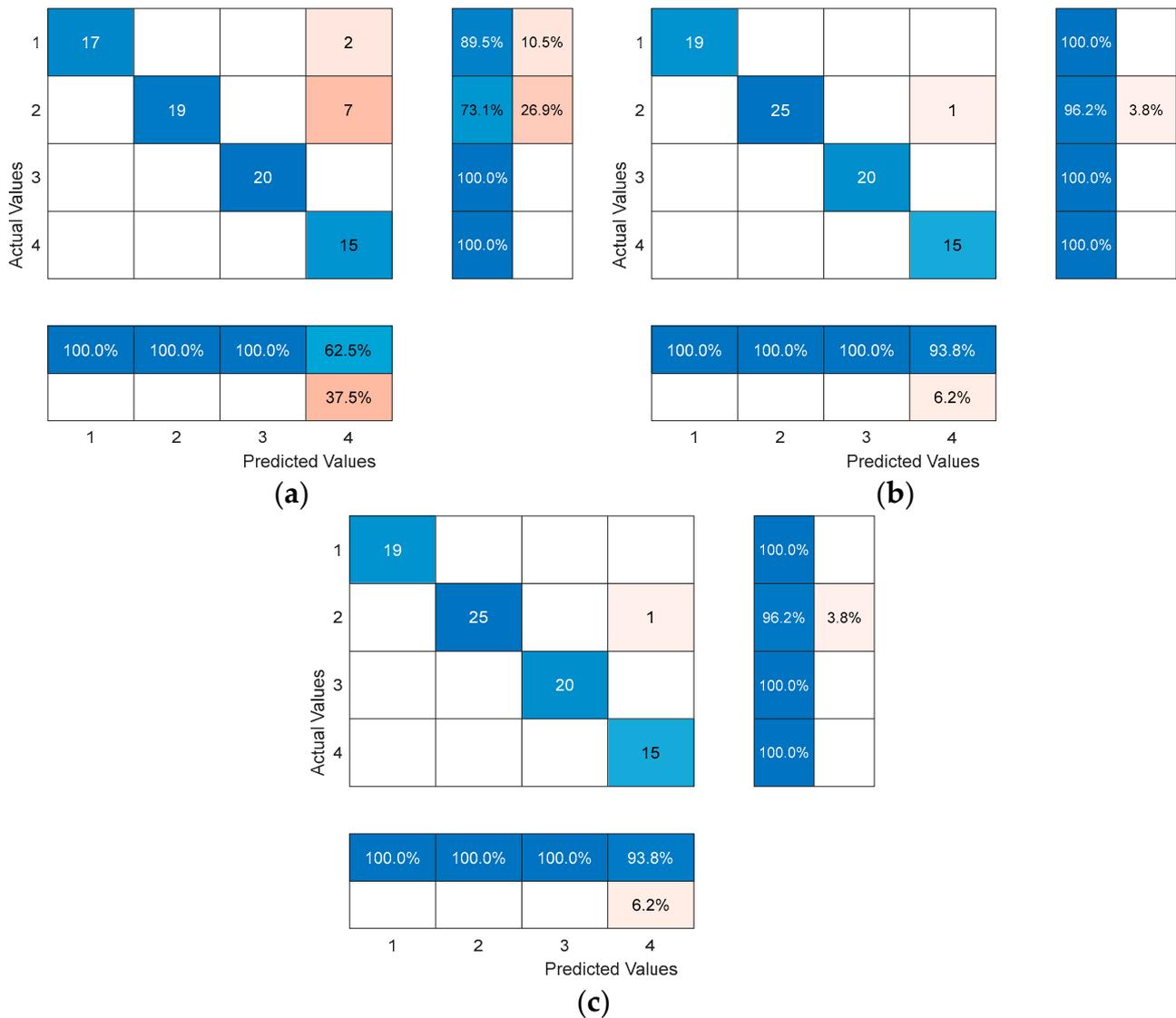
**Figure 15.** The confusion matrix of the PNN, MRFO-PNN, and ITMRFO-PNN models for the classification of the pressure fluctuation signal test sets. (**a**) PNN. (**b**) MRFO-PNN. (**c**) ITMRFO-PNN.

**Table 11.** Comparison of the characteristic identification results of the three models.

| Model | Training/Testing Results | Accuracy % | Precision | Recall | F1-Score |
|---|---|---|---|---|---|
| PNN | Training | 96.67 | 96.30 | 97.43 | 96.87 |
| | Testing | 88.75 | 92.63 | 90.64 | 90.63 |
| MRFO-PNN | Training | 100 | 100 | 100 | 100 |
| | Testing | 98.75 | 98.44 | 99.04 | 98.74 |
| ITMRFO-PNN | Training | 100 | 100 | 100 | 100 |
| | Testing | 98.75 | 98.44 | 99.04 | 98.74 |

Figure 16 shows the training sample identification error rates for the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) and the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm (ITMRFO-PNN). According to Figure 16 and Table 11, the identification accuracy of the training sample (100%) and the test sample

(98.75%) were consistent. However, compared with the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) (11 iterations achieved an error rate of zero), the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) achieved an error rate of zero for training sample identification in fewer iterations (only seven iterations achieved an error rate of zero). In addition, the identification error rate of the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) at the beginning of the iteration was less than 0.04, while the identification error rate of the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN) at the end of the iteration was higher than 0.06. In other words, the identification error rate of the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) at the beginning of iteration was lower than that of the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN).
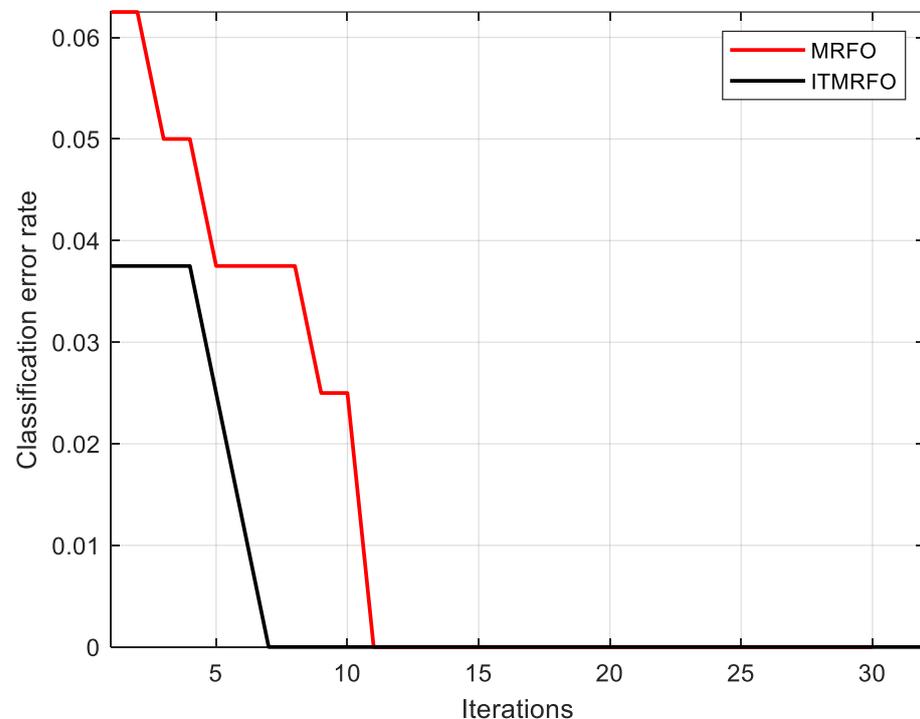


**Figure 16.** Training error rate of the MRFO-PNN and the ITMRFO-PNN.

## 4. Conclusions

(1) Discrete wavelet transform was used to decompose and reconstruct the collected pressure fluctuation signal, and the maximum, minimum, square sum, and standard deviation of the nine coefficients $ca1$–$ca3$, $ch1$–$ch3$, $a1$, $v1$, and $d1$ were taken as characteristics. This is a new characteristic extraction method, which provides a new research idea for subsequent data characteristic extraction methods. Following this, the fuzzy c-means (FCM) clustering algorithm was utilized to automatically classify the signal based on its own properties.

(2) Aiming to solve the problem of the manta ray foraging optimization (MRFO) algorithm often falling into the local optimum, the algorithm was improved four times. The elite-opposition-based learning strategy was used to optimize the initial population. The first 50% of the initial population was chosen as the new population to obtain a high-quality population. Adaptive t distribution was used instead of the chain factor to optimize the individual renewal strategy at the chain foraging site.

In chain foraging and spiral foraging, the partial expressions for multiplying by r were removed to ensure the stability of the algorithm. In order to evaluate the performance of the improved manta ray foraging optimization (ITMRFO) algorithm, it was compared with three other algorithms including particle swarm optimization (PSO). The results showed that the improved manta ray foraging optimization (ITMRFO) algorithm has high accuracy and efficiency.

(3) To enhance the identification accuracy of the probabilistic neural network (PNN), an improved manta ray foraging optimization (ITMRFO) algorithm was employed to optimize the smoothing factor of the probabilistic neural network, and an ITMRFO-PNN identification model was developed. The model was used to identify the characteristics of pressure fluctuation signals in the draft tube of a hydraulic turbine. The identification results of the probabilistic neural network (PNN), the optimized probabilistic neural network model based on a manta ray foraging optimization algorithm model (MRFO-PNN), and the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) were compared. The identification accuracy for the training samples was 96.67%, 100%, and 100% for the PNN, MRFO-PNN, and ITMRFO-PNN models, respectively. For the test samples, the identification accuracy was 88.75%, 98.75%, and 98.75% for the PNN, MRFO-PNN, and ITMRFO-PNN models, respectively. And we compared these three models through confusion matrix, accuracy, precision, recall, and F1-score. The experimental results show that both MRFO-PNN and ITMRFO-PNN models have better performance than PNN models. However, compared to the MRFO-PNN model (which achieved a zero-bit error rate in eleven iterations), the ITMRFO-PNN model achieved a zero-bit error rate for training sample identification in fewer iterations (only seven iterations achieved a zero-bit error rate). In addition, the identification error rate of the ITMRFO-PNN model at the beginning of the iteration was lower than that of the MRFO-PNN model. Therefore, compared with other algorithms, the improved manta ray foraging optimization (ITMRFO) algorithm has obvious advantages in optimizing a probabilistic neural network (PNN) to identify the pressure fluctuation signal in the draft tube of a hydraulic turbine.

Although the optimized probabilistic neural network model based on an improved manta ray foraging optimization algorithm model (ITMRFO-PNN) demonstrates excellent performance, it also has certain limitations:

(1) The ITMRFO-PNN identification model has much higher identification accuracy than the optimized PNN network model. However, the identification accuracy of this model has not yet reached 100%, and it needs to be improved.

(2) Due to the limitation of the experiment, the amount of data in this study was too small, and using only the pressure fluctuation signal of the Francis turbine draft tube led to poor comparability of the data. This also means that data from other types of hydraulic turbine units cannot be verified, and further verification is needed for state identification of other types.

Therefore, future research can select additional types of sample data and conduct further in-depth research based on the optimized probabilistic neural network model based on the improved manta ray foraging optimization algorithm model (ITMRFO-PNN) with the feature identification methods proposed in this study.

**Author Contributions:** Conceptualization, X.L., L.Z., W.Z. and L.W.; methodology, X.L. and L.W.; software, W.Z., H.Y. and X.L.; validation, X.L., L.W. and L.Z.; formal analysis, X.L.; investigation, X.L.; resources, L.W.; data curation, X.L. and L.W.; writing—original draft preparation, X.L.; writing—review and editing, X.L., H.Y. and L.W.; visualization, X.L. and Q.C.; supervision, X.L. and L.W.; funding acquisition, H.Y.; project administration, Q.C., W.Z. and L.W. All authors have read and agreed to the published version of the manuscript.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The numerical and experimental data used to support the findings of this study are included within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Yang, B.; Bo, Z.; Yawu, Z.; Xi, Z.; Yalan, J. The vibration trend prediction of hydropower units based on wavelet threshold denoising and bi-directional long short-term memory network. In Proceedings of the 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA), Shenyang, China, 22–24 January 2021. [CrossRef]
2. Feng, Z.; Chu, F. Nonstationary Vibration Signal Analysis of a Hydroturbine Based on Adaptive Chirplet Decomposition. *Struct. Health Monit.* **2007**, *6*, 265–279. [CrossRef]
3. Zhao, W.; Presas, A.; Egusquiza, M.; Valentín, D.; Egusquiza, E.; Valero, C. On the use of Vibrational Hill Charts for improved condition monitoring and diagnosis of hydraulic turbines. *Struct. Health Monit.* **2022**, *21*, 2547–2568. [CrossRef]
4. An, X.; Zeng, H. Pressure fluctuation signal analysis of a hydraulic turbine based on variational mode decomposition. *Proc. Inst. Mech. Eng. Part A J. Power Energy* **2015**, *229*, 978–991. [CrossRef]
5. Pan, H. Analysis on vibration signal of hydropower unit based on local mean decomposition and Wigner-Ville distribution. *J. Drain. Irrig. Mach. Eng.* **2014**, *32*, 220–224.
6. Lu, S.; Ye, W.; Xue, Y.; Tang, Y.; Guo, M.; Lund, H.; Kaiser, M.J. Dynamic feature information extraction using the special empirical mode decomposition entropy value and index energy. *Energy* **2020**, *193*, 116610. [CrossRef]
7. Oguejiofor, B.N.; Seo, K. PCA-based Monitoring of Power Plant Vibration Signal by Discrete Wavelet Decomposition Features. In Proceedings of the 2023 Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, 23–26 January 2023; pp. 1–5.
8. Shomaki, A.; Alkishriwo, O.A.S. Bearing Fault Diagnoses Using Wavelet Transform and Discrete Fourier Transform with Deep Learning. In Proceedings of the International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA, Tripoli, Libya, 25–27 May 2021.
9. Lu, S.; Zhang, X.; Shang, Y.; Li, W.; Skitmore, M.; Jiang, S.; Xue, Y. Improving Hilbert–Huang Transform for Energy-Correlation Fluctuation in Hydraulic Engineering. *Energy* **2018**, *164*, 1341–1350. [CrossRef]
10. Sun, H.; Si, Q.; Chen, N.; Yuan, S. HHT-based feature extraction of pump operation instability under cavitation conditions through motor current signal analysis. *Mech. Syst. Signal Process.* **2020**, *139*, 106613. [CrossRef]
11. Luo, P.; Hu, N.; Zhang, L.; Shen, J.; Chen, L. Adaptive Fisher-Based Deep Convolutional Neural Network and Its Application to Recognition of Rolling Element Bearing Fault Patterns and Sizes. *Math. Probl. Eng.* **2020**, *2020*, 3409262. [CrossRef]
12. Lan, C.; Li, S.; Chen, H.; Zhang, W.; Li, H. Research on running state recognition method of hydro-turbine based on FOA-PNN. *Measurement* **2021**, *169*, 108498. [CrossRef]
13. Ma, Y.; Wang, Q.; Wang, R.; Xiong, X. Optical fiber perimeter vibration signal recognition based on SVD and MPSO-SVM. *Syst. Eng. Electron.* **2020**, *42*, 1652–1661.
14. Cao, Q.; Wang, L.; Zhao, W.; Yuan, Z.; Liu, A.; Gao, Y.; Ye, R. Vibration State Identification of Hydraulic Units Based on Improved Artificial Rabbits Optimization Algorithm. *Biomimetics* **2023**, *8*, 243. [CrossRef] [PubMed]
15. Wang, Y.; Yang, M.; Li, Y.; Xu, Z.; Wang, J.; Fang, X. A multi-input and multi-task convolutional neural network for fault diagnosis based on bearing vibration signal. *IEEE Sens. J.* **2021**, *21*, 10946–10956. [CrossRef]
16. Li, X.; Zheng, J.; Li, M.; Ma, W.; Hu, Y. One-shot neural architecture search for fault diagnosis using vibration signals. *Expert Syst. Appl.* **2022**, *190*, 116027. [CrossRef]
17. Ravikumar, K.; Yadav, A.; Kumar, H.; Gangadharan, K.; Narasimhadhan, A. Gearbox fault diagnosis based on Multi-Scale deep residual learning and stacked LSTM model. *Measurement* **2021**, *186*, 110099. [CrossRef]
18. Prosvirin, A.E.; Maliuk, A.S.; Kim, J.-M. Intelligent rubbing fault identification using multivariate signals and a multivariate one-dimensional convolutional neural network. *Expert Syst. Appl.* **2022**, *198*, 116868. [CrossRef]
19. Ji, M.; Peng, G.; He, J.; Liu, S.; Chen, Z.; Li, S. A two-stage, intelligent bearing-fault-diagnosis method using order-tracking and a one-dimensional convolutional neural network with variable speeds. *Sensors* **2021**, *21*, 675. [CrossRef] [PubMed]
20. Zhou, F.; Wang, Y.; Jiang, S.; Hao, T. Research on an early warning method for bearing health diagnosis based on EEMD-PCA-ANFIS. *Electr. Eng.* **2023**, *105*, 2493–2507. [CrossRef]
21. Zhao, W.; Wang, L.; Zhang, Z.; Fan, H.; Zhang, J.; Mirjalili, S.; Khodadadi, N.; Cao, Q. Electric Eel Foraging Optimization: A new bio-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2023**, *238*, 122200. [CrossRef]
22. Sánchez, D.; Melin, P.; Castillo, O. Optimization of modular granular neural networks using a firefly algorithm for human recognition. *Eng. Appl. Artif. Intell.* **2017**, *64*, 172–186. [CrossRef]
23. Sairamya, N.; Subathra, M.; George, S.T. Automatic identification of schizophrenia using EEG signals based on discrete wavelet transform and RLNDiP technique with ANN. *Expert Syst. Appl.* **2022**, *192*, 116230. [CrossRef]

24.	Paul, J.G.; Rawat, T.K.; Job, J. Selective brain MRI image segmentation using fuzzy C mean clustering algorithm for tumor detection. *Int. J. Comput. Appl.* **2016**, *144*, 28–31.

25.	Khairi, R.; Fitri, S.; Rustam, Z.; Pandelaki, J. Fuzzy c-means clustering with minkowski and euclidean distance for cerebral infarction classification. *J. Phys. Conf. Ser.* **2021**, *1752*, 012033. [CrossRef]

26.	Zhang, H.; Zhang, H. A Novel Segmentation Method for Brain MRI Using a Block-Based Integrated Fuzzy C-Means Clustering Algorithm. *J. Med. Imaging Health Inform.* **2020**, *10*, 579–585. [CrossRef]

27.	Zhao, W.; Zhang, Z.; Wang, L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103300. [CrossRef]

28.	Jain, S.; Indora, S.; Atal, D.K. Rider manta ray foraging optimization-based generative adversarial network and CNN feature for detecting glaucoma. *Biomed. Signal Process. Control* **2022**, *73*, 103425. [CrossRef]

29.	Zouache, D.; Abdelaziz, F.B. Guided manta ray foraging optimization using epsilon dominance for multi-objective optimization in engineering design. *Expert Syst. Appl.* **2022**, *189*, 116126. [CrossRef]

30.	Houssein, E.H.; Zaki, G.N.; Diab, A.A.Z.; Younis, E.M. An efficient Manta Ray Foraging Optimization algorithm for parameter extraction of three-diode photovoltaic model. *Comput. Electr. Eng.* **2021**, *94*, 107304. [CrossRef]

31.	Hu, G.; Li, M.; Wang, X.; Wei, G.; Chang, C.-T. An enhanced manta ray foraging optimization algorithm for shape optimization of complex CCG-Ball curves. *Knowl. Based Syst.* **2022**, *240*, 108071. [CrossRef]

32.	Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 28–30 November 2005; pp. 695–701.

33.	Zhou, X.-y.; Wu, Z.-j.; Wang, H.; Li, K.-s.; Zhang, H.-y. Elite opposition-based particle swarm optimization. *Acta Electonica Sin.* **2013**, *41*, 1647.

34.	Yuan, Y.; Mu, X.; Shao, X.; Ren, J.; Zhao, Y.; Wang, Z. Optimization of an auto drum fashioned brake using the elite opposition-based learning and chaotic k-best gravitational search strategy based grey wolf optimizer algorithm. *Appl. Soft Comput.* **2022**, *123*, 108947. [CrossRef]

35.	Elgamal, Z.M.; Yasin, N.M.; Sabri, A.Q.M.; Sihwail, R.; Tubishat, M.; Jarrah, H. Improved equilibrium optimization algorithm using elite opposition-based learning and new local search strategy for feature selection in medical datasets. *Computation* **2021**, *9*, 68. [CrossRef]

36.	Qiang, Z.; Qiaoping, F.; Xingjun, H.; Jun, L. Parameter estimation of Muskingum model based on whale optimization algorithm with elite opposition-based learning. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2020; p. 022013.

37.	Shafiei, A.; Saberali, S.M. A Simple Asymptotic Bound on the Error of the Ordinary Normal Approximation to the Student's t-Distribution. *IEEE Commun. Lett.* **2015**, *19*, 1295–1298. [CrossRef]

38.	Yan, Z.; Zhang, J.; Tang, J. Path planning for autonomous underwater vehicle based on an enhanced water wave optimization algorithm. *Math. Comput. Simul.* **2021**, *181*, 192–241. [CrossRef]

39.	Zhao, W.; Wang, L.; Zhang, Z.; Mirjalili, S.; Khodadadi, N.; Ge, Q. Quadratic Interpolation Optimization (QIO): A new optimization algorithm based on generalized quadratic interpolation and its applications to real-world engineering problems. *Comput. Methods Appl. Mech. Eng.* **2023**, *417*, 116446. [CrossRef]

40.	Specht, D.F. Applications of probabilistic neural networks. In *Applications of Artificial Neural Networks*; SPIE: Bellingham, WA, USA, 1990; pp. 344–353.

41.	Wu, S.G.; Bao, F.S.; Xu, E.Y.; Wang, Y.-X.; Chang, Y.-F.; Xiang, Q.-L. A leaf recognition algorithm for plant classification using probabilistic neural network. In Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology, Cairo, Egypt, 15–18 December 2007; pp. 11–16. [CrossRef]

42.	Toğaçar, M. Detecting attacks on IoT devices with probabilistic Bayesian neural networks and hunger games search optimization approaches. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4418. [CrossRef]

43.	Song, M.-J.; Kim, B.-H.; Cho, Y.-S. Estimation of Maximum Tsunami Heights Using Probabilistic Modeling: Bayesian Inference and Bayesian Neural Networks. *J. Coast. Res.* **2022**, *38*, 548–556. [CrossRef]