Hindawi
Journal of Electrical and Computer Engineering
Volume 2023, Article ID 4137614, 20 pages
https://doi.org/10.1155/2023/4137614

Research Article

A Time Delay Prediction Model of 5G Users Based on the BiLSTM Neural Network Optimized by APSO-SD

Xiaozheng Dang,[1] **Di He**,[2] **and Cong Xie**[3]

[1]*Tianjin Vocational Institute, Tianjin 300000, China*
[2]*Vipshop Holdings Limited, Guangzhou, Guangdong 510000, China*
[3]*Guangxi Vocational University of Agriculture, Nanning, Guangxi 530005, China*

Correspondence should be addressed to Cong Xie; cong_xie2022@163.com

Received 10 January 2023; Revised 20 May 2023; Accepted 23 May 2023; Published 13 June 2023

Academic Editor: B. Rajanarayan Prusty

To address the problems of 5G network planning and optimization, a 5G user time delay prediction model based on the BiLSTM neural network optimized by APSO-SD is proposed. First, a channel generative model based on the ray-tracing model and the statistical channel model is constructed to obtain a large amount of time delay data, and a 5G user ray data feature model based on three-dimensional stereo mapping is proposed for input feature extraction. Then, an adaptive particle swarm optimization algorithm based on a search perturbation mechanism and differential enhancement strategy (APSO-SD) is proposed for the parameters' optimization of BiLSTM neural networks. Finally, the APSO-SD-BiLSTM model is proposed to predict the time delay of 5G users. The experimental results show that the APSO-SD has a better convergence performance and optimization performance in benchmark function optimization compared with the other PSO algorithms, and the APSO-SD-BiLSTM model has better user time delay prediction accuracy in different scenarios.

1. Introduction

Cellular systems are evolving towards 5G wireless systems due to the explosive growth of mobile devices and mobile traffic. Compared to 4G networks, the 5G network make new standards across generations in spectrum, air interface, and network architectures to meet future 5G application scenarios. These new standards and technologies bring challenges to 5G wireless network planning [1–3]. In addition, large-scale antenna arrays increase the difficulty of network optimization, and the complex spatial correlation makes it difficult for traditional fixed-point road tests and quantitative coverage models to meet the requirements of fast, comprehensive, and accurate assessment of raster-level network performance. Although traditional network performance simulations can achieve accurate analysis, the massive computation and time overhead make the simulation impossible to apply in the actual field [4].

With the widely used application of neural networks in information, automation, medicine, economy, and other fields, many researchers used the neural networks to predict channel characteristics and network performance [5–7]. The literature [8] used neural networks to predict the channel impulse response in wireless orthogonal frequency division multiplexing systems. The literature [9] used neural networks to predict the quality of downlink channels. The literature [5] used neural networks to predict the deviation angle of the transmission path in the user channel. Su et al. [10] used the whale optimization algorithm (WOA) and the long short-term memory neural network (LSTM) to predict network delay in narrowband communication networks. Wang and Zhang [11] proposed a least squares support vector machines (LS-SVMs) optimized by an improved PSO algorithm to predict the communication network time delay. Shi and Guo [12] proposed a wavelet neural network (WNN) delay prediction algorithm based on an improved set-averaged empirical modal decomposition and cuckoo

search (CS) optimization. Owing to the high cost of sample collection and the need to regenerate the sample data when the network parameters are changed, these methods are difficult to apply in actual networks.

The literature [13] used neural networks to predict 5G channel state information online, but only the wireless channel characteristics were predicted without network performance, and there has been shortcoming such as limited prediction parameters, small application range, and long data collection cycle. The literature [14] predicted the path loss of a wireless channel using a feedforward neural network algorithm based on the path loss data. This approach had a good performance in path loss prediction, but it could not collect data before the existing network was set up, and the cost of obtaining data through transmitters is relatively high. The network time delay prediction model based on gated recurrent neural networks was designed and validated based on real delay data between offsite server rooms at Amazon in the literature [15]. The model had a good result in terms of enterprise device latency but had a narrow scope of application for the complex 5G user usage scenario. Zhu et al. [16] implemented network latency prediction by deep neural networks for grid-level users. This method was effective in realistic 5G network planning and optimization, but it did not consider the impact of deep neural network parameter settings on the network time delay prediction performance.

From the previous research, it can be seen that the current research on 5G user time delay prediction had achieved certain results, but there had some problems in the user time delay modeling process and model prediction accuracy. Addressing the shortcomings of the previous methods, this paper proposes a 5G user ray data feature model based on three-dimensional stereo mapping, and it uses the deep neural network BiLSTM to learn the ray data features. The trained network model is used to predict the grid-level user network time delay. However, BiLSTM has shortcomings such as slow convergence speed and easy to fall into local optimization due to its random assignment of initial weight thresholds. In order to improve the accuracy of 5G user time delay prediction, this paper proposes an adaptive particle swarm optimization algorithm based on a search perturbation mechanism and differential enhancement strategy (APSO-SD) to optimize the initial parameters of the BiLSTM network.

The main contributions of this paper are follows:

(1) An adaptive particle swarm algorithm with a migration strategy and search disturbance mechanism (APSO-SD) for the parameter optimization of BiLSTM neural networks is proposed

(2) A 5G user time delay data feature model based on three-dimensional stereo mapping is proposed

(3) A 5G user time delay data prediction model (APSO-SD-BiLSTM) based on BiLSTM optimized by APSO-SD is proposed

The context of the article is as follows: Section 2 introduces the correlation theory. Section 3 introduces the 5G user ray data feature model based on three-dimensional stereo mapping. Section 4 introduces the adaptive particle swarm optimization based on the search perturbation mechanism and differential enhancement strategy. Section 5 introduces the 5G user time delay prediction based on the BiLSTM neural network optimized by APSO-SD. The experimental and simulation results are introduced in Section 6. The conclusion is put at the end.

2. Correlation Theory

2.1. Time Delay Simulation Model. The main flowchart of the time delay simulation model used in this article is shown in Figure 1. This model first utilizes the fusion of the ray-tracing model and the statistical model to generate a channel matrix and then inputs the channel matrix to the 5G simulation platform to generate grid-level user delay. A large amount of the 5G user time delay data can be obtained through the delay simulation model.

2.1.1. Channel Generative Model Based on the Ray-Tracing Model and Statistical Channel Model. Statistical channel models can generate channel matrices based on specific scenario configuration model parameters. The ray-tracing model can output the geometric information of all emission points between the starting and ending points of the ray, including the three-dimensional coordinate information of the starting point, ending point, and reflection point, as well as the large-scale path loss information of each ray [17]. Due to the ray-tracing model not containing small-scale information, only the geometric position information of the rays cannot calculate multipath effects. Therefore, the missing part must rely on the probability distribution information of the statistical model. To calculate the complete channel fading coefficient matrix, it is necessary to supplement the radiation data with antenna layout information, antenna pattern, and power delay distribution information. The antenna layout information and antenna pattern support the standard antenna configuration of 3GPP38.901, as well as antenna template input. The power delay distribution information is based on the probability distribution and parameter generation success rate delay distribution information of different scenarios defined by 3GPP38.901. This article proposed a channel generative model based on the ray-tracing model and the statistical channel model to generate the channel matrix. The new model calculates the channel matrix according to the process and method defined in 3GPP38.901. The information that the ray-tracing model can provide is based on the ray-tracing model, while the remaining information is based on a statistical model. The specific parameters of the channel matrix are shown in Table 1.

The final equation of the channel coefficient is shown as follows:

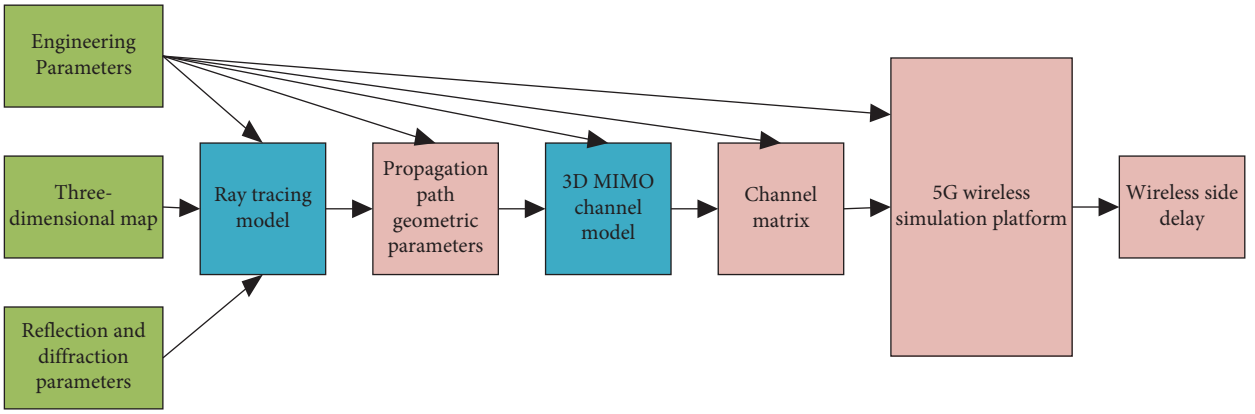

Figure 1: Flowchart of the delay simulation model.

Table 1: Specific parameters of the channel matrix.

Name	Related settings
Channel model	3D-UMa
Antenna configuration for base stations	Number 64, planar array 16 ∗ 4, cross-polarisation
User's antenna configuration	Number 2, cross-polarisation
Terminal movement speed	0 km/h
Multiuser launch precoding algorithm	JSDM
Base station transmitting power	46 dBM
Simulation bandwidth	20 Mkz
Carrier frequency	2 GHz
Intercell interference elimination	Support
Number of scheduling layers	8

$$H_{u,s,n}(t) = \sqrt{\frac{P_n}{M}} \sum_{m=1}^{M} \begin{bmatrix} R_{rx,u,\theta}\left(\theta_{n,m,ZOA}, \varphi_{n,m,AOA}\right) \\ R_{rx,u,\varphi}\left(\theta_{n,m,ZOA}, \varphi_{n,m,AOA}\right) \end{bmatrix}^T \begin{bmatrix} \exp\left(j\phi_{n,m}^{\theta\theta}\right) & \sqrt{\kappa_{n,m}^{-1}} \exp\left(j\phi_{n,m}^{\theta\varphi}\right) \\ \sqrt{\kappa_{n,m}^{-1}} \exp\left(j\phi_{n,m}^{\varphi\theta}\right) & \exp\left(j\phi_{n,m}^{\varphi\varphi}\right) \end{bmatrix} \begin{bmatrix} R_{tx,s,\theta}\left(\theta_{n,m,ZOD}, \varphi_{n,m,AOD}\right) \\ R_{tx,s,\varphi}\left(\theta_{n,m,ZOD}, \varphi_{n,m,AOD}\right) \end{bmatrix} \exp\left(j2\pi\lambda_0^{-1}\left(r_{rx,n,m}^T . d_{rx,u}\right)\right) \exp\left(j2\pi\left(r_{tx,n,m}^T . d_{tx,s}\right)\right) \exp\left(j2\pi\nu_{n,m}t\right), \tag{1}$$

where N represents the ray cluster, m represents the rays within the cluster, P_n represents the number of receiving and transmitting antennas, M represents the number of rays within the cluster, $\theta_{n,m,ZOA}$ and $\varphi_{n,m,AOA}$ represent the horizontal and vertical azimuth angles of the receiving antenna u, $\theta_{n,m,ZOD}$ and $\varphi_{n,m,AOD}$ represent the horizontal and vertical azimuth angles of the transmitting antenna s, $R_{rx,u,\theta}$ and $R_{rx,u,\varphi}$ are directional maps of the field of the receiving antenna u in the vector direction of the spherical base θ and φ, $R_{tx,s,\theta}$ and $R_{tx,s,\varphi}$ are directional maps of the field of the transmitting antenna s in the vector direction of the spherical base θ and φ, $r_{rx,n,m}$ and $r_{tx,n,m}^T$ are the spherical unit vectors with azimuths $\varphi_{n,m,AOA}$ and $\varphi_{n,m,AOD}$, and $d_{rx,u}$ and $d_{tx,s}$ are the direction vectors of the receiving and transmitting antennas.

2.1.2. Time Delay Calculation Method. The latency in this paper refers to the airport delay on the wireless side, mainly considering the single downstream delay of the packet from the MAC layer of the base station to the MAC layer of the user. It is assumed that packets do not need to be split and merged, the FULL BUFFER model is used in the service model, the hybrid automatic retransmission request (HARQ) channel configuration is 8, the packet single transmission duration (no retransmission) is 4 ms (1TTI), and the maximum number of packet retransmissions is 4. The packet retransmission process is shown in Figure 2.

This paper calculates the downlink unidirectional transmission delay of the user's wireless side by counting the number of packet retransmissions. By counting the number of retransmissions of a single packet, the transmission delay of the packet in this transmission task can be calculated, and the user's delay can be obtained by counting the transmission delay of all the user's packets during the simulation period, as shown in the following equation:

$$T^* = t + N * T_d, \tag{2}$$

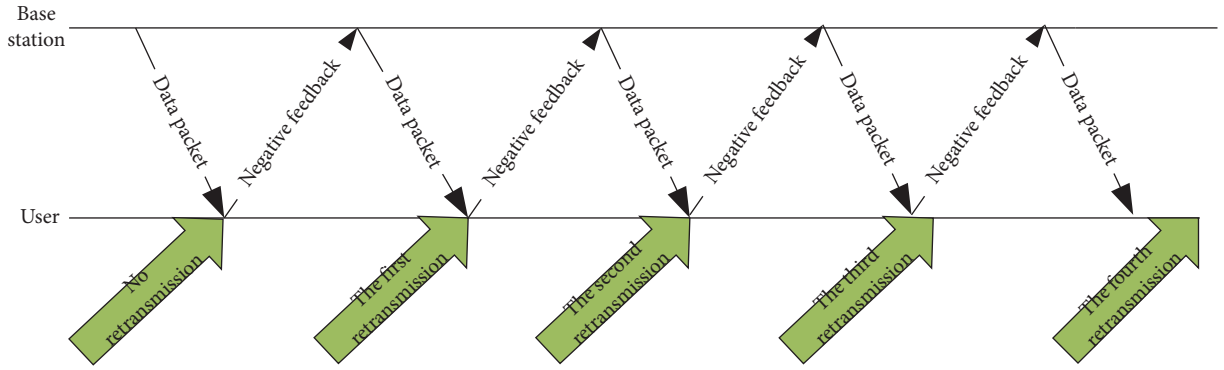

Figure 2: Packet retransmission process.

where T denotes the transmission delay of a single packet, t denotes the transmission delay of the first successful transmission of a packet, N denotes the number of retransmissions of a packet, and T_d denotes the time required for a retransmission to occur. The user's time delay calculation process is shown in the following equation:

$$T = \frac{T_S}{P_n}, \tag{3}$$

where T denotes the user's transmission delay, T_S denotes the total transmission delay of all packets from the user during the simulation, and P_N denotes the total number of packets sent during the simulation.

2.2. Bidirectional Long Short-Term Memory (BiLSTM) Network. In 1997, Schmidhuber et al. proposed a variant recurrent neural network long short-term memory (LSTM) network [18], which introduces a gating mechanism to simply and effectively solve the gradient explosion or disappearance problem of traditional recurrent neural networks. The LSTM controls the information transfer between each cell by means of a gating mechanism. The calculation of each gate in the LSTM model is shown in the following equations:

$$i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right), \tag{4}$$

$$f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right), \tag{5}$$

$$c_t = f_t \oplus c_{t-1} + i_t \oplus \tanh\overline{c_t}, \tag{6}$$

$$\overline{c_t} = \tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right), \tag{7}$$

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right), \tag{8}$$

$$h_t = o_t \oplus \tanh\left(c_t\right), \tag{9}$$

where i_t, f_t, and o_t represent the input gate, forgetting gate, and output gate, c_t represents the cell unit, σ and tanh are two activation functions, W_f represents the weight matrix connected by the forgetting gate, b_f represents the offset value of the forgetting gate, W_i represents the weight matrix of the input gate connection, b_i represents the offset value of the input gate, W_o represents the weight matrix of the output gate connection, b_o represents the offset value of the output gate, and $\oplus$ represents the multiplication of two matrix elements.

Although LSTM solves the problems of gradient vanishing and long-term dependence, for user delay prediction problems, the current state of the network is not only related to the previous state but may also be related to the subsequent state. In order to improve the prediction effect, the bidirectional long short-term memory (BiLSTM) network is introduced to predict user delay [19]. BiLSTM is composed of two LSTM layers stacked forward and backward, and its structure is shown in Figure 3.

The output of BiLSTM is determined by both LSTM layers together, the forward LSTM layer can be seen as a forward calculation from the starting moment to the last moment and the reverse LSTM layer can be seen as a reverse calculation from the last moment to the starting moment, with both layers being processed in the same way during calculation. Finally, the outputs of the forward and reverse layers are combined at each moment to obtain the output for that moment.

The state calculation at each moment in the BiLSTM model is shown in equations (10) and (11). The output is jointly determined by the states of LSTM in these two directions, as shown in equation (12):

$$\overrightarrow{h}_t = \mathrm{LSTM}\left(x_t, \overrightarrow{h}_{t-1}\right), \tag{10}$$

$$\overleftarrow{h}_t = \mathrm{LSTM}\left(x_t, \overleftarrow{h}_{t-1}\right), \tag{11}$$

$$h_t = W_t \overrightarrow{h}_t + v_t \overleftarrow{h}_t + b_t, \tag{12}$$

where W_t represents the weight matrix of the forward output, v_t represents the weight matrix of the reverse output, and b_t represents the offset at time t.

3. The 5G User Ray Data Feature Model Based on Three-Dimensional Stereo Mapping

In Section 2.1.1, the ray data are the input of the channel model and the frequency-domain channel matrix is the

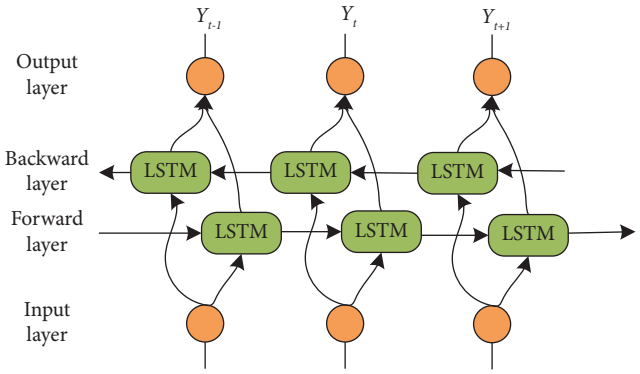

Figure 3: Basic structure of BiLSTM.

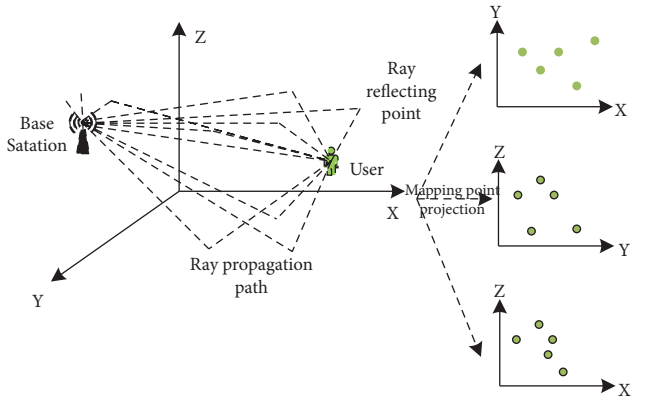

Figure 4: Schematic diagram of the feature model.

output of the channel model. The ray data contain the spatial characteristics of ray propagation, a large-scale path loss, and delay information of rays. The spatial feature vector of the channel matrix is contained in the geometric parameters of ray propagation. Once the position of the ray reflection point relative to the user and base station is determined, the spatial feature vector of the channel matrix is determined accordingly. Using the frequency-domain channel matrix generated by channel generation as an input to the 5G wireless simulation platforms, combined with other influencing factors, the user delay on the wireless side can be output.

Using neural network models to predict user delay requires designing reasonable input features. For this reason, this paper proposes a three-dimensional feature model that extracts the features of ray data as input features for the neural network model's training.

As shown in Figure 4, the user's ray data contain spatial features of ray propagation. Once the location of the base station, user, and ray reflection point is determined, the propagation path of the user ray is determined. The three-view feature model projects all reflection points of user ray data onto three planes: XOY, XOZ, and YOZ. Each plane contains a part of the spatial features of user ray data. By combining the spatial features of the three planes, the spatial features of user ray data can be fully restored.

Taking the reflection points on the XOY plane as an example, all reflection points of user ray data are projected onto the XOY plane. Simple processing of the XOY plane is required before projection. Because the base station coverage range in this article is 1000 m, therefore, the base station is used as the coordinate origin to extend 1000 m to the positive and negative directions of the X and Y axes, respectively, to form a coverage area. Similar to the pixel data in image processing, the $1000 * 1000$ base station coverage area of the XOY plane is divided into grid data such as $32 * 32$. After dividing the cells, it is necessary to determine the projection position of the reflection point, which is the small grid on the XOY plane where the reflection point falls. In the XOY plane, there are some cells without projection points and some cells with multiple projection points. After the projection position of the reflection point is determined, if there are no projection points in the cell, zero will be filled in the cell. If there are multiple projection points in the cell, the large-scale path loss or delay information of the ray will be averaged and filled in the cell to construct grid data as the input data for subsequent neural network models. In addition, due to the reflection points being all above the ground, the Z-axis only extends 100 m in its positive direction (height) to construct the coverage area.

4. Adaptive Particle Swarm Optimization Based on the Search Perturbation Mechanism and Differential Enhancement Strategy

PSO is a global search algorithm proposed by Kennedy and Eberhart by observing the foraging behavior of birds [20]. The algorithm initializes a group of particles randomly and assigns each particle a random speed and position, and each particle represents a random solution. During the iteration process, each particle completes the update by tracking the individual extreme value and the global extreme value. The standard PSO is described as follows:

$$V_{i,n}^{k+1} = \omega V_{i,n}^{k} + c_1\left(\text{Pbest}_{i,n}^{k} - X_{i,n}^{k}\right) + c_2\left(\text{Gbest}_{i,n}^{k} - X_{i,n}^{k}\right), \tag{13}$$

$$X_{i,n}^{k+1} = V_{i,n}^{k+1} + X_{i,n}^{k}, \tag{14}$$

where ω represents the inertia weight, k represents the number of iterations, n represents the vector dimension, c_1 and c_2 are the random numbers between 0 and 1, $V_{i,n}^{k}$, $X_{i,n}^{k}$, $\text{Pbest}_{i,n}^{k}$, and $\text{Gbest}_{i,n}^{k}$ are, respectively, the speed, position, individual extreme value, and global extreme value of the ith particle in the nth dimension of the kth iteration.

Similar to other intelligent algorithms, the PSO algorithm is prone to premature convergence and falling into local optimizations when solving complex high-dimensional functions. From the perspective of improving the particle iteration mechanism, this paper proposes an adaptive particle swarm optimization based on the search perturbation mechanism and differential enhancement strategy (APSO-SD). The APSO-SD algorithm mainly proposes corresponding optimization strategies for the different problems that exist in the population generation and optimization stages of the PSO algorithm. The optimization strategy is divided into the following two aspects:

(1) In order to expand the search space of particles and increase the diversity of solutions, disturbance factors are introduced to perturb the optimization state of particles to achieve intelligent search.

(2) To avoid the loss of excellent genes from poorer individuals, genetic mechanisms are utilized to prevent ineffective operations caused by "inbreeding." Then, particle irrelevance is introduced, and the "distant relatives" of the differential individuals were found for differential variation to increase the population diversity.

4.1. Search Disturbance Mechanism. In APSO-SD, in order to balance the global search and local search of particles, this paper introduces the conversion mechanism in the flower pollination algorithm (FPA) [21]. In the FPA, the simulated pollen heterogeneous pollination method is global search and the simulated pollen self-pollination is local search. The two search methods are controlled by the transition probability P, where P is a random number between 0 and 1. The smaller the transition probability P is, the easier it is for the particle to perform a local search. The larger the transition probability P is, the easier it is for the particle to perform a global search. To address the above problems, this paper uses a linearly varying transition probability to make it decrease linearly from the maximum value $P_{\max}$ to $P_{\min}$, as in the following equation:

$$P = P_{\max} - \frac{t \bullet \left(P_{\max} - P_{\min}\right)}{T}. \tag{15}$$

Based on experimental data in the literature [21], generally, it is better to set $P_{\max}$ at 0.95 and $P_{\min}$ at 0.4.

In the process of global search, in order to strengthen the algorithm's search capability, this paper will modify the search strategy of the particles by combining the advantages of the iterative mechanism of lioness foraging in the lioness algorithm [22] to randomly select a particle from the population to assist the current particle in the global search, and it increases the exchange of information between particles, as shown in equation (16). In the process of local search, to help the particles jump out of the local optimum, a sinusoidal disturbance factor is introduced in this paper, as in Eq. (17):

$$x_i^{t+1} = \frac{x_i^t + x_i^c}{2}\left(1 + \alpha_f \gamma\right) + v_i^{t+1}, \tag{16}$$

$$x_i^{t+1} = x_i^t + \sin\left(r_1\right) \bullet v_i^{t+1}, \tag{17}$$

where x_i^c is the historical best position of a randomly selected collaborating partner from the remaining particles, α_f is the disturbance factor, $\alpha_f = \text{step} \bullet \exp\left(-(30t/T)\right)^{10}$. $\text{step} = 0.1 \bullet \left(x_{\max} - x_{\min}\right)$, $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of the particle activity space, γ is a (0.1) uniformly distributed random number, and r_1 is a random number generated by the uniform distribution of $(0, 2\pi)$.

4.2. Differential Enhancement Strategy. The entire optimization process of the PSO algorithm is only driven by iterations of the historical optimal positions of individuals and society. The evaluation results are only used as a measure of the optimization effect. The results after each iteration cannot be effectively fed back to the population, so the population cannot make corresponding adjustments to the next operation based on the current search results. In order to effectively feed back each search result of the PSO algorithm to the population and make the population energetic, the population is required to perform corresponding transformations based on the optimization results to achieve the adaptive adjustment. This paper introduces the reconstruction probability, which is the probability that a particle is selected to rebuild the intermediate population based on the fitness value of the particle. According to the reconstruction probability, some poor individuals are selected in the optimization process of the PSO algorithm to build the intermediate population. By optimizing and enhancing some poor individuals, particles can accelerate their convergence to the global optimal solution.

Second, all operations of the population are only aimed at excellent individuals, which is easy to ignore other particle information in the population, especially for poor individuals. The genes in poor individuals and excellent individuals often differ significantly, which leads to the gradual loss of genes, and the population is prone to falling into local extreme values and unable to escape. In order to continue the excellent genes of the poorer individuals in the intermediate population, this paper proposes a differential enhancement strategy, which introduces particle irrelevance based on the genetic concept of "distant breeding, hybridizing to yield advantages." Particle irrelevance is used to calculate the selection probability. The selection probability is used to identify the "distant" individuals for hybrid breeding, effectively ensure the inheritance of excellent genes, keep the population active during the search process, and avoid the algorithm from falling into local extremes.

4.2.1. The Reconstruction Probability. Assuming the fitness function value of the ith particle generation k is f_i^k, the reconstruction probability of the particle being selected to form the intermediate population is shown in the following equation:

$$P_i^k = \frac{\sum_i^{NP} f_i^k - f_i^k}{2\sum_i^{NP} f_i^k}, \tag{18}$$

where $i = (1, 2, 3, \ldots, NP/3)$ and NP is the population size. Equation (18) indicates that when constructing an intermediate population, particles in the population are generated based on the reconstruction probability. When the particle fitness value is lower, the corresponding reconstruction probability is greater, and the probability of particles being selected to form an intermediate population is greater.

4.2.2. The Irrelevance of Particles. Assuming that $X_{i1,n}^k$ and $X_{j1,n}^k$ are the nth dimensional vectors corresponding to the $i1$th and $j1$th particle in the k-generation population, the irrelevance between individuals $X_{i1,n}^k$ and $X_{j1,n}^k$ is shown in the following equation:

$$R\left(X_{i1,n}^k, X_{j1,n}^k\right) = \sum_{i}^{n} \left|X_{i1,n}^k, X_{j1,n}^k\right|, \tag{19}$$

where t=(1.2, . . ., n) and t is the specific one dimension of the nth dimensional vector. Equation (19) indicates that if there is a significant difference in the values of variables among individuals, irrelevance is greater.

4.2.3. Selection Probability. Assuming that $Pc\,(X_{j1,n}^k/X_{i1,n}^k)$ is the probability that an individual $X_{j1,n}^k$ will be selected and undergo differential variation with $X_{i1,n}^k$, the selection probability calculation equation is shown in the following:

$$\mathrm{Pc}\left(\frac{X_{j1,n}^k}{X_{i1,n}^k}\right) = \frac{1}{np}\left(1 + \frac{R\left(X_{i1,n}^k, X_{j1,n}^k\right) - Ravg}{R\max - R\min}\right), \tag{20}$$

where np is the remaining individuals in the intermediate population except the individuals $X_{i1,n}^k$ and $1/np$ is the average probability of individuals being selected, and $Ravg$, R max, and R min are, respectively, the average, maximum, and minimum values of irrelevance between the remaining individuals and the selected individuals.

The differential enhancement strategy first uses the reconstruction probability to reconstruct the intermediate population based on the roulette wheel to select some individuals with low fitness values and then randomly selects the differential individuals in the intermediate population. Based on the irrelevance of the particles, the selection probability of the remaining individuals is calculated. Using the selection probability, two distantly related individuals with significant genetic differences from the differential individuals are found in the intermediate population, and differential mutation is performed to form temporary individuals. The retained individual genes with high fitness values enter the next generation of the population. When all individuals in the intermediate population have completed the local distant relative differential enhancement operation, the average fitness values before and after the intermediate population are compared. If the average fitness value after the local distant relative differential enhancement operation is high, the initial intermediate population is replaced with the existing intermediate population. If the distant relative differential enhancement strategy does not optimize the state of the intermediate population, the population does not change.

4.3. Implementation Steps and Pseudocode of APSO-SD. The APSO-SD algorithm uses a randomization method to initialize the population. After the population is generated, first the current local and global optimal solutions of the population are calculated, the particle search perturbation mechanism are used to update the speed and position of the particles, and the population enters the local differential enhancement stage. Second, the updated population fitness value is calculated, and the reconstruction probability is used to select individuals with lower fitness values to form an intermediate population. Finally, the average fitness value of the initial intermediate population is calculated, the selection probability is calculated based on the irrelevance of particles in the population, and two individuals with significant differences from the mutated individuals are selected for mutation, crossover, and selection operations. The calculation equation for the mutation operation is shown as follows:

$$\begin{aligned} V_i^{k+1} &= x_{r1}^k + F\left(x_{r2}^k - x_{r3}^k\right) \\ \text{S.T.} \quad & i \neq r1 \neq r2 \neq r3, \end{aligned} \tag{21}$$

where x_{r1}^k is the ith individual of the kth generation, which is the selected individual in the current population, x_{r2}^k、x_{r3}^k is a randomly selected different individual, $x_{r2}^k - x_{r3}^k$ is a difference vector generated by two random individuals, and F is a mutation operator that controls the scaling scale of the difference vector. Reasonable scaling can balance the search step size and search rate of individuals. V_i^{k+1} is a mutant intermediate of x_{r1}^k in the (k+1)th generation.

The cross-operation calculation equation is shown as follows:

$$U_{i,n}^{k+1} = \begin{cases} v_{i,n}^{k+1}, \mathrm{rand}\,(0,1) \leq CR_i \quad \text{or} \quad n = n_{\mathrm{rand}}, \\ x_{i,n}^k, \text{otherwise}, \end{cases} \tag{22}$$

where CR_i is the crossover probability. When the random number rand (0, 1) generated by individual i is less than the crossover probability, the check vector $U_{i,n}^{k+1}$ selects the mutation intermediate individual $v_{i,n}^{k+1}$; otherwise, it inherits the parent vector $x_{i,n}^k$.

The selection operation calculation equation is shown as follows:

$$x_i^{k+1} = \begin{cases} u_i^{k+1}, f\left(u_i^{k+1}\right) \leq f\left(x_i^k\right), \\ x_i^k, \text{otherwise}. \end{cases} \tag{23}$$

When the fitness value of the test vector u_i^{k+1} is smaller than that of the parent vector x_i^k, the algorithm selects the test vector u_i^{k+1} to enter the next generation population, and the population undergoes a successful update in generation kth.

After the local difference enhancement of all individuals in the intermediate population is completed, the average fitness value of the intermediate population is recalculated and compared with the average fitness value of the initial population. If the fitness value is optimized, the algorithm returns to the particle swarm optimization algorithm, replaces the initial intermediate population with the enhanced intermediate population, and continues the particle swarm optimization algorithm. Otherwise, it continues to perform local differential enhancement operations on these intermediate individuals until the fitness value of the intermediate population is optimized or the calculation times

of the intermediate population reach a preset value and then ends the local differential enhancement operation. When the evolution number of the entire particle swarm optimization algorithms reaches a preset value, the APSO-SD algorithm ends its optimization.

The implementation steps of the APSO-SD algorithm are as follows:

(1) Step 1: Initialize the population.

(2) Step 2: Calculate the fitness value of each particle based on the speed and position of the current particle and obtain the individual historical best position *Pbest* and the global best position *Gbest*.

(3) Step 3: Judge that the condition $rand < P_1$ is satisfied or not. If the condition is satisfied, the particle position is updated by equation (16). Otherwise, the particle position is updated by equation (17). Recalculate the fitness value for each particle and update *Gbest* and *Pbest*.

(4) Step 4: Use equation (18) to calculate the population reconstruction probability.

(5) Step 5: Initialize the intermediate population using roulette.

(6) Step 6: Use equation (19) to calculate particle irrelevance.

(7) Step 7: Use equation (20) to calculate the selection probability and select the differential individuals.

(8) Step 8: Use equations (21–23) to perform mutation, crossover, and selection operations, respectively.

(9) Step 9: Recalculate the average fitness value of the intermediate population and compare it with the average fitness value of the initial population. If the fitness value is optimized, replace the initial intermediate population with the intermediate population after the difference enhancement. Otherwise, continue with the particle swarm optimization algorithm.

(10) Step 10: Determine whether the algorithm meets the termination conditions for the iteration. If the algorithm meets the termination conditions for the iteration, the algorithm jumps to the next step. Otherwise, the algorithm jumps to Step 3 for the next iteration optimization.

(11) Step 11: Output the global optimal value and the algorithm ends.

The pseudocode implementation of the APSO-SD algorithm is shown in Table 2.

5. 5G User Time Delay Prediction Based on the BiLSTM Neural Network Optimized by APSO-SD

For BiLSTM networks, the selection of parameters in the structure is crucial to the effect of the model, such as the number of hidden layers, weights, the number of hidden layer cells, and learning rate. Many researchers determine these parameters based on experience or trial-and-error methods, which makes the robustness and accuracy of the model unreliable. Therefore, in this paper, a particle swarm algorithm with simple principles, low complexity, fast convergence, and suitable for dealing with real-valued problems is selected to optimize the structure parameters of the BiLSTM network.

5.1. Construction of the BiLSTM Model. The BiLSTM model used for the experiments in this paper is shown in Figure 5.

(1) BiLSTM layers: by two BiLSTM layers, their combined before-and-after capabilities can be fully exploited to enhance the model's learning capability

(2) Dropout layer: avoid overfitting of the model and improve generalisation

(3) Dense layer: set the last layer as dense, transform the output dimension, and get the prediction result

5.2. APSO-SD-BiLSTM Model. The APSO-SD-BiLSTM model is based on the fast optimization ability of APSO-SD to optimize the related parameters of BiLSTM, and it improves the situation prediction effect of BiLSTM. The specific steps of the APSO-SD-BiLSTM model are as follows:

(1) Step 1: According to the size of the sliding window, the training set samples and test set samples are constructed.

(2) Step 2: The range of values for each dimension in the particles to be optimized is set. The particle dimensions, *iterator*, n_1, n_2, and s represent the learning rate, the number of model iterations, the number of cells in the first hidden layer, the number of cells in the second hidden layer of the LSTM, and the random seeds in the BiLSTM model, respectively.

(3) Step 3: The relevant parameters in APSO-SD are initialized, which include the maximum and minimum values of the search dimension D, the number of particles P_N, the acceleration factors c_1 and c_2, the maximum number of iterations *max_iter*, the initial position of the particles x_i^0 and the initial velocity v_i^0, the inertia weight factor ω, and the learning factors r_1 and r_2.

(4) Step 4: The APSO-SD algorithm is carried out to optimize the BiLSTM parameters.

(5) Step 5: The optimal parameters obtained are assigned to the BiLSTM model to obtain the time delay prediction results.

The flowchart of the APSO-SD-BiLSTM model is shown in Figure 6.

6. Experimental and Simulation Results

The experimental environment for this paper is Ubuntu 18.04.5 (64 bit), a 2.90 GHz 8-core CPU, a GTX 1080Ti GPU,

Table 2: Pseudocode for the APSO-SD algorithm.

Algorithm: APSO-SD

Initialization:

for $i = 1$ **to** NP **do**
  for $n = 1$ **to** D **do**
    $X_{i,n}^k = \text{rand}(n, i)$
    $V_{i,n}^k = \text{rand}(n, i)$
  end for
  $\text{Pbest}_{i,n}^k = X_{i,n}^k$
  if $f(\text{Pbest}_{i,n}^k) < f(\text{Gbest}_{i,n}^k)$ **then**
    $\text{Gbest}_{i,n}^k = \text{Pbest}_{i,n}^k$
  end if
end for

Interation Steps:

for $i = 1$ **to** NP **do**
  if $P_1 > rand()$ **then**
    $x_i^{t+1} = x_i^t + x_i^c/2(1 + \alpha_f \gamma) + v_i^{t+1}$
  end if
  if $P_1 \le rand()$ **then**
    $x_i^{t+1} = x_i^t + \sin(r_1) \bullet v_i^{t+1}$
  end if
  $P_i^k = \sum_i^{NP} f_i^k - f_i^k/2 \sum_i^{NP} f_i^k$
  Constructing an intermediate population with a scale of (NP/3) through roulette
  $R(X_{i1,n}^k, X_{j1,n}^k) = \sum_i^n |X_{i1,n}^k, X_{j1,n}^k|$
  $Pc(X_{j1,n}^k / X_{i1,n}^k) = 1/np(1 + R(X_{i1,n}^k, X_{j1,n}^k) - Ravg/R\max - R\min)$

$$V_i^{k+1} = x_{r1}^k + F(x_{r2}^k - x_{r3}^k), \quad S.T. \quad i \neq r1 \neq r2 \neq r3$$

$$U_{i,n}^{k+1} = \begin{cases} v_{i,n}^{k+1}, \text{rand}(0,1) \le CR_i \quad or \quad n = n_{\text{rand}} \\ x_{i,n}^k, \text{otherwise} \end{cases},$$

$$x_i^{k+1} = \begin{cases} u_i^{k+1}, f(u_i^{k+1}) \le f(x_i^k) \\ x_i^k, \text{otherwise} \end{cases}$$

  if $\frac{3}{NP} f(\sum_{j1=1}^{NP/3} X_{j1,n}^k) < 3/NP f(\sum_{i1=1}^{NP/3} X_{i1,n}^k)$ **then**
  $X_{j1,n}^k = X_{i1,n}^k$
  end if
  $V_{i,n}^{k+1} = \omega V_{i,n}^k + c_1(\text{Pbest}_{i,n}^k - X_{i,n}^k) + c_2(\text{Gbest}_{i,n}^k - X_{i,n}^k);$
  $X_{i,n}^{k+1} = V_{i,n}^{k+1} + X_{i,n}^k;$
  $k = k + 1$
  if $k > Iter_{max}$ **then**
  Save the $\text{Gbest}_{i,n}^k$ and $\text{Pbest}_{i,n}^k$
  end if
end for

16GB of RAM, a 500GB hard drive, and Python 3.8.2 as the development language.

6.1. Particle Swarm Optimization Comparison Experiments

6.1.1. Benchmark Function. In order to verify the performance of APSO-MS, PSO-HS, PSONHM, CS-PSO, SE-PSO, E-PSO, AERPSO, and AdPSO are chosen as the comparison algorithms for the comparative analysis of performance [23–29]. In APSO-SD, the inertia weight $\omega = 0.3$, the learning factor $c_1 = c_2 = 1.5$, the maximum transition probability $P_{\max} = 0.95$, the minimum transition probability $P_{\min} = 0.4$, $\gamma = 0.5$, $r_1 = 0.2$, and the mutation operator $F = 0.5$. The parameter settings of other algorithms are consistent with those in the literature [23–29]. In this section, the performance of the PSO algorithms is tested in experiments using six benchmark tests. The detailed description of each function is shown in Table 3, and the image of each function is shown in Figure 7.

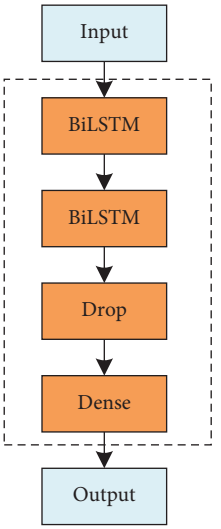

Figure 5: BiLSTM model.

In order to reduce the influence of the randomness of the PSO algorithm on the experimental results, the average of 30 independent run trials is used to evaluate the performance of the algorithm in this paper. For all the benchmark functions, the search dimension of the algorithm is set to 30 and 50 for experiments. The maximum number of iterations per run is set to 3000.

6.1.2. Function Optimization Results. In this section, a comparison of the PSO algorithms over six benchmark functions is carried out, and the mean values of the optimization results for each algorithm are shown in Tables 4-5.

It can be observed that whether $D = 30$ or $D = 50$, the quality and stability of APSO-SD search solutions are superior to the other seven algorithms for most functions, which indicates that APSO-SD uses a differential enhancement strategy to maintain the diversity of the population and enable it to continuously search for the optimal solution. The accuracy of the search solution is also better than that of other algorithms, which indicates that the search perturbation mechanism adopted by APSO-SD is conducive to maintaining the vitality of the population and the diversity of particles.

6.1.3. The Effectiveness Verification of Individual Improvement Measures

(1) Search Perturbation Mechanism. In order to verify the effectiveness of the search perturbation mechanism (SPM), this section uses test functions f_2 and f_6 to verify the basic

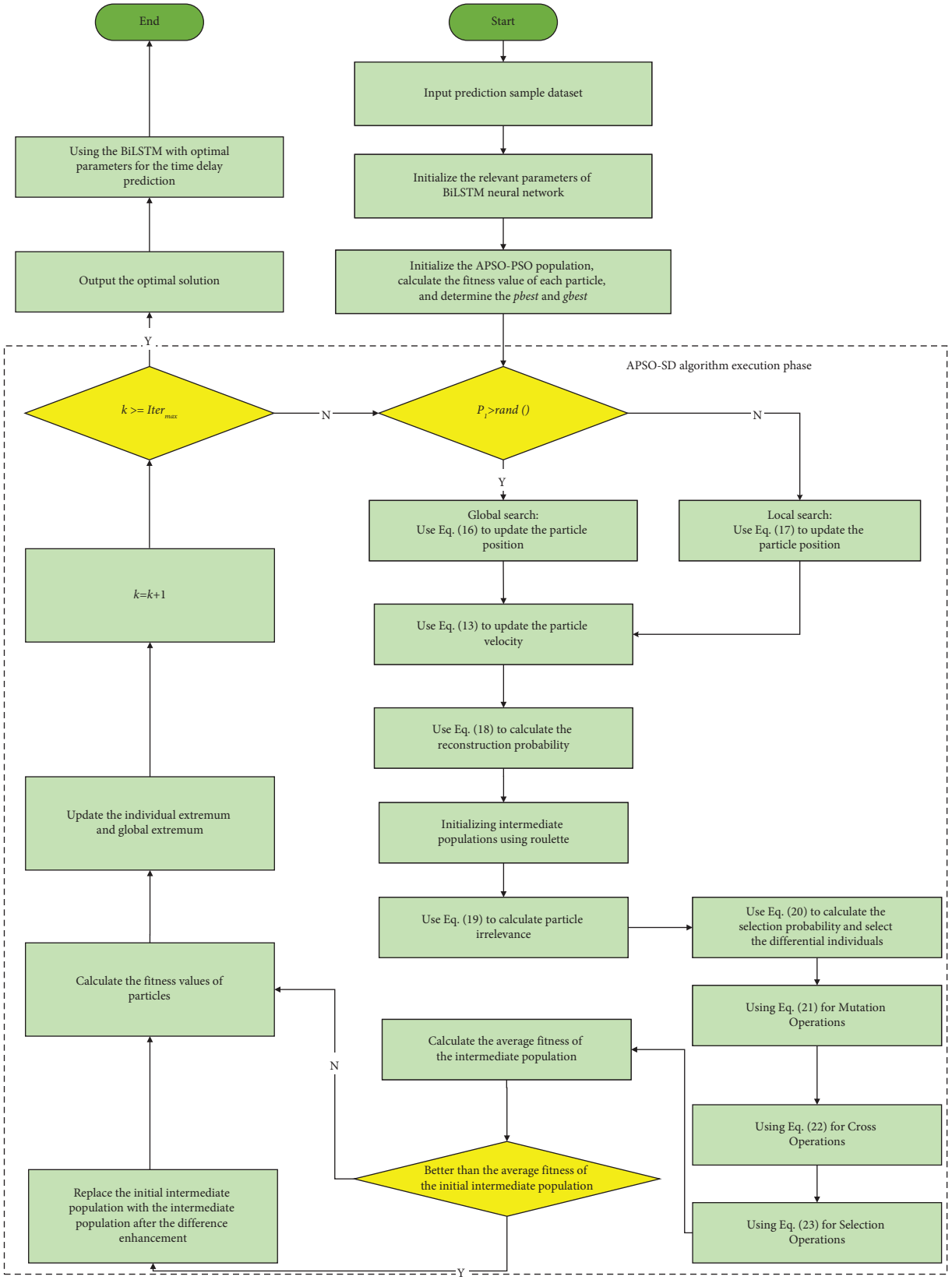

Figure 6: Flowchart of the APSO-SD-BiLSTM model.

Table 3: Detailed description of each function.

Functions	Name	Formulation	Range
f_1	Sphere	$F_1(X) = \sum_{i=1}^{n} X_i^2$	$[-5.12, 5.12]$
f_2	Rastrigin	$F_2(X) = 10n + \sum_{i=1}^{n} (X_i^2 - 1 - \cos(2\pi X_i))$	$[-5.12, 5.12]$
f_3	Beale	$F_3(X) = (1.5 - X_1 + X_1X_2)^2 + (2.25 - X_1 + X_1X_2^2)^2 + (2.625 - X_1 + X_1X_2^2)^2$	$[-3, 0.5]$
f_4	Booth	$F_4(X) = (X_1 + 2X_2 - 7)^2 + (2X_1 + X_2 - 5)^2$	$[1, 3]$
f_5	Holder table	$F_5(X) = -\lvert\sin(X_1)\cos(X_i)\exp(\lvert 1 - \sqrt{X_1^2 + X_2^2}/\pi\rvert)$	$[-10, 10]$
f_6	Ackley	$F_6(X) = 20 + e - 20\exp(-0.2\sqrt{(1/n\sum_{i=1}^{n} X_i^2)}) - \exp(\sqrt{(1/n\sum_{i=1}^{n}\cos(2\pi X_i))})$	$[-16, 32]$

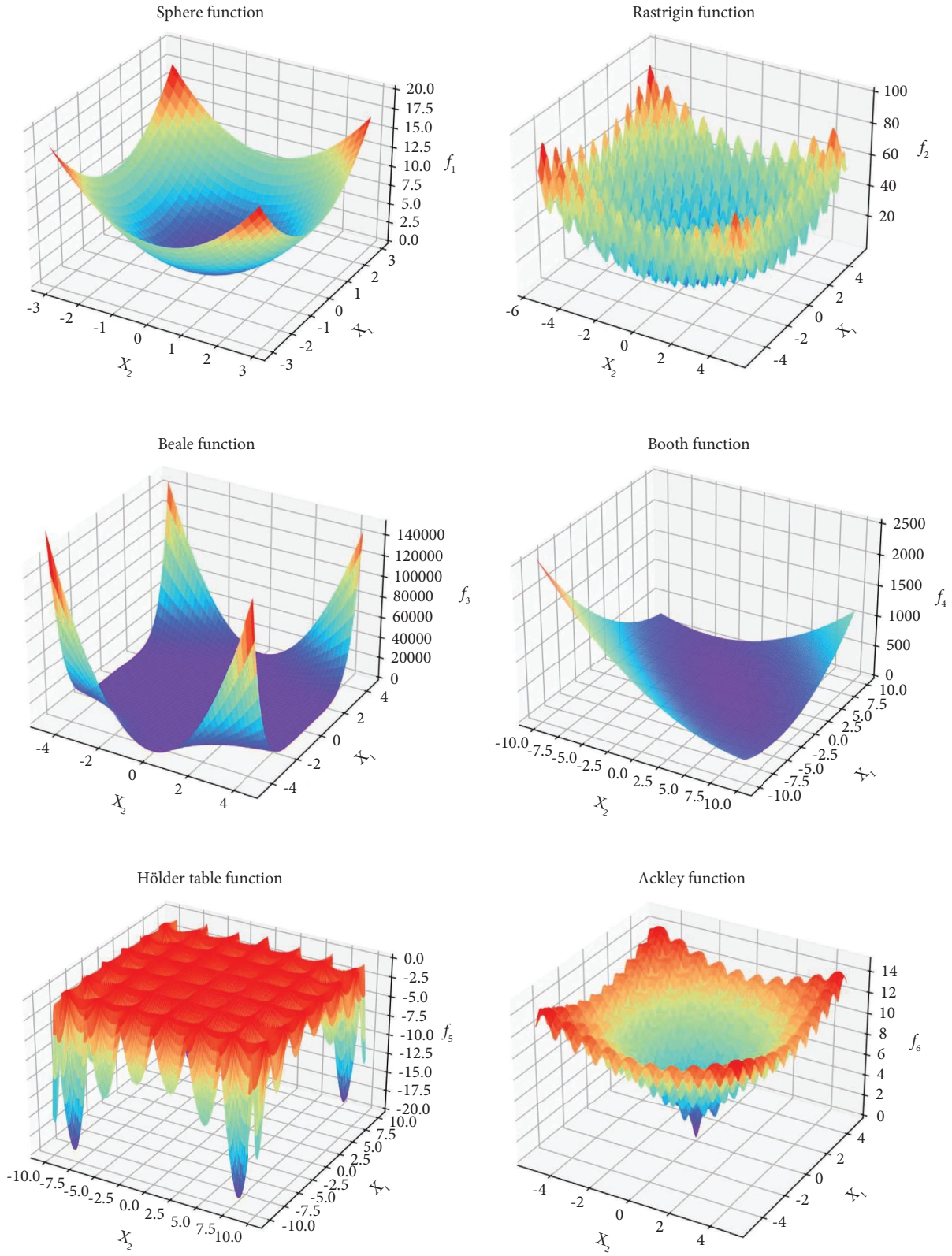

Figure 7: Benchmark function images.

PSO, APSO-SD-SPM (cancel the SPM mechanism in APSO-SD), and APSO-SD. The experimental results are shown in Table 6.

It can be seen from Table 6 that the variance of the fitness value of PSO in function f_2 and f_6 is $6.57E+02$ and $3.44E+06$ times of APSO-SD. When operating for 1000,

Table 4: Experimental results of the function test ($D = 30$ dimensions).

	PSO-HS	PSONHM	CS-PSO	SE-PSO	E-PSO	AERPSO	AdPSO	APSO-SD
f_1	$1.02E-12$	$1.43E-05$	$9.37E-06$	$7.06E-08$	$2.37E-07$	$4.04E-05$	$1.02E-12$	$1.43E-05$
f_2	$1.63E-01$	$6.13E-01$	$2.09E-01$	$5.26E-02$	$1.31E+01$	$1.20E+00$	$1.63E-01$	$6.13E-01$
f_3	$5.32E-02$	$6.87E+00$	$2.24E+00$	$4.16E+00$	$1.18E+01$	$1.26E-01$	$5.32E-02$	$6.87E+00$
f_4	$6.81E-05$	$8.78E-03$	$1.28E-02$	$8.94E-04$	$1.88E-02$	$5.55E-03$	$6.81E-05$	$8.78E-03$
f_5	$1.02E-12$	$1.43E-05$	$9.37E-06$	$7.06E-08$	$2.37E-07$	$4.04E-05$	$1.02E-12$	$1.43E-05$
f_6	$2.86E-01$	$3.92E+00$	$4.36E-03$	$3.44E+00$	$3.65E+00$	$7.00E-02$	$2.86E-01$	$3.92E+00$

Table 5: Experimental results of the function test ($D = 50$ dimensions).

	PSO-HS	PSONHM	CS-PSO	SE-PSO	E-PSO	AERPSO	AdPSO	APSO-SD
f_1	$3.67E+02$	$3.49E+02$	$2.02E+03$	$4.25E+02$	$7.73E+01$	$6.09E+02$	$8.97E+02$	$1.23E+02$
f_2	$1.92E-03$	$8.58E+01$	$1.13E+02$	$1.19E-03$	$4.35E+02$	$2.26E+02$	$2.73E+01$	$3.92E-05$
f_3	$7.51E+02$	$2.06E+04$	$1.04E+04$	$8.30E+00$	$2.02E+03$	$7.30E+02$	$4.05E+03$	$8.39E+02$
f_4	$3.96E-02$	$2.70E+01$	$6.92E+01$	$8.33E+00$	$1.98E+01$	$3.06E-02$	$1.14E+02$	$2.01E-03$
f_5	$3.67E+02$	$3.49E+02$	$2.02E+03$	$4.25E+02$	$7.73E+01$	$6.09E+02$	$8.97E+02$	$1.23E+02$
f_6	$3.16E+02$	$3.01E+02$	$1.75E+03$	$3.67E+02$	$1.06E+01$	$5.26E+02$	$7.74E+02$	$6.68E+01$

Table 6: Diversity and rate comparison of the algorithm.

Functions	Algorithm	Variance	Number of evaluation		
			1,000	2,000	3,000
f_2	PSO	$6.08E+05$	$6.65E+00$	$2.16E-01$	$2.31E-03$
	APSO-SD-SPM	$1.18E+03$	$1.39E+00$	$2.02E-02$	$2.42E-05$
	APSO-SD	$9.26E+02$	$4.91E-06$	$4.96E-07$	$3.87E-08$
f_6	PSO	$4.27E+02$	$4.57E+01$	$3.03E+01$	$2.12E+01$
	APSO-SD-SPM	$4.93E-03$	$9.13E-03$	$2.12E-04$	$8.82E-16E$
	APSO-SD	$1.24E-04$	$5.67E-15$	$4.38E-15$	$0.00E+00$

2000, and 3000 times, the fitness values of PSO in function f_2 are $1.35E+06$, $4.36E+05$, and $5.96E+04$ times of APSO-SD. The fitness values of function f_6 are $8.05E+15$ and $6.93E+15$ times of APSO-SD. The fitness variance of APSO-SD-SPM in functions f_2 and f_6 is $1.27E+00$ and $3.98E+01$ times of APSO-SD. When operating for 1000, 2000, and 3000 times, the fitness values of APSO-SD-SPM in function f_2 are $4.77E+00$, $1.07E+01$, and $9.54E+01$ times of APSO-SD. The fitness values of function f_6 are $1.61E+12$ and $4.84E+10$ times.

In conclusion, the optimization results of APSO-SD-SPM are significantly improved compared to the basic PSO, but there is still a certain gap between APSO-SD and APSO-SD. This is because the APSO-SD and APSO-SD-SPM is through crossover and mutation to generate new individuals. However, the APSO-SD algorithm uses the search perturbation mechanism to increase the diversity of the particle population in each iteration, which makes it have better results in the process of optimization.

(2) Differential Enhancement Strategy. To verify the effectiveness of the differential enhancement strategy (DES), this section uses test functions f_2 and f_6 to validate the basic PSO and APSO-SD-DES (canceling the DES mechanism in APSO-SD) and APSO-SD algorithms. The experimental results are shown in Table 7.

Table 7 shows that compared to the basic PSO, the optimization results of APSO-SD-DES are significantly improved, but there is still a certain gap between APSO-SD-DES and APSO-SD. This is because both APSO-SD and APSO-SD-DES increase the diversity of particle populations through search perturbation mechanisms. However, the APSO-SD algorithm generates two populations through DE and PSO in each iteration, and it compares the two populations to select the best *Pbest* and *Gbest*, thus achieving better optimization results.

From the results of the previous two single validation experiments, it can be concluded that the perturbation strategy and the local distantly related differential enhancement strategy proposed in this paper have significant advantages in both the convergence rate of the algorithm and the maintenance of the population diversity.

6.1.4. T-Test and Friedman Test. In the comparative analysis of swarm intelligence optimization algorithms, researchers usually use the *T*-test [30], Friedman test [31], Wilcoxon signed-rank test [32], and Mann–Whitney *U* test [33] to compare the significant differences between the algorithms. Based on this, this article selects the *T*-test and Friedman test to test the performance of eight algorithms on six test functions. The experimental results are shown in Table 8.

Here, "+" indicates that the APSO-SD algorithm outperforms the other algorithms, " = " indicates that there is no significant difference between the algorithms, "–" indicates inferiority to the other algorithms, and *w/t/l* indicates the

Table 7: Diversity and rate comparison of the algorithm.

	Algorithm	Variance	Number of evaluation		
			10,000	20,000	30,000
f_2	PSO	$3.90E+06$	$4.26E+01$	$1.39E+00$	$1.48E-02$
	APSO-SD-D*E*S	$7.55E+03$	$8.93E+00$	$1.30E-01$	$1.55E-04$
	APSO-SD	$5.93E+03$	$3.15E-05$	$3.18E-06$	$2.48E-07$
f_6	PSO	$2.73E+03$	$2.92E+02$	$1.94E+02$	$1.36E+02$
	APSO-SD-DES	$3.16E-02$	$5.85E-02$	$1.36E-03$	$6.38E-14E$
	APSO-SD	$7.93E-04$	$3.63E-14$	$2.80E-14$	$0.00E+00$

Table 8: Results of the T test and Friedman test for four algorithms.

	f_1	f_2	f_3	f_4	f_5	f_6	$w/t/l$	Rank mean
PSO-HS	+	+	+	=	+	=	5/3/0	2.87
PSONHM	=	=	—	+	=	+	3/4/1	1.89
CS-PSO	+	+	=	+	—	+	5/2/1	2.13
SE-PSO	+	+	+	+	+	+	7/1/0	1.96
E-PSO	+	=	—	+	+	+	6/1/1	2.53
AERPSO	+	+	+	=	+	+	6/0/2	2.18
AdPSO	+	+	—	+	+	+	4/4/0	2.27
APSO-SD								1.68

number of statistics for each of these three comparative results.

The T-test results show that APSO-SD has better performance on four test functions compared with that of the PSO-HS algorithm, and the two tests have no difference. Compared with that of PSONHM, APSO-SD has better performance on two test functions, three with no difference and one with worse performance. Compared with that of CS-PSO, APSO-SD has better performance on four test functions, one with no difference and one with worse performance. The performance difference between SE-PSO and APSO-SD is significant. APSO-SD has better performance on four test functions, one with no difference and one with worse performance. APSO-SD has better performance on five test functions and one worse compared to that of AERPSO. APSO-SD has better performance on five test functions, one with no difference, compared to that of AdPSO. From the Friedman test results, it can be concluded that APSO-SD has the smallest rank mean value and the best performance compared with other algorithms.

Based on the previous experimental results, it can be seen that, to the single-peak and multipeak function, the APSO-SD algorithm can obtain high-quality optimization results. Compared with other algorithms, the proposed algorithm has better stability and search ability. The algorithm alleviates the contradiction between precocity and convergence speed and balances the global search and local search effectively.

6.2. User Delay Prediction Comparison Experiments

6.2.1. Dataset Construction. In this section, the reflective points are first projected onto a 3-dimensional plane. Then, the reflective points are extended 1000 m in both the positive and negative directions on the X and Y axes with the base station as the center and 100 m above the base station in the Z-axis direction starting from the ground. Finally, all the three planes are divided into $64 * 64$ grids to determine the intervals where the reflective points are located and to build the dataset for neural network training.

6.2.2. Evaluation Indicators. The expression for the relative error in time delay can be given by the following equation:

$$\xi = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|X_i - Y_i\right|}{Y_i}, \tag{24}$$

where X_i is the predicted value of the network model, Y_i is the actual value, and n is the number of users.

The relative error in the number of message packet retransmissions could be expressed as follows:

$$\xi = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|X_i - Y_i\right|}{N_{\max}}, \tag{25}$$

where $N_{\max}$ is the maximum value of the sum of the number of message packet retransmissions.

The expressions for the mean and standard deviation of the errors are shown as follows:

$$\begin{aligned}\tau &= \frac{1}{n}\sum_{i=1}^{n}\left|X_i - Y_i\right|,\\ d &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\left|X_i - Y_i\right| - \tau\right)^2}\end{aligned}. \tag{26}$$

6.2.3. Parameter Selection. In the APSO-SD BiLSTM model, the parameter settings of the APSO-SD algorithm are consistent with those in Section 5.1. The number of input layer nodes in the BiLSTM neural network is 5, the depth of the hidden layer is 2, and the number of output layer nodes is 3. This section uses different time steps and batch sizes to train the APSO-SD-BiLSTM model.

Figure 8 shows the training results of the APSO-SD-BiLSTM model. The time step and batch size gradually converge to the optimal values as the algorithm is updated. Figure 8 shows that the batch size of the model training data is 2 and that the optimal time step is 5.

6.2.4. Wireless Side Network Latency Prediction. In this section, network latency is first predicted directly using the APSO-SD-BiLSTM model in the experiments. Then, the number of message packet retransmissions predicted by the APSO-SD-BiLSTM model is processed as a regression task to predict network latency indirectly. Finally, the number of message packet retransmissions predicted by the APSO-SD-BiLSTM model is processed as a classification task to predict network latency indirectly. Among them, the number of training samples is 90,000 and the number of prediction samples is 9,000.

(1) Direct Prediction of User Network Latency. In this section, the APSO-SD-BiLSTM model is used to directly predict the network delay, and the predicted delay is shown in Figure 9. From Figure 9, it can be seen that the trend of the predicted value and the actual value is basically matched.

The evaluation metrics for the network delay prediction results are shown in Table 9. As can be seen from Table 9, both the relative error and the mean value of the error are small.

(2) Regression-Oriented Tasks User Network Latency Indirect Prediction. The number of message packet retransmissions predicted by the APSO-SD-BiLSTM model is treated as a regression task to indirectly predict network latency. The prediction results are shown in Figure 10. As can be seen from Figure 10, the trend of the predicted and actual values is generally consistent.

The evaluation metrics for the prediction results are shown in Table 10. Table 10 shows that the relative error and the mean value of the error for the indirect prediction of the network delay for the regression-oriented task are large compared to the results for the direct prediction of the network delay.

(3) Indirect Prediction of User Network Latency for Classification Tasks. The number of message packet retransmissions predicted by the APSO-SD-BiLSTM model is treated as a classification task to indirectly predict network latency. The prediction results are shown in Figure 11. As can be seen from Figure 11, the trend of the predicted and actual values is generally consistent, and only a few points differ.

The evaluation metrics for the prediction results are shown in Table 11. Table 11 shows that the relative error and the mean error of the indirect prediction results for the classification-oriented task network time delay are slightly improved compared to the indirect prediction results for the regression-oriented task network time delay.

(4) Error Analysis for Predicting Network Delay. The relative error probability distributions for direct prediction of network latency and indirect prediction of the number of message packet retransmissions are shown in Figures 12 and 13, respectively. From Figure 12, it can be seen that the relative error of direct prediction of network latency is concentrated at 1%. From Figure 13, the relative errors of indirect prediction of user network latency for regression task and indirect prediction of user network latency for classification task are not significantly different, and the relative errors of message packet retransmission number for both are concentrated at 10%.

6.2.5. Comparison with Existing Forecasting Methods. To verify the validity of the models, this section replaces BiLSTM [34], PSO-BiLSTM [35], APSO-BiLSTM [36], IPSO-BiLSTM [37], ASPSO-BiLSTM [38], APSO-MS-LSTM (BiLSTM in the APSO-MS-BiLSTM model with LSTM), APSO-MS-RNN (BiLSTM replaced by RNN in APSO-MS-BiLSTM model), CNN [16], CNNDP [39], AMCA [40], PABAFT [41], OCEAN [13], CNN-LSTM [13], and CNN-KF [42] with APSO-SD-BiLSTM models for direct prediction of network delay, respectively. The experimental results are shown in Table 12.

The results of the experimental comparison are as follows:

(1) Compared with BiLSTM, PSO-BiLSTM, APSO-BiLSTM, IPSO-BiLSTM, and ASPSO-BiLSTM, the prediction results of APSO-SD-BiLSTM achieved better results in all three evaluation metrics. This is mainly because the APSO-SD algorithm has better outcome-seeking performance relative to the PSO algorithm and can find the relevant parameters that give the BiLSTM neural network better prediction performance.

(2) Compared with APSO-SD-LSTM and APSO-SD-RNN, the prediction results of APSO-SD-BiLSTM have a large degree of improvement in all three evaluation metrics. This is mainly due to the fact that BiLSTM can better capture the premoment and postmoment dependence information of time-delay data compared to LSTM and RNN, which is conducive to improving the prediction accuracy of the temporal data.

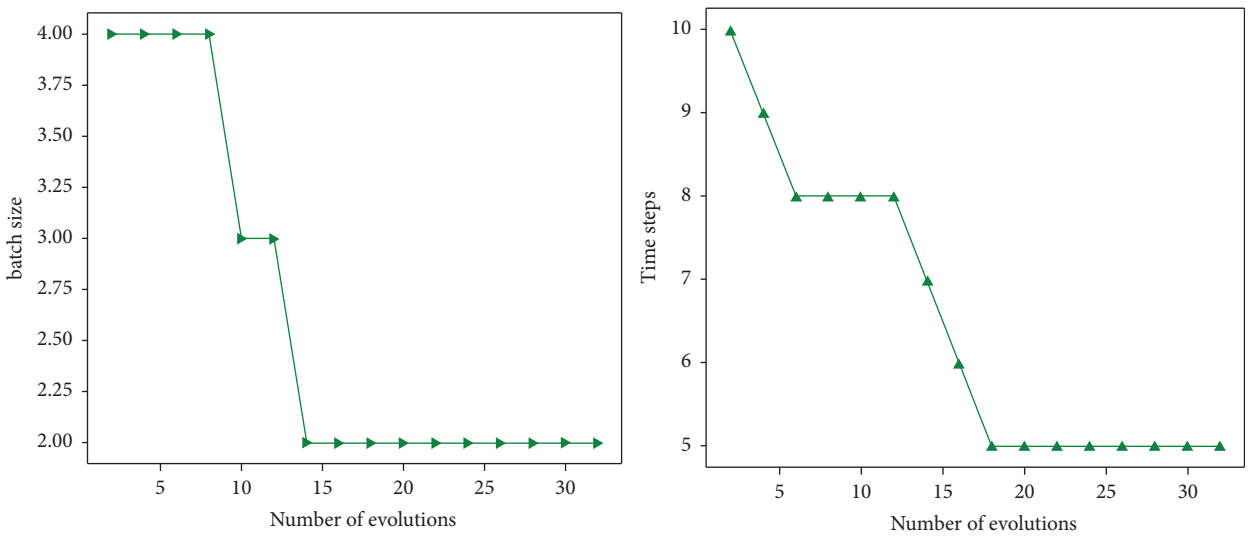

Figure 8: The parameters of BiLSTM.

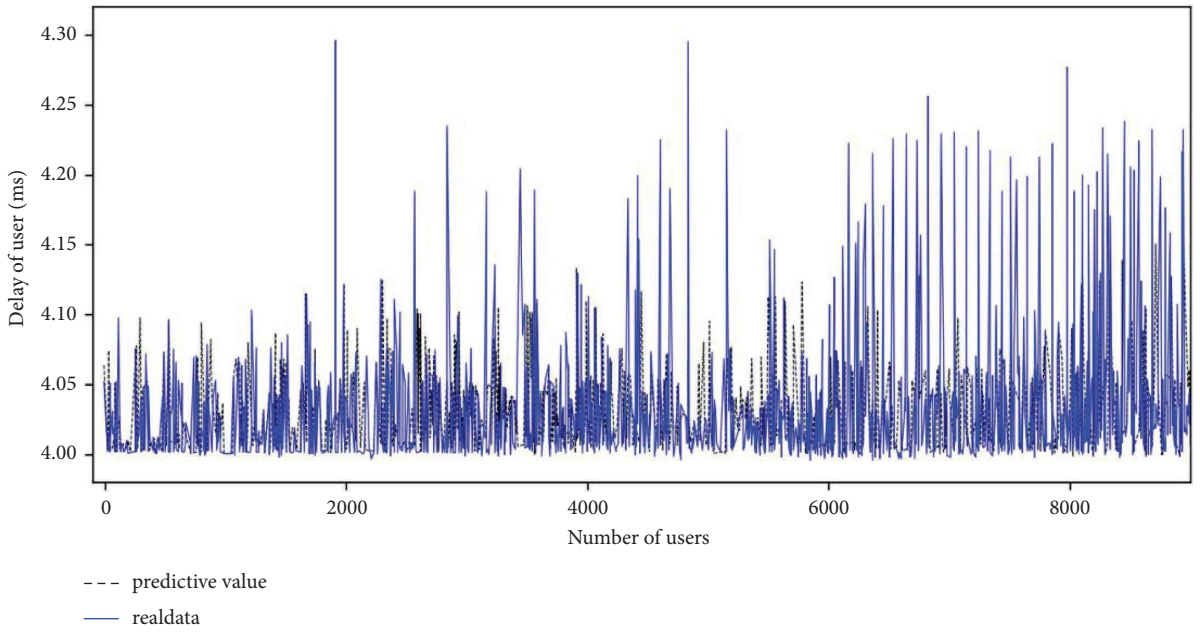

Figure 9: Results of direct network delay prediction.

Table 9: Evaluation results of direct network delay prediction.

Indicators	Assessment results
Relative error (%)	5.9235
Mean value of error (time delay difference) (ms)	0.0361
Error standard deviation (time delay difference) (ms)	0.0312
Predicted hours used by 9000 users	13.0778

(3) Compared with existing models such as CNN, CNNDP, AMCA, PABAFT, OCEAN, CNN-LSTM, and CNN-KF, the APSO-SD-BiLSTM model has the highest prediction accuracy and can effectively achieve the user's time-delay prediction. This is mainly because the APSO-SD-BiLSTM model makes full use of the dependence of user latency data on the latency data of the preceding and following

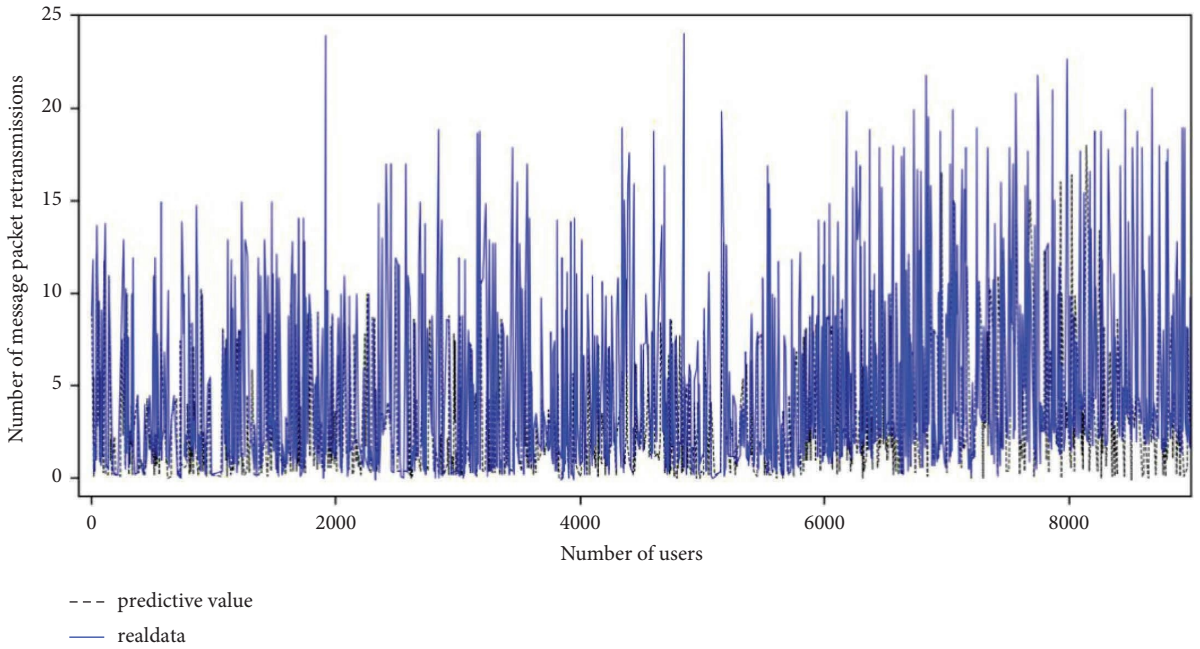

Figure 10: Predictions for the regression-oriented task number of message packet retransmissions.

Table 10: Evaluation results of the predicted number of message packet retransmissions under the regression task.

Indicators	Assessment results
Relative error (%)	10.762503
Mean error (number of message packet retransmissions) (time)	3.228742
Error standard deviation (number of message packet retransmissions) (time)	3.303858
Predicted hours used by 9,000 users	11.850617

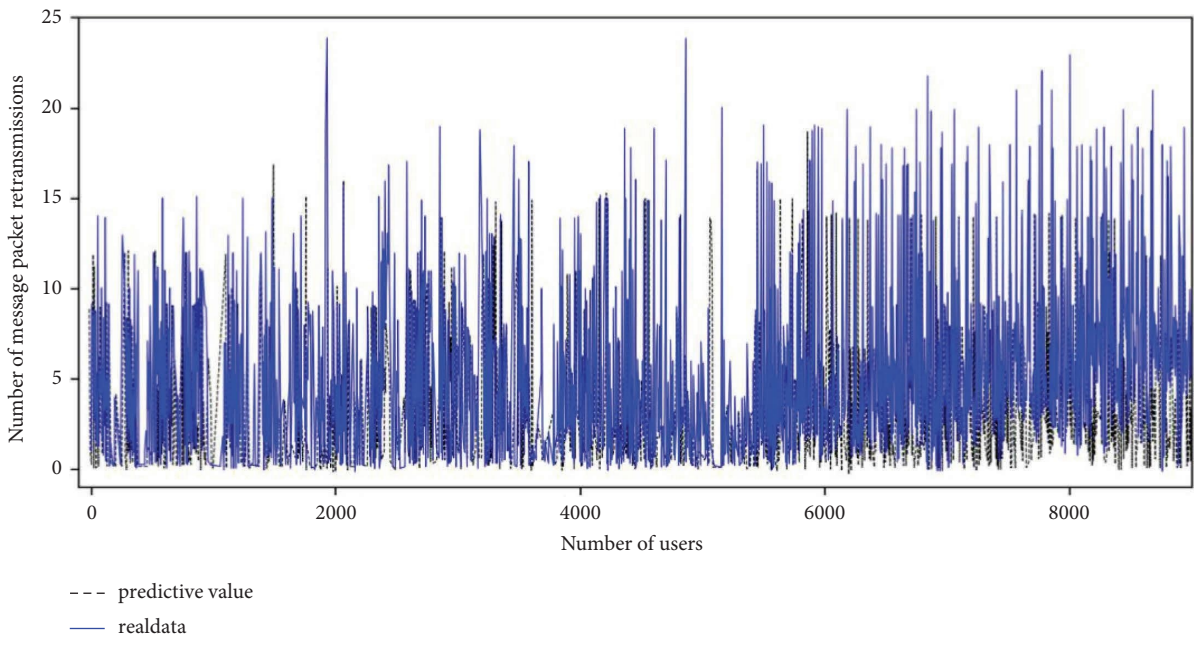

Figure 11: Prediction of the number of message packet retransmissions for classification tasks.

Table 11: Evaluation results of the predicted message packet retransmission count metric under the classification task.

Indicators	Assessment results
Relative error (%)	11.725394
Mean error (number of message packet retransmissions) (time)	3.517547
Error standard deviation (number of message packet retransmissions) (time)	3.661282
Predicted length of time for 10,000 users	14.165685

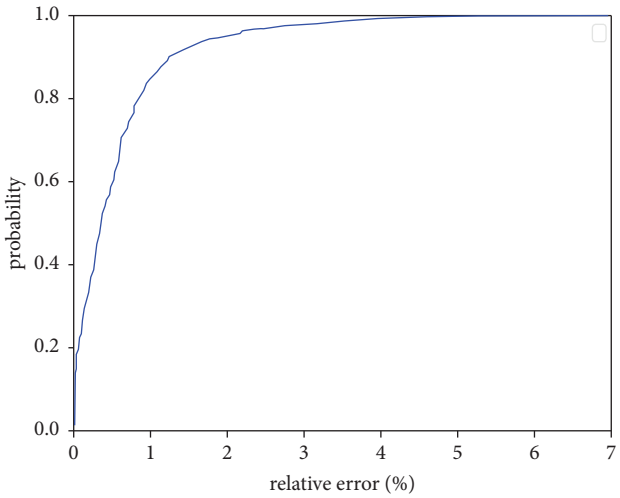

Figure 12: Relative error probability distribution of direct prediction network delay.

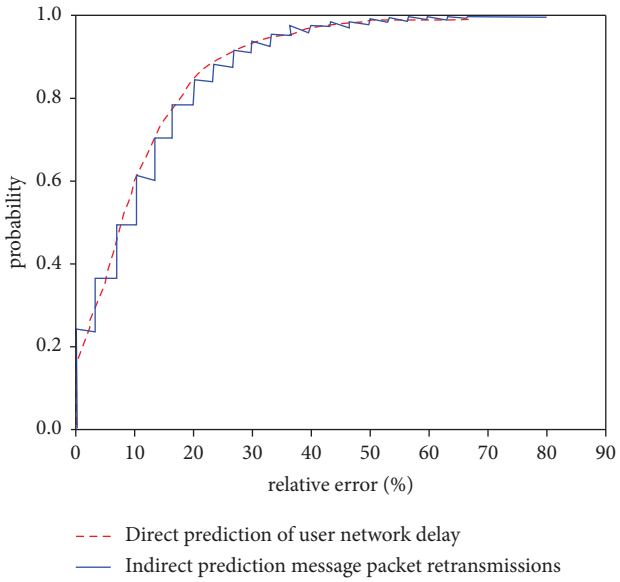

Figure 13: Relative error probability distribution of indirect prediction message packet retransmissions.

Table 12: Test results of the models.

Models	Evaluation indicators		
	Relative error (%)	Mean value of error (time delay difference) (ms)	Error standard deviation (time delay difference) (ms)
BiLSTM	9.1282	0.0600	0.0506
PSO-BiLSTM	8.0581	0.0516	0.0474
APSO-BiLSTM	7.2241	0.0477	0.0407
IPSO-BiLSTM	9.2003	0.0639	0.0525
ASPSO-BiLSTM	10.0092	0.0708	0.0595
APSO-SD-LSTM	11.3953	0.091	0.0683
APSO-SD-RNN	11.6042	0.0960	0.0736
CNN	14.6404	0.1615	0.0810
CNNDP	10.8311	0.1048	0.0823
AMCA	12.0359	0.1117	0.0843
PABAFT	13.7760	0.1009	0.0891
OCEAN	11.7894	0.1413	0.0956
CNN-LSTM	12.1966	0.1202	0.0916
CNN-KF	12.8941	0.1583	0.0934
APSO-SD-BiLSTM	5.9235	0.0361	0.0312

moments, and it effectively achieves the mining of user historical latency data.

7. Conclusions and Future Work

In this paper, a 5G user time delay data prediction model is proposed based on the BiLSTM neural network optimized by APSO-SD. First, a large amount of delay data is obtained by using the delay simulation model based on the ray-tracing model and statistical model fusion. Then, a user ray data feature model is proposed based on 3D stereo mapping. Finally, the 5G user network time delay prediction model (APSO-SD-BiLSTM) is carried out based on the BiLSTM neural network optimized by APSO-SD. The experimental results show that the APSO-SD-BiLSTM model has better prediction accuracy than the existed prediction models, and it can effectively achieve the network delay prediction.

The APSO-SD BiLSTM model proposed in this paper has two main limitations. First, the feature model of 5G user ray data proposed in this paper only considers the user's temporal characteristics; second, the APSO-SD BiLSTM model only focuses on optimizing the structural parameters of BiLSTM. Notwithstanding its limitation, the APSO-SD BiLSTM model can still effectively predict 5G user time delay.

In the future research, the following three aspects can be carried out:

(1) On the basis of extracting ray reflection point features, the temporal and spatial features of 5G user time delay data are fused and user time delay prediction is performed based on the fused features.

(2) The APSO for the joint optimization of the initial weight parameters and structural parameters of BiLSTM is used, and then, optimized BiLSTM to further improve the accuracy of user latency data prediction is used.

(3) The difference between 5G user delay characteristics and 6G user delay characteristics is studied, and 6G user delay based on the combination of the swarm intelligence algorithm and neural network is predicted.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Xiaozheng Xiang performed the experiments and analyzed the data. Di He supervised the research and critically revised the paper. Cong Xie wrote the paper. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was supported by the joint fund for enterprise innovation and development of the National Natural Science Foundation of China (U21B1325) and the Guangxi Agricultural Science and Technology Self-Financing Project (Z202234).

References

[1] S. Lavanya, A. Prasanth, S. Jayachitra, and A. Shenbagarajan, "A Tuned classification approach for efficient heterogeneous fault diagnosis in IoT-enabled WSN applications," *Measurement*, vol. 183, Article ID 109771, 2021.

[2] P. Nayak, G. K. Swetha, S. Gupta, and K. Madhavi, "Routing in wireless sensor networks using machine learning techniques: challenges and opportunities," *Measurement*, vol. 178, 2021.

[3] K. Karunanithy and B. Velusamy, "An efficient data collection using wireless sensor networks and internet of things to monitor the wild animals in the reserved area," *Peer-to-Peer Networking and Applications*, vol. 15, no. 2, pp. 1105–1125, 2022.

[4] H. Esmaeili, B. M. Bidgoli, and V. Hakami, "CMML: combined metaheuristic-machine learning for adaptable routing in clustered wireless sensor networks," *Applied Soft Computing*, vol. 118, Article ID 108477, 2022.

[5] S. Navabi, C. Wang, O. Y. Bursalioglu, and H. Papadopoulos, "Predicting wireless channel features using neural networks," in *Proceedings of the 2018 IEEE international conference on communications (ICC)*, pp. 1–6, IEEE, Kansas City, MO, USA, May 2018.

[6] S. Jonas, A. O. Rossetti, M. Oddo, S. Jenni, P. Favaro, and F. Zubler, "EEG-based outcome prediction after cardiac arrest with convolutional neural networks: performance and visualization of discriminative features," *Human Brain Mapping*, vol. 40, no. 16, pp. 4606–4617, 2019.

[7] Y. Kong and E. Saeedi, "The investigation of neural networks performance in side-channel attacks," *Artificial Intelligence Review*, vol. 52, no. 1, pp. 607–623, 2019.

[8] N. Taşpinar and M. N. Seyman, "Back propagation neural network approach for channel estimation in OFDM system," in *Proceedings of the 2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, pp. 265–268, IEEE, Beijing, China, June 2010.

[9] Z. Huang, D. Corrigan, and S. Narayanan, "Distributed and dynamic spectrum management in airborne networks," in *Proceedings of the MILCOM 2015-2015 IEEE Military Communications Conference*, pp. 786–791, IEEE, Tampa, FL, USA, October 2015.

[10] P. F. Su, S. Y. Xu, and X. L. Yu, "Network delay prediction algorithm for narrowband communication networks based on WOA-LSTM," *Hebei Industrial Science and Technology*, vol. 39, no. 1, pp. 9–15, 2022.

[11] Z. Wang and T. Zhang, "A delay prediction method for train communication networks based on improved LS-SVM algorithm," *Urban Rail Transit Research*, vol. 24, no. 1, pp. 101–106, 2021.

[12] W. G. Shi and M. Guo, "Network delay prediction based on MEEMD-PE and CS-WNN models," *Systems Engineering and Electronics Technology*, vol. 42, no. 1, pp. 184–190, 2020.

[13] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: a deep learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 227–236, 2020.

[14] S. I. Popoola, E. Adetiba, A. A. Atayero, N. Faruk, and C. T. Calafate, "Optimal model for path loss predictions using feed-forward neural networks," *Cogent Engineering*, vol. 5, no. 1, Article ID 1444345, 2018.

[15] S. Wu, H. J. Deng, and Y. Q. Sheng, "A network delay prediction model based on gated recurrent neural networks," *Web New Media Technology*, vol. 10, no. 06, pp. 29–37, 2021.

[16] J. Zhu, S. Zang, L. Jian, L. Shi, and G. Ye, "Network delay prediction in large-scale MIMO networking scenarios," *Journal of Anhui University (Natural Science Edition)*, vol. 44, no. 02, pp. 35–42, 2020.

[17] M. K. Samimi and T. S. Rappaport, "3-D millimeter-wave statistical channel model for 5G wireless system design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 7, pp. 2207–2225, 2016.

[18] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[19] W. Lu, J. Li, J. Wang, and L. Qin, "A CNN-BiLSTM-AM method for stock price prediction," *Neural Computing and Applications*, vol. 33, no. 10, pp. 4741–4753, 2021.

[20] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015.

[21] Z. M. He and W. J. Li, "Flower pollination algorithm based on dynamic global search and Cauchy mutation," *Computer Engineering and Application*, vol. 55, no. 19, pp. 74–80, 2019.

[22] S. J. Liu, Y. Yang, and Y. Q. Zhou, "A group intelligence algorithm lion group algorithm," *Pattern Recognition and Artificial Intelligence*, vol. 31, no. 5, pp. 431–441, 2018.

[23] S. P. Zhu, B. Keshtegar, M. E. A. Ben Seghier, E. Zio, and O. Taylan, "Hybrid and enhanced PSO: novel first order reliability method-based hybrid intelligent approaches," *Computer Methods in Applied Mechanics and Engineering*, vol. 393, Article ID 114730, 2022.

[24] W. Li, "Improving particle swarm optimization based on neighborhood and historical memory for training multi-layer perceptron," *Information*, vol. 9, no. 1, p. 16, 2018.

[25] P. Ghosh, A. Karmakar, J. Sharma, and S. Phadikar, "CS-PSO based intrusion detection system in cloud environment," *Emerging Technologies in Data Mining and Information Security*, pp. 261–269, Springer, Singapore, 2019.

[26] T. Ma, C. Xu, Z. Zhou, and L. Zhong, "SE-PSO: resource scheduling strategy for multimedia cloud platform based on security enhanced virtual migration," in *Proceedings of the 2019 15th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 650–655, IEEE, Tangier, Morocco, June 2019.

[27] S. P. Usha Kirana and D. A. D'Mello, "Energy-efficient enhanced Particle Swarm Optimization for virtual machine consolidation in cloud environment," *International Journal of Information Technology*, vol. 13, no. 6, pp. 2153–2161, 2021.

[28] V. Garg, A. Shukla, and R. Tiwari, "AERPSO—an adaptive exploration robotic PSO based cooperative algorithm for multiple target searching," *Expert Systems with Applications*, vol. 209, Article ID 118245, 2022.

[29] S. Nabi, M. Ahmad, M. Ibrahim, and H. Hamam, "AdPSO: adaptive PSO-based task scheduling approach for cloud computing," *Sensors*, vol. 22, no. 3, p. 920, 2022.

[30] W. Y. Gong, Z. H. Cai, C. X. Ling, and L. Hui, "Enhanced differential evolution with adaptive strategies for numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, pp. 397–413, 2011.

[31] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[32] S. Couch, Z. Kazan, K. Shi, A. Bray, and A. Groce, "A differentially private Wilcoxon signed-rank test," 2018, https://arxiv.org/abs/1809.01635.

[33] P. E. Mcknight and J. Najab, *Mann-Whitney U Test*, John Wiley and Sons, Inc, Hoboken, NJ, USA, 2010.

[34] M. H. Essai Ali and I. B. M. Taha, "Channel state information estimation for 5G wireless communication systems: recurrent neural networks approach," *PeerJ Computer Science*, vol. 7, 2021.

[35] X. Gao, Y. Guo, D. A. Hanson, Z. Liu, M. Wang, and T. Zan, "Thermal error prediction of ball screws based on PSO-LSTM," *The International Journal of Advanced Manufacturing Technology*, vol. 116, no. 5-6, pp. 1721–1735, 2021.

[36] K. Chen, "APSO-LSTM: an improved LSTM neural network model based on APSO algorithm," in *Proceedings of the Journal of Physics: Conference Series*, vol. 1651, no. 1, August 2020.

[37] Y. Liu, Y. Liu, and Q. Song, "Continuous prediction of lower-limb joint torque based on IPSO-LSTM," in *Proceedings of the 2022 4th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP)*, pp. 45–49, IEEE, Hangzhou, China, July 2022.

[38] S. P. Ji, Y. L. Meng, L. Yan, G. S. Dong, and D. Liu, "GRU-Corr neural network optimized by improved PSO algorithm for time series prediction," *The International Journal on Artificial Intelligence Tools*, vol. 29, Article ID 2040010, 2020.

[39] A. M. Mahmood, A. Al-Yasiri, and O. Y. Alani, "Cognitive neural network delay predictor for high speed mobility in 5G C-RAN Cellular Networks," in *Proceedings of the 2018 IEEE 5G World Forum (5GWF)*, pp. 93–98, IEEE, Silicon Valley, CA, USA, July 2018.

[40] R. Tripathi and K. Rajawat, "Adaptive network latency prediction from noisy measurements," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 807–821, 2021.

[41] K. Glinskiy, A. Kureev, A. Krasilov, and E. Khorov, "PABAFT: channel prediction approach based on autoregression and flexible TDD for 5G systems," *Electronics*, vol. 11, no. 12, p. 1853, 2022.

[42] B. N. Alsunbuli, W. Ismail, and N. M. Mahyuddin, "Convolutional neural network and Kalman filter-based accurate CSI prediction for hybrid beamforming under a minimized blockage effect in millimeter-wave 5G network," *Applied Nanoscience*, vol. 13, no. 2, pp. 1539–1560, 2021.