IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 January 2023, accepted 23 January 2023, date of publication 30 January 2023, date of current version 2 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3240444

RESEARCH ARTICLE

A Capability Fitting and Data Reconstruction Model Based on Particle Swarm Optimization-Bidirectional Deep Neural Network for Search and Rescue System of Systems

YAN GAO[1,2], HU LIU[1], FU NIU[2], AND YONGLIANG TIAN[1]
[1]School of Aeronautic Science and Engineering, Beihang University, Beijing 100083, China
[2]Academy of Systems Engineering, Beijing 100166, China

Corresponding author: Yongliang Tian (tianyongliang_buaa@163.com)

ABSTRACT Search and rescue (SAR) is an important part of joint operations and a key support for combat effectiveness. Because of the complex composition of the SAR system of systems (SoS), sensitivity analysis method is usually used to carry out sensitivity analysis of SoS capability, so as to determine the main design indicators affecting SoS capability. Reliable sensitivity analysis results are often based on the analysis for sufficient data. However, the SAR SoS capability is affected by many factors and there are numerous design indicators. Even if a small number of design points are selected for each design indicator, tens of thousands of test schemes will be formed, and carrying out all simulation tests will bring huge workload and time cost. To solve this contradiction, this paper introduces a bidirectional deep neural network (BDNN), and takes advantage of its better self-learning and adaptive features and unique structure to train the existing test data, Through strong feature extraction ability of BDNN, the network model between the design indicator and capability indicator is formed, namely, the capability fitting and data reconstruction (CFDR) model, so that the implicit relationship between the two is fixed into the model. In the training process, the number of hidden layers and neurons in each hidden layer, and the amount of training data are explored according to the training effect, so as to obtain a better parameter combination. In order to avoid introducing large cumulative errors accumulated during BDNN pre-training into DNN, particle swarm optimization (PSO) was introduced to optimize weight parameters and avoid large training errors being transmitted to deep neural network (DNN). Meanwhile, three basic functions were used to verify the strong global optimization and convergence abilities of the BDNN after optimized by the PSO (PSO-BDNN). Finally, the new test scheme is applied to the CFDR model to obtain the SoS capability value. The reconstructed data obtained from the CFDR model based on BDNN and PSO-BDNN respectively were compared with the simulation test data.The results show that the reconstruction accuracy of the CFDR model based on the PSO-BDNN is greatly improved than that of the BDNN. And the feasibility of this model as a reconstruction data generation model and the effectiveness of this model as an analysis data extension method applied to the sensitivity analysis of insufficient data to obtain reliable analysis results are verified.

INDEX TERMS Capability fitting and data reconstruction model, bidirectional deep neural network, particle swarm optimization, search and rescue, SoS capability sensitivity analysis.

The associate editor coordinating the review of this manuscript and approving it for publication was Mitra Mirhassani.

I. INTRODUCTION

In modern high-tech wars and local conflicts, search and rescue (SAR) in distress is not only an action of rescuing an

individual life, but also often develops into severe military and political events. In other words, it is of great significance to successfully perform SAR actions to improve the action capability of the joint force.

With the development of information technology and the improvement of simulation modeling technology, it is more and more complicated to evaluate the SoS capability. It is not only necessary to investigate the SoS capability values that can be achieved based on the current SAR equipment and its performance, but also to evaluate which factors have a greater influence on the capability indicators.

Traditional methods based on experts, intuition, knowledge, and logic, such as the analytic hierarchy process (AHP) [1], rule-based reasoning methods, and cloud models [2], are more subjective in the evaluation process and do not provide sufficient depth for the analysis of the key factors affecting SAR capability. However, sensitivity analysis, as a data-based analysis method, is often used to study the degree to which the model output or model itself is influenced by changes in the input factors. This method is not limited by the model structure, and is more suitable for the SoS capability analysis of high-dimensional, nonlinear, and complex SoS [3], [5], [6], where the study of nonlinear and nonmonotonic models can be carried out using analytical or simulation methods, and all input parameters can be varied simultaneously; thus, the method is more suitable for capability evaluation of this type of SAR SoS.

When using the sensitivity analysis method to evaluate the SAR SoS capability, sufficient analytical data is the basis for obtaining reliable sensitivity analysis results. However, the SAR SoS capability is affected by many factors and there are numerous design indicators. Even if a small number of design points are selected for each design indicator, tens of thousands of test schemes will be formed, and carrying out all simulation tests will bring huge workload and time cost. Therefore, for SAR simulation systems that cannot quickly generate new data, it is important to study how to use existing data to generate new data to meet the requirements and complete SAR SoS capability analysis using sensitivity analysis methods.

To solve the above problems, the common way is adopting interpolation and other methods to generate new data to expand the analytical data to meet the requirements of sensitivity analysis for the data volume and data form. Recently, some researches have been conducted on this topic. For example, Li et al. [7] used the Bootstrap method to transform the small-sample problem into a large-sample problem to evaluate the performance storage reliability of equipment components. Zhang et al. [8] used SVM regression to replace the least square method for data fitting and then carried out a reliability analysis using the Weibull distribution. In the case of small-sample data, good curve-fitting results were obtained. However, the Bootstrap method mainly uses the method of putting back and repeated sampling to obtain new samples, which results in a large deviation, and there is a certain dependence on the selection of the empirical distribution and the number of samples. The problem that SVM is sensitive to noise or outliers still needs to be solved, and this method needs to consider the influence of the dispersion of small sample data and kernel function selection on the regression accuracy and generalization ability of the SVM method.

The SAR SoS has the characteristics of object diversification, spatial multi-dimensionality, and time real-time, which leads to great uncertainties in simulation results and distribution. Therefore, the above methods are no longer suitable for the analysis of such issues and it is difficult to stably obtain effective results.

Deep learning has better self-learning, self-organization and adaptive ability [9], can automatically extract features from original data without human participation, reducing the instability and randomness caused by human factors. After ten years of development, some deep learning models have been proposed successively, such as convolutional neural network (CNN), recurrent neural network (RNN), etc. In general, when selecting the deep learning model, the selection and training of neural network model should according to the actual application scene, so as to achieve the best results. Although CNN and RNN can also realize feature extraction, they have their own characteristics and can play an important role in different scenes. For example, CNN is a neural network specially used to process data with similar grid structure (which can be regarded as two-dimensional pixel grid), so it performs exceptionally well in large-scale image processing problem [10]. However, the complex network structure may greatly increase the training time. RNN is a neural network used to process sequential data, such as audio data and language representation, so RNN has excellent performance in machine translation, speech recognition and other fields. However, RNN shares the same weight parameters at multiple levels and steps [11], which is not suitable for the application scenario in this article.

Different from the CNN and RNN, BDNN owns the following characteristics: Firstly, its multi-layer structure characteristic together with more information processing units results in a strong learning and fitting ability for complex and nonlinear models under multidimensional space, and the process does not depend on the accurate mathematical model. Also, it allows input samples with larger dispersion [12], [13]. Secondly, BDNN learns one hidden layer feature representation each time through greedy learning. Then, stacks the restricted Boltzmann machine (RBM) layer by layer to obtain the more advanced and abstract feature representation of the original input data, which can better describe the complex data structure. Thirdly, adding unsupervised pre-training learning process can make full use of unlabeled data to obtain good initial model, which can effectively improve the training effect and generalization ability of the model. At the same time, due to the pre-training for deep belief network (DBN), parameters of each layer and the connection weights between layers in the network can be obtained through training, which is different from the random initialization of traditional neural network. Therefore, it has high efficiency and avoids the problem that the traditional neural network is easy to fall into the local optimization in the training process.

Therefore, this paper introduces the BDNN, based on the existing test data, takes advantage of strong feature extraction ability of the BDNN to construct the network model between the design indicators and SoS capability. Meanwhile, PSO was introduced to optimize weight parameters and avoid large training errors being transmitted to DNN. Through the formed model can effectively reconstruct the data required for SoS capability sensitivity analysis, which can effectively solve the problem of existing data not meeting the requirements of sensitivity analysis, reducing the number and cycle of simulation tests, and improving the effectiveness of sensitivity analysis. The research framework is shown in Fig. 1.

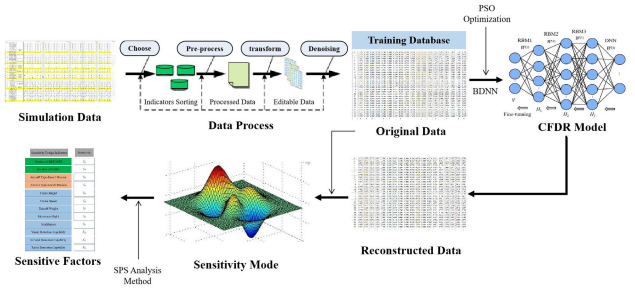


FIGURE 1. Research framework.

II. SIMULATION TEST ENVIRONMENT AND INDICATORS SYSTEM CONSTRUCTION

In this study, we take land SAR of pilots in distress as the background, on the basis of SAR concept describing the connection relationship and SAR mission of SAR nodes, as shown in Fig. 2, based on the process of SAR action, carry out the simulation of SAR action in the SAR simulation system to obtain the simulation data.

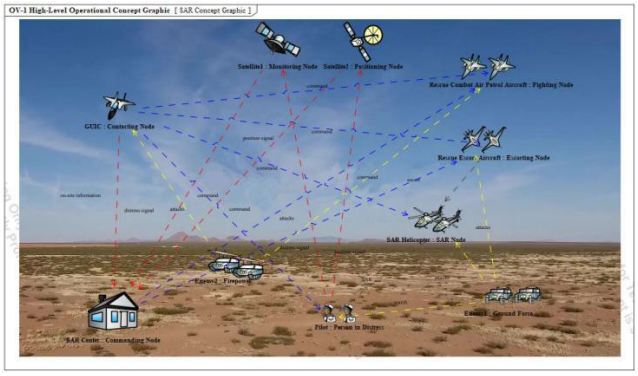


FIGURE 2. SAR concept graphic.

A. SAR SIMULATION SYSTEM DESIGN AND DEVELOPMENT

First, based on the composite modeling method combining multi-agent with DEVS (discrete event system), the SAR scenario was constructed as shown in Fig. 3.

Second, based on the SAR simulation system framework shown in Fig. 4 and the simulation design process shown in Fig. 5, simulation and deduction of SAR action was carried out, as shown in Fig. 6.

FIGURE 3. SAR scenario construction.

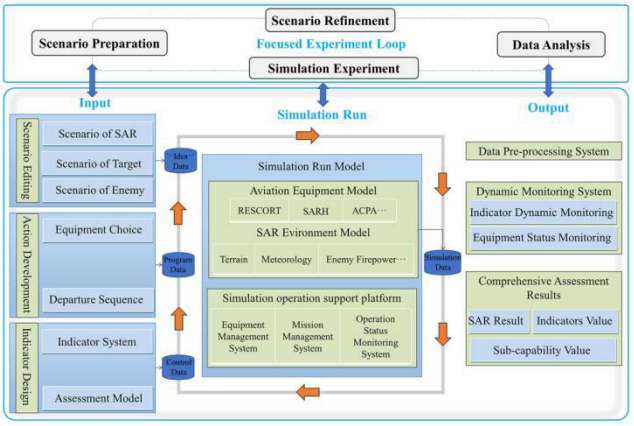


FIGURE 4. SAR simulation system framework.

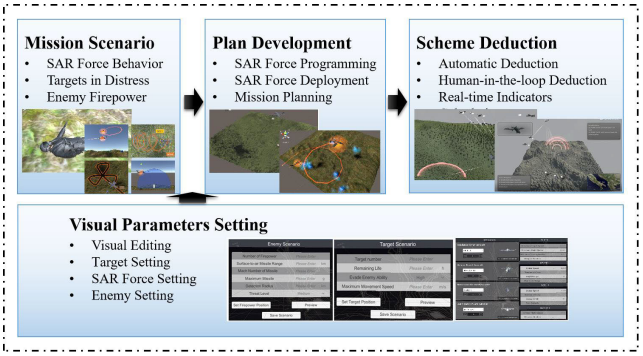


FIGURE 5. Simulation design process.

The simulation design process includes visual parameters setting module (including visual editing, target setting, SAR force setting, enemy setting function), mission scenario editing module (including SAR force behavior, targets in distress, enemy firepower), plan development module (including SAR force programming, SAR force deployment, mission planning) and scheme deduction module (including scheme automatic deduction, human-in-the-loop command deduction, real-time indicators caculation). The visual parameters setting module provides support for the mission scenario editing module and plan development module. After completing the two modules, enter the scheme deduction module. Simulation data will be generated after the simulation deduction is completed and stored in the simulation database.

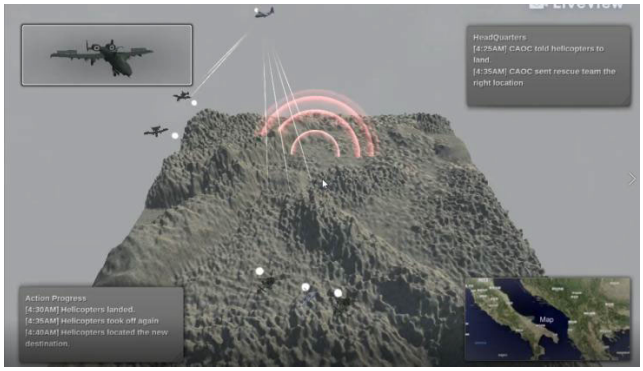


FIGURE 6. The process of simulation evolution.

B. SAR CAPABILITY ANALYSIS AND INDICATOR SYSTEM CONSTRUCTION

1) SAR CAPABILITY ANALYSIS

To analyze the degree of influence of aviation equipment on the SAR SoS, it is necessary to construct an indicator system in a selective manner according to the research objectives, the needs of the analysis or evaluation, and the level and content of the analysis [14], [15], [16]. Before constructing the indicator system, the SAR SoS capability must be analyzed.

Capability analysis is the core link between strategic objectives and development programs for aviation equipment and is used to clarify the capabilities required for aviation equipment to accomplish its mission and to determine indicators reflecting capabilities and the specific equipment and its performance to achieve them [17], [18].

According to reference [19], the survival rate of injured personnel after 24 h was reduced by 80%, and the survival rate of uninjured personnel after 3 days decreases significantly. For a rescue time of more than 5 h, the possibility of survivors being successfully rescued was 20%. If the rescue time is reduced to 1.8 h, the probability of a successful rescue is increased to 60%, which means that the time is reduced by 1/3 and the possibility of distressed personnel being rescued is increased by a factor of three. Thus, time is the largest enemy of a SAR mission, and it is crucial to reduce the rescue time to ensure the smooth implementation of SAR action. Second, in the process of SAR, if there is damage to SAR equipment or casualties of SAR personnel, the purpose fails to achieve a successful SAR of personnel in distress but also causes greater losses. Therefore, rescuing as many casualties as possible means that the possibility of personnel being captured is greatly reduced, thus improving the morale and combat capabilities of the side.

Therefore, the core of SAR action is a fast, efficient, successful, and safe process, which means using as few SAR forces as possible, in the shortest possible time, to successfully SAR for the greatest number of combat casualties.

2) INDICATOR SYSTEM CONSTRUCTION

For SAR SoS capability analysis, the accuracy of analysis results is closely related to the construction of indicator system. The more detailed and accurate the indicators, the more in-depth the SAR SoS will be described, and the final analysis results will be more accurate. However, too many indicators will increase the amount of calculation and waste unnecessary resources. This paper constructed the indicator system according to the following principles:

1) Simplicity: In order to reduce the complexity of analysis, unnecessary indicators should be removed as far as possible to evaluate the whole SoS;

2) Testability: the selected indicators should be objective as far as possible, and specific values can be obtained through tests or other methods;

3) Integrity: The selected indicators cover all aspects of evaluation as far as possible and can measure all contents to be evaluated.

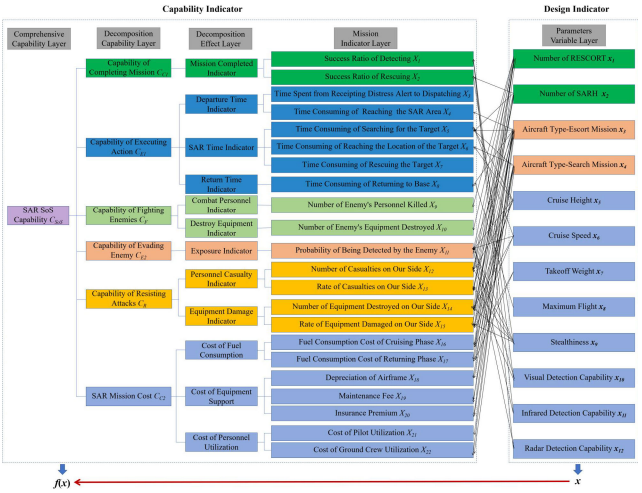


FIGURE 7. Indicator decomposition hierarchy model.

The indicator system is shown in Fig.7, it consists of two parts: capability indicator and design indicator. The index decomposition hierarchy model is adopted for capability indicator. Specifically, based on the SAR capability analysis, SAR SoS capability is decomposed into the following six sub-capabilities: completing mission, executing action, fighting enemies, evading enemies, resisting attacks and SAR mission cost, then they are decomposed layer-by-layer until the specific mission indicators [20], [21] whose values can be obtained through simulation, and then six sub-capabilities and the whole SoS capability can be obtained through capability aggregation., The CFDR model based on BDNN or PSO-BDNN in this paper mainly fits the relationship between input variables and SoS capability through the existing simulation data. Then through the CFDR model, the SoS capability under the new input variables are obtained. The existing and reconstructed data are used as the analysis data for sensitivity analysis, the indicators that have a great influence on the SoS capability are obtained, which can provide guidance for design optimization and decision support for actual SAR mission.

This paper mainly focuses on the design indicators shown in Fig.7 and takes them as input factors of SAR simulation system. The output factor selects the SoS capability (if it

is necessary to examine the influence degree of the design indicators on a certain sub-capability or mission indicator, then they can also be selected as the output factor). The output factor value can be obtained by simulation after the input factor is determined. The capability analysis model connecting input factors and output factor is shown in (1).

$$Y = f(x_1, x_2, \cdots, x_i, \cdots, x_n) = f(x) \tag{1}$$

where, Y is output vector, x is input vector, $x = [x_1, x_2, \cdots, x_i, \cdots, x_n]$, x_i is the ith design indicator.

III. BDNN STRUCTURE AND TRAINING

A. BDNN STRUCTURE

The BDNN can learn high-level features more abstractly and can show excellent data features extraction ability not only for image datasets but also for other types of datasets. In this study, the existing historical simulation dataset was trained to extract features from the input data and obtain the output from the last hidden layer, which is the forward training phase. Then, a back-propagation algorithm was introduced to adjust the initial network parameters obtained from the previous phase, which avoids the drawbacks of slow convergence and local optimality in simple networks.

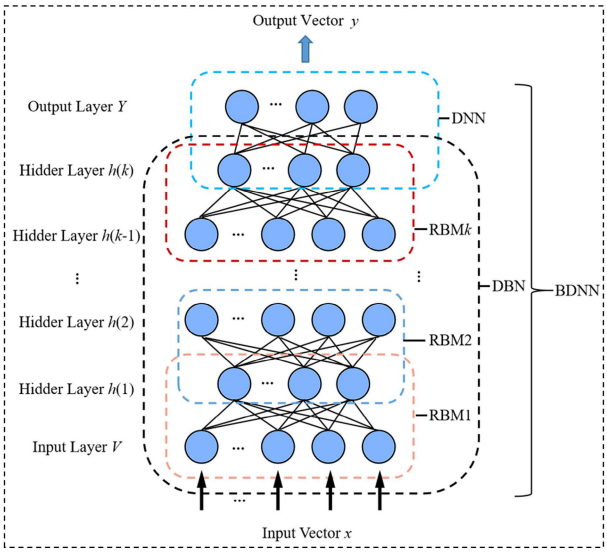


FIGURE 8. BDNN structure.

Fig. 8 shows the BDNN structure schematic diagram. The BDNN consists of a deep belief network (DBN) [22], [23], [24] and a deep neural network (DNN).

The DBN was first proposed by Hinton et al., and promoted the development of deep learning networks and was first successfully applied in the fields of image and speech recognition [25], and later fields of textual and visual data representation. A DBN has great advantages in feature extraction compared with traditional deep learning methods. The original data enter the network through the input layer, and the features are automatically abstracted by processing the weights layer-by-layer. In this process, the weights are pre-trained and not randomly assigned, and no expert experience is required, which reduces the uncertainty caused by human intervention in the traditional method and thus improves accuracy.

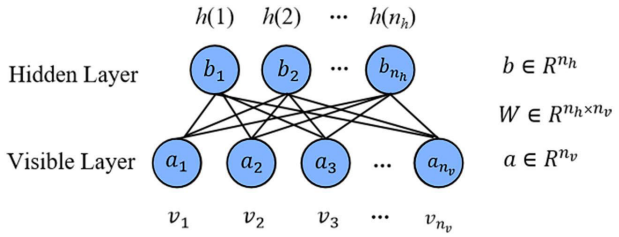


FIGURE 9. RBM structure.

Further, the DBN consists of many restricted Boltzmann machine (RBM) [26] whose structure [27], [28], [29] is shown in Fig. 9, where the hidden layer of the 1st RBM serves as the visible layer of the 2nd RBM, and by stacking layer by layer, the DBN structure is formed. There is no connection between the same layer or cross-layers except the connection between adjacent layers, which reduces the complexity and achieves better results.

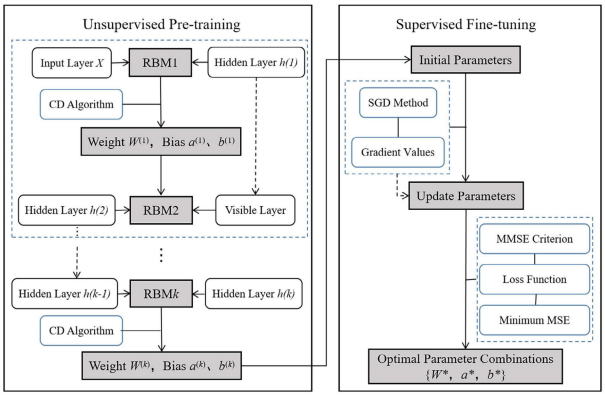
The DNN consisted of the hidden layer of the last RBM and a fully connected BP neural network acting as the regression layer of the entire network. The output feature vector of the last hidden layer of the DBN is used as the input vector of the DNN, and the pre-training process of the DBN can be regarded as the process of initializing the parameters of the DNN.

B. BDNN TRAINING PROCESS

This section refers to the training of the BDNN, and the specific process is as follows: after unsupervised pre-training of the DBN through the layer-by-layer greedy method, and then supervised tuning using the back propagation algorithm. The training process and pseudo code are shown in Fig. 10.

In algorithm 1, x_0 is the training samples, n is the number of visible layers, m is the number of hidden layers, η is the learning rate of the unsupervised pre-training process, k is the number of parameters updates, W is the connection weight matrix, a is the bias vector of the visible layer, b is the bias vector of the hidden layer, X is the input sample of the DNN, s is the number of input sample, Y is the output sample, $W^{(k)}$ is the connection weight matrix of the kth RBM, $a^{(k)}$ is the bias vector of visible layer of the kth RBM, $b^{(k)}$ is the bias vector of hidden layer of the kth RBM, ε is the learning rate of the supervised tuning process, l is the number of iterations, p is the number of samples for each update, d is the number of layers of DNN, I is the regularization coefficient, f is the momentum factor, X' is the reconstruction input, Y' is the reconstruction output, $\{W^*, a^*, b^*\}$ is optimal parameter combinations.

During the unsupervised pre-training process, the training method for the RBM can be applied to the DBN. Every two neighbouring layers of the DBN from the bottom to the top can be considered as one RBM. Then, the RBM is trained using the contrastive divergence (CD) algorithm to obtain corresponding parameters. The output parameters of



Algorithm 1: Main training steps of BDNN network

Step1:Unsupervised pre-training of DBN based on CD algorithm;
input : x_0, n, m, η, k.
output: $W^{(k)}$, $a^{(k)}$, $b^{(k)}$.
Initialization: Set $v^{(0)} \leftarrow x_0$; W, a and b are the smaller randomly selected values.
for $t \leftarrow 0$ **to** $k-1$ **do**
  for $i \leftarrow 1$ **to** n **do**
    for $j \leftarrow 1$ **to** m **do**
      $P(h_j^{(t)} = 1|v^{(t)}) \leftarrow \sigma(b_j + \sum_{i=1}^{n} v_i^{(t)} \times W_{ij})$;
      Drawing $h_j^{(t)} \in \{0,1\}$ from $P(h_j^{(t)} = 1|v^{(t)})$
    end
    $P(v_i^{(t+1)} = 1|h^{(t)}) \leftarrow \sigma(a_i + \sum_{j=1}^{m} W_{ij} \times h_j^{(t)})$;
    Drawing $v_i^{(t+1)} \in \{0,1\}$ from $P(v_i^{(t+1)} = 1|h^{(t)})$
  end
  $W^{(t+1)} \leftarrow W^{(t)} + \eta \times (P(h_j^{(t)} = 1|v^{(t)}) \times v^{(t)^T} - P(h_j^{(t+1)} = 1|v^{(t+1)}) \times v^{(t+1)^T})$;
  $a^{(t+1)} \leftarrow a^{(t)} + \eta \times (v^{(t)} - v^{(t+1)})$;
  $b^{(t+1)} \leftarrow b^{(t)} + \eta \times (P(h_j^{(t)} = 1|v^{(t)}) - P(h_j^{(t+1)} = 1|v^{(t+1)}))$;
end
Step2:Supervised fine-tuning of the DNN network based on back-propagation algorithm;
input : X, s, Y, $W^{(k)}$, $a^{(k)}$, $b^{(k)}$, ε, l, k, p, d, I, f.
output: X', Y', $\{W^*, a^*, b^*\}$.
Initialization: Set $W^* \leftarrow W^{(0)} \leftarrow W^{(k)}$, $a^* \leftarrow a^{(0)} \leftarrow a^{(k)}$, $b^* \leftarrow b^{(0)} \leftarrow b^{(k)}$.
for $u \leftarrow 0$ **to** $d-2$ **do**
  if $I > 0$ **then**
    $dW \leftarrow \frac{\partial E}{\partial W^{(u)}} + I \times W^{(u)}$;
    $da \leftarrow \frac{\partial E}{\partial a^{(u)}} + I \times a^{(u)}$;
    $db \leftarrow \frac{\partial E}{\partial b^{(u)}} + I \times b^{(u)}$;
  end
  if $f > 0$ **then**
    $dW \leftarrow f \times dW + dW$;
    $da \leftarrow f \times da + da$;
    $db \leftarrow f \times db + db$;
    $W^{(u)} \leftarrow W^{(u)} - \varepsilon \times dW$;
    $a^{(u)} \leftarrow a^{(u)} - \varepsilon \times da$;
    $b^{(u)} \leftarrow b^{(u)} - \varepsilon \times db$;
  end
end
for $i \leftarrow 1$ **to** l **do**
  for $j \leftarrow 1$ **to** k **do**
    for $q \leftarrow (j-1) \times p + 1$ **to** $j \times p$ **do**
      $X' \leftarrow X \cdot randperm(q)$;
      $Y' \leftarrow Y \cdot randperm(q)$;
    end
  end
end
$E \leftarrow 1/N \times \sum_{u=1}^{N} (Y'(W^{(u)}, a^{(u)}, b^{(u)}) - Y)^2$; /* Loss function */
if $E(u+1) < E(u)$ **then**
  $W^* \leftarrow W^{(u+1)}$;
  $a^* \leftarrow a^{(u+1)}$;
  $b^* \leftarrow b^{(u+1)}$;
end

FIGURE 10. Training process and pseudo code of BDNN.

the previous RBM are used as the input parameters of the next RBM until the last RBM is trained, and the parameters of the entire DBN are then obtained. The specific training process is illustrated in Fig. 11.

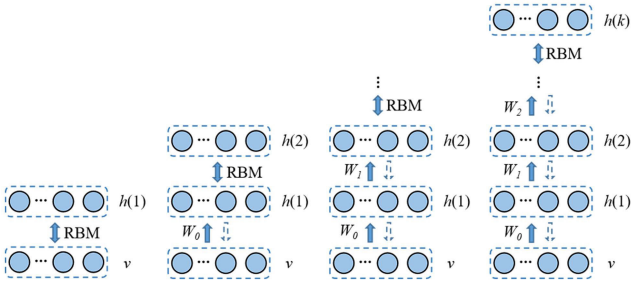


FIGURE 11. Unsupervised pre-training process of DBN.

First, the input layer v and the 1st hidden layer $h(1)$ are taken as the RBM, and the parameters including the connection weights w between the input layer v and the hidden layer $h(1)$ and the bias a of the input layer and the bias b of the hidden layer are trained using the CD algorithm. After obtaining the parameters of the current RBM, they are no longer changed; then the current hidden layer $h(1)$ is used as the visible layer to form a new RBM with the hidden layer $h(2)$, and the current RBM is trained and the parameters are obtained, until the end.

The pre-training process for the DBN is mainly through layer-by-layer training, and the parameters obtained can only ensure that the feature extraction for the input is optimal for current layer, to ensure the optimality of the overall results, the weights of the network need to be further optimized in the following process which is named as fine-tuning process. This paper introduced DNN to adjust the parameters obtained from the pre-training process.

The DNN is used in various fields because of its low requirements for mathematical models, strong nonlinear mapping ability and fault tolerance, and good robustness. Typically, a DNN is considered as a deep neural network, and the parameters derived from the previous training step are used as the initial parameters of the DNN. Certain criteria are chosen and based-errors to adjust the parameters [30], and the data are reconstructed so that the data back-propagated to the visible layer is closer to the original data. In this study, the minimum mean-square error (MMSE) criterion was used to determine the direction of parameter optimization, at which point the parameter-tuning problem can be transformed into a mean-square error minimization optimization problem. The gradient descent method can be used to solve extremal problems. While the training trajectory of the traditional gradient descent method would show a jagged shape, which significantly prolongs the training time. Simultaneously, owing to the oscillation phenomenon, only a small learning rate can be set to avoid deviating from the minimum value caused by the large step size. Therefore, in this study, stochastic gradient descent (SGD) was used and the weights of the overall network were learned in a supervised manner using the back propagation method treating the weight error as a high-dimensional function with each element of the weight vector as a variable, so that it seeks the minimum value of the training error by continuously adjusting the weights,

and updates the weights in the direction of the gradient descent of the loss function. This results in a better network performance and a more stable training process for the entire network. This process is known as supervised tuning. The specific steps are as follows:

(1) The MMSE criterion is used to measure the effect of parameters update, and the parameters update is completed when the mean square error is minimum. Where the loss function is defined as follows.

$$E = \frac{1}{N}\sum_{\mathrm{i}=1}^{N}\left(\hat{Y}_{\mathrm{i}}\left(W^{(k)}, a^{(k)}, b^{(k)}\right) - Y_{\mathrm{i}}\right)^2 \tag{2}$$

where, E is the mean square error; $\hat{Y}_i$ and Y_i, are the actual and ideal values of the ith output layer, respectively; $W^{(k)}$ is the weight, $a^{(k)}$ and $b^{(k)}$ are biases of the kth RBM, respectively.

(2) The back propagation algorithm is used to obtain the gradient values of each layer to update the weights and bias parameters of the network, and the updating process is as follows.

$$W^{(k)} = W^{(k)} - \varepsilon \cdot dW \tag{3}$$
$$a^{(k)} = a^{(k)} - \varepsilon \cdot da \tag{4}$$
$$b^{(k)} = b^{(k)} - \varepsilon \cdot db \tag{5}$$

where, ε is the learning rate.

(3) The weights are gradually adjusted to minimize the mean square error of the loss function by the above weight update rule, which results in the optimal combination of parameters W^*, a^* and b^*.

IV. CFDR MODEL CONSTRUCTION AND ANALYSIS

A. CFDR MODEL BASED ON BDNN FRAMEWORK AND TEST DESIGN

The SoS capability evaluation based on the sensitivity analysis method requires high data completeness, and sometimes it is not possible to directly evaluate the SoS ability using existing data. The BDNN has a strong feature extraction capability owing to its unique composition, and its core is to automatically learn from the data set to extract the appropriate features. Thus, this study applies deep learning to existing simulation data to build a CFDR model based on a BDNN to improve the data needed for SAR SoS capability sensitivity analysis.

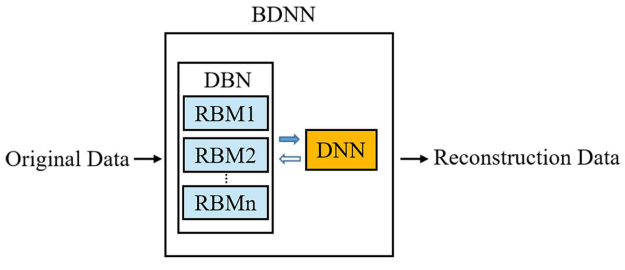


FIGURE 12. CFDR model framework based on BDNN.

The framework of the BDNN-based CFDR model is illustrated in Fig. 12. The original data form the input vector and the output vector represents the reconstructed data. The sample data are input from the visible layer of the RBM, and through the visible layer and the hidden layer neurons, the output of the hidden layer maximally fits the input data of the visible layer to complete the feature extraction and form CFDR model. To verify the accuracy of the extracted features, Input factors of test data are put into the CFDR model, and the results obtained are compared with the output factors.

TABLE 1. Setting for design points of design indicators.

Design Indicators	Variation Range	Design Point 1	Design Point 2	Design Point 3
Number of RESCORT x_1	2 ~ 4	2	3	4
Number of SARH x_2	1 ~ 3	1	2	3
Aircraft Type-Escort Mission x_3	-1 ~ 1	-1	0	1
Aircraft Type-Search Mission x_4	-1 ~ 1	-1	0	1
Cruise Height x_5	2800~3200 m	2800	3000	3200
Cruise Speed x_6	240~350 km/h	240	290	350
Takeoff Weight x_7	4819~5953 kg	4819	5178	5953
Maximum Flight x_8	600~1159 km	600	1010	1159
Stealthiness x_9	0.73~0.75 m^2	0.73	0.74	0.75
Visual Detection Capability x_{10}	0.70~0.98	0.70	0.84	0.98
Infrared Detection Capability x_{11}	0.73~0.96	0.73	0.84	0.96
Radar Detection Capability x_{12}	0.74~0.98	0.74	0.86	0.98

In the indicator system (as shown in Fig.7), multiple design points can be set for each design indicator. Assuming that each design indicator has m design points, so x can be expressed as follows:

$$x = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \tag{6}$$

When $n = 12$, $m = 3$, there are $3^{12} = 531441$ groups of test scheme, If each group of test scheme is simulated and deduced in the SAR simulation system, the corresponding mission indicator value is obtained, and then through capability aggregation to obtain the SoS capability value, it is undoubtedly a huge project and a long period with several minutes each time.

Table 1 shows the setting for design points of the design indicators in Fig.7, where RESCORT and SARH respectively represent rescue escort aircraft and search and rescue helicopter. The type of aircraft in escort mission and search mission includes fixed-wing aircraft, armed helicopter and unmanned aerial vehicle (UAV) which are assigned the values 1,0 and -1, respectively. x_5-x_{12} are the performance or functional parameters of SARH. In this paper, design indicators are randomly sampled to obtain 8000 groups of scheme, and corresponding output factor values are obtained through SAR simulation system. The 8000 sets of data including input factor and output factor value constitute the test data, which are divided into 6000 sets of training data and 2000 sets of test data to train the model.

The test data used in this part of the experiment were obtained by processing the simulation data of the SAR simulation system. The specific data processing steps were shown in Fig. 13.

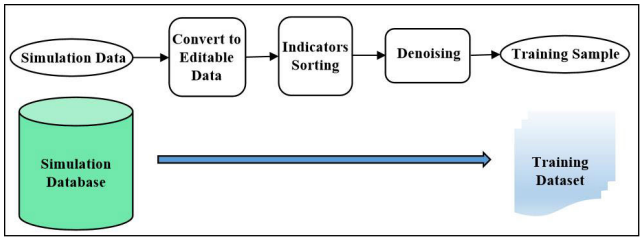


FIGURE 13. Data processing steps.

B. CFDR MODEL PARAMETERS OPTIMIZATION-SEEKING

1) EFFECTS OF STRUCTURAL PARAMETERS ON RECONSTRUCTION RESULTS

First, structural parameters such as the number of hidden layers and neurons in each hidden layer of the data reconstruction model were analyzed to study the effects of different structural parameters on the network performance. Second, the effects of the amount of training data on the network performance were investigated.

The root mean square error (RMSE) and mean absolute error (MAE) were mainly used as indicators to evaluate the performance of the network. The RMSE and MAE were calculated as shown in (7) and (8), respectively. The calculation process is presented in algorithm 2 which is shown in Fig. 14.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|Y_i' - Y_i\right|^2} \tag{7}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i' - Y_i\right| \tag{8}$$

where, Y_i' is the reconstructed data, Y_i is the original data, N is the number of batches in the training process.

For the CFDR model based on BDNN, this section first focuses on the selection of two parameters: the number of hidden layers and the number of neurons in each hidden layer.

The number of neurons in the input layer was determined by the dimension of the input data [31] which was 12 in this experiment, thus setting the number of neurons in the input layer to 12. To obtain the best model structure, the number of neurons in the 1st hidden layer is first set, and the number is fixed after finding the best value according to the evaluation index. A hidden layer is then added to determine the best number of neurons in the new hidden layer, and so on, until the reconstruction accuracy is no longer improved. The changes in the RMSE and MAE values during the BDNN training process when the number of neurons in the 1st hidden layer was 20, 30 and 40, respectively, are shown in Fig. 15.

From Fig. 15, it can be seen that when the number of neurons is small (e.g., 20), the BDNN tends to be stable

Algorithm 2: Calculating reconstruction error

output: $RMSE$, MAE

$Error_1 \leftarrow Error_2 \leftarrow 0$ /* Initializing the error */

/* For each sample */

for $t \leftarrow 1$ **to** N **do**

  $h \sim P(|v^{(t)})$ /* Sampling the hidden layer */

  $v \sim P(|h)$ /* Sampling the visible layer */

  $Error_1 \leftarrow Error_1 + (v^{(t)} - v)^2$;

  $Error_2 \leftarrow Error_2 + abs(v^{(t)} - v)$;

end

$RMSE \leftarrow sqrt(1/N \times Error_1)$; /* root mean square error */

$MAE \leftarrow 1/N \times Error_2$; /* mean absolute error */

FIGURE 14. Pseudo code for calculating reconstruction error.

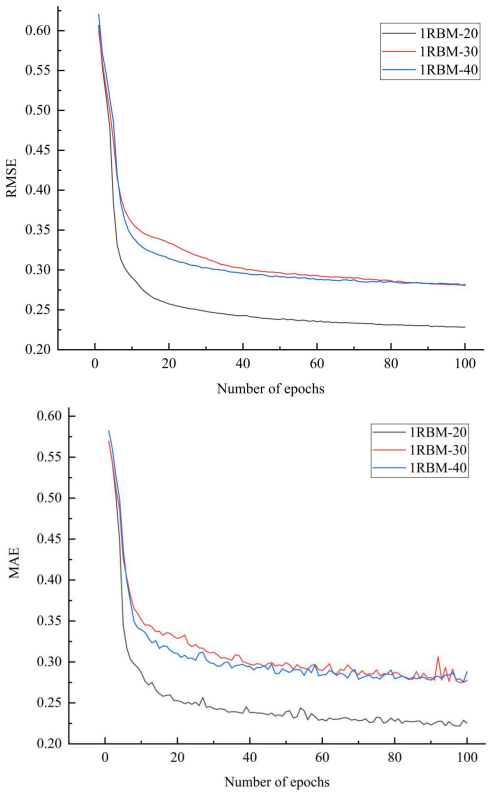


FIGURE 15. The change of RMSE and MAE value during BDNN training process under different number of neurons.

with a smaller number of training times, and the error is smaller. However, when the number of neurons was increased (e.g., 30 or 40), the error increased. This may be due to the fact that on the one hand, when the number of neurons is increased, the model is complex, the problem of overfitting and the need for constant dynamic adjustment of training parameters may occur, which leads to a low accuracy instead; on the other hand, when the number of neurons is set to 20 that close to the dimension of the input data, which has a strong data carrying capacity and can completely extract the features from the input data. Therefore, in this study, the number of neurons in the 1st hidden layer was set to 20, and the number of neurons in the remaining hidden layers was obtained in the same way.

Table 2 lists the values of the evaluation index of the data reconstruction model for different numbers of hidden layers and different numbers of neurons in each hidden layer, with

TABLE 2. Model evaluation index values under different model parameters.

Number of Hidden Layer	Number of Neurons	$RMSE_{min}$	MAE_{min}	Training Time of One Epoch
1 (1RBM)	20	0.235	0.224	1.5008
	30	0.292	0.300	1.5325
	40	0.283	0.283	1.5717
2 (2RBM)	20	0.315	0.283	1.6258
	30	0.277	0.276	1.6415
	40	0.286	0.279	1.6736
3 (3RBM)	20	0.261	0.265	1.7002
	30	0.224	0.235	1.7094
	40	0.276	0.268	1.8319

three cases of 1, 2 and 3 layer set for the hidden layers and three choices of 20, 30, and 40 for the number of neurons in each hidden layer. To reduce the error of randomly selected experimental samples, 10 times are conducted under each iteration, and the model is trained separately. Then the values of the evaluation index are calculated separately, and finally, the average value is obtained.

As can be seen from Table 2, the accuracy of the reconstructed data obtained increases within a certain range as the number of hidden layers and the number of neurons in each hidden layer increase, and the time to train the network also increases, but the training time increases insignificantly compared to the increase in accuracy, thus the network structure containing three hidden layers is chosen in this study.

It can also be seen that when the number of hidden layers is increased from 1 to 2, the RMSE of the reconstructed data is larger, indicating a greater volatility of the results, which may be related to the change of the RBM type.

Thus, on the basis of the training data selected in this section, the reconstruction effect of the model is better when choose the data reconstruction model with a 3 hidden layer structure and the 1st hidden layer, the 2nd hidden layer and the 3rd hidden layer contains 20 neurons, 30 neurons, and 30 neurons, respectively.

The training process and training time of the network under different number of hidden layers are given in Fig. 16, where the number of neurons is set to 20 in the 1st hidden layer, 30 in the 2nd hidden layer, and 30 in the 3rd hidden layer, respectively. As can be seen from Fig. 16, the training results under different number of hidden layers match the trends in Table 2.

In summary, the network structure of the obtained CFDR model is a 5-layer structure with three hidden layers whose neurons number are 20, 30, and 30, respectively, as shown in Fig. 17.

2) EFFECTS OF THE AMOUNT OF TRANING DATA ON RECONSTRUCTION RESULTS

After analyzing the effects of model parameters and structure on the CFDR model performance, the effects of the amount of training data on the CFDR model performance are analyzed below.

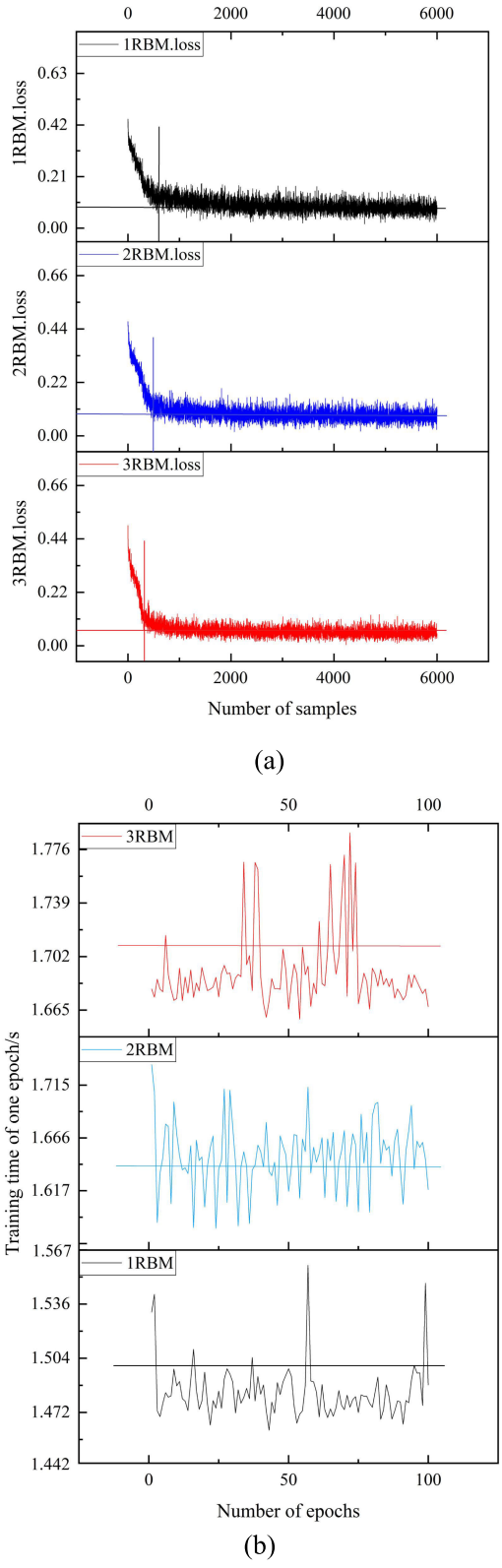


FIGURE 16. Training process and training time of one epoch of BDNN under different number of hidden layers. (a) training process; (b)training time.

In this section, 6000 and 7000 sets of training data are used to train the BDNN, respectively. 10 times are conducted

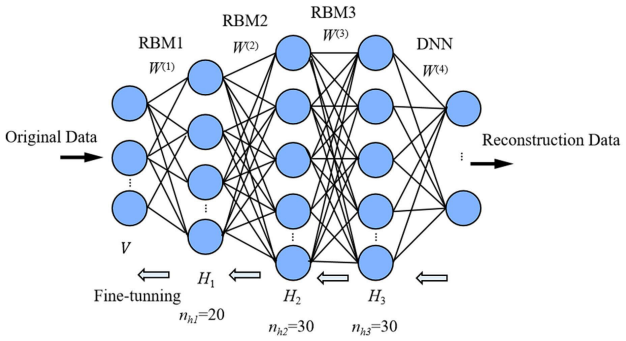


FIGURE 17. Better BDNN structure obtained.

under each iteration, and the results are averaged to obtain a comparison plot of the original and reconstructed curves, as shown in Fig. 18. To reduce the amount of calculation, this section conducts comparison and analysis for misssion indicators.

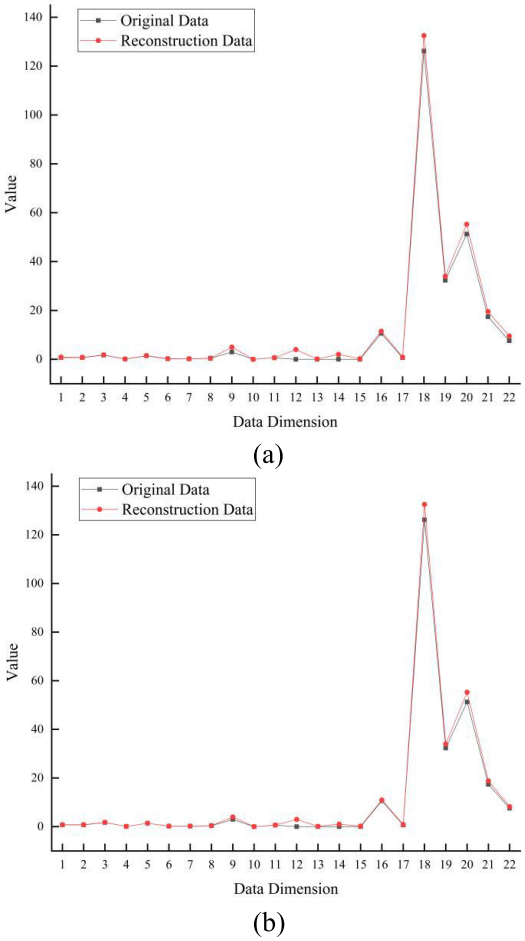


FIGURE 18. Reconstruction effects under different amount of training data. (a) 6000 sets of training data; (b)7000 sets of training data.

As can be seen from Fig. 18 and Table 3, the difference between the reconstructed data and the original input data decreases as the amount of training data increases, indicating that when the amount of training data increases, the feature

TABLE 3. Model evaluation index values under different amount of training data.

Amount of Training Data	$RMSE_{ave}$	$RMSE_{min}$	MAE_{ave}	MAE_{min}
6000	0.685	0.224	0.579	0.235
7000	0.573	0.217	0.501	0.220

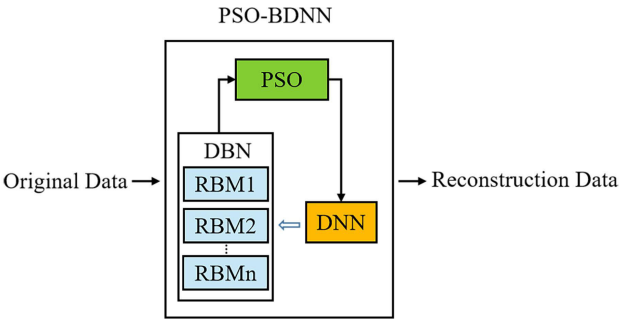


FIGURE 19. The framework of the CFDR model based on PSO-BDNN.

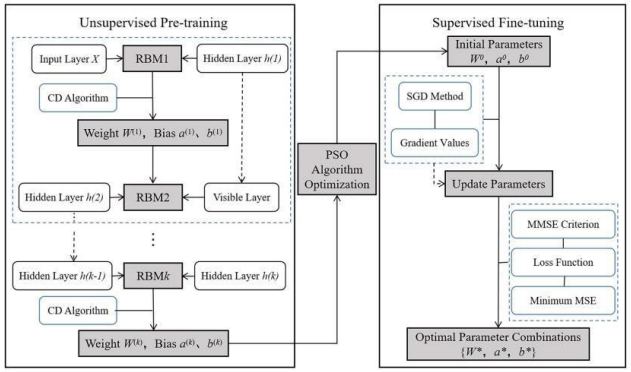
extraction ability is enhanced and the extracted features can better fit the input data, and thus the fitting accuracy of the CFDR model improves, which also illustrates the validity of the CFDR model based on the BDNN.

V. NETWORK OPTIMIZATION OF CFDR MODEL BASED ON BDNN

A. CFDR MODEL BASED ON PSO-BDNN FRAMEWORK AND NETWORK PERFORMANCE ANALYSIS

BDNN has good application prospects in the field of predictive and reconstruction analysis owing to its strong approximation ability for nonlinear functions and are not dependent on accurate mathematical model, strong fault tolerance and robustness, and high accuracy. However, there are also some problems, for example, the pre-training process for the DBN is mainly through layer-by-layer training, and the parameters obtained can only ensure that the feature extraction for the input is optimal for current layer, but with the increase of the number of training layers, the training error of the previous layer will be gradually transferred to the next layer during the training process, resulting in a larger error of the whole network at the end.

To solve these problems, some optimization algorithms have been proposed in recent years, such as the chaotic optimization algorithm (COA) and genetic algorithm (GA), which are mainly used to optimize model parameters. COA is more complex, sensitive to the initial parameters, and has a long search time [32], [33]. GA needs to set more initial parameters, and its selected values are dependent on manual experience. The final results obtained are random and uncertain, with poor reliability and slow convergence [34]. PSO is simpler, requires fewer initial parameters to be set, has high reliability [35], and is more advantageous for cases with multiple variables, non-linearity, discontinuity, and non-integrability features compared to COA and GA.



Algorithm 3: Main training steps of BDNN network

Step1:Unsupervised pre-training of DBN based on CD algorithm;
input : x_0, n, m, η, k.
output: $W^{(k)}$, $a^{(k)}$, $b^{(k)}$.
Step2:Optimize parameters of DBN network based on PSO algorithm;
input : C_1, C_2, D, M, G_{max}, V_{max}, V_{min}, Pos_U, Pos_L, W, η.
output: global optimal value P_g.
// Training Stage
Initialization: Set $P_{gt} \leftarrow Pos_1$, $V_i \leftarrow 0$.
Parameters Mapping: Map the parameters {W,a,b} according to equation (10) and equation (11).
Training: Input sample data for training.
for $t \leftarrow 1$ **to** G_{max} **do**
  /* Initializing particles' position and velocity */
  /* for all particles */
  for $i \leftarrow 1$ **to** M **do**
    /* for all independent variables */
    for $j \leftarrow 1$ **to** D **do**
      $Pos_i(j) \leftarrow Pos_L(j) + (Pos_U(j) - Pos_L(j)) \times rand(1)$;
      $V_i(j) \leftarrow V_{min}(j) + (V_{max}(j) - V_{min}(j)) \times rand(1)$;
    end
    $P_{bi} \leftarrow Pos_i$;
    $P_{gt} \leftarrow P_{bi}$;
  end
  /* Calculating particles' fitness value */
  $fitness(Pos_i) = \sum_{j=1}^{D}(Pos_i(j))^2$
  /* Calculating particles' velocity and position */
  for $i \leftarrow 1$ **to** M **do**
    for $j \leftarrow 1$ **to** D **do**
      $V_i(j) = W \times V_i(j) + C_1 \times rand(1) \times (P_b - Pos_i(j)) + C_2 \times rand(1) \times (P_g - Pos_i(j))$;
      $Pos_i(j) \leftarrow Pos_i(j) + V_i(j)$;
    end
  end
  /* Seeking the individual optimal position */
  for $i \leftarrow 1$ **to** M **do**
    if $fitness(Pos_i) < fitness(P_{bi})$ **then**
      $P_{bi} \leftarrow Pos_i$;
    end
  end
  /* Seeking the global optimal position */
  for $i \leftarrow 1$ **to** M **do**
    if $fitness(P_{b(i+1)}) < fitness(P_{bi})$ **then**
      $P_g \leftarrow P_{b(i+1)}$;
    end
  end
  if $fitness(P_{g(t+1)}) < fitness(P_{gt})$ **then**
    $P_g \leftarrow P_{g(t+1)}$;
  end
end
$(W^0, a^0, b^0) \leftarrow P_g$
Step3:Supervised fine-tuning of the DNN network based on back-propagation algorithm;
input : X,s, Y, W^0, a^0, b^0, ε, l, k, p, d, I, f.
output: $X^{'}$, $Y^{'}$, {W*, a^*, b^*}.

FIGURE 20. Training process and pseudo code of PSO-BDNN.

The BDNN constructs a multi-hidden layer network structure with excellent performance in regression prediction analysis through an unsupervised greedy layer-by-layer training algorithm, however, the BDNN is susceptible to fall into a local optimum situation due to random initialization when calculating the weights, thus this paper uses the PSO algorithm to optimize the weights to obtain better BDNN performance. The framework of the CFDR model based on PSO-BDNN in this section is shown in Fig. 19.

The optimization of the BDNN through the PSO algorithm mainly involves treating the parameters to be optimized in the DBN, such as connection weights and biases, as the parameters of the individual position vectors of the particle swarm (the mapping relationship is in the form of floating-point encoding, as shown in (7) and (8)), and putting them into the particle swarm for iterative optimization, and seeking the most suitable value by comparing the final error, at which time the position vector parameters of the particle represents the optimal parameters value of the DBN network, and the optimized parameters is used as the initial parameters of the DNN.

$$Pos = \left(W_1^{(k)}, W_2^{(k)}, \cdots, W_m^{(k)}, a_1^{(k)}, a_2^{(k)}, \cdots, a_m^{(k)}, b_1^{(k)}, b_2^{(k)}, \cdots, b_m^{(k)}\right) \tag{9}$$

$$Pos_i = \left(W_i^{(k)}, a_i^{(k)}, b_i^{(k)}\right) \tag{10}$$

where, m is the number of neurons in the last hidden layer of the DBN, W is the connection weight, a is the bias of the visible layer, and b is the bias of the hidden layer.

As shown in Fig. 20, the training process of PSO-BDNN is as follows.

First, the DBN was conducted unsupervised pre-training layer-by-layer to obtain the weight parameters.

Second, the PSO algorithm is used to optimize the parameters which are mapped with the parameters of the individual position vector of the particle population and then adjusted by the PSO algorithm.

Finally, the optimized and adjusted weight parameters were used as the initial parameters of the DNN, and the parameters were fine-tuned by selecting certain criteria. Take the fine-tuned weight parameters as the parameters of CFDR model.

where, C_1 is the individual experience learning factor, C_2 is the group experience learning factor, D is the number of independent variables, M is the group scale, G_{max} is maximum iterations, V_{max} is the particle maximum velocity, V_{min} is the particle minimum velocity, Pos_U is the particle position upper boundary, Pos_L is the particle position lower boundary, W is the inertia factor, η is the learning rate, P_b is the individual optimal position, P_g is the global optimal position.

To verify the convergence speed and global optimization-seeking ability of PSO, three typical benchmark functions such as sphere function, shaffer function and griewank function are selected for performance testing of BDNN and PSO-BDNN respectively, and MAE is used to characterize the accuracy of the algorithm.

To carry out the algorithm comparison, the number of iterations were all chosen to be 6000. In the PSO algorithm, the particle scale was set as 30, the two learning factors C1

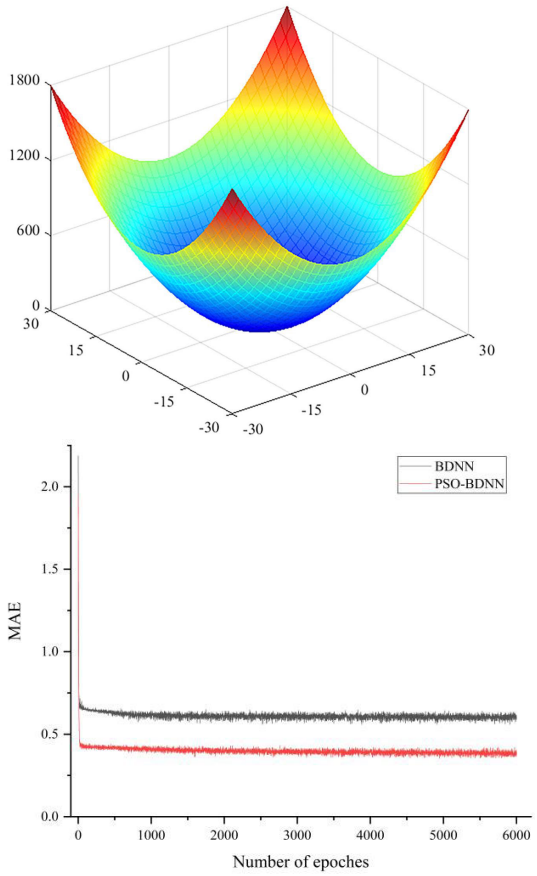


FIGURE 21. Sphere function image and convergence process of BDNN and PSO-BDNN based on the sphere function.

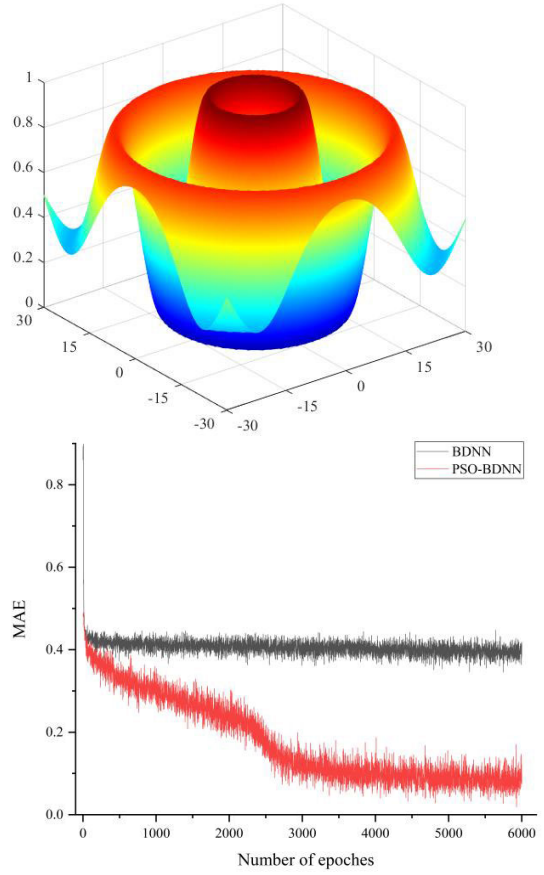


FIGURE 22. Shaffer function image and convergence process of BDNN and PSO-BDNN based on the shaffer function.

TABLE 4. The range of the normalized Xi value.

X_i	X_1	X_2	X_3	X_6	X_7
Range	0.10-0.88	0-0.72	0.10-0.55	0.28-0.61	0.38-0.55
X_i	X_8	X_9	X_{11}	X_{12}	X_{13}
Range	0.30-0.80	0.30-0.80	0-0.60	0-1	0-1
X_i	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}
Range	0-1	0-1	0.19-0.98	0.06-0.88	0.06-0.85
X_i	X_{19}	X_{20}	X_{21}	X_{22}	
Range	0.02-0.85	0.14-0.95	0.08-0.81	0.05-0.92	

and C2 were both 2, the initial inertia factor was 1.5, and the inertia factor was 0.4 at the maximum number of iterations. To eliminate randomness, 10 experiments were conducted each time and the average value was taken to compare the training accuracy of BDNN and PSO-BDNN.

Sphere function is shown in (11).

$$f(x) = \sum_{i=1}^{n} x_i^2, -30 \le x_i \le 30 \tag{11}$$

The sphere function belongs to the symmetric nonlinear function with a single peak, it is simpler and mainly used to test the algorithm's optimization-seeking accuracy. The function achieves the minimal value when x_i =0. Fig. 21 shows the three-dimensional image of sphere function when the dimension is 2 and the convergence process of BDNN and PSO-BDNN under the sphere function.

Shaffer function is shown in (12).

$$f(x, y) = 0.5 + \frac{\left(sin\sqrt{x^2 + y^2}\right)^2 - 0.5}{1.0 + 0.001\left(x^2 + y^2\right)^2}, -30 \le x, y \le 30 \tag{12}$$

The shaffer function belongs to the multi-peaked function. Fig. 22 shows the three-dimensional image of the shaffer function when the dimension is 2. As can be seen from the Fig.22, the shaffer function undergoes strong oscillations and is harder to converge to the global optimum. Fig. 22 also shows the convergence process of BDNN and PSO-BDNN under the shaffer function.

Griewank function is shown in (13).

$$f(x, y) = 0.5 + \frac{\left(sin\sqrt{x^2 + y^2}\right)^2 - 0.5}{1.0 + 0.001\left(x^2 + y^2\right)^2}, \quad -30 \le x, y \le 30 \tag{13}$$

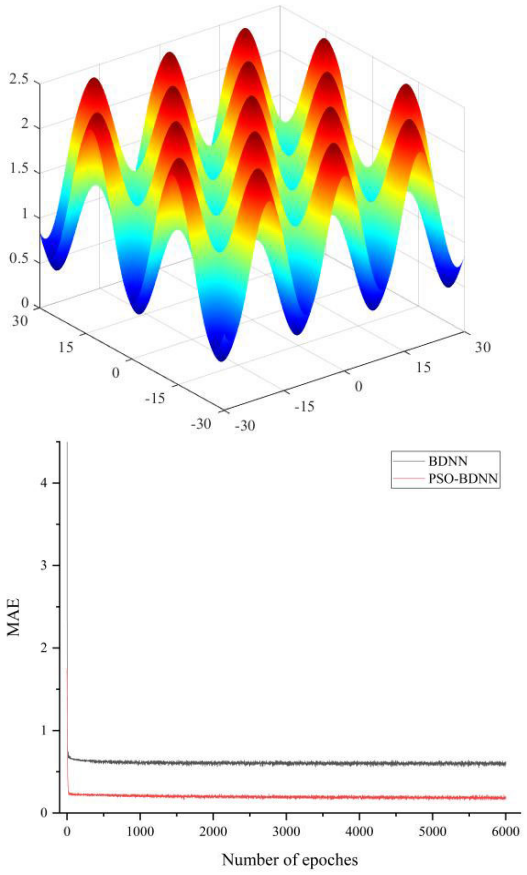


FIGURE 23. Griewank function image and convergence process of BDNN and PSO-BDNN based on the griewank function.

The griewank function is also a multi-peaked function, its image has the characteristics of rotation and non-separability. Fig. 23 shows the three-dimensional image of the griewank function when the dimension is 2 and the convergence process of BDNN and PSO-BDNN under the griewank function. It can be seen that the griewank function has unlimited number of local optima, which is easy to fall into local optima.

From Fig. 21, Fig. 22 and Fig. 23, it can be seen that both BDNN and PSO-BDNN fastly converge to the optimal value, but PSO-BDNN converges faster and with higher accuracy. Even under the shaffer function, PSO-BDNN gradually converges to the equilibrium state and maintains a higher accuracy level with the increase of the number of iterations, although a stronger oscillation occurs.

B. ANALYSIS OF THE RECONSTRUCTION EFFECT OF CFDR MODEL BASED ON PSO-BDNN

To further verify the performance of the PSO-BDNN, the reconstruction effects of the CFDR models based on the BDNN and the PSO-BDNN are compared, as shown in Fig. 24(a) and Fig. 24(b). Similarly, this section conducts comparison and analysis for misssion indicators. To make the observation clearer, the first 7-dimensional data are taken

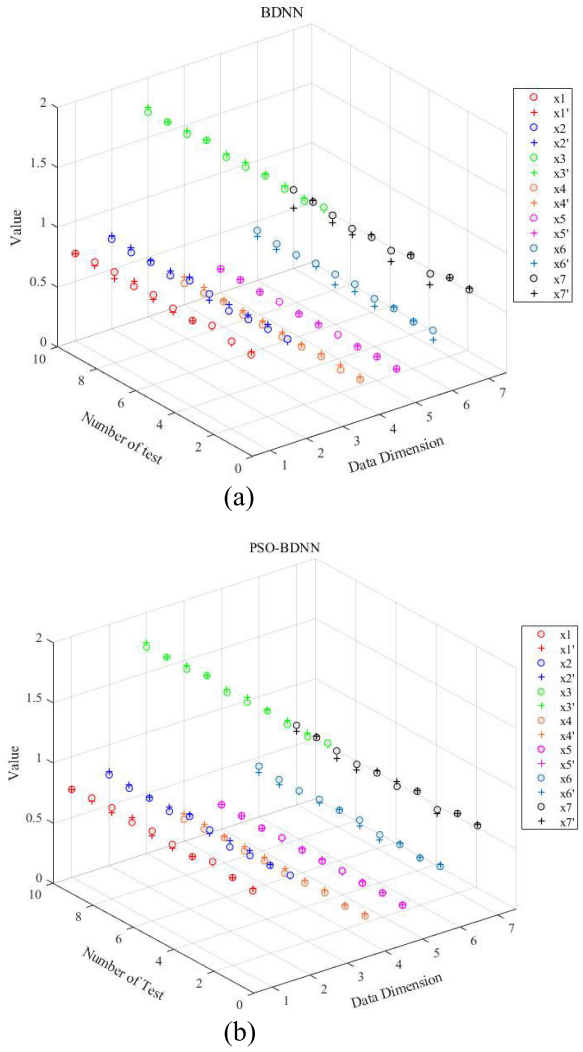


FIGURE 24. Reconstruction effects of the CFDR models based on the BDNN and the PSO-BDNN. (a) BDNN; (b) PSO-BDNN.

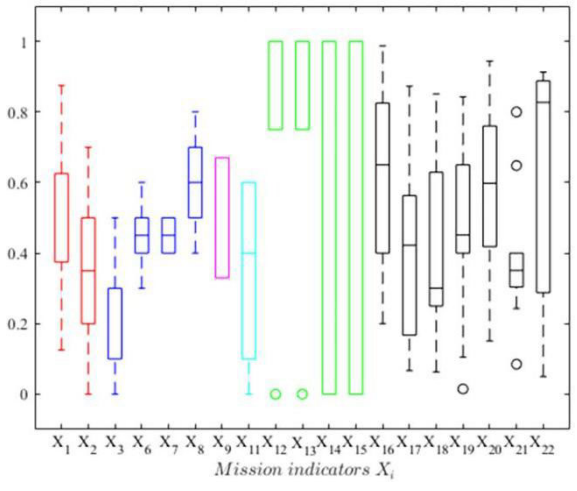


FIGURE 25. Distribution of the range of mission indicators values.

for comparative analysis in this subsection, where x_i is the normalized original data value of ith mission indicator, x_i' is the reconstruction value.

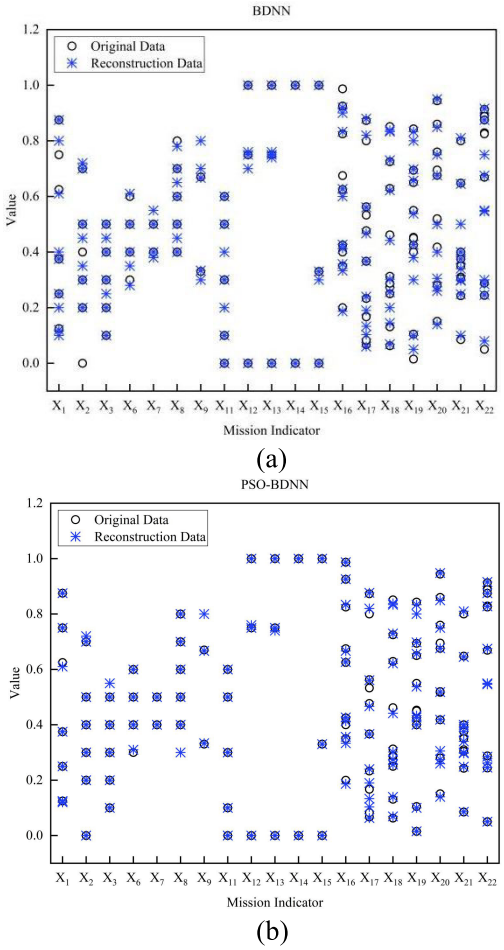


FIGURE 26. Reconstruction effect of the mission indicators under the two CFDR models. (a) BDNN; (b) PSO-BDNN.

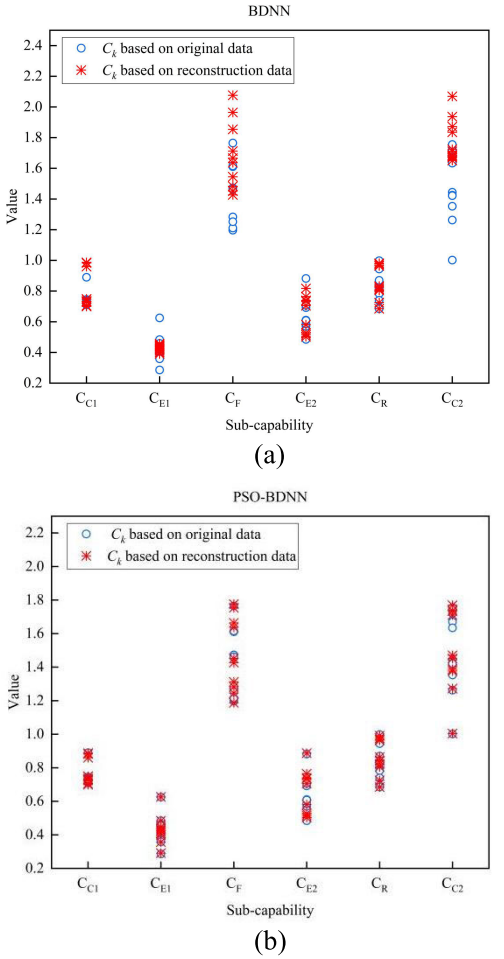


FIGURE 27. Sub-capability values under the two CFDR models. (a) BDNN; (b) PSO-BDNN.

It can be seen from Fig. 24 that the reconstruction accuracy is improved to a greater extent after the optimization of the BDNN through PSO, which is due to the weight parameters obtained from the unsupervised pre-training process are optimized and adjusted by PSO, then they are used as the initial parameters of the DNN, which avoids bringing the large accumulated errors during the training process of the DBN into the DNN, and thus the accuracy is improved to a greater extent.

VI. CFDR MODEL VALIDITY VERIFICATION

A. CAPABILITY FITTING EFFECT COMPARISON

The network of the CFDR model is optimized and a better network structure is determined. On this basis, 1000 groups of test schemes are selected to input the CFDR model to obtain the corresponding capability indictor (including mission indicators, sub-capabilities and SoS capability) values which are compared to the values obtained through SAR simulation system, then analyzed results to verify the effectiveness of the CFDR model.

The mission indicators of SAR SoS focus on the 22 indicators in Fig. 7 which are derived from the SAR simulation system, and the expressions of the SAR SoS capability are shown in (14) and (15), which is CEFERC SoS capability calculation model, and the expression of the sub-capabilities is shown in (16).

$$C_{SoS} = f\left(C_{C_1}, C_{E_1}, C_F, C_{E_2}, C_R, C_{C_2}\right) \tag{14}$$

$$C_{SoS} = k \cdot C_{C_1}^{W_{C_{C_1}}} \cdot C_{E_1}^{W_{C_{E_1}}} \cdot C_F^{W_{C_F}} \cdot C_{E_2}^{W_{C_{E_2}}} \cdot C_A^{W_{C_R}} \cdot C_{C_2}^{W_{C_{C_2}}} \tag{15}$$

$$C_k = \prod_{j=1}^{n} q_{kj}^{w_j} \tag{16}$$

where, C_{sos} is the capability of the whole SAR SoS, $C_{C_1}, C_{E_1}, C_F, C_{E_2}, C_R$ and C_{C_2} are the capability of completing mission, the capability of executing action, the capability of fighting enemies, the capability of evading enemies, the capability of resisting attacks and the SAR mission cost, respectively, $W_{C_{C_1}}, W_{C_{E_1}}, W_{C_F}, W_{C_{E_2}}, W_{C_R}, W_{C_{C_2}}$ are the weight corresponding to each sub-capability, respectively, which can be derived from the AHP analysis method, and k is the correction coefficient, C_k is the kth sub-capability, $k = 1, 2, \ldots, 6$, q_{kj} is the jth mission indicator within the kth sub-capability, w_j is the weight of q_{kj}(i.e. normalized X_i value).

TABLE 5. Mission indicator values from the three types of data.

X_i	Original Data				Reconstruction Data (BDNN)				Reconstruction Data (PSO-BDNN)			
X_1	0.125	0.375	0.875	…	0.120	0.375	0.800	…	0.125	0.375	0.750	…
X_2	0.400	0.300	0.200	…	0.450	0.350	0.200	…	0.400	0.300	0.250	…
X_3	0.400	0.200	0.500	…	0.450	0.250	0.500	…	0.400	0.200	0.550	…
X_6	0.500	0.400	0.600	…	0.400	0.500	0.500	…	0.500	0.310	0.500	…
X_7	0.400	0.500	0.400	…	0.380	0.500	0.400	…	0.500	0.400	0.400	…
X_8	0.600	0.400	0.600	…	0.600	0.500	0.700	…	0.600	0.500	0.600	…
X_9	0.670	0.670	0.330	…	0.800	0.800	0.333	…	0.667	0.667	0.333	…
X_{11}	0.500	0.600	0.100	…	0.600	0.600	0.100	…	0.500	0.500	0.100	…
X_{12}	0.000	1.000	0.750	…	0.000	0.750	0.760	…	0.000	0.760	1.000	…
X_{13}	0.750	0.000	1.000	…	0.760	0.000	1.000	…	0.740	0.000	1.000	…
X_{14}	1.000	1.000	0.000	…	1.000	1.000	0.000	…	1.000	0.000	1.000	…
X_{15}	1.000	0.000	1.000	…	1.000	0.000	0.000	…	1.000	0.000	0.000	…
X_{16}	0.675	0.200	0.987	…	0.625	0.187	0.833	…	0.627	0.187	0.926	…
X_{17}	0.367	0.067	0.873	…	0.367	0.060	0.820	…	0.467	0.063	0.877	…
X_{18}	0.462	0.063	0.851	…	0.442	0.069	0.730	…	0.456	0.069	0.838	…
X_{19}	0.425	0.015	0.843	…	0.427	0.050	0.657	…	0.427	0.016	0.833	…
X_{20}	0.418	0.151	0.676	…	0.400	0.140	0.725	…	0.418	0.140	0.676	…
X_{21}	0.310	0.085	0.648	…	0.249	0.100	0.645	…	0.340	0.086	0.645	…
X_{22}	0.288	0.050	0.669	…	0.285	0.080	0.675	…	0.246	0.050	0.675	…

TABLE 6. Sub-capability values and SoS capability values calculated from the three types of data.

Type of Data	Sub-capability						C_{SoS}
	C_{C_1}	C_{E_1}	C_F	C_{E_2}	C_A	C_{C_2}	
Original Data	0.825	0.443	1.575	0.643	0.806	1.508	0.802
Reconstruction Data（CFDR model based on BDNN）	0.712	0.399	1.425	0.752	0.756	1.116	0.696
Reconstruction Data（CFDR model based on PSO-BDNN）	0.748	0.410	1.725	0.720	0.788	1.802	0.724
Accuracy（BDNN）	86.3%	90.0%	90.5%	83.0%	93.8%	74.0%	86.8%
Accuracy（PSO-BDNN）	90.7%	92.5%	90.5%	88.0%	97.8%	81.5%	91.3%

The range of the normalized X_i value (i.e. value distribution of mission indicators) is shown in Fig. 25 and Table 4. In addition, the indicators with small variation intervals or take essentially constant values would not be listed in this study.

The mission indicator values from three ways are shown in Table 5. Fig. 26 gives the reconstruction effect of mission indicators under the two CFDR models, and it can be seen from the Fig. 26 that the CFDR model based on PSO-BDNN fits the original data better and has higher reconstruction accuracy compared to the BDNN, which is related to the introduction of the PSO algorithm to optimize the weight parameters of the BDNN.

According to (15) and (16), the mission indicators are calculated to obtain the six sub-capability values and SoS capability values, the comparison results are shown in Fig. 27 and Fig. 28.

It can be seen from Fig.27 and Fig.28 that the sub-capability values and SoS capability values calculated from the reconstructed data obtained from CFDR model based on PSO-BDNN are closer to those values calculated from the original data. Therefore, for SAR simulation systems that cannot quickly generate new data, appling the CFDR model based on PSO-BDNN to it to get the data that meet the requirements of sensitivity analysis to complete

TABLE 7. Comparison of main effect index of design indicators under the three ways.

Design Indicators	S_{x_i}		
	Original Data	Reconstruction Data (BDNN)	Reconstruction Data (PSO-BDNN)
x_1	0.45985	0.37013	0.46051
x_2	0.36956	0.37055	0.37073
x_3	0.23978	0.24881	0.29289
x_4	0.25848	0.22358	0.26915
x_5	0.11284	0.11186	0.12966
x_6	0.06529	0.05137	0.06925
x_7	0.07353	0.08329	0.06548
x_8	0.10866	0.07571	0.10198
x_9	0.03981	0.04048	0.03029
x_{10}	0.14225	0.15137	0.14048
x_{11}	0.14461	0.14024	0.14047
x_{12}	0.15160	0.15997	0.15569

TABLE 8. Sorting the main effect index of design indicators calculated under the three ways.

Type of Data \ Ranking	1	2	3	4	5	6	7	8	9	10	11	12
Original Data	x_1	x_2	x_4	x_3	x_{12}	x_{11}	x_{10}	x_5	x_8	x_7	x_6	x_9
Reconstruction Data （CFDR model based on BDNN）	x_2	x_1	x_3	x_4	x_{12}	x_{10}	x_{11}	x_5	x_7	x_8	x_6	x_9
Reconstruction Data （CFDR model based on PSO-BDNN）	x_1	x_2	x_3	x_4	x_{12}	x_{10}	x_{11}	x_5	x_8	x_6	x_7	x_9

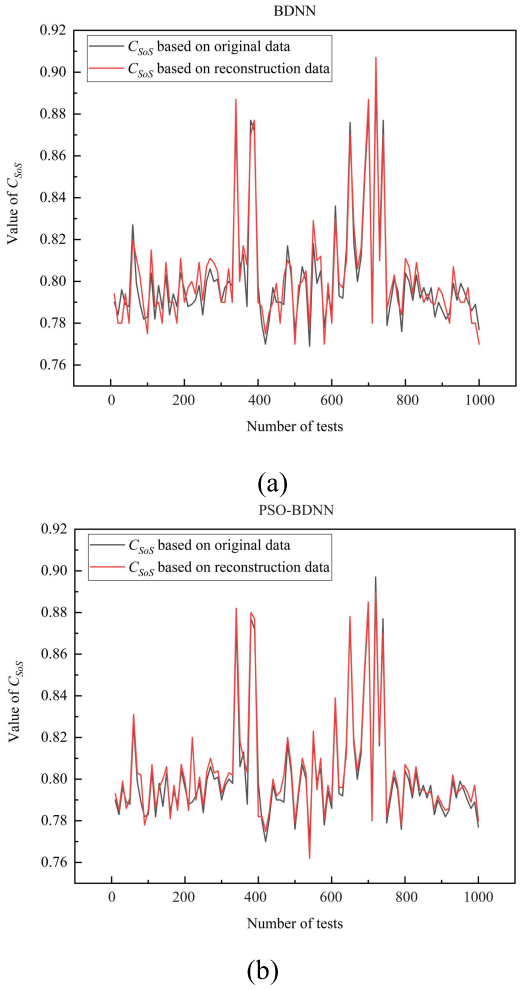


FIGURE 28. SoS capability values under the two CFDR models. (a) BDNN; (b) PSO-BDNN.

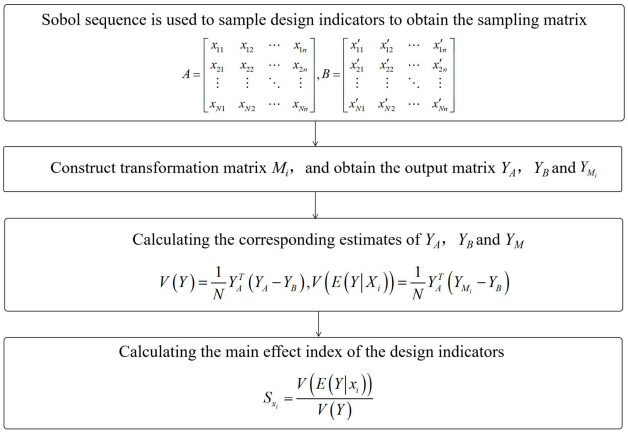


FIGURE 29. Main steps of SPS analysis method.

the capability sensitivity analysis of the SAR SoS is an effective way.

Table 6 shows the comparison results of the sub-capability values and SoS capability values calculated from the three types of data, respectively, and the data in the table are the average value of the results of 1000 trials. From Table 6, it can be seen that the accuracy of sub-capability and SoS capability under the CFDR model based on BDNN are 86.3% and 86.8%, respectively, and the accuracy of sub-capability and SoS capability under the CFDR model based on PSO-BDNN are 90.2% and 91.3%, respectively. Thus, the feasibility of CFDR model as a data generation model is verified

B. SENSITIVITY ANALYSIS RESULTS COMPARISON

Sensitivity analysis, as a data-based analysis method, is often used to study the degree to which the model output or model itself is influenced by changes in input factors, and is more suitable for SoS capability analysis of high-dimensional, nonlinear, and complex SoS.

Based on the above research, sobol power sensitivity (SPS) analysis method was used to calculate the main effect index of the input variable (design indicator) and the second-order interaction effect index of the input variable sum under the original data and the reconstructed data. The main effect index and second-order interaction effect index can be regarded as the sensitivity factor.

The main steps of SPS analysis method is shown in Fig.29.

The main effect index of the design indicators obtained under the three ways are shown in Table 7, and their ranking (descending order) is shown in Table 8.

As can be seen from Table 7 and Table 8, compared with the original data, there is a certain difference in the value of the main effect index of each design indicator obtained by CFDR model. However, in terms of ranking, the first 5 design indicators with the largest main effect index value are the same, as are the last 5 design indicators with the smallest

value. The overall ranking under the three ways is not much different, but the ranking under the CFDR model based on PSO-BDNN is closer to that of the original data. Thus, the effectiveness of CFDR model specially the model based on PSO-BDNN as an analysis data extension method applied to the sensitivity analysis of insufficient data to obtain reliable analysis results is verified.

VII. CONCLUSION

This paper introduces CFDR model to solve the problem that sensitivity analysis method requires high data quantity and form, while SAR simulation system cannot generate new data quickly, and obtain the following conclusions:.

1) The CFDR model based on the BDNN can effectively reconstruct the data required for SoS capability sensitivity analysis, which solves the problem of existing data not meeting the requirements of sensitivity analysis, so as to provide corresponding support for the optimization of the SAR SoS and improve the effectiveness of the SoS capability sensitivity analysis.
2) With an increase in the number of hidden layers and the number of neurons in each hidden layer, the reconstruction accuracy is improved within a certain range, and the time for training the network increases; however, compared with the improvement in accuracy, the increase in the training time is not obvious.
3) When the network structure was increased from one hidden layer to two hidden layers, the RMSE of the reconstruction was larger, indicating that the results had greater volatility, which may be related to the change in the RBM type.
4) The optimization of the BDNN through PSO improves the reconstruction accuracy to a greater extent, which is due to the weight parameters obtained from the unsupervised pre-training process are optimized and adjusted by PSO, then they are used as the initial parameters of the DNN, which avoids bringing the large cumulative errors during the training process of the DBN into the DNN, and thus the accuracy is improved to a greater extent.
5) Through the comparison and analysis of capability indicator values and sensitivity analysis results obtained from CFDR model and results from simulation, the feasibility of this model as a reconstruction data generation model and the effectiveness of this model as an analysis data extension method applied to the sensitivity analysis of insufficient data to obtain reliable analysis results are verified.

In this paper, the CFDR model based on PSO-BDNN is used to achieve the fitting of capability analysis model and data reconstruction, which provides a favorable support for the analysis of SoS capability sensitivity, and also provides a certain reference value for other types of SAR SoS or combat SoS. Meanwhile, the ranking of the sensitivity of design indicators has a certain engineering significance. On one hand, it provides strategic support for the commander in the face of the actual SAR mission. On the orther hand, it provides reference for future equipment design focus and development directionfor for aviation equipment designers. In this paper, BDNN is used for feature extraction, which does not require manual participation and automatically extracts features from original data. However, the initial parameter setting of BDNN still relies on manual experience. Therefore, for the next research, some optimization algorithms can be used to optimize initial parameters, and the optimal parameters can be dynamically selected according to actual training errors. So that the features BDNN extracts can better characterize the input data. The CFDR model proposed in this paper is more suitable for sensitivity analysis of SoS effectiveness or capability when indicators can be measured (specific values can be obtained through tests and other methods), output is comprehensively affected by multidimensional parameters.

ACKNOWLEDGMENT

The authors are grateful to the Academy of Systems Engineering for providing valuable experience of the SAR mission. They also thank the anonymous reviewers for their critical and constructive review of the manuscript.

REFERENCES

[1] Y. Liu, C. M. Eckert, and C. Earl, "A review of fuzzy AHP methods for decision-making with subjective judgements," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113738, doi: 10.1016/j.eswa.2020.113738.
[2] Z. L. Lei, K. B. Qin, M. Xu, H. Shan, and Y. C. Bian, "Efficiency evaluation of radar countermeasure equipment joint netting operation based on AHP-Cloud model," *Shipboard Electron. Countermeasure*, vol. 37, no. 6, pp. 77–82, 2014, doi: 10.16426/j.cnki.jcdzdk.2014.06.020.
[3] E. Fock, "Global sensitivity analysis approach for input selection and system identification purposes—A new framework for feedforward neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1484–1495, Aug. 2014, doi: 10.1109/TNNLS.2013.2294437.
[4] R. Y. Xie, J. J. Zhao, and T. Jiang, "Error analysis of weapon system calibration based on Monte Carlo method," *J. Ordnance Equip. Eng.*, vol. 40, no. 1, pp. 130–134, 158, 2019.
[5] Y. Y. Guo, Y. C. Sun, and L. B. Li, "Failure risk evaluation method of civil aircraft based on Monte Carlo method," *Acta Aeronautica Astronautica Sinica*, vol. 38, no. 10, pp. 155–163, 2017.
[6] S. L. Avila, A. C. Lisboa, L. Krahenbuhl, W. P. Carpes, J. A. Vasconcelos, R. R. Saldanha, and R. H. C. Takahashi, "Sensitivity analysis applied to decision making in multiobjective evolutionary optimization," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1103–1106, Apr. 2006, doi: 10.1109/TMAG.2006.871447.
[7] Y. Li, S. Zheng, and H. Jiang, "Research on reliability assessment method based on bootstrap method," in *Proc. Int. Conf. Quality, Rel., Risk, Maintenance, Saf. Eng.*, Jun. 2012, pp. 871–874, doi: 10.1109/ICQR2MSE.2012.6246365.
[8] X. F. Zhang, Y. Zhao, S. C. Wang, and S. Z. Wang, "Smallsample Weibull reliability analysis based on support vectormachine (SVM)," *Mech. Sci. Technol.*, vol. 31, no. 8, pp. 1359–1362, Aug. 2012, doi: 10.3969/j.issn.1004-132X.2012.16.017.
[9] L. Q. Han, *Artificial Neural Networks Course*. Beijing, China: Beijing Univ. Posts Telecommun. Press, 2006, p. 12.
[10] Y.-L. Chang, T.-H. Tan, W.-H. Lee, L. Chang, Y.-N. Chen, K.-C. Fan, and M. Alkhaleefah, "Consolidated convolutional neural network for hyperspectral image classification," *Remote Sens.*, vol. 14, no. 7, p. 1571, Mar. 2022, doi: 10.3390/rs14071571.
[11] R. Murata, F. Okubo, T. Minematsu, Y. Taniguchi, and A. Shimada, "Recurrent neural network-FitNets: Improving early prediction of student performanceby time-series knowledge distillation," *J. Educ. Comput. Res.*, vol. 2022, Oct. 2022, Art. no. 073563312211297, doi: 10.1177/07356331221129765.
[12] Y. S. Yao and K. J. Xu, "Quality evaluation of flight training based on BP neural network," *Acta Aeronautica Astronautica Sinica*, vol. 38, no. S1, pp. 24–32, 2017, doi: 10.7527/S1000-6893.2017.721513.

[13] M. Zoutendijk and M. Mitici, "Probabilistic flight delay predictions using machine learning and applications to the flight-to-gate assignment problem," *Aerospace*, vol. 8, no. 6, p. 152, May 2021, doi: 10.3390/aerospace8060152.

[14] Y.-J. Lu, L.-L. Chang, K.-W. Yang, Q.-S. Zhao, and Y.-W. Chen, "Study on system of systems capability modeling framework based on complex relationship analyzing," in *Proc. IEEE Int. Syst. Conf.*, Apr. 2010, pp. 23–28, doi: 10.1109/SYSTEMS.2010.5482495.

[15] C. B. Keating, J. J. Padilla, and K. Adams, "System of systems engineering requirements: Challenges and guidelines," *Eng. Manage. J.*, vol. 20, no. 4, pp. 24–31, Apr. 2015, doi: 10.1080/10429247.2008.11431785.

[16] J. B. Kristen, *Systems Engineering Guide for Systems of Systems*. Washington, DC, USA: Office Deputy Under Secretary Defense for Acquisition Technol., 2008, pp. 33–34.

[17] B. Ge, K. W. Hipel, K. Yang, and Y. Chen, "A novel executable modeling approach for system-of-systems architecture," *IEEE Syst. J.*, vol. 8, no. 1, pp. 4–13, Mar. 2014, doi: 10.1109/JSYST.2013.2270573.

[18] Y. J. Wang, W. Fang, C. Lin, and C. X. Xie, "Air-defense decision-making model of single ship CGF based on the theory of fuzzy mathematics and behavior tree," in *Proc. 2nd Int. Conf. Inf. Technol. Comput. Appl. (ITCA)*, Dec. 2021, pp. 755–759, doi: 10.1109/ITCA52113.2020.00162.

[19] D. C. Blair, "Joint tactics, techniques and procedures for combat search and rescue," Joint Pub., America, Tech. Rep. 3-50.21, 2021.

[20] W. Kross, "Doctrine for joint combat search and rescue," Joint Pub., America, Tech. Rep. 3-50.2, 1996.

[21] M. G. T. A. Kinnan, *Combat Search and Rescue*, Air Force Doctrine document 2-1.6, 1998.

[22] X. H. Zhang, X. W. Cao, and S. T. Geng, "Research on intelligence of military auxiliary decision-making system based on deep learning," *J. Ordnance Equip. Eng.*, vol. 39, no. 10, pp. 162–167, 2018, doi: 10.1145/3503047.3503057.

[23] A. H. Al-fatlawi, S. H. Ling, and M. H. Jabardi, "Diagnosis system for parkinson's disease using speech characteristics of patients and deep belief network," *CAAI Trans. Intell. Technol.*, vol. 2, no. 9, pp. 246–253, Dec. 2017.

[24] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2014, doi: 10.1162/neco.2006.18.7.1527.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, p. 436, Nov. 2016, doi: 10.1038/nature14539.

[26] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006, doi: 10.1126/science.1127647.

[27] A. Fischer and C. Igel, "Training restricted Boltzmann machines: An introduction," *Pattern Recognit.*, vol. 47, no. 1, pp. 25–39, Jan. 2014, doi: 10.1016/j.patcog.2013.05.025.

[28] S. Hochreiter and M. C. Mozer, "Monaural speech separation by support vector machines: Bridging the divide between supervised and unsupervised learning methods," in *Blind Speech Separation*. Berlin, Germany: Springer, 2008, p. 18.

[29] I. Sutskever and T. Tieleman, "On the convergence properties of contrastive divergence," *J. Mach. Learn. Res.*, vol. 9, no. 4, pp. 789–795, Mar. 2010.

[30] X. Yang, F. Li, and H. Liu, "A survey of DNN methods for blind image quality assessment," *IEEE Access*, vol. 7, pp. 123788–123806, 2019, doi: 10.1109/ACCESS.2019.2938900.

[31] M. Memarzadeh, B. Matthews, and I. Avrekh, "Unsupervised anomaly detection in flight data using convolutional variational auto-encoder," *Aerospace*, vol. 7, no. 8, p. 115, Aug. 2020, doi: 10.3390/aerospace7080115.

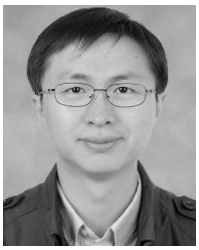
[32] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014, doi: 10.1109/JSTARS.2014.2329330.

[33] Y. Zhu, Q. Li, Y. Feng, W. Han, F. Liu, C. Han, J. Zhou, and F. Si, "Hierarchical economic load dispatch based on chaotic-particle swarm optimization," in *Proc. 9th Int. Conf. Natural Comput. (ICNC)*, Jul. 2013, pp. 517–521, doi: 10.1109/ICNC.2013.6818031.

[34] A. Bakar, L. Ke, H. Liu, Z. Xu, and D. Wen, "Design of low altitude long endurance solar-powered UAV using genetic algorithm," *Aerospace*, vol. 8, no. 8, p. 228, Aug. 2021, doi: 10.3390/aerospace8080228.

[35] J. Karwowski, M. Okulewicz, and J. Legierski, "Application of particle swarm optimization algorithm to neural network training process in the localization of the mobile terminal," in *Proc. EANN*, 2013, pp. 122–131, doi: 10.1007/978-3-642-41013-0_13.

YAN GAO was born in 1990. She received the B.E. degree in material forming and control engineering and the M.E. degree in material processing engineering from the Hefei University of Technology, Hefei, China, in 2013 and 2016, respectively. She is currently pursuing the Ph.D. degree in aircraft design with the School of Aeronautic Science and Engineering, Beihang University. Her research interests include system simulation and operational effectiveness evaluation.



HU LIU received the B.S. and Ph.D. degrees in aircraft design from Beihang University (formerly the Beijing University of Aeronautics and Astronautics), Beijing, China, in 2000 and 2004, respectively. He is currently the Head of the Aircraft Department and a Professor with Beihang University. He is also a main Instructor of Aircraft Conceptual Design, which is a national quality curriculum. His research interests include aircraft conceptual design and training support technologies. He is a member of the American Institute of Aeronautics and Astronautics (AIAA) and the Chinese Society of Aeronautics and Astronautics (CSAA).

FU NIU was born in 1976. He received the B.S. and M.E. degrees in health equipment from the Academy of Military Medical Sciences, Tianjin, in 2000 and 2004, respectively, and the Ph.D. degree in vehicle man–machine dynamics from Shanghai Jiao Tong University, Shanghai, China. His research interest includes search and rescue technology and equipment.

YONGLIANG TIAN received the B.S. degree in aircraft design and engineering and the Ph.D. degree in aircraft design from Beihang University, Beijing, in 2009 and 2015, respectively.

From 2013 to 2014, he was a Visiting Student with the School of Aeronautics and Aerospace Engineering, Purdue University, West Lafayette, IN, USA. From 2016 to 2018, he worked at the Shanghai Aircraft Design and Research Institute, Commercial Aircraft Corporation of China Ltd. Since 2018, he has been a Postdoctoral Researcher of mechanics at Beihang University. He has published more than 15 papers in academic publications. His main research interests include aircraft conceptual design and system engineering, including conceptual software systems, virtual simulation technology, and intelligent design technology. He is a member of the Chinese Society of Aeronautics and Astronautics (CSAA). He received the "FRONTRUNNER 5000—Top Articles in Outstanding *Science and Technology* Journal of China" Award by the China Institute of Science and Technology Information, in 2013.