



## Research Article

# Air Target Threat Assessment: A Kernel Extreme Learning Machine Based on a Multistrategy Improved Sparrow Search Algorithm

Ruiqi Song <sup>1</sup>, Bailin Liu <sup>1</sup>, Suqin Xue,<sup>2</sup> Hong Li,<sup>1</sup> Jingyi Li,<sup>1</sup> and Zehua Zhang<sup>1</sup>

<sup>1</sup>*Xi'an Technological University, School of Computer Science and Engineering, Xi'an 710038, China*

<sup>2</sup>*Yan'an University, Network Information Center, Yan'an 716000, China*

Correspondence should be addressed to Bailin Liu; [xatulbl@xatu.edu.cn](mailto:xatulbl@xatu.edu.cn)

Received 15 October 2022; Revised 10 December 2022; Accepted 24 December 2022; Published 4 January 2023

Academic Editor: Rajesh Kaluri

Copyright © 2023 Ruiqi Song et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Air strikes are among the main means of attack in modern warfare. To improve air defense capabilities and aid military decision-making, threat assessment models have been introduced. As the parameters of the kernel extreme learning machine (KELM) model need to be set individually, this study proposes a parameter learning strategy based on a multistrategy improved sparrow search algorithm (MISSA). First, a reasonable threat assessment model was established based on the capability and situation factors of air targets. Second, the sparrow search algorithm was improved in terms of population position initialization and position update strategy, incorporating tent chaos reverse learning, nonlinear inertia weights, a global search strategy, and adaptive t-distribution. The effectiveness of the MISSA strategy was verified using nine common benchmark functions. The results show that the proposed MISSA finds an effective balance between global and local searches. Moreover, when the MISSA is applied to solve the tuning problem of KELM, the values of mean absolute percentage error, mean square error, root mean square error, and mean absolute error for MISSA-KELM in the air target threat assessment problem are  $2.013 \times 10^{-2}$ ,  $1.282 \times 10^{-4}$ ,  $1.132 \times 10^{-2}$ , and  $8.316 \times 10^{-3}$ , respectively, all of which are higher than that of the other metaheuristic algorithms (e.g., ACWOA-KELM and HGWO-KELM). Therefore, the method proposed in this study can be used as a parameter-tuning tool for KELM, enabling KELM to perform better in practical applications.

## 1. Introduction

Threat assessment is the foundation of air defense systems and is crucial for improving air defense capabilities. Threat assessment measures the threat value of incoming targets to assets based on data acquired using battlefield sensors, thereby providing decision support for the subsequent allocation of firepower. Threat assessment models can be divided into statistical and artificial intelligence models.

Statistical models are programmatic representations of relationships between variables in the form of mathematical equations that enable knowledge to be gained from data. Commonly used statistical models include Bayesian inference, fuzzy sets, rough sets, multiattribute decision-making, and game theory. Previous literature [1] has proposed that interval-

valued intuitionistic fuzzy multiattribute group decision-making results in inconsistent decision-making. Bayesian network-based threat assessment models have been proposed for decision-making related to battlefield uncertainty [2]; however, the prior probability is highly dependent on expert experience. As such, combinations of interval-valued intuitionistic fuzzy sets, game theories, and evidential reasoning methods have been used for dynamic threats [3]; however, their generalization capability may be insufficient. Threat assessment based on statistical methods has high accuracy in specific battlefield environments; however, the targets of information warfare are complicated, and thus it is difficult for statistical methods to analyze the data in real time as they lack self-adaptation and self-learning capabilities. For this reason, researchers have proposed artificial intelligence approaches.

Artificial intelligence-based approaches have recently been widely used in various industries [4]. For example, literature [5] applied data mining techniques for studying weather forecasting and climate change, and literature [6] studied the impact of air pollution on agricultural communities and crop yields. Literature [7] proposed particle swarm optimization and nondominated ranking techniques for breast cancer prediction. Furthermore, Literature [8] used the Co-Active Neuro Fuzzy Expert System (CANFES) classification method model to expose botnets in cloud environments and enhance the security of cloud networks. Preliminary exploration of big data-based threat assessments show that artificial intelligence methods can discover regularity among data. In view of this, previous studies [9] have used support vector machines for threat assessment, employing the PSO algorithm to optimize the SVM parameters. Regrettably, the PSO algorithm suffers from local optimality. Literature [10] has proposed deep learning-based methods for target threat assessment; however, this may lead to problems of gradient disappearance during the process of backpropagation. Target threat assessment based on normalized fully connected residual networks [11] may also be advantageous; however, their networks are more complex, and their calculation speeds are slow compared with other networks.

In view of the above shortcomings of neural networks, Huang et al. [12] proposed a new machine learning method named extreme learning machine (ELM). ELM randomly selects the input layer weights and hidden layer biases during the calculation process, thereby calculating output weights based on Moore–Penrose generalized inverse matrix theory. By virtue of its excellent predictive capabilities, ELM has been widely used in various fields, such as medicine [13], engineering [14], and meteorology [15]. In addition, ELM has been widely used in military applications such as information security [16], ship detection [17], and threat assessment [18]. However, one of the limitations of ELM is that it requires more hidden layer neurons, which can lead to very complex network structures, reducing its accuracy in terms of calculation speed. Thus, Huang et al. proposed the KELM method [19] based on ELM. Compared with ELM, KELM exhibits better results in wind power prediction [20], fault prediction [21], and water quality detection [22]. Studies [23] have shown that KELM performs better than ELM and BP neural networks. It is of concern that, for KELM a with radial basis function kernel, it is required to set kernel penalty parameters  $C$  and  $\gamma$  separately. The setting of the hyperparameters of KELM greatly influences its performance [24]. Therefore, in practical applications, the hyperparameters of KELM should be optimized appropriately. The grid search method is used to optimize the parameters; however, setting the parameter range in this method is difficult, making the model fall into local optimal solutions. Therefore, many scholars have proposed metaheuristic algorithms for solving this problem [25]. Literature [26] proposed the bankruptcy prediction model of the LSEFOA-KELM, and Literature [27] proposed an improved gray wolf algorithm and KELM model, which were used for students' second major selection, thyroid cancer

diagnosis, and financial stress prediction. Literature [28] proposed the MFO and KELM model, which can be effectively used for soil erosion prediction. Literature [29] proposed a hybrid artificial fish particle swarm optimization and KELM model for type II diabetes prediction. The sparrow search algorithm (SSA) is a new type of swarm optimization algorithm [30], mimicking the foraging and antipredation behaviors of a sparrow group.

Due to its excellent optimization capabilities, SSA (sparrow search algorithm) has been widely used in various fields. A previous study [31] used SSA to optimize BP neural networks, which demonstrated that the SSA algorithm has the advantage of strong optimization capability. Combining the PSO and SSA algorithms [32] indicates that the optimization performance of the hybrid algorithm was better than that of each single algorithm. Literature [33] optimized deep trust networks based on SSA and applied it to fault diagnosis, and Literature [34] used a center of gravity reverse learning mechanism to initialize the population, introduced learning coefficients to update the positions of discoverers, and used variation operators to update the positions of joiners. Literature [35] proposed a chaotic SSA based on Bernoulli chaotic mapping, dynamic adaptive weighting, Cauchy variance, and reverse learning, which can better solve the optimal scheduling problem of microgrid clusters, including wind turbines, photovoltaics, and energy storage batteries. A multiobjective SSA (MOSSA) based on 2k congestion distance entropy and location optimization was proposed in Literature [36], and the results showed that MOSSA has unique advantages in solving complex problems. Literature [37] proposed an improved SSA (ISSA), which first uses hybrid search to generate initialized populations, then combines quantum revolving gate and positive cosine algorithms to increase the search capability, and finally uses adaptive adjustment strategies and variable domain search to enhance the diversity of ISSA. Although ISSA has significant advantages in solving scheduling problems, it repeatedly uses greedy ideas (simultaneously finding multiple groups of solutions and selecting the optimal solution) to select the optimal solution, inevitably resulting in relatively high time complexity. According to the no free lunch algorithm [38], one optimization algorithm cannot solve all problems, and the potential of SSA for KELM parameter tuning has not been fully exploited. Therefore, this study proposes a multistrategy improved sparrow search algorithm (MISSA), which is improved in terms of population position initialization and position update strategies. Its utility is based on a tent chaos reverse learning initialization strategy, nonlinear inertia weights, a global optimal guidance strategy, and an adaptive t-distribution.

We first establish a threat assessment model and use a combined assignment method to determine the weights of each influencing factor. Then, the MISSA algorithm is proposed. The optimization of MISSA is verified by comparing its results in the benchmark function with those of other algorithms. Finally, MISSA is compared with seven metaheuristic algorithms for optimizing KELM. Our experimental results show that the MISSA method proposed herein outperforms other algorithms in terms of both convergence speed and prediction accuracy. The main contributions of this study are as follows: (1) A threat

assessment model is scientifically constructed in terms of capability factors and situation factors, considering realistic air defense situations. (2) An effective parameter optimization algorithm termed MISSA is proposed. (3) MISSA effectively solves the parameter selection problem of KELM. (4) The MISSA–KELM model established is used for air target threat assessment.

This remainder of the paper is structured as follows: Section 2 introduces the threat assessment model, Section 3 presents the related algorithms used in this study and their improvements, Section 4 verifies the performance of the proposed algorithm, which is applied to threat assessment, and Section 5 summarizes the conclusions.

## 2. Threat Assessment Model

**2.1. Threat Assessment Metrics.** The establishment of metrics is key to threat assessment. Metrics should not only be selected by considering dynamics but also by considering target attributes. This section divides the metrics used herein into capability factor metrics and situation factor metrics, covering both dynamic and static threats. The threat assessment metric system considered is shown in Figure 1.

### 2.2. Capability Factor Metrics

**2.2.1. Target Type.** Different types of targets have different threat values. The quantitative values of target types are shown in Table 1.

**2.2.2. Jamming Capability.** Air raids are accompanied by electronic jamming during the attack. The quantitative values of the jamming capability are shown in Table 2.

**2.2.3. Maneuverability.** Maneuverability refers to the ability of a flight target to change its flight speed, altitude, and direction in a certain period of time. The quantitative values of maneuverability are shown in Table 3.

**2.3. Situation Factor Metrics.** The situation factor refers to dynamic information regarding air targets and is mainly used to measure the threat value from the relative position, as shown in Figure 2.

**2.3.1. Target Speed.** The affiliation function is shown in the following equation. The threat value reaches its maximum when the flight speed is greater than 2.5 Ma.

$$T(v) = \begin{cases} 0.2, & v < 0.2, \\ \frac{(v - 0.2)}{1.25 \times (2.5 - 0.2)^2} + 0.2, & 0.2 \leq v \leq 2.5, \\ 1, & 1v > 2.5. \end{cases} \quad (1)$$

**2.3.2. Target Height.** The affiliation function is shown in the following equation. The threat value reaches its maximum when the height is lower than 100 m.

$$T(h) = \begin{cases} 1, & h < 100, \\ e^{100-h/5000} & h \geq 100. \end{cases} \quad (2)$$

**2.3.3. Target Heading Angle.** The affiliation function is shown in the following equation. When the target heading angle is large, the threat value increases.

$$T(\theta) = e^{-k(\theta-a)^2} \quad 0 \leq \theta \leq \pi, \quad (3)$$

where  $a = 180$  and  $k = 0.0005$ .

**2.3.4. Target Distance.** The affiliation function is shown in the following equation. The threat value reaches its maximum when the target distance is less than 10 km.

$$T(d) = \begin{cases} 1, & d < 10, \\ e^{10-d/100} & d \geq 10. \end{cases} \quad (4)$$

According to the capability and situation factors of the air target, a linear weighting method, as shown in equation (5), was used to determine the comprehensive threat value.

$$\text{Threat}_i = \omega_1 \cdot T_i(a) + \omega_2 \cdot T_i(m) + \omega_3 \cdot T_i(j) + \omega_4 \cdot T_i(v) + \omega_5 \cdot T_i(h) + \omega_6 \cdot T_i(\theta) + \omega_7 \cdot T_i(d), \quad (5)$$

where  $\text{Threat}_i$  denotes the threat value of the  $i$ th target to the asset and  $\omega_i$  ( $i = 1, 2, 3, 4, 5, 6, 7$ ) denotes the weight of each metric, in which the value is determined by the importance of individual metrics.

**2.4. Determination of the Weight of Each Metric.** The combined subjective–objective assignment method considers expert experience and data objectivity. Therefore, this method combines the analytic hierarchy process (AHP) and the entropy weighting method.

#### Step 1. Objective weights

The entropy weighting method determines weights according to the variability in their metrics. Assuming that there are  $m$  air targets and  $n$  threat assessment metrics, after preprocessing to obtain the normative decision matrix  $R = (r_{ij})$ , the entropy weight of the  $j$ th metric is defined as follows:

$$\omega_j = \frac{1 - H_j}{n - \sum_{j=1}^n H_j}, \quad (6)$$

$$H_j = -k \sum_{i=1}^m p_{ij} \ln p_{ij}. \quad (7)$$

In equations (6) and (7),  $p_{ij} = r_{ij} / \sum_{i=1}^m r_{ij}$ ,  $0 \leq r_{ij} \leq 1$ ,  $k = 1/\ln m$ , and  $k$  denotes the uncertainty of the decision. When  $p_{ij} = 0$ .

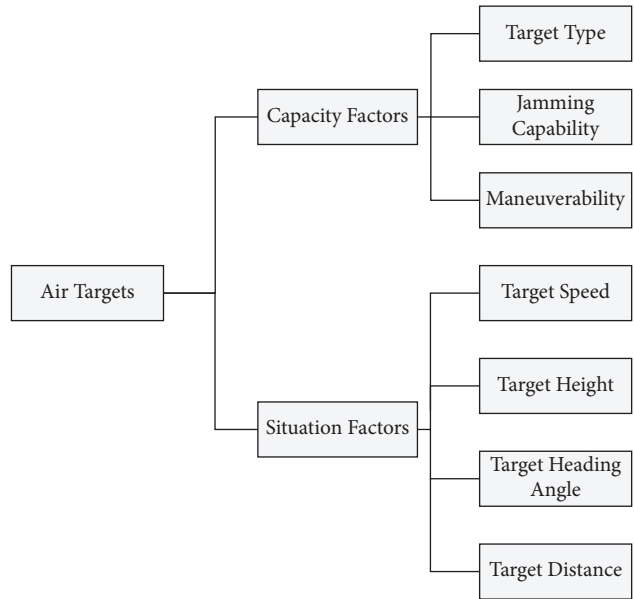


FIGURE 1: Threat assessment metrics.

TABLE 1: Target type quantification.

Target type	Missile	Bomber	Fighter	Micro UAV swarm	Helicopter	Early warning aircraft
$T(a)$	0.9	0.7	0.6	0.5	0.3	0.2

TABLE 2: Jamming capabilities quantification.

Jamming capability	Stronger	Strong	Medium	Weak	None
$T(j)$	0.9	0.7	0.5	0.3	0.1

TABLE 3: Maneuvering capability quantification.

Maneuverability	Stronger	Strong	Medium	Weak	Weaker
$T(m)$	0.9	0.7	0.5	0.3	0.1

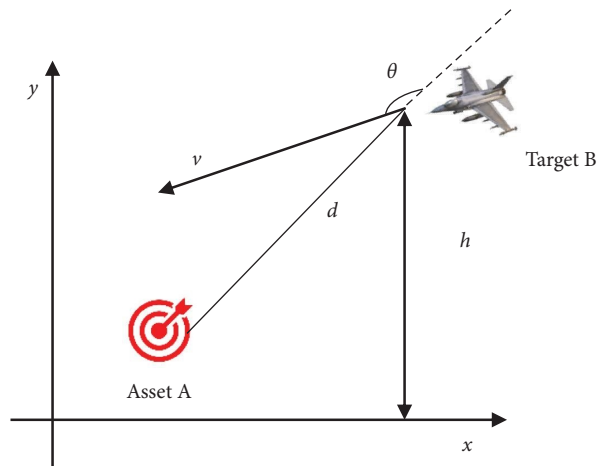


FIGURE 2: Asset and target locations.

The objective weight vector  $\omega_o = (\omega_1, \omega_2, \dots, \omega_7)$  of threat metrics can be obtained using the above method.

Step 2. Subjective weights

We used the more mature AHP to obtain the subjective weights of threat assessment metrics. As this method is widely used [39], it will not be introduced here. The subjective weight vector  $\omega_s = (\omega_1, \omega_2, \dots, \omega_7)$  of the threat assessment is calculated using AHP.

Step 3. Combined empowerment

According to the objective and subjective weights calculated in Steps 1 and 2, a linear weighting method was used to calculate the combined weights, as shown in the following equation:

$$\omega_i = \alpha \cdot \omega_o + \beta \cdot \omega_s, \quad (8)$$

where  $\alpha + \beta = 1$ , in which  $\alpha$  is the percentage of subjective weights and  $\beta$  is the percentage of objective weights.  $\omega_o, \omega_s,$  and  $\omega_i$  denote the subjective, objective, and comprehensive weights, respectively.

### 3. MISSA–KELM Algorithm

3.1. *Kernel Extreme Learning Machine.* KELM is based on ELM with the addition of kernel functions. Kernel functions are used to map input training data into high-dimensional feature spaces, thereby replacing the kernel function operation in the original space with the inner product operation in the transformed high-dimensional space. The kernel matrix  $\Omega_{ELM}$  of KELM is shown in the following equations:

$$\Omega_{ELM} = HH^T, \quad (9)$$

$$\begin{aligned} \Omega_{ELM}(i, j) &= h(x_i) * h(x_j) \\ &= K(x_i, x_j), \end{aligned} \quad (10)$$

where  $x_i$  and  $x_j$  are the input vectors and  $K(x_i, x_j)$  is the kernel function. In this study, the radial basis kernel function with strong localization and good generalization is selected; this function is as follows:

$$K(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right). \quad (11)$$

In equation (11),  $\gamma$  is the kernel parameter.

From equation (11), the output function of KELM can be expressed as follows:

$$\begin{aligned} F(x) &= h(x)\beta \\ &= h(x)H^T \left( HH^T + \frac{I}{C} \right)^{-1} T \\ &= \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix} \left( \Omega_{ELM} + \frac{I}{C} \right)^{-1} T. \end{aligned} \quad (12)$$

3.2. *Sparrow Search Algorithm.* To accomplish foraging, individual sparrows are usually divided into explorers and followers. In the natural state, individuals will monitor each other and followers in a flock will usually compete for the food resources of their high-feeding companions to increase their own predation rate. While foraging, all individuals are alert to their surroundings to prevent the arrival of natural predators.

In the SSA, we assume that the initial size of the sparrow population is  $n$ , dimension of the variable to be optimized is  $d$ , and position of the  $i$ th sparrow in  $d$ -dimensional space can be expressed as shown in the following equation:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix}. \quad (13)$$

Explorers are responsible for searching food in the environment and providing foraging directions and locations to all followers. During each iteration, the position of the explorer is updated as described in the following equation:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot \text{iter}_{\max}}\right), & \text{if } R < ST, \\ X_{i,j}^t + Q \cdot L, & \text{if } R \geq ST, \end{cases} \quad (14)$$

where  $X_{i,j}$  denotes the position information of the  $i$ th sparrow in the  $j$ th dimension,  $t$  denotes the current number of iterations,  $\text{iter}_{\max}$  denotes the maximum number of iterations,  $\alpha \in (0, 1]$  is a random number,  $R (R \in [0, 1])$  denotes the warning value,  $ST (ST \in [0.5, 1])$  is the threshold warning value above which an individual sparrow in the algorithm will move to a safe place to forage,  $Q$  is a random number obeying a normal distribution, and  $L$  denotes a matrix of  $1 \times d$ , in which each matrix element is 1.

With the exception of the explorer, the remaining sparrows are all followers, and their locations are updated based on the following equation:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^t - X_{i,j}^t}{t^2}\right), & i > \frac{n}{2}, \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L, & i \leq \frac{n}{2}, \end{cases} \quad (15)$$

where  $X_p$  is the optimal position currently occupied by the explorer,  $X_{\text{worst}}$  denotes the current global worst position,  $A$  is a matrix of  $1 \times d$ , with each matrix element randomly assigned to 1 or  $-1$ , and  $A^+ = A^T (AA^T)^{-1}$ . When  $i > n/2$ , it denotes that the  $i$ th follower did not obtain food and needs to fly to other places to forage.

When aware of a danger, sparrow populations engage in antipredatory behavior, which is mathematically expressed in the following equation:

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t + \beta \cdot |X_{i,j}^t - X_{\text{best}}^t|, & f_i > f_g, \\ X_{i,j}^t + K \cdot \left( \frac{|X_{i,j}^t - X_{\text{worst}}^t|}{(f_i - f_w) + \varepsilon} \right), & f_i = f_g, \end{cases} \quad (16)$$

where  $X_{\text{best}}$  is the location of optimal fitness in the global solution space,  $\beta$  denotes the step control parameter, a random number obeying a normal distribution with mean zero and variance one,  $K$  is a random number of  $(-1, 1)$ , indicating the direction of movement of individuals,  $f_i$  is the fitness value of the  $i$ th individual under the current iteration,  $f_g$  and  $f_w$  are the global current best and worst fitness values, respectively, and  $\varepsilon$  is the smallest constant to avoid zero in the denominator.

**3.3. Multistrategy Improved Sparrow Search Algorithm.** SSA has many advantages; however, it also has disadvantages such as slow convergence and susceptibility to local optimal solutions. To overcome these disadvantages, this study improves the algorithm in terms of population position initialization and position update strategy.

**3.4. Tent Chaos Reverse Learning Initialization Strategy.** Random initialization of sparrow individuals in the search space may lead to an uneven distribution among individuals, affecting the quality of the optimal global solution. Previous studies [40] have applied a quasirandom population initialization strategy to a genetic algorithm, and although the method was able to improve the quality of the final result, little change in the speed of convergence was observed. In this study, we use the tent chaos reverse learning initialization strategy to initialize the sparrow population, thereby increasing the speed of the algorithm in the global optimal search process. The tent chaos mapping function is as follows:

$$X_{d+1} = \begin{cases} 2X_d, & 0 \leq X_d \leq 0.5, \\ 2(1 - X_d), & 0.5 < X_d < 1, \end{cases} \quad (17)$$

where  $d = 1, 2, \dots, L$  denotes the dimensions of the chaos variables and  $X_d$  is the chaos variable  $X_d \in [0, 1]$ . The population individuals are denoted as follows:

$$X_{i,d} = X_{\min} + (X_{\max} - X_{\min}) \cdot Z_{i,d}, \quad (18)$$

where  $X_{i,d}$  denotes the  $d$ th dimension value of the  $i$ th population individual and  $Z_{i,d}$  is the chaos sequence obtained after  $t$  iterations of tent mapping.  $X_{\max}$  and  $X_{\min}$  define the upper and lower bounds of the search, respectively.

Finally, population  $X_{i,d}$  obtained from tent mapping based on the reverse learning of  $OX_{i,d}$  is calculated and expressed as follows:

$$OX_{i,d} = X_{\min} + X_{\max} - X_{i,d}. \quad (19)$$

Population  $X$  is merged with the reverse population  $OX$  to obtain a new population  $\{X \cup OX\}$ . The objective function values of the new population are calculated and ranked, and the  $N$  individuals with the best fitness values are selected as the initial population.

**3.5. Nonlinear Inertia Weights.** Adaptive inertia weights affect the search ability of the sparrow algorithm. The global search ability of the algorithm is stronger when the inertia weight is larger; conversely, the local search ability of the algorithm is stronger when the inertia weight is smaller. Consequently, a nonlinear inertia weight is proposed in this study, as shown in the following equation:

$$\omega = 1 - \sin\left(\mu \frac{t}{T_{\max}} \pi\right), \quad (20)$$

where  $\mu$  is a constant and is taken as  $\mu = 0.5$ . The value of  $\omega$  decreases nonlinearly as the number of iterations increases. The value of  $\omega$  decreases rapidly with an increasing number of iterations in the early iterations and then decreases slowly for the middle and late iterations.

**3.6. Global Search Strategy.** To improve its search ability, our approach drew on the ‘‘compass’’ of the pigeon-inspired optimization. The global guide term is added in the process of updating the explorer’s position to improve the global search ability of the algorithm. The improved explorer’s position update is shown in the following equation:

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

$$V_{i,j}^t = X_{\text{best}}^t - X_{i,j}^t, \quad (22)$$

where  $R$  is the map compass operator, the random number of  $R \in [0, 1]$ ,  $V_{i,j}^t$  is the velocity of the  $i$ th sparrow in the  $j$ th dimension, as shown in equation (22),  $\omega$  is the nonlinear inertia weight, and  $t$  is the number of iterations.

**3.7. Adaptive T-Distribution.** The adaptive t-distribution mutation operator was used to perturb the position of the sparrow, and iter was used as the degree of freedom parameter of the t-distribution. In the early stages of the algorithm, the value of iter is small and its t-distribution variation approximates a Cauchy distribution variation; in the late stage of the algorithm, the value of iter is large and its t-distribution variation approximates a Gaussian distribution variation. Consequently, the mutation operator based on the t-distribution combines the advantages of both Gaussian and Cauchy distributions, simultaneously improving the global exploration and local exploitation of the

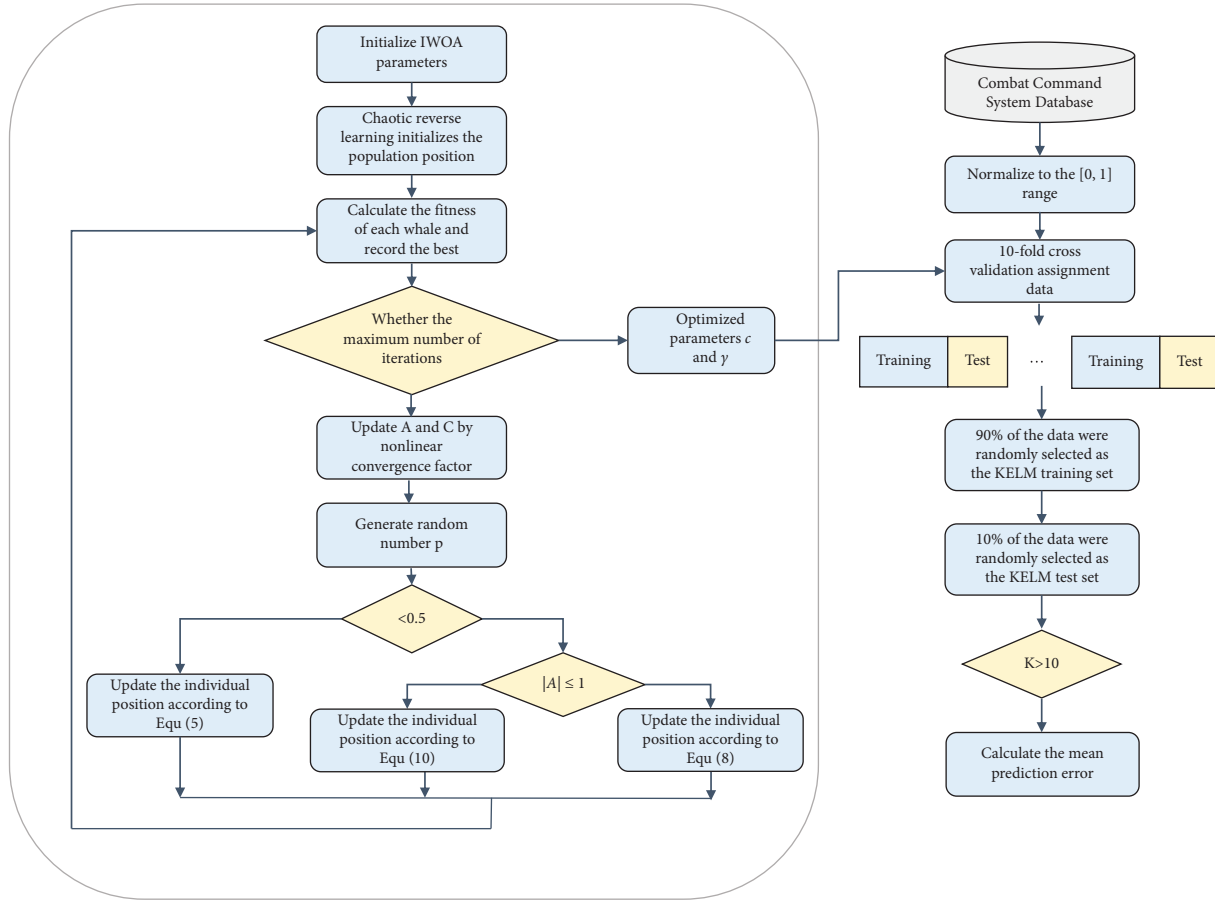


FIGURE 3: Diagram showing the multistrategy improved sparrow algorithm optimization kernel and its limit-learning machine framework.

algorithm. The location update is shown in the following equation:

$$X_i^{t+1} = X_i^t + X_i^t \cdot t(\text{iter}), \quad (23)$$

where  $X_i^{t+1}$  is the position of the sparrow after perturbation and  $X_i^t$  is the position of the  $i$ th sparrow after  $t$  iterations.

The pseudo code of IMSSA is as follows:

#### 4. Air Target Threat Assessment Based on MISSA–KELM

In summary, the key to improving the performance of KELM lies in the choice of parameters  $C$  and  $\gamma$ . In this study, we introduced a relatively new swarm optimization algorithm (SSA) and proposed a multistrategy improved SSA. The resulting KELM with a multistrategy improved SSA is termed MISSA–KELM. To ensure model stability, 10-fold cross-validation was introduced during training. The general framework of MISSA–KELM is shown in Figure 3, and its constituent steps are as follows:

- Step 1. Input the air target dataset
- Step 2. Using a 10-fold cross-validation scheme, the dataset is divided into a training set (90%) and a test set (10%)
- Step 3. Input data normalized to the range of  $[-1, 1]$

- Step 4. Initialize the parameters of MISSA, including the population size and number of iterations
- Step 5. Initialize the sparrow position by tent chaos reverse learning
- Step 6. Update the sparrow position
- Step 7. Conduct adaptive t-distribution to perturb the sparrow’s position and update to the best adaptation
- Step 8. If the termination condition is not reached, go to Step 6; otherwise, output the optimal solution
- Step 9. Use the optimal solution and values for KELM training
- Step 10. Predict the unknown test sample using the optimal solution of MISSA–KELM

#### 5. Experimental Simulation Analysis

To evaluate the effectiveness of the MISSA–KELM, 4,000 data sets were obtained from research institutions in related fields, such as air target threat assessment studies.

5.1. Optimization Performance Test of MISSA. All experiments were conducted using MATLAB 2021a. The experimental environment was an Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, 64.0 GB RAM, and 64-bit OS. In

```

Begin
Setting the parameters of IMSSA, the population size ( $X_i, i = 1, 2, \dots, N$ ), maximum number of iterations ( $T$ ), number of sparrows ( $N$ ), number of explorers ( $PD$ ), and number of warners ( $SD$ );
Initialization of populations using Equations (17)–(19);
Update the best fitness and optimal position up to now;
While
For
If ( $R < ST$ ) then
The location of the sparrow is updated using Equation (21);
Else
The location of the sparrow is updated using Equation (21);
End if
End for
For
If ( $i > n/2$ ) then
The location of the sparrow is updated using Equation (15);
Else
The location of the sparrow is updated using Equation (15);
End for
For
If ( $f_i > f_g$ )
The location of the sparrow is updated using Equation (16);
Else
The location of the sparrow is updated using Equation (16);
End if
End for
Random perturbation of sparrow populations through the adaptive t-distribution of Equation (23);
End while
Return: Best position as of current location
Output: Sparrow location and adaptation
End.

```

ALGORITHM: Pseudo code of IMSSA.

this section, nine more representative benchmark functions were selected from literature [41] as a means to test the performance of the algorithm, including three unimodal functions (F1, F2, and F3), three multimodal functions (F9, F10, and F11), and three fixed-dimensional multimodal functions (F14, F15, and F23). Table 4 shows the benchmark functions used and Table 5 shows the results, with the best fitness value (Best), average fitness value (Mean), and standard deviation (Std) as performance evaluation metrics. To demonstrate the optimization abilities of MISSA, we selected PSO [42], GA [43], SOA [44], WOA [45], GWO [46], and SSA as comparisons, making several attempts to select the optimal parameters of each algorithm. The average value of 20 runs was taken as the test result.

For F1, F2, and F3, the best of MISSA is 0, indicating that MISSA successfully takes the optimal value; this is not the case for the other algorithms. This suggests that MISSA has a better ability to find optimal value in the unimodal function. From F9 and F10, the best of SOA and MISSA are optimal and the mean and std of MISSA are better than SOA. In F15, the best, mean, and std of MISSA are optimal. In F14 and F23, PSO, HGWO, and MISSA find the appropriate value of best; however, MISSA performs better in determining mean and std.

Analyzing the results of the benchmark functions shows that the performance of the MISSA algorithm is better than that of the other algorithms, which is sufficient to show that MISSA has stronger stability and better ability to find optima. In the case of some benchmark functions, MISSA and other algorithms found the optimal fitness value; however, their convergence speed was different; representative convergence curves are shown in Figure 4. Figure 4 also shows that the convergence speed of the MISSA algorithm is significantly faster than that of the other algorithms.

*5.2. Combined Assignment Method to Determine Weights.* To ensure that the results are as accurate as possible, the structural entropy weighting method was used to verify the results of the combined assignment method. The weights obtained by each method are shown in Table 6.

According to the results shown in Table 5, the results obtained by the two methods are approximately equivalent.

*5.3. Performance Analysis of Each Algorithm.* Previous literature [47] indicates that KELM outperforms BP neural networks, SVM, and ELM. Therefore, this experiment focused on comparing the optimal finding ability of swarm



TABLE 4: Benchmark function.

Name	Equation	Minimum
F1 Sphere	$f(x) = \sum_{i=1}^n x_i^2$	0
F2 Schwefel 2.2	$f(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	0
F3 Schwefel 1.2	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	0
F9 Generalized rastrigin	$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	0
F10 Ackley	$f(x) = -20e^{(-0.2\sqrt{1/n} \sum_{i=1}^n x_i^2)} - e^{(1/n \sum_{i=1}^n \cos(2\pi x_i))} + 20 + e$	0
F11 Generalized griewank	$f(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	0
F14 Shekel foxholes	$f(x) = (1/500 + \sum_{j=1}^{25} 1/j + \sum_{i=1}^2 (x_i - a_{ij})^6)^{-1}$	1
F15 Kowalik	$f(x) = \sum_{i=1}^{11} [a_i - x_i (b_j^2 + b_i x_2)]/b_i^2 + b_i x_3 + x_4]^2$	0.0003075
F23 Shekel	$f(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	-10.5364

TABLE 5: Results of the benchmark function.

Fun	Metric	PSO	GA	SOA	WOA	GWO	SSA	MISSA
F1	Best	3.5161E+02	1.5685E+02	4.9426E-114	6.8071E-73	2.4531E-28	1.7822E-37	<b>0</b>
	Mean	4.1103E+03	1.9127E+03	819.3379	7.1327E-02	6.2968E+02	24.3247	<b>0.2768</b>
	Std	6.7765E+03	4.9055E+03	6.5516E+03	4.4152E+03	4.6784E+03	5.4381E+02	<b>5.9147</b>
F2	Best	42.0048	30.4263	7.6915E-84	6.1823E-48	1.688E-15	1.4711E-35	<b>0</b>
	Mean	1.6237E+10	1.0521E+38	5.2308E+38	2.761E+39	5.854E+40	6.2739E+37	<b>5.3169E-02</b>
	Std	3.3904E+11	2.3489E+39	2.3489E+39	5.1349E+40	1.3089E+42	1.4027E+39	<b>1.167</b>
F3	Best	1.2707E+04	9.7146E+04	1.5975E-79	3.7009E+04	3.2357E-07	4.7811E-17	<b>0</b>
	Mean	2.3477E+04	1.7581E+04	9.4932E+03	7.7998E+04	2.7397E+03	17.2380	<b>2.0904</b>
	Std	1.1549E+04	1.0706E+04	3.1057E+04	3.0829E+04	1.3099E+04	2.7386E+02	<b>44.2849</b>
F9	Best	1.0876E+02	16.7023	<b>0</b>	1.1369E-14	1.018	<b>0</b>	<b>0</b>
	Mean	2.1203E+02	50.5318	12.6611	40.9501	25.664	0.5877	<b>0.2627</b>
	Std	52.689	51.3866	58.8522	78.4899	69.1854	9.0273	<b>5.7253</b>
F10	Best	7.4461	3.378	<b>8.8818E-16</b>	6.5725E-15	1.0036E-13	1.5987E-15	<b>8.8818E-16</b>
	Mean	11.7179	6.1081	0.4132	1.2179	0.7846	1.7247E-02	<b>1.3079E-02</b>
	Std	2.7612	3.2157	1.6439	3.3558	3.1562	0.1678	<b>0.2177</b>
F11	Best	2.8874	1.9325	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Mean	38.042	18.724	7.0065	6.7754	5.5319	<b>1.5376E-03</b>	1.8725E-03
	Std	61.5164	49.4481	46.8056	41.5895	41.1837	<b>2.5654E-02</b>	3.1357E-02
F14	Best	<b>0.998</b>	<b>0.998</b>	<b>0.998</b>	3.1503	2.1885	6.8543	<b>0.998</b>
	Mean	1.9794	1.3679	4.9061	4.0743	2.9348	9.2355	<b>1.1273</b>
	Std	11.5069	1.9097	18.0304	5.5155	10.4193	1.2446	<b>0.7625</b>
F15	Best	9.4203E-03	1.4657E-02	1.6744E-03	1.2402E-03	4.3666E-03	4.0585E-04	<b>3.1056E-04</b>
	Mean	1.4657E-02	1.6138E-02	4.5151E-03	1.6766E-03	5.2104E-03	5.5348E-04	<b>5.5143E-04</b>
	Std	1.6507E-02	5.0922E-03	8.7819E-03	<b>3.6449E-03</b>	7.2148E-033	1.5248E-03	<b>3.6055E-03</b>
F23	Best	<b>-10.5233</b>	-6.3073	-1.6278	-6.7398	<b>-10.5344</b>	-10.3564	<b>-10.5364</b>
	Mean	-8.3547	-4.8829	-1.627	-6.4703	-7.599	-9.3705	<b>-10.3585</b>
	Std	2.356	1.5781	1.7038E-02	0.6486	2.1577	1.3155	<b>0.8954</b>

The meanings in bold are best values.

optimization algorithms. To evaluate the convergence speed of the MISSA-KELM algorithm, we recorded the trend of the fitness of relevant models with the relevant number of iterations. We selected the first 3,600 data sets as the training set and the remaining 400 data sets as the test set. The population size and the number of iterations were each set to 20, and mean square error (MSE) was used as the fitness value of the iterations. IPSO [48], IGA [49], ISOA [50], ACWOA [51], AGWO [52], HGWO [53], and HSSA [54] were selected to determine the average results after each algorithm was run 20 times independently, and the main parameters of each algorithm are shown in Table 7. The corresponding change in fitness is shown in Figure 5.

Figure 5 shows that the fitness of each algorithm varies with the number of iterations. The average fitness of IGA-KELM, ISOA-KELM, and HSSA-KELM are relatively close and fall within the local optimum solution. MISSA-KELM reaches its optimum fitness value after eight iterations, its convergence speed is the fastest among all algorithms, and its accuracy is the highest. ACWOA-KELM, IAGWO-KELM, and HGWO-KELM have relatively similar best fitness values; however, the convergence speed of ACWOA-KELM is significantly faster than that of the other two algorithms.

In summary, it can be preliminarily concluded that the MISSA-KELM model can quickly and consistently overcome the local optimum to achieve the best accuracy. To

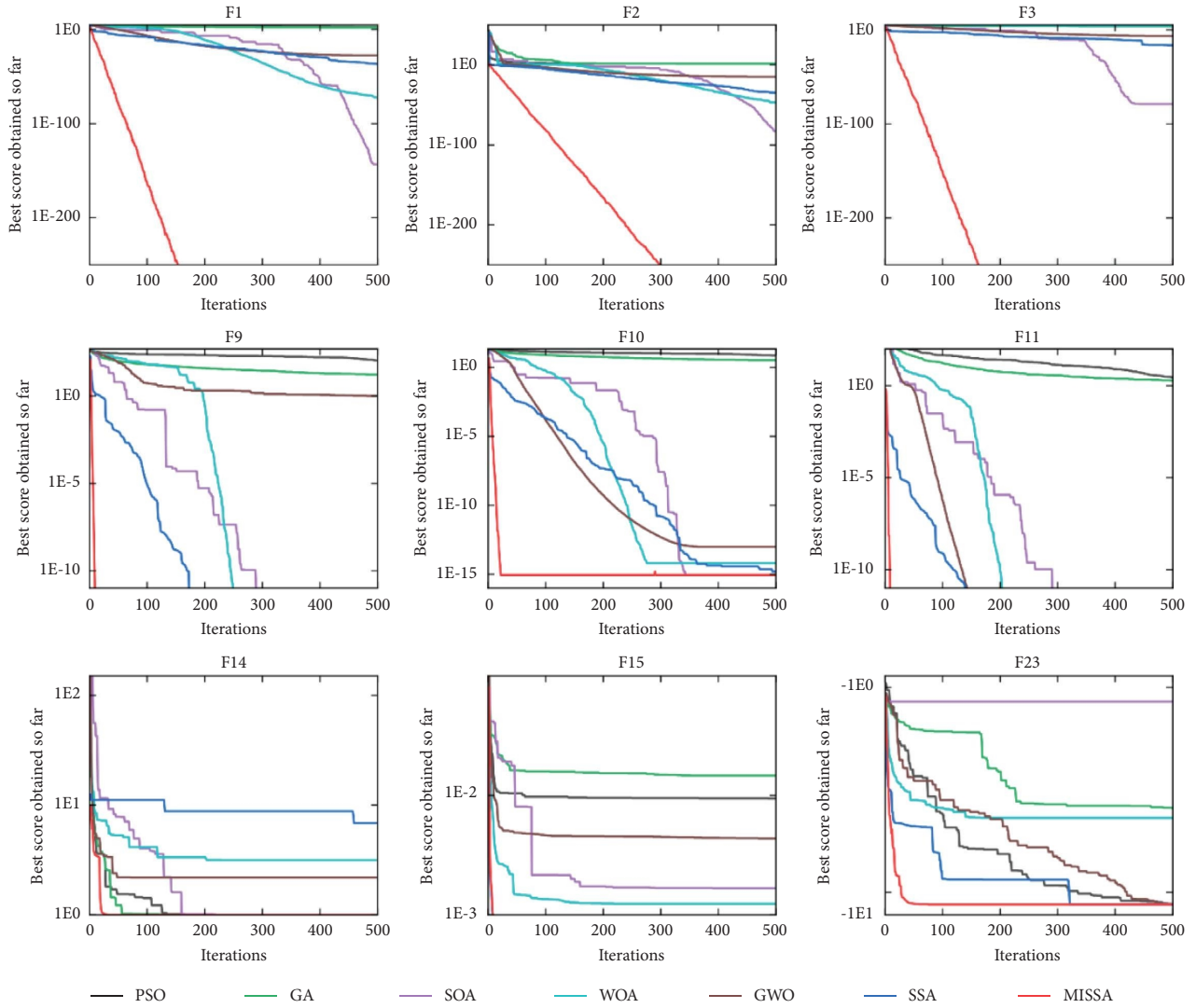


FIGURE 4: Convergence curve of the benchmark function.

TABLE 6: Weight of each metric.

Metrics	Combined empowerment	Structural entropy weighting
Target type $\omega_1$	0.2006	0.1979
Jamming capability $\omega_2$	0.0925	0.0812
Maneuverability $\omega_3$	0.1151	0.1103
Speed $\omega_4$	0.1543	0.1673
Height $\omega_5$	0.1325	0.1311
Heading angle $\omega_6$	0.1231	0.1421
Distance $\omega_7$	0.1792	0.1701

further improve the stability of the results, 10-fold cross-validation was used in the training process and mean absolute percentage error (MAPE), MSE, root mean square error (RMSE), and mean absolute error (MAE) were used as performance evaluation metrics. The performance metrics of each model are listed in Table 7 and compared in Figure 6. Table 8 shows that MISSA-KELM exhibits the best performance with MAPE of 2.013%, MSE of  $1.282 \times 10^{-4}$ , RMSE

of 1.132%, and MAE of  $9.316 \times 10^{-3}$ ; its corresponding area share in Figure 6 is the smallest. The IGA-KELM model performed the worst, having the largest area share in Figure 6.

Our results show that the MISSA-KELM method has powerful local and global search ability, which is mainly due to the incorporation of multistrategy improvement in the algorithm.

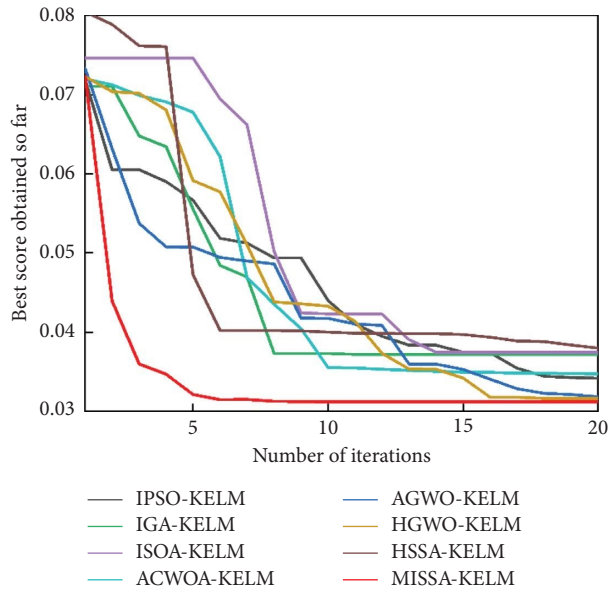


FIGURE 5: Variation curve of adaptation degree according to the number of iterations.

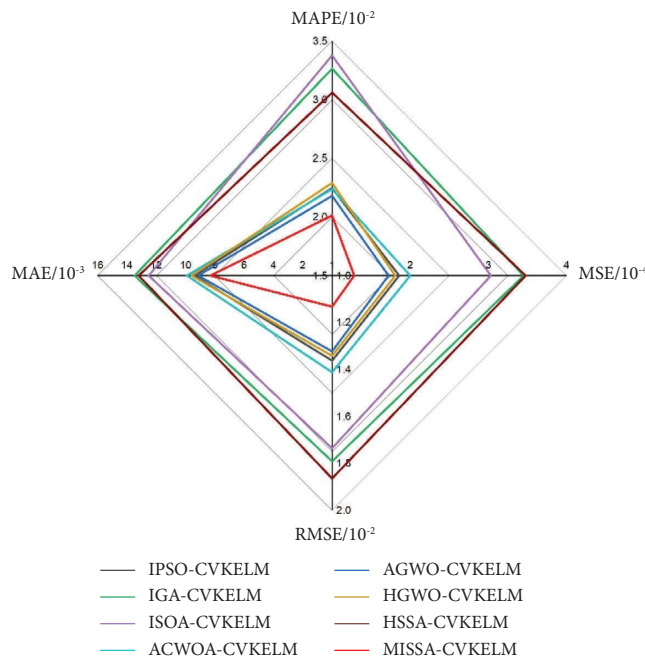


FIGURE 6: Comparison of performance metrics for each model.

TABLE 7: Main parameters of each algorithm.

Algorithm	Main parameter settings
IPSO	$c_1 = 2; c_2 = 2; \omega_{\max} = 1.2; \omega_{\min} = 0.8$
IGA	$p_1 = 0.8; p_2 = 0.1$
ISOA	$f_c = 2; u = 1; v = 1$
ACWOA	$a_1 = [2, 1]; C = [0, 2]; \omega = [0.5, 1]$
AGWO	$a = [2, 0]$
HGWO	$a = [2, 0]; p = 0.5; \text{bata}_{\min} = 0.2; \text{bata}_{\max} = 2$
HSSA	$ST = 0.8; PD = 0.2; SD = 0.1; \omega = 0.5; c_1 = 2; c_2 = 2$
MISSA	$ST = 0.8; PD = 0.2; SD = 0.1$

TABLE 8: Performance metrics of each model.

Algorithm	MAPE/ $10^{-2}$	MSE/ $10^{-4}$	RMSE/ $10^{-2}$	MAE/ $10^{-3}$
IPSO-KELM	2.2468	1.8582	1.3624	9.411
IGA-KELM	3.2639	3.4512	1.7913	13.481
ISOA-KLEM	3.3768	3.0313	1.7354	12.518
ACWOA-KELM	2.538	2.2028	1.4758	10.625
AGWO-KELM	2.1801	1.7211	1.323	9.146
HGWO-KELM	2.2896	1.8073	1.3418	9.594
HSSA-KELM	3.0955	3.477	1.8647	13.2025
MISSA-KELM	2.013	1.282	1.132	8.316

## 6. Conclusion

An air target threat assessment model based on MISSA-KELM was developed, and the following conclusions can be made.

According to realistic air defense situations, the main factors influencing air defense capabilities are reasonably well-described capability and situation factors; these metrics may be scientifically quantified, and a threat assessment framework can be established.

The combined empowerment method considers the subjective experience of experts while ensuring objective authenticity and high credibility.

A MISSA is proposed to improve the convergence speed and global search capability of the algorithm.

The MISSA optimization algorithm can well optimize the kernel penalty parameters,  $C$  and  $\gamma$ , in KELM, thereby effectively improving the prediction accuracy of the model.

Ultimately, our simulation results show that MISSA-KELM can effectively solve threat assessment problems and can form the premise of air defense combat target assignment.

The information in the highly informationized battlefield is hundreds of times more than that in the traditional battlefield, which contains several deceptive information. Therefore, the artificial intelligence-based method is more suitable for aiding decision-making. However, it has certain limitations, such as requiring a large amount of data to train a model with higher accuracy. Currently, there are few available data. Thus, we will consider the method of sand table deduction to simulate battlefield data.

In the future, in the area of threat assessment, we will work on researching and proposing better threat assessment models. The research will focus on understanding how to integrate threat assessment with weapon assignment into the air defense command and operation system for fully automated fire distribution and precision strikes. For improving the algorithm, we will research more metaheuristics and consider combining multiple algorithms to compensate for each other's shortcomings.

## Data Availability

The data involved in this study is confidential in nature and therefore cannot be made public at this time. If necessary, the first author will consider requests for the data in the future.

## Conflicts of Interest

The authors declare that there is no conflicts of interest regarding the publication of this article.

## Acknowledgments

This work was supported by the Natural Science Basic Research Program of Shaanxi Province (grant numbers 2019JM-603).

## References

- [1] Y. Deng, "A threat assessment model under uncertain environment," *Mathematical Problems in Engineering*, vol. 2015, no. 9, 12 pages, Article ID 878024, 2015.
- [2] F. Johansson and G. Falkman, "A Bayesian network approach to threat evaluation with application to an air defense scenario," in *Proceedings of the 2008 11th International Conference on Information Fusion*, pp. 1–7, Cologne, Germany, June 2008.
- [3] R. Zhao, F. Yang, L. Ji, and Y. Bai, "Dynamic air target threat assessment based on interval-valued intuitionistic fuzzy sets, game theory, and evidential reasoning methodology," *Mathematical Problems in Engineering*, vol. 2021, no. 8, pp. 1–13, 2021.
- [4] P. Ray, R. Kaluri, T. Reddy, K. R. M. Praveen, and L. Kuruva, "Contemporary developments and technologies in deep learning-based IoT," *Deep Learning for Internet of Things Infrastructure*, pp. 61–82, CRC Press, Boca Raton, FL, USA, 2021.
- [5] T. R. Gadekallu, B. Kidwai, S. Sharma, P. Rishabh, and K. Sudheer, "Application of data mining techniques in weather forecasting," *Sentiment Analysis and Knowledge Discovery in Contemporary Business*, pp. 162–174, IGI Global, Pennsylvania, USA, 2019.
- [6] S. Pandya, T. R. Gadekallu, P. K. R. Maddikunta, and R. Sharma, "A study of the impacts of air pollution on the agricultural community and yield crops (Indian context)," *Sustainability*, vol. 14, no. 20, Article ID 13098, 2022.
- [7] S. Mohan, S. Bhattacharya, R. Kaluri et al., "Multi-modal prediction of breast cancer using particle swarm optimization with non-dominating sorting," *International Journal of Distributed Sensor Networks*, vol. 16, no. 11, Article ID 155014772097150, 2020.
- [8] N. P. Selvaraj, S. Paulraj, P. Ramadass et al., "Exposure of botnets in cloud environment by expending trust model with CANFES classification approach," *Electronics*, vol. 11, no. 15, p. 2350, 2022.

- [9] S. Y. Jing, "A target threat assessment approach based on improved support vector machine," *Ship Electronic Engineering*, vol. 38, no. 1, pp. 29–32+76, 2018.
- [10] H. M. Chai, Y. Zhang, X. Li, and S. Yanan, "Aerial target threat assessment method based on deep learning," *Journal of System Simulation*, vol. 34, no. 7, pp. 1459–1467, 2021.
- [11] X. Y. Zhai, F. B. Yang, L. N. Ji, S. Q. Li, and S. H. Lv, "Air combat targets threat assessment based on standardized fully connected network and residual network," *Fire Control and Command Control*, vol. 45, no. 6, pp. 39–44, 2020.
- [12] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.
- [13] B. Çil, H. Ayyıldız, and T. Tuncer, "Discrimination of  $\beta$ -thalassemia and iron deficiency anemia through extreme learning machine and regularized extreme learning machine based decision support system," *Medical Hypotheses*, vol. 138, Article ID 109611, 2020.
- [14] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1033–1044, 2014.
- [15] R. C. Deo and M. Şahin, "Application of the extreme learning machine algorithm for the prediction of monthly Effective Drought Index in eastern Australia," *Atmospheric Research*, vol. 153, pp. 512–525, 2015.
- [16] W. Zhang, H. Ren, Q. Jiang, and K. Zhang, "Exploring feature extraction and ELM in malware detection for android devices," *International Symposium on Neural Networks*, pp. 489–498, Springer Cham, Manhattan, NY, USA, 2015.
- [17] W. Zhang, Q. Li, Q. M. J. Wu, Y. Yang, and M. Li, "A novel ship target detection algorithm based on error self-adjustment extreme learning machine and cascade classifier," *Cognitive Computation*, vol. 11, no. 1, pp. 110–124, 2019.
- [18] Y. Cao, Y. X. Kou, A. Xu, and ZF. Xi, "Target threat assessment in air combat based on improved glowworm swarm optimization and ELM neural network," *International Journal of Aerospace Engineering*, vol. 202119 pages, Article ID 4687167, 2021.
- [19] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.
- [20] L. Jun and L. Dachao, "Wind power time series prediction using optimized kernel extreme learning machine method," *Acta Physica Sinica*, vol. 65, no. 13, pp. 39–48, 2016.
- [21] Z. Yin, L. Wang, B. Zhang, L. Meng, and Y. Zhang, "An integrated DC series arc fault detection method for different operating conditions," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 12, Article ID 12720, 2021.
- [22] H. Liu, Y. Zhang, and H. Zhang, "Prediction of effluent quality in papermaking wastewater treatment processes using dynamic kernel-based extreme learning machine," *Process Biochemistry (Oxford, United Kingdom)*, vol. 97, no. 17, pp. 72–79, 2020.
- [23] R. Liu, Y. Wang, H. Zhou, and Z. Qian, "Short-term passenger flow prediction based on wavelet transform and kernel extreme learning machine," *IEEE Access*, vol. 7, Article ID 158025, 2019.
- [24] N. Veronese, C. Luchini, A. Nottegar et al., "Prognostic impact of extra-nodal extension in thyroid cancer: a meta-analysis," *Journal of Surgical Oncology*, vol. 112, no. 8, pp. 828–833, 2015.
- [25] M. Wang, H. Chen, H. Li et al., "Grey wolf optimization evolving kernel extreme learning machine: application to bankruptcy prediction," *Engineering Applications of Artificial Intelligence*, vol. 63, pp. 54–68, 2017.
- [26] Y. Zhang, R. Liu, A. A. Heidari et al., "Towards augmented kernel extreme learning models for bankruptcy prediction: algorithmic behavior and comprehensive analysis," *Neurocomputing*, vol. 430, pp. 185–212, 2021.
- [27] Z. Cai, J. Gu, J. Luo et al., "Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy," *Expert Systems with Applications*, vol. 138, Article ID 112814, 2019.
- [28] C. Chen, X. Wang, C. Wu, M. Mafarja, H. Turabieh, and H. Chen, "Soil erosion prediction based on moth-flame optimizer-evolved kernel extreme learning machine," *Electronics*, vol. 10, no. 17, p. 2115, 2021.
- [29] N. Kanimozhi and G. Singaravel, "Hybrid artificial fish particle swarm optimizer and kernel extreme learning machine for type-II diabetes predictive model," *Medical, & Biological Engineering & Computing*, vol. 59, no. 4, pp. 841–867, 2021.
- [30] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.
- [31] P. Yan, S. Shang, C. Zhang et al., "Research on the processing of coal mine water source data by optimizing BP neural network algorithm with sparrow search algorithm," *IEEE Access*, vol. 9, Article ID 108718, 2021.
- [32] L. Yang, Z. Li, D. Wang, H. Miao, and Z. Wang, "Software defects prediction based on hybrid particle swarm optimization and sparrow search algorithm," *IEEE Access*, vol. 9, Article ID 60865, 2021.
- [33] J. Gai, K. Zhong, X. Du, K. Yan, and J. Shen, "Detection of gear fault severity based on parameter-optimized deep belief network using sparrow search algorithm," *Measurement*, vol. 185, Article ID 110079, 2021.
- [34] J. Yuan, Z. Zhao, Y. Liu et al., "DMPPT control of photovoltaic microgrid based on improved sparrow search algorithm," *IEEE Access*, vol. 9, Article ID 16623, 2021.
- [35] P. Wang, Y. Zhang, and H. Yang, "Research on economic optimization of microgrid cluster based on chaos sparrow search algorithm," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 5556780, 18 pages, 2021.
- [36] B. Li and H. Wang, "Multi-objective sparrow search algorithm: a novel algorithm for solving complex multi-objective optimisation problems," *Expert Systems with Applications*, vol. 210, Article ID 118414, 2022.
- [37] F. Luan, R. Li, S. Q. Liu, B. Tang, S. Li, and M. Masoud, "An improved sparrow search algorithm for solving the energy-saving flexible job shop scheduling problem," *Machines*, vol. 10, no. 10, p. 847, 2022.
- [38] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [39] M. Tyagi, P. Kumar, and D. Kumar, "A hybrid approach using AHP-TOPSIS for analyzing e-SCM performance," *Procedia Engineering*, vol. 97, pp. 2195–2203, 2014.
- [40] H. Maaranen, K. Miettinen, and M. M. Mäkelä, "Using quasi random sequences in genetic algorithms," *Optimization and Inverse Problems in Electromagnetism*, pp. 33–44, Springer, Dordrecht, Netherlands, 2003.
- [41] Y. Xin, L. Yong, and L. Guangming, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

- [42] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33–57, 2007.
- [43] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.
- [44] G. Dhiman and V. Kumar, "Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems," *Knowledge-Based Systems*, vol. 165, pp. 169–196, 2019.
- [45] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [46] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [47] N. Li, F. He, W. Ma, R. Wang, and X. Zhang, "Wind power prediction of kernel extreme learning machine based on differential evolution algorithm and cross validation algorithm," *IEEE Access*, vol. 8, Article ID 68874, 2020.
- [48] Y. Jiang, T. Hu, C. C. Huang, and X. Wu, "An improved particle swarm optimization algorithm," *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 231–239, 2007.
- [49] H. Liang, J. Zou, K. Zuo, and M. J. Khan, "An improved genetic algorithm optimization fuzzy controller applied to the wellhead back pressure control system," *Mechanical Systems and Signal Processing*, vol. 142, Article ID 106708, 2020.
- [50] X. Chen, Y. Li, Y. Zhang, X. Ye, X. Xiong, and F. Zhang, "A novel hybrid model based on an improved seagull optimization algorithm for short-term wind speed forecasting," *Processes*, vol. 9, no. 2, p. 387, 2021.
- [51] M. A. Elhousseini, A. Y. Haikal, M. Badawy, and N. Khashan, "Biped robot stability based on an A-C parametric whale optimization algorithm," *Journal of Computational Science*, vol. 31, pp. 17–32, 2019.
- [52] W. Zhang, S. Zhang, F. Wu, and Y. Wang, "Path planning of UAV based on improved adaptive grey wolf optimization algorithm," *IEEE Access*, vol. 9, Article ID 89400, 2021.
- [53] A. Hoballah, D. E. A. Mansour, and I. B. M. Taha, "Hybrid grey wolf optimizer for transformer fault diagnosis using dissolved gases considering uncertainty in measurements," *IEEE Access*, vol. 8, Article ID 139176, 2020.
- [54] Y. Fan, Y. Zhang, B. Guo, X. Luo, Q. Peng, and Z. Jin, "A hybrid sparrow search algorithm of the hyperparameter optimization in deep learning," *Mathematics*, vol. 10, no. 16, p. 3019, 2022.