

## Research Article

# Adaptive Differential Evolution Algorithm with Simulated Annealing for Security of IoT Ecosystems

Qianqian Liu <sup>1</sup>, Xiaoyan Zhang <sup>1</sup>, Qiaozhi Hua <sup>2</sup>, Zheng Wen <sup>3</sup> and Haipeng Li <sup>4</sup>

<sup>1</sup>College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China

<sup>2</sup>Computer School, Hubei University of Arts and Science, Xiangyang 441000, China

<sup>3</sup>School of Fundamental Science and Engineering, Waseda University, Tokyo 169-8050, Japan

<sup>4</sup>Capinfo Company Ltd, Beijing 100010, China

Correspondence should be addressed to Xiaoyan Zhang; zhangxy@xust.edu.cn

Received 12 January 2022; Revised 11 February 2022; Accepted 28 February 2022; Published 12 April 2022

Academic Editor: Yan Huo

Copyright © 2022 Qianqian Liu 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.

With the wide application of the Internet of Things (IoT) in real world, the impact of the security on its development is becoming incrementally important. Recently, many advanced technologies, such as artificial intelligence (AI), computational intelligence (CI), and deep learning method, have been applied in different security applications. In intrusion detection system (IDS) of IoT, this paper developed an adaptive differential evolution based on simulated annealing algorithm (ASADE) to deal with the feature selection problems. The mutation, crossover, and selection processes of the self-adaptive DE algorithm are modified to avoid trapping in the local optimal solution. In the mutation process, the mutation factor is changed based on the hyperbolic tangent function curve. A linear function with generation is incorporated into the crossover operation to control the crossover factor. In the selection process, this paper adopts the Metropolis criterion of the SA algorithm to accept poor solution as optimal solution. To test the performance of the proposed algorithm, numerical experiments were performed on 29 benchmark functions from the CEC2017 and six typical benchmark functions. The experimental results indicate that the proposed algorithm is superior to the other four algorithms.

## 1. Introduction

With the popularity of the Internet of Things (IoT) [1, 2], the applications of the IoT in various industries have gradually increased, e.g., autonomous vehicles [3, 4], the medical-care IoT [5], the satellite-based IoT [6], and the industrial Internet of Things [7]. The widespread deployments and advancements in IoT have simultaneously increased threats of security, such as online spamming [8], advanced persistent threats [9], and some malicious activities. IoT also adopts many protection measures for security, including intrusion detection system (IDS) [10], antiforensic technology [11], encryption [12–14], privacy-preserving [15] technology, etc. To further ensure the security of the IoT, various technologies such as artificial intelligence (AI), blockchain technology [16–22], security and detection mechanisms [23], and key management schemes [24] are used to enhance and optimize these protection measures.

The Artificial Intelligence Internet of Things (AIoT) makes the intercommunication of various networks and systems more efficient [25–29]. Deep learning has also made many contributions to the realization of AIoT [30–32] and other field [33, 34]. Many scholars have adopted different computational intelligences, including fuzzy system, neural networks [35–37], swarm intelligence [38], differential evolution algorithm [39], and other evolutionary computation [40], to resolve differential optimization problems. The optimization and improvement of differential evolution (DE) algorithm have become a trend in the application of IoT. Xue et al. [41] adopted a self-adaptive differential evolution algorithm (SaDE) to deal with feature selection problems. In the nonuniform IoT node deployments, to solving nonlinear real-parameter problems, Ghorpade et al. [42] proposed an enhanced particle swarm optimization algorithm and adopted differential crossover quantum in this algorithm. In heterogeneous resource allocation, to minimize service cost

and service time, Fang et al. [43] proposed a dynamic multiobjective evolutionary algorithm to allocating IoT services. Iwendi et al. [44] proposed a metaheuristic optimization approach for energy efficiency in the IoT networks. Yang et al. [45] proposed an intelligent trust cloud management method for secure and reliable communication in Internet of Medical Things (IoMT) systems. Qureshi et al. [46] proposed enhanced differential evolution (EDE) and adaptive EDE algorithms to effectively improve the topology robustness of the IoT network while keeping the node degree distribution unchanged.

In 1995, Storn and Price first proposed the DE algorithm on the basis of the genetic algorithm to solve global optimization problems over continuous space [47]. As a metaheuristic algorithm, the DE algorithm utilizes the individuals in the population to present the solutions of problem and updates the individuals through the mutation operation, crossover operation, and selection operation. Due to its easy implementation, high convergence speed, and superior robustness, the DE algorithm has been widely used in many fields, including in solving dynamic optimization problems [48], constraint optimization problems [49], multiobjective optimization problems [50], and engineering design problems in practical applications [51]; it can also be used as scheduling algorithm in CPS system [52].

To achieve better improvement on the performance of the DE algorithm, many scholars have developed different optimized DE algorithms, which adopt adaptive mutation strategy and crossover strategy to optimize the mutation and crossover process. Mohamed and Suganthan [53] proposed an enhanced DE algorithm and introduced a new triangular mutation operator and two adaptive schemes to change the values of the mutation factor and crossover factor. Mohamed and Mohamed [54] proposed a new DE algorithm, namely, AGDE, to prepare two candidate pools of crossover factor and adaptively update the parameter value. Mohamed et al. [49] proposed an enhanced DE algorithm (EDDE) to solve constrained engineering optimization problems. EDDE uses individual information with different fitness function values in the population to generate a mutation vector. Wu et al. [55] realized an ensemble of three DE variants (EDEV). In each generation of the ensemble, the optimal variant is obtained by competition among three variants, and the final evolution is carried out by the optimal variant. Elquliti and Mohamed [56] proposed the nonlinear integer goal programming problem with binary and real variables and developed an improved real-binary differential evolution (IRBDE) algorithm for solving constrained optimization problems. In the developed algorithm, a new binary mutation strategy is introduced to deal with binary variables. Fu et al. [57] proposed an adaptive DE algorithm with aging leader and challenger mechanism to solve optimization problem. Sun et al. [58] proposed a hybrid adaptive DE algorithm, namely, HADE, which develops a mutation process with a disturbance factor and adjusts the crossover factor according to the fitness function values. Huynh et al. [59] added a Q-learning model to generate the values of the mutation factor and the crossover factor in the DE algorithm. Zeng et al. [60] combined the

DE algorithm with the SA algorithm to generate a new individual with a Markov chain length of  $L$  time in the mutation operation.

In this study, we proposed an adaptive simulated annealing differential evolution (ASADE) algorithm based on the SA algorithm and DE algorithm. In the ASADE algorithm, the mutation factor is modified with reference to the hyperbolic tangent function curve, the crossover factor is changed to linear variation of generation, and we combine the selection operation and the Metropolis criterion of the SA algorithm. In the early evolution stages of the proposed algorithm, the mutation factor and crossover factor maintain a relatively large value, and the ability of the algorithm to get rid of a local optimal solution is enhanced. In the middle evolution stages, the values of the mutation factor and crossover factor are decreasing and the algorithm speeds up the convergence rate and obtains the trade-off between global and local abilities. In the later stage, the mutation factor and crossover factor maintain a relatively small value; the search continues until the optimal solution is found. ASADE was tested on the 2017 IEEE Evolutionary Computing Conference (IEEE CEC2017) [61] and six typical benchmark functions. The experiments and comparisons show that ASADE is superior to two typical population-based algorithms and two DE optimized algorithms.

This paper is arranged as follows: the DE algorithm and SA algorithm are introduced in Section 2. Section 3 proposes the ASADE algorithm. The experimental testing results are discussed in Section 4. Finally, the conclusions of this paper are presented in Section 5.

## 2. Related Algorithms

*2.1. DE Algorithm.* The DE algorithm is a direct search algorithm based on biological ideas to solve global optimization problems. It utilizes evolutionary process such as the mutation and crossover operation to obtain a new individual as a new solution to the optimization function. The DE algorithm focuses on the diversity of solutions and the effectiveness of convergence. Compared with other optimization algorithms, the DE algorithm has fewer control parameters, faster convergence speed, stronger robustness in optimization results, and wider application in various fields.

The DE algorithm includes four processes: initialization, mutation, crossover, and selection. In the initialization process, the initial parameters include the population size (NP), mutation factor ( $F$ ), crossover factor (CR), maximum evolutionary generations ( $G_m$ ), number of variables ( $D$ ), and range of variables ( $[x_{\min}, x_{\max}]$ ) in individuals. The fitness function for specific problem ( $f(X)$ ) and the initial population ( $X^0$ ) as the target population in the first generation are also obtained. In the mutation and crossover process, each individual in the target population is mutated and crossed to generate trial individual and the fitness function value of each individual can be obtained. In the selection process, by comparing the fitness function values, the better individuals between the target population and the trial population are selected to form the target population of the next

generation. Algorithm 1 presents the algorithmic process of the DE algorithm.

**2.2. SA Algorithm.** The SA algorithm is a stochastic intelligent optimization algorithm based on the Monte Carlo method to solve an optimized problem; the name of the SA algorithm comes from the annealing and cooling process in metallurgy [62]. The algorithm treats a feasible solution of the optimized problem as a particle in the solid. The particle will reach the final ground state in the process of cooling and annealing, and the internal energy will be reduced to the minimum value, which is similar to the process of finding the optimal value of the problem. In the process of particle cooling, at high temperature, a new state that differs significantly from the current temperature is more acceptable to be an important state, while at a low temperature, a new state with a smaller temperature difference from the current state is more inclined to accept as an important state. And as the temperature tends toward a constant, no new state can be accepted. The above criterion for accepting a new state is called the Metropolis criterion. According to the Metropolis criterion, the probability that a particle will go to equilibrium at temperature  $T$  is  $e^{-(\Delta E/(kT))}$ , where  $E$  represents the internal energy,  $\Delta E$  is its changed energy, and  $k$  is the Boltzmann constant. A new solution is accepted or rejected according to this probability while finding solution of the problem. The SA process can be described in the following steps.

*Step 1.* Set the initial parameters, objective function  $f$ , initial temperature  $T$ , cooling function  $T_k$ , and Markov chain length  $L$ . In a single evolution, the number of iterations to generate new solutions is set by the Markov chain length.

*Step 2.* Choose an initial viable solution  $X$  randomly, which can be regarded as a particle in a solid. At present, the optimal solution of the objective function is  $f(X)$ .

*Step 3.* Perform the process of generating a new solution with a Markov chain length of  $L$  times, which is called the Markov process. The method of generating new solutions is as follows:

(1) Part 1

$$\begin{aligned} X(x_1, x_2, \dots, x_k, \dots, x_l, \dots, x_n), \\ X_{\text{new}}(x_1, x_2, \dots, x_l, \dots, x_k, \dots, x_n) \end{aligned} \quad (1)$$

(2) Part 2

Assuming that the resulting new solution is  $P(p_1, p_2, \dots, p_k, \dots, p_l, \dots, p_n)$ , for each of the unknown variables,  $p_i (i = 1, 2, \dots, n)$  can be expressed as

$$p_i = \begin{cases} |x_i - 1|, & w > v, \\ x_i, & \text{otherwise,} \end{cases} \quad (2)$$

where  $v \in [0, 1]$  and  $w$  is a random number between 0 and 1.

*Step 4.* After generating a new solution, decide whether or not to accept the new solution according to the Metropolis criterion. The criterion equation is as follows:

$$\begin{aligned} p &= \begin{cases} 1, & f(X_{\text{new}}) < f(X), \\ e^{-(\Delta E/T_k)}, & \text{otherwise,} \end{cases} \\ T_k &= \begin{cases} -\frac{f(X)}{\log 0.2}, & k = 1, \\ T_{k-1} \cdot \mu, & k > 1, \end{cases} \end{aligned} \quad (3)$$

where  $\Delta E = f(X_{\text{new}}) - f(X)$  and  $\mu$  is the cooling coefficient. If  $p > \text{rand}(0, 1)$ , accept the new solution, otherwise, reject the new solution. Decrease the Markov chain length by 1, and repeat the process of producing new solutions until the Markov chain length equals to 0. Choose the solution corresponding to the greatest  $p$  value as the optimal solution of the iteration.

*Step 5.* Obtain the current optimal solution from Step 4, execute the cooling function  $T_k$ , and determine whether the temperature remains unchanged. If so, output the current optimal solution. If not, go to Step 3.

### 3. The ASADE Algorithm

The DE algorithm is a stochastic direct search evolutionary algorithm. In the process of evolution, the mutation operation and crossover operation greatly impact the diversity of the solutions. In the selection operation, different optimal solutions can affect the optimization process of the next evolution. Based on the literature [63], the ASADE algorithm is proposed.

**3.1. Adaptive Mutation.** Previous research [64] found that the mutation factor is closely related to the search step size. In the early stages of evolution, in the scope of the global feasible solution, a large mutation factor can search the solutions widely, the structure of the solution will be more directional and diversified, and it will be easy to get rid of the local optimal solution. In the middle and late stages of evolution, when the global optimal solution range has been found, a small mutation can help to accurately search better solutions, and the performance of the DE algorithm is more effective. According to the analysis of the range of the mutation factor, a hyperbolic tangent function between  $[-4, 4]$  was adopted in this paper to adjust the value of the mutation factor, and its equation is as follows:

$$F = \frac{F_{\max} + F_{\text{mix}}}{2} + \frac{\tanh(-4 + 8 \cdot ((G_m - G)/G_m))(F_{\max} - F_{\min})}{2}, \quad (4)$$

```

Input: the initialization parameter:  $NP, F, CR, G_m, D, [x_{min}, x_{max}]$  and  $f(X)$ 
output: the optimal solution of the problem
population initialization
generation=0
for  $i=1 \rightarrow NP$  do
  assign initial values to each variable in each individual  $X^0$ 
  for  $j=1 \rightarrow D$  do
     $X_{i,j}^0 = x_{min} + rand(0,1) \cdot (x_{max} - x_{min})$ 
while generation <  $G_m$  do
  k=generation
  mutate the target population  $X^k$ 
  for  $i=1 \rightarrow NP$  do
     $V_i^k = X_i^k + F(X_m^k - X_n^k)$ 
  cross the mutant population  $V^k$ 
  for  $i=1 \rightarrow NP$  do
     $k_i = rand(1,2,\dots, D)$ 
    for  $j=1 \rightarrow D$  do
      k=rand(0,1)
      if  $k < CR$  or  $j == k_i$ 
        then  $U_{i,j}^k = V_{i,j}^k$ 
      else
         $U_{i,j}^k = X_{i,j}^k$ 
    Select from the trial population  $U^k$  and the target population  $X^k$ 
    if  $f(X_i^k) < f(U_i^k)$ 
      then  $X_i^{k+1} = X_i^k$ 
    else  $X_i^{k+1} = U_i^k$ 
    generation = generation+1
  obtain  $X_{best}^{G_m}$  and  $f(X_{best}^{G_m})$ 

```

ALGORITHM 1: The algorithmic description of DE.

where  $G_m$  is the maximum evolution generation,  $G$  is the current generation,  $F_{max}$  and  $F_{min}$  are the maximum and minimum ranges of the mutation factor, respectively. Taking the maximum evolution generation as 1000, the variation trend of the mutation factor in this paper is compared with that in the study by Sun et al. [58], and the graph is plotted in Figure 1.

In Figure 1, in the mutation process of the proposed algorithm, the mutation factor is maintained at approximately 0.75 in the first 350 generations. Through a period of global searching, the algorithm can get rid of the local optimal solution continually and find the range of the global optimal solution. In the other two cases, the rate of decreasing the mutation factors at the early stages of evolution is fast, which shortens the algorithm's global search time and makes the global search less effective. From 350 to 700 generations, the mutation factor of the proposed algorithm decreases from 0.75 to 0.25, and at the same time, the search scope also reduces from global to local. After 700 generations, the mutation factor remains at 0.25. After performing a local search for a sufficiently long period of time, the better optimal solution is found in the local scope.

**3.2. Adaptive Crossover.** In the DE algorithm, the mutation factor has a great impact on the global search, while the crossover factor can increase the diversity of solutions and has the ability to affect the search range, but the influence

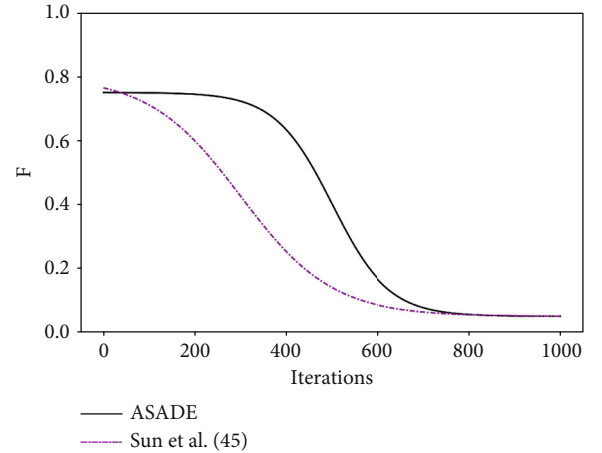


FIGURE 1: Change tendency curve of the mutation factor.

of the crossover factor is slightly smaller. To improve the efficiency of the solution, we introduced a linear function with the number of generations in equation (5) to express the crossover factor. The value range of the crossover factor  $CR$  is  $CR_{min} < CR < CR_{max}$ .

$$CR = CR_{max} - \frac{G(CR_{max} - CR_{min})}{G_m}. \quad (5)$$

In the early stage of evolution, the value of the crossover factor is relatively large, and the diversity of feasible solutions is relatively rich, while in the later stage of evolution, the crossover factor gradually decreases. According to the crossover process in the DE algorithm, the probability that the value randomly generated between [0, 1] is smaller than the crossover factor decreases, the probability that the crossover vector selects the parent vector increases, and the diversity of the population decreases.

**3.3. SA Selection.** In the selection process of the DE algorithm, the fitness function values of individuals are compared between the trial population and the target population, and individuals with small fitness function values are selected to form the target population in next generation. After the selection process, the individual with the smallest fitness function value is the optimal value obtained by this evolution. In the process of selection, it is easy to ignore a poor solution in the trial population. To consider the impact of a poor solution could produce more diverse mutation vectors, affecting the value of optimal solution. In this paper, the Metropolis criterion of the SA algorithm is introduced in selection process to accept a poor solution as the individual in the next population. The selection of the poor solution can make the algorithm get rid of a local optimal solution in the evolutionary process and mutate in a wider direction in the next evolutionary process. The selection operation with Metropolis criterion in the proposed algorithm is defined as follows:

If  $p_i > \text{rand}(0, 1)$ ,

$$X_i^{k+1} = U_i^k. \quad (6)$$

Else

$$X_i^{k+1} = X_i^k \quad (7)$$

End where the equation of  $p_i$  is as follows:

$$p_i = \begin{cases} 1 & f(X_i^k) < f(U_i^k), \\ e^{\left(-\frac{\Delta E}{T_k}\right)}, & \text{otherwise,} \end{cases} \quad (8)$$

where  $E = |f(X_i^k) - f(U_i^k)|$  and  $T_k$  is the temperature in the  $k$ th generation; the initial temperature and cooling function are set as follows:

$$T_k = \begin{cases} -\frac{f(X_{\text{best}})}{\log 0.2}, & k = 1, \\ T_{k-1} \cdot \mu, & k > 1, \end{cases} \quad (9)$$

where  $k$  is the number of generation and  $\mu$  is the cooling coefficient. The temperature decreases with evolution increment until the temperature remains constant, and each drop is related only to the value of the previous temperature.

## 4. Numerical Experiment and Result Analysis

The experiment in this paper used a 64-bit Windows 10 operating system. The processor is an Intel(R) Core (TM) i5-5200U CPU @ 2.20 GHz with an Intel(R) HD Graphics 5500 GPU. Python 3.5.2 is selected as the experimental code language, and the experiment is run in PyCharm software to complete the experimental process.

**4.1. Experiment Setup.** The performance of the ASADE algorithm in this paper was tested on the IEEE Congress on Evolutionary Computation17 test suite (CEC2017). The CEC2017 test includes 29 benchmark functions, a detailed introduction, and description of CEC2017, and its specific functions can be found in [61]. The test dimensions ( $D$ ) of functions are  $D = 10$ ,  $D = 30$ ,  $D = 50$ , and  $D = 100$ , respectively, and the solution error  $fi = f(x) - f(x^*)$  is regarded as the objective function and the fitness function, with  $f(x)$  as the calculated optimal solution and  $f(x^*)$  as the known optimal solution. The maximum number of fitness evaluations was  $D * 10000$ . The objective function values of each benchmark function are calculated over 51 runs.

The statistics results including the best, mean, and standard deviation (Std) for all dimensions are presented in Tables 1 and 2.

To compare the results of different algorithms on test functions, this paper used the Friedman test and Wilcoxon signed-rank test to analyze and compare the solution quality. The two tests use  $\alpha = 0.05$  as the significance level. The Friedman test generates the final ranks of different algorithms on test functions' the Wilcoxon signed-rank test compares the specific differences between two algorithms for test functions of CEC2017. Comparing the solution solved by the former algorithm and the comparison algorithm, "R+" is the sum of ranks for the functions in the first algorithm solutions that are more than the second algorithm solutions in the row, "R-" is the sum of ranks for the opposite situation, a plus (+) sign indicates the function number of CEC2017 in which the first algorithm solutions are more than the second algorithm solutions, a minus (-) sign indicates the function number of CEC2017 in the opposite situation, and the approximation ( $\approx$ ) presents the number of the remaining functions.  $p$  values less than the significance level are marked in italic. SPSS 26.0 was used as an experimental tool for the statistical tests.

**4.2. ASADE Parametric Study.** The ASADE algorithm optimized the mutation, crossover, and selection processes in the DE algorithm. To analyze the impact of the adaptive mutation, adaptive crossover, and SA selection on the performance of the ASADE algorithm, experiments were conducted. Three different versions of ASADE were tested and compared against the proposed version on 29 functions of CEC2017 on  $D = 10$ ,  $D = 30$ ,  $D = 50$ , and  $D = 100$ .

- (1) *Version 1.* To test the individual effect of adaptive mutation on the performance of the ASADE algorithm, an ASADE version with adaptive crossover, SA selection, and a basic mutation strategy was

TABLE 1: The results of the ASADE algorithm for  $D = 10$  and  $D = 30$  on CEC2013.

Function	$D = 10$				$D = 30$			
	Best	Worst	Mean	Std	Best	Worst	Mean	Std
f1	6.68E-05	2.61E+00	2.21E-01	4.67E-01	4.02E-01	1.91E+01	3.21E+01	4.64E+01
f2	1.13E-10	3.17E-02	1.86E-03	6.00E-03	1.59E+00	1.17E+02	5.58E+01	2.25E+00
f3	1.48E+00	3.42E+00	2.48E+00	4.98E-01	8.57E+01	1.18E+02	8.88E+01	5.50E+00
f4	9.95E-01	1.19E+01	4.99E+00	2.22E+00	1.80E+01	7.47E+01	3.89E+01	1.20E+01
f5	0.00E+00	4.09E-10	1.39E-11	5.70E-11	1.14E-13	1.10E-06	2.93E-08	1.53E-07
f6	1.10E+01	2.01E+01	1.36E+01	1.91E+00	2.24E+01	1.32E+02	6.39E+01	2.20E+01
f7	1.00E+00	9.01E+00	4.15E+00	1.99E+00	2.01E+01	7.06E+01	4.09E+01	1.30E+01
f8	0.00E+00	9.23E-08	3.13E-09	1.35E-08	0.00E+00	1.35E-06	3.02E-08	1.86E-07
f9	1.02E+02	1.07E+03	4.46E+02	1.28E+02	1.61E+03	5.16E+03	3.28E+03	8.81E+02
f10	2.07E-01	4.24E+00	2.09E+00	9.82E-01	5.67E+00	9.44E+01	4.39E+01	3.29E+01
f11	1.99E+01	2.86E+02	1.20E+02	6.74E+01	2.45E+04	7.75E+05	1.62E+05	1.65E+05
f12	1.15E+00	8.32E+00	5.93E+00	1.29E+00	4.52E+01	2.92E+02	1.95E+02	5.47E+01
f13	0.00E+00	1.99E+00	3.31E-01	5.07E-01	3.14E+01	7.60E+01	5.66E+01	1.09E+01
f14	5.80E-02	1.53E+00	4.29E-01	3.91E-01	1.11E+01	4.76E+01	2.90E+01	8.78E+00
f15	2.42E-02	3.81E+01	2.89E+00	6.33E+00	2.77E+01	1.43E+03	6.28E+02	2.86E+02
f16	3.67E-06	2.05E+01	1.73E+00	4.24E+00	2.77E+01	5.56E+02	2.38E+02	1.40E+02
f17	2.29E-03	8.63E-01	1.66E-01	1.76E-01	9.92E+01	6.16E+02	2.19E+02	8.37E+01
f18	1.68E-06	2.54E-01	1.89E-02	3.40E-02	1.20E+01	2.98E+01	2.23E+01	3.23E+00
f19	0.00E+00	3.12E-01	1.20E-02	6.00E-02	2.18E+01	6.73E+02	2.37E+02	1.44E+02
f20	1.00E+02	2.17E+02	1.60E+02	5.37E+01	2.14E+02	2.67E+02	2.41E+02	1.34E+01
f21	8.28E-11	1.01E+02	8.89E+01	3.15E+01	1.00E+02	5.32E+03	3.09E+03	1.32E+03
f22	3.04E+02	3.16E+02	3.09E+02	2.66E+00	3.67E+02	4.15E+02	3.90E+02	1.15E+01
f23	1.00E+02	3.49E+02	3.36E+02	3.32E+01	4.40E+02	5.49E+02	4.71E+02	2.22E+01
f24	3.98E+02	3.98E+02	3.98E+02	1.07E-01	3.83E+02	3.87E+02	3.87E+02	4.66E-01
f25	3.00E+02	3.00E+02	3.00E+02	2.07E-07	1.18E+03	1.70E+03	1.45E+03	1.21E+02
f26	3.87E+02	3.91E+02	3.89E+02	5.80E-01	4.62E+02	5.37E+02	4.95E+02	1.28E+01
f27	3.00E+02	5.84E+02	3.13E+02	4.58E+01	4.03E+02	1.83E+03	4.63E+02	2.24E+02
f28	2.31E+02	2.50E+02	2.38E+02	5.04E+00	3.74E+02	8.81E+02	5.39E+02	1.10E+02
f29	4.22E+02	8.83E+05	1.54E+05	3.14E+05	2.83E+03	9.25E+03	5.84E+03	1.49E+03

experimentally investigated. This version was called ASADE-1

- (2) *Version 2.* To test the individual effect of adaptive crossover on the performance of the ASADE algorithm, an ASADE version with adaptive mutation, SA selection, and a basic crossover strategy was experimentally investigated. This version was called ASADE-2
- (3) *Version 3.* To test the individual effect of SA selection on the performance of the ASADE algorithm, an ASADE version with adaptive mutation, adaptive crossover, and basic selection process was experimentally investigated. This version was called ASADE-3

The statistical test results of the ASADE algorithm against its alternate versions (ASADE-1, ASADE-2, and

ASADE3) on CEC2017 are presented in Tables 3 and 4. Table 3 shows the average ranks of four ASADE versions calculated by the Friedman test. In the table, the  $p$  values obtained by the Friedman test for each dimension are 0.003, 0.000, 0.000, and 0.000, which are all less than 0.05. It can be drawn that the performance of these ASADE versions has a significant difference. Compared with ASADE-1, ASADE-2, and ASADE-3, the ranks of 2.02, 1.24, 1.17, and 1.10 obtained by the ASADE algorithm for all dimensions are all the smallest, and the mean rank value 1.38 is also the smallest, which proves that ASADE is better than the other three algorithms in all dimensions. In addition, ASADE-3 ranks second, followed by ASADE-2 and ASADE-1. This proves that the adaptive modified mutation factor in the mutation process plays a key role in the ASADE algorithm. The ASADE algorithm integrates three optimization strategies to obtain the best optimization effect.

TABLE 2: The results of the ASADE algorithm for  $D = 50$  and  $D = 100$  on CEC2013.

Function	$D = 50$				$D = 100$			
	Best	Worst	Mean	Std	Best	Worst	Mean	Std
f1	1.26E+02	3.03E+03	9.53E+02	1.02E+02	9.95E+03	1.11E+03	1.05E+03	4.15E+02
f2	2.62E+01	5.43E+02	4.02E+02	8.57E+01	3.94E+02	4.96E+02	4.35E+02	4.39E+01
f3	4.84E+01	2.08E+02	1.54E+02	5.81E+01	7.81E+02	8.49E+02	8.22E+02	2.44E+01
f4	8.46E+01	1.17E+02	1.03E+02	1.26E+01	7.19E-02	1.51E-01	9.22E-02	2.70E-02
f5	5.68E-13	2.20E-01	4.43E-02	7.92E-02	9.47E+02	9.75E+02	9.62E+02	1.02E+01
f6	1.13E+02	1.81E+02	1.49E+02	2.48E+01	7.94E+02	8.64E+02	8.28E+02	2.66E+01
f7	5.17E+01	1.36E+02	8.67E+01	2.66E+01	4.40E+01	8.31E+02	4.15E+02	2.67E+02
f8	1.34E-08	4.64E+02	8.68E+01	1.70E+02	3.12E+04	3.17E+04	3.14E+04	2.12E+02
f9	3.37E+03	8.03E+03	5.61E+03	1.79E+03	8.64E+04	1.60E+05	1.22E+05	2.54E+04
f10	4.58E+01	6.04E+01	5.35E+01	4.38E+00	4.12E+04	5.45E+05	1.73E+05	1.83E+04
f11	1.86E+05	4.45E+06	2.27E+06	1.53E+06	1.74E+06	2.31E+07	1.10E+07	8.12E+06
f12	2.95E+03	1.92E+04	1.01E+04	6.02E+03	4.49E+06	1.53E+07	9.21E+06	3.41E+06
f13	1.23E+02	1.67E+02	1.51E+02	1.43E+01	6.84E+03	1.12E+05	3.62E+04	3.46E+04
f14	5.68E+01	2.04E+02	1.24E+02	5.71E+01	7.85E+03	8.91E+03	8.39E+03	3.85E+02
f15	4.96E+02	2.12E+03	1.24E+03	5.96E+02	4.97E+03	5.45E+03	5.22E+03	1.53E+02
f16	5.92E+02	9.40E+02	7.71E+02	1.08E+02	2.63E+07	5.46E+07	3.94E+07	1.14E+07
f17	2.05E+04	6.49E+04	3.70E+04	1.53E+04	6.20E+03	1.18E+06	2.18E+05	4.29E+05
f18	6.45E+01	1.37E+02	8.33E+01	2.54E+01	5.49E+03	6.06E+03	5.80E+03	1.79E+02
f19	4.73E+02	1.64E+03	1.00E+03	4.05E+02	9.58E+02	1.08E+03	1.02E+03	4.26E+01
f20	2.76E+02	3.18E+02	2.99E+02	1.69E+01	3.04E+04	3.28E+04	3.18E+04	8.73E+02
f21	4.98E+03	8.62E+03	7.37E+03	1.25E+03	6.45E+02	1.10E+03	8.61E+02	1.72E+02
f22	4.94E+02	5.67E+02	5.24E+02	2.36E+01	1.47E+03	1.72E+03	1.63E+03	8.55E+01
f23	5.39E+02	7.02E+02	6.11E+02	5.45E+01	1.03E+03	1.37E+03	1.22E+03	1.15E+02
f24	4.81E+02	5.80E+02	5.36E+02	3.51E+01	9.64E+03	1.17E+04	1.08E+04	7.72E+02
f25	1.86E+03	2.57E+03	2.20E+03	2.82E+02	7.47E+02	9.07E+02	8.31E+02	5.70E+01
f26	5.28E+02	8.99E+02	6.51E+02	1.34E+02	1.17E+04	1.44E+04	1.31E+04	9.65E+02
f27	4.59E+02	5.14E+02	4.83E+02	2.22E+01	3.87E+03	6.19E+03	5.29E+03	7.38E+02
f28	4.29E+02	9.16E+02	7.37E+02	1.71E+02	1.41E+04	1.01E+05	5.78E+04	2.85E+04
f29	6.26E+05	8.53E+05	7.00E+05	7.39E+04	9.95E+05	1.11E+06	1.05E+06	4.15E+04

Table 4 presents the comparison results between ASADE and the other three versions on different dimensions according to the Wilcoxon test. The best results are distinguished in italic. In the table, the  $p$  value of the comparison between ASADE and ASADE-1 is less than 0.05 in each dimension, “R+” is less than “R-” and the function number of “-” is more than the function number of “+.” ASADE is significantly better than ASADE-1 in performance. Additionally, the  $p$  values of the comparisons of ASADE with ASADE-3 and ASADE-2 on  $D = 10$  are 0.545 and 0.940, respectively, both of which are greater than 0.05. However, as the dimension increases, the  $p$  value tends toward 0.000. This indicates that the optimization of the adaptive crossover process and the SA selection process has an increasing influence on the performance of the ASADE algorithm.

The final comparison results are recorded in the last column of the table; a plus (+) sign indicates that the former algorithm is superior to the compared algorithm. According to the last column, ASADE is improved to the compared algorithm in 83% of the rows.

4.3. *Comparison against State-of-the-Art Algorithms.* The proposed ASADE algorithm was compared with four evolutionary algorithms, i.e., PSO, DE, HADE [58], and adaptive DE with disturbance factor algorithm (ADE-D) [65] on six typical benchmark functions; the functions are presented in Table 5. These benchmark functions have many local optimal values, a large search space, and strong deception. The function values are all positive, and the global optimal values of these functions are all zero; then, the fitness

TABLE 3: Average ranks calculated by the Friedman test for ASADE, ASADE-1, ASADE-2, and ASADE-3 across all problems and all dimensions using CEC2017.

Algorithm	$D = 10$	$D = 30$	$D = 50$	$D = 100$	Mean rank	Rank
ASADE	2.02	1.24	1.17	1.10	1.38	1
ASADE-1	3.21	3.83	3.79	3.79	3.66	4
ASADE-2	2.52	2.76	2.72	2.76	2.69	3
ASADE-3	2.26	2.17	2.31	2.34	2.27	2
Friedman $p$ value	<i>0.000</i>	<i>0.000</i>	<i>0.000</i>			

TABLE 4: Results of the Wilcoxon signed-rank test between ASADE and the three compared versions across all problems and all dimensions using CEC2017.

$D$	ASADE vs.	R+	R-	$p$ value	+	$\approx$	-	Dec.
10	ASADE-1	72.00	334.00	<i>0.003</i>	5	1	23	+
	ASADE-2	221.00	214.50	0.940	13	0	16	$\approx$
	ASADE-3	140.00	185.00	0.545	9	4	16	$\approx$
30	ASADE-1	0.00	435.00	<i>0.000</i>	0	0	29	+
	ASADE-2	2.00	433.00	<i>0.000</i>	1	0	28	+
	ASADE-3	68.00	367.00	<i>0.001</i>	6	0	23	+
50	ASADE-1	2.00	433.00	<i>0.000</i>	1	0	28	+
	ASADE-2	28.00	407.00	<i>0.000</i>	2	0	27	+
	ASADE-3	29.00	406.00	<i>0.000</i>	2	0	27	+
100	ASADE-1	2.00	433.00	<i>0.000</i>	1	0	28	+
	ASADE-2	3.00	432.00	<i>0.000</i>	1	0	28	+
	ASADE-3	5.00	430.00	<i>0.000</i>	1	0	28	+

TABLE 5: Six typical benchmark functions.

Function name	Expression	Variable range
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	$x_i \in [-5.12, 5.12]$
Rastrigin	$f_2(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$x_i \in [-5.12, 5.12]$
Salomon	$f_3(x) = -\cos(2\pi \sqrt{\sum_{i=1}^D x_i^2}) + 0.1 \sqrt{\sum_{i=1}^D x_i^2} + 1$	$x_i \in [-100, 100]$
Griewank	$f_4(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-600, 600]$
Ackley	$f_5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	$x_i \in [-32, 32]$
Levy	$f_6 = (\sin \pi w_1)^2 + \sum_{i=1}^{D-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1)] + (w_D - 1)^2 [1 + \sin^2(2\pi w_D)]$ , $w_i = 1 + \frac{x_i - 1}{4}$	$x_i \in [-10, 10]$

function is defined as the function itself. The closer the solution is to zero, the closer it is to the global optimal value. The results including the best (BST), worst (WST), and average (AVG) values and the number of evolution generations (NEG) that reach the specified convergence precision for each function are recorded in

Table 6, the values smaller than  $10^{-8}$  are taken as zero, and the smallest values are marked in italic. The parameter setting of DE, PSO, HADE, and ADE-D can be found in original paper, and the parameter values of ASADE are  $NP = 100$ ,  $G_m = D * 1000$ ,  $F_{\max} = 0.8$ ,  $F_{\min} = 0.05$ ,  $CR_{\max} = 1$ , and  $CR_{\min} = 0.9$ .



TABLE 6: Comparative results of benchmark functions for  $D = 30$ .

Function	Value type	DE	PSO	HADE	ADE-D	ASADE
Sphere	BST	$7.71E-04$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	WST	$2.23E-03$	$1.34E-08$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	AVG	$1.35E-03$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	NEG	589	379	615	165	134
Rastrigin	BST	$7.77E+01$	$7.06E+01$	$6.92E-04$	$2.81E-06$	$0.00E+00$
	WST	$1.20E+02$	$1.57E+02$	$3.99E+00$	$9.95E-01$	$8.96E-01$
	AVG	$9.82E+01$	$8.78E+01$	$1.37E+00$	$1.99E-02$	$0.69E-02$
	NEG	980	787	969	661	576
Salomon	BST	$1.01E+00$	$1.10E+00$	$2.00E-01$	$2.00E-01$	$2.00E-01$
	WST	$1.33E+00$	$1.70E+00$	$3.00E-01$	$3.00E-01$	$2.00E-01$
	AVG	$1.17E+00$	$1.18E+00$	$2.60E-01$	$2.22E-01$	$2.00E-01$
	NEG	997	680	885	937	285
Griewank	BST	$5.74E-01$	$3.20E-02$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	WST	$8.69E-01$	$3.20E-02$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	AVG	$7.26E-01$	$3.20E-02$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	NEG	962	598	838	302	236
Ackley	BST	$1.53E+01$	$0.00E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	WST	$1.87E+01$	$1.34E+00$	$4.91E-07$	$0.00E+00$	$0.00E+00$
	AVG	$1.77E+01$	$8.08E-01$	$1.49E-07$	$0.00E+00$	$0.00E+00$
	NEG	978	745	975	272	191
Levy	BST	$1.85E-02$	$2.64E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	WST	$6.30E-02$	$1.62E+01$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	AVG	$4.11E-02$	$7.38E+00$	$0.00E+00$	$0.00E+00$	$0.00E+00$
	NEG	882	908	758	221	121

TABLE 7: Average ranks calculated by the Friedman test for ASADE, DE, PSO, HADE, and ADE-D on  $D = 10$ ,  $D = 30$ , and  $D = 50$ .

Algorithm	$D = 10$	$D = 30$	$D = 50$	Mean rank	Rank
ASADE	2.00	1.92	1.75	1.89	1
DE	4.67	3.67	3.08	3.81	4
PSO	3.83	5.00	5.00	4.61	5
HADE	2.33	2.50	3.00	2.61	3
ADE-D	2.17	1.92	2.17	2.09	2
Friedman $p$ value	$0.001$	$0.001$	$0.001$		

In Table 6, the solutions for all benchmark functions of the ASADE algorithm are the smallest and only ASADE obtains the optimal solutions on the Rastrigin and Salomon function. The NEG of the ASADE algorithm on six functions are 134, 576, 236, 191, 121, and 285; among them, the probability of finding the global optimal solution before 500 generations is 83.33%, while the probability of DE, PSO, HADE, and ADE-D is 0%, 16.67%, 0%, and 66.67%. Table 7 shows the outcome of the Friedman test. The average ranks of ASADE on 10, 30, and 50 dimensions are 2.00, 1.92, and 1.75, respectively. The mean rank of ASADE is 1.89, which is the smallest among the five algorithms. The second and third best algorithms are ADE-D and HADE, with the mean rank as 2.09 and 2.61.

When  $D = 30$ , the convergent tendency curves of ASADE and other four compared algorithms for six benchmark functions are depicted in Figure 2. In Figure 2, the rate of searching the best objection function value of ASADE is faster than other algorithms; compared with ADE-D, ASADE firstly finds the optimal value in the solutions of all functions. On Rastrigin and Ackley functions, DE, PSO, and HADE easily fall into local optimal solutions, while ASADE continually presents monotonic downward trend until it find the smallest solution. Therefore, we can conclude that compared with DE, PSO, HADE, and ADE-D, ASADE has faster convergence speed and more accurate solution in the process of solving these functions.

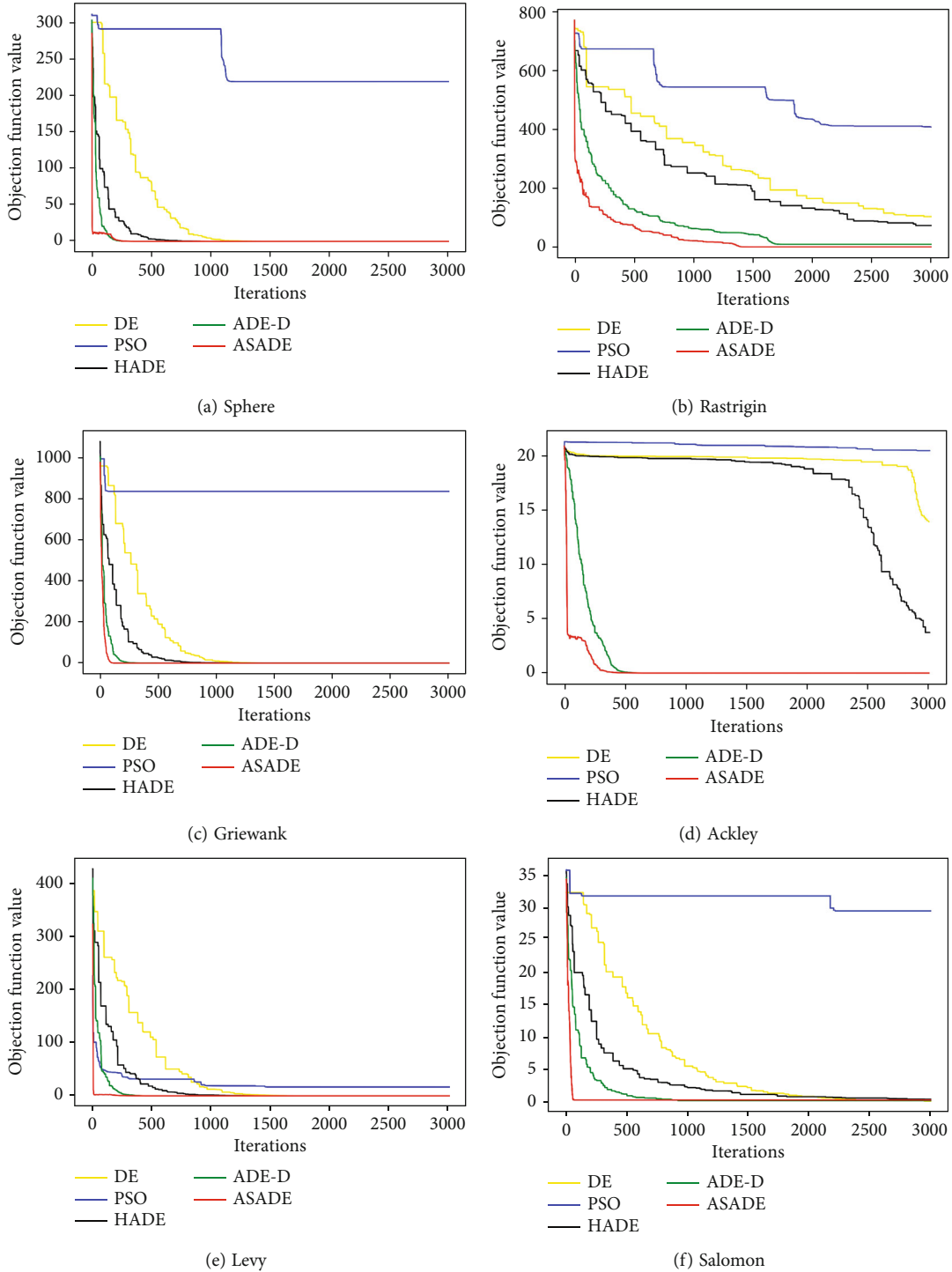


FIGURE 2: The convergent tendency curve of differential algorithms on typical benchmark functions for  $D = 30$ .

### 5. Conclusion

In the development of Internet of Things (IoT), intrusion detection systems (IDS) play a vital role in data security. The IDS dataset has dimensional problems of irrelevant and redundant, and feature selection is employed to reduce dimensions. An adaptive simulated annealing differential evolution algorithm (ASADE) is proposed to generate

multiple candidate solutions to find the global optima in the feature selection process. The ASADE algorithm optimized the basic DE algorithm in three aspects. First, in the process of mutation, the hyperbolic tangent function is used as a variable-factor change trend function to balance global exploration and local exploitation abilities in the evolution process. Second, we adapt a linearly varied crossover factor in the crossover operation; with the increase in evolutionary

time, the crossover ability gradually changes from strong to weak. Finally, in selection process, the Metropolis criterion of the SA algorithm is used to accept a poor solution as optimal solution, which gives the DE algorithm an enhanced ability to enrich population diversity and get rid of the local optimum. To test the performance of the ASADE algorithm, we analyze the effectiveness of three ASADE versions on CEC2017 test and compare it with four improved evolution algorithms on six typical benchmark functions. The experimental results demonstrate that the performance of the ASADE algorithm is improved compared with other algorithms. In the future, the population reduction strategy, success-history slots, and a uniform distribution or a Cauchy distribution for the parameters can be employed to the ASADE algorithm to improve the performance of the algorithm. In addition, more and more security problems of IoT Ecosystem could be transformed into nonlinear real-parameter optimization problems and the ASADE algorithm could be applied to resolve them with high accuracy and fast convergence.

### Data Availability

The data used to support the findings of this study are included within the article.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This work was supported in part by the Fundamental Research on Safe and High-Efficiency Mining at Large Inclined Longwall Faces (No. 51634007), the Hubei Natural Science Foundation under grant 2021CFB156, and the Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research (KAKENHI) under grant JP21K17737.

### References

- [1] N. M. Balamurugan, S. Mohan, M. Adimoolam, A. John, and W. Wang, "DOA tracking for seamless connectivity in beamformed IoT-based drones," *Computer Standards & Interfaces*, vol. 79, article 103564, 2022.
- [2] H. Li, Q. Zheng, W. Yan, R. Tao, X. Qi, and Z. Wen, "Image super-resolution reconstruction for secure data transmission in Internet of Things environment," *Mathematical Biosciences and Engineering*, vol. 18, no. 5, pp. 6652–6671, 2021.
- [3] L. Tan, K. Yu, L. Lin, G. Srivastava, J. C. Lin, and W. Wei, "Speech emotion recognition enhanced traffic efficiency solution for autonomous vehicles in a 5G-enabled space-air-ground integrated intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2830–2842, 2022.
- [4] F. Ding, K. Yu, Z. Gu, X. Li, and Y. Shi, "Perceptual enhancement for autonomous vehicles: restoring visually degraded images for context prediction via adversarial training," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 4, pp. 1–12, 2021.
- [5] D. Wang, Y. He, K. Yu, G. Srivastava, L. Nie, and R. Zhang, "Delay sensitive secure NOMA transmission for hierarchical HAP-LAP medical-care IoT networks," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.
- [6] L. Zhen, Y. Zhang, K. Yu, N. Kumar, A. Barnawi, and Y. Xie, "Early collision detection for massive random access in satellite-based internet of things," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 5, pp. 5184–5189, 2021.
- [7] Y. Gong, L. Zhang, R. Liu, K. Yu, and G. Srivastava, "Nonlinear MIMO for industrial internet of things in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5533–5541, 2021.
- [8] Z. Guo, K. Yu, A. Jolfaei, F. Ding, and N. Zhang, "Fuz-Spam: label smoothing-based fuzzy detection of spammers in Internet of Things," *IEEE Transactions on Fuzzy Systems*, p. 1, 2021.
- [9] K. Yu, L. Tan, S. Mumtaz et al., "Securing critical infrastructures: deep-learning-based threat detection in IIoT," *IEEE Communications Magazine*, vol. 59, no. 10, pp. 76–82, 2021.
- [10] A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. AI-Qaness, and S. Lu, "IoT intrusion detection system using deep learning and enhanced transient search optimization," *IEEE Access*, vol. 9, pp. 123448–123464, 2021.
- [11] F. Ding, G. Zhu, Y. Li, X. Zhang, P. K. Atrey, and S. Lyu, "Anti-forensics for face swapping videos via adversarial training," *IEEE Transactions on Multimedia*, 2021.
- [12] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the Internet of Health Things," *IEEE Journal of Biomedical and Health Informatics*, p. 1, 2021.
- [13] C. Feng, K. Yu, M. Aloquail, M. Alazab, Z. Lv, and S. Mumtaz, "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13784–13795, 2020.
- [14] D. Xu, K. Yu, and J. A. Ritcey, "Cross-layer device authentication with quantum encryption for 5G enabled IIoT in Industry 4.0," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.
- [15] Y. Sun, J. Liu, K. Yu, M. Alazab, and K. Lin, "PMRSS:privacy-preserving medical record searching scheme for intelligent diagnosis in IoT healthcare," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1981–1990, 2022.
- [16] C. Feng, B. Liu, Z. Guo, K. Yu, Z. Qin, and K. K. R. Choo, "Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones," *IEEE Internet of Things Journal*, p. 1, 2021.
- [17] W. Wang, H. Xu, M. Alazab, T. R. Gadelallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE transactions on industrial informatics*, p. 1, 2021.
- [18] L. Tan, K. Yu, N. Shi, C. Yang, W. Wei, and H. Lu, "Towards secure and privacy-preserving data sharing for COVID-19 medical records: a blockchain-empowered approach," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 1, pp. 271–281, 2022.
- [19] K. Yu, L. Tan, C. Yang et al., "A blockchain-based Shamirs threshold cryptography scheme for data protection in industrial Internet of Things settings," *IEEE Internet of Things Journal*, p. 1, 2021.
- [20] W. Wang, C. Qiu, Z. Yin et al., "Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, p. 1, 2021.

- [21] L. Tan, N. Shi, K. Yu, M. Aloqaily, and Y. Jararweh, "A blockchain-empowered access control framework for smart devices in green Internet of Things," *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1–20, 2021.
- [22] C. Feng, B. Liu, K. Yu, S. K. Goudos, and S. Wan, "Blockchain-empowered decentralized horizontal federated learning for 5G-enabled UAVs," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3582–3592, 2022.
- [23] M. Vashishtha, P. Chouksey, D. S. Rajput et al., "Security and detection mechanism in IoT-based cloud computing using hybrid approach," *International Journal of Internet Technology and Secured Transactions*, vol. 11, no. 5/6, pp. 436–451, 2021.
- [24] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 8, pp. 2072–2085, 2015.
- [25] T. Guo, K. Yu, M. Aloqaily, and S. Wan, "Constructing a prior-dependent graph for data clustering and dimension reduction in the edge of AIoT," *Future Generation Computer Systems*, vol. 128, pp. 381–394, 2022.
- [26] K. Yu, Z. Guo, Y. Shen, W. Wang, J. C. Lin, and T. Sato, "Secure artificial intelligence of things for implicit group recommendations," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2698–2707, 2021.
- [27] L. Xu, X. Zhou, X. Li, R. H. Jhaveri, T. R. Gadekallu, and Y. Ding, "Mobile collaborative secrecy performance prediction for artificial IoT networks," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.
- [28] L. Tan, K. Yu, F. Ming, X. Cheng, and G. Srivastava, "Secure and resilient artificial Intelligence of Things: a honeyNet approach for threat detection and situational awareness," *IEEE Consumer Electronics Magazine*, p. 1, 2021.
- [29] X. Shang, L. Tan, K. Yu, J. Zhang, K. Kaur, and M. M. Hassan, "Newton-interpolation-based zk-SNARK for Artificial Internet of Things," *Ad Hoc Networks*, vol. 123, p. 102656, 2021.
- [30] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5GB remote e-health," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 54–61, 2021.
- [31] F. Ding, G. Zhu, M. Alazab, X. Li, and K. Yu, "Deep-learning-empowered digital forensics for edge consumer electronics in 5G HetNets," *IEEE Consumer Electronics Magazine*, vol. 11, no. 2, pp. 42–50, 2022.
- [32] L. Tan, K. Yu, A. K. Bashir et al., "Toward real-time and efficient cardiovascular monitoring for COVID-19 patients by 5G-enabled wearable medical devices: a deep learning approach," *Neural Computing and Applications*, pp. 1–14, 2021.
- [33] H. Li, M. Zhang, Z. Yu, Z. Li, and N. Li, "An improved pix2pix model based on Gabor filter for robust color image rendering," *Mathematical Biosciences and Engineering*, vol. 19, no. 1, p. 86101, 2022.
- [34] H. Li, Q. Zheng, J. Zhang, Z. Du, Z. Li, and B. Kang, "Pix2-Pix-based grayscale image coloring method," *Journal of Computer-Aided Design & Computer Graphics*, vol. 33, no. 6, pp. 929–938, 2021.
- [35] H. I. Ahmed, N. A. Elfeshawy, S. F. Elzoghdy, H. S. Elsayed, and O. S. Faragallah, "A neural network-based learning algorithm for intrusion detection systems," *Wireless Personal Communications*, vol. 97, no. 2, pp. 3097–3112, 2017.
- [36] Q. Zhang, K. Yu, Z. Guo et al., "Graph neural networks-driven traffic forecasting for connected Internet of Vehicles," *IEEE Transactions on Network Science and Engineering*, p. 1, 2021.
- [37] H. Li, Q. Zheng, X. Qi et al., "Neural network-based mapping mining of image style transfer in big data systems," *Computational Intelligence and Neuroscience*, vol. 2021, 11 pages, 2021.
- [38] I. Ahmad and F. E. Amin, "Towards feature subset selection in intrusion detection," in *Proceedings of 2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference*, pp. 69–74, Beijing, 2014.
- [39] A. A. Aburomman and M. B. Ibne Reaz, "A novel weighted support vector machines multiclass classifier based on differential evolution for intrusion detection systems," *Information Sciences*, vol. 414, pp. 225–246, 2017.
- [40] S. Rastegari, P. Hingston, and C.-P. Lam, "Evolving statistical rulesets for network intrusion detection," *Applied Soft Computing*, vol. 33, pp. 348–359, 2015.
- [41] Y. Xue, W. Jia, X. Zhao, and W. Pang, "An evolutionary computation based feature selection method for intrusion detection," *Security and Communication Networks*, vol. 2018, 10 pages, 2018.
- [42] S. N. Ghorpade, M. Zennaro, B. S. Chaudhari, R. A. Saeed, H. Alhumyani, and S. Abdel-Khalek, "Enhanced differential crossover and quantum particle swarm optimization for IoT applications," *IEEE Access*, vol. 9, pp. 93831–93846, 2021.
- [43] S. S. Fang, Z. Y. Chai, and Y. L. Li, "Dynamic multi-objective evolutionary algorithm for IoT services," *Applied Intelligence*, vol. 51, no. 3, pp. 1177–1200, 2021.
- [44] C. Iwendi, P. K. R. Maddikunta, T. R. Gadekallu, K. Lakshmana, A. K. Bashir, and M. J. Piran, "A metaheuristic optimization approach for energy efficiency in the IoT networks," *Software: Practice and Experience*, vol. 51, no. 12, pp. 2558–2571, 2021.
- [45] L. Yang, K. Yu, S. X. Yang, C. Chakraborty, Y. Liu, and T. Guo, "An intelligent trust cloud management method for secure clustering in 5G enabled Internet of Medical Things," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.
- [46] T. N. Qureshi, N. Javaid, A. Almogren, A. U. Khan, H. Almajed, and I. Mohiuddin, "An adaptive enhanced differential evolution strategies for topology robustness in internet of things," *International Journal of Web and Grid Services*, vol. 18, no. 1, pp. 1–33, 2022.
- [47] R. Storn and K. Price, *Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces*, Technical Report TR-95-012. ICSI, 1995.
- [48] S. Das, A. Mandal, and R. Mukherjee, "An adaptive differential evolution algorithm for global optimization in dynamic environments," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 966–978, 2014.
- [49] A. W. Mohamed, A. K. Mohamed, E. Z. Elfeky, and M. Saleh, "Enhanced directed differential evolution algorithm for solving constrained engineering optimization problems," *International Journal of Applied Metaheuristic Computing*, vol. 10, no. 1, pp. 1–28, 2019.
- [50] M. H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, and H. Faris, "MTDE: an effective multi-trial vector-based differential evolution algorithm and its applications for engineering design problems," *Applied Soft Computing Journal*, vol. 97, p. 106761, 2020.

- [51] X. Yu, C. Li, and J. Zhou, "A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios," *Knowledge-Based Systems*, vol. 204, p. 106209, 2020.
- [52] Y. Peng, A. Jolfaei, and K. Yu, "A novel real-time deterministic scheduling mechanism in industrial cyber-physical systems for energy internet," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.
- [53] A. W. Mahamed and P. N. Suganthan, "Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation," *Soft Computing*, vol. 22, no. 10, pp. 3215–3235, 2018.
- [54] A. W. Mahamed and A. K. Mahamed, "Adaptive guided differential evolution algorithm with novel mutation for numerical optimization," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 2, pp. 253–277, 2019.
- [55] G. Wu, X. Shen, H. Li, H. Chen, A. Lin, and P. N. Suganthan, "Ensemble of differential evolution variants," *Information Sciences*, vol. 423, pp. 172–186, 2018.
- [56] S. A. H. ElQuliti and A. W. Mahamed, "A large-scale nonlinear mixed-binary goal programming model to assess candidate locations for solar energy stations: an improved real-binary differential evolution algorithm with a case study," *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 11, pp. 7909–7921, 2016.
- [57] C. Fu, C. Jiang, G. Chen, and Q. Liu, "An adaptive differential evolution algorithm with an aging leader and challengers mechanism," *Applied Soft Computing*, vol. 57, pp. 60–73, 2017.
- [58] Z. Sun, N. Wang, Y. Bi, and D. Strinivasan, "Parameter identification of PEMFC model based on hybrid adaptive differential evolution algorithm," *Energy*, vol. 90, pp. 1334–1341, 2015.
- [59] T. Huynh, D. Do, and J. Lee, "Q-learning-based parameter control in differential evolution for structural optimization," *Applied Soft Computing*, vol. 107, no. 11, article 107464, 2021.
- [60] Y. Zeng, J. Zhang, L. Peng, and L. Wang, "Hybrid differential evolution supported by simulated annealing and its application in integrated joint replenishment-delivery problems," *Application Research of Computers*, vol. 35, no. 4, pp. 1037–1041, 2018.
- [61] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, *Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-Parameter numerical optimization*, Nanyang Technological University, Jordan University of Science and Technology and Zhengzhou University, Tech. Rep, 2016.
- [62] Y. Lu, Y. Lin, Q. Peng, and Y. Wang, "Summary of simulated annealing algorithm improvement and parameter exploration," *College Mathematics*, vol. 31, no. 6, pp. 96–103, 2015.
- [63] Q. Yan, R. Ma, Y. Ma, and J. Wang, "Adaptive simulated annealing particle swarm optimization algorithm," *Journal of Xidian University*, vol. 48, no. 4, pp. 1–9, 2021, (in Chinese).
- [64] W. Deng, J. Xu, Y. Song, and H. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Applied Soft Computing*, vol. 100, p. 106724, 2020.
- [65] Z. Sun, Y. Ling, H. Qu, Z. Sun, and F. Wu, "An adaptive DE algorithm based fuzzy logic anti-swing controller for overhead crane systems," *International Journal of Fuzzy Systems*, vol. 22, no. 6, pp. 1905–1921, 2020.