Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9530274, 14 pages
https://doi.org/10.1155/2021/9530274

Research Article

DIAMOND: A Structured Coevolution Feature Optimization Method for LDDoS Detection in SDN-IoT

Wencheng Yin,[1] Yunhe Cui,[1] Qing Qian,[2] Guowei Shen,[1] Chun Guo,[1] and Saifei Li[3]

[1]*State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550000, China*
[2]*School of Information, Guizhou University of Finance and Economics, Guiyang 550000, China*
[3]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu 610000, China*

Correspondence should be addressed to Yunhe Cui; yhcui@gzu.edu.cn

Received 5 August 2021; Revised 8 October 2021; Accepted 24 November 2021; Published 29 December 2021

Academic Editor: Zhihan Lv

Software-defined networking for IoT (SDN-IoT) has become popular owing to its utility in smart applications. However, IoT devices are limited in computing resources, which makes them vulnerable to Low-rate Distributed Denial of Service (LDDoS). It is worth noting that LDDoS attacks are extremely stealthy and can evade the monitoring of traditional detection methods. Therefore, how to choose the optimal features to improve the detection performance of LDDoS attack detection methods is a key problem. In this paper, we propose DIAMOND, a structured coevolution feature optimization method for LDDoS detection in SDN-IoT. DIAMOND is consisted of a reachable count sorting clustering algorithm, a group structuring method, a comutation strategy, and a cocrossover strategy. By analysing the information of SDN-IoT network features in the solution space, the relationship between different SDN-IoT network features and the optimal solution is explored in DIAMOND. Then, the individuals with associated SDN-IoT network features are divided into different subpopulations, and a structural tree is generated. Further, multiple structural trees evolve in concert with each other. The evaluation results show that DIAMOND can effectively select optimal low-dimension feature sets and improve the performance of the LDDoS detection method, in terms of detection precision and response time.

1. Introduction

Internet of Things (IoT) ecosystem is one of the most critical aspects of human lives, which facilitates a wide variety of applications in different domains such as smart homes, agriculture, healthcare, smart cities, smart grids, industrial automation (Industry 4.0), smart driving, and elderly assistance [1]. Hence, guaranteeing communication quality for IoT devices is an issue worth exploring. In this work, WiFi is utilized as the communication protocol between IoT devices and access points (APs), which are implemented through Mininet-wifi. In fact, WiFi provides low latency and low latency jitter necessary for IoT applications such as industrial production lines, IP telephony, video conferencing, and virtual desktop infrastructures. However, the computing resources, storage resources, and network capacity of IoT devices are limited to carry high-speed data transmission.

Recently, SDN-IoT is proposed to improve transmission quality [2]. Figure 1 shows the SDN-IoT architecture. In SDN-IoT, SDN changes the limitation of the network infrastructure, gives the network more flexibility, and simplifies policy implementation and network configuration [3]. SDN-IoT is anticipated to smartly route traffic and use underutilized network resources to deliver IoT data to the Internet. However, due to the stronger flexibility that SDN gives to the network, SDN-IoT faces many security threats [4].

LDDoS (Low-rate Distributed Denial of Service) attack is a serious threat for SDN-IoT, which exploits the vulnerabilities in network protocols to launch attacks and often realizes superior attack effects at a smaller attack cost. LDDoS attacks can elude the monitor of the traditional detection approach by sending low-rate packets in the way of periodic pulse to a victim [5]. Hence, how to effectively detect LDDoS

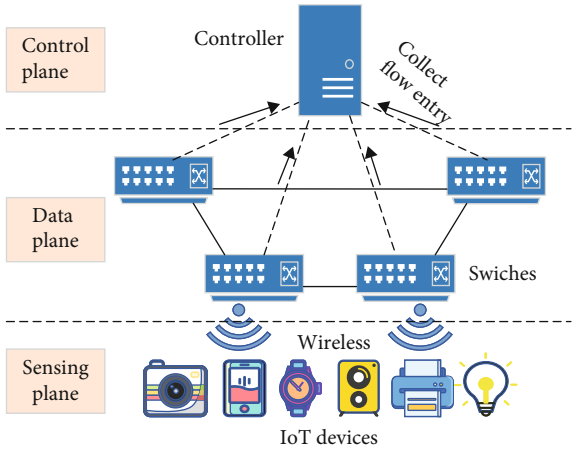

Figure 1: The SDN-IoT architecture.

is a huge challenge [6]. Feature selection makes an outstanding contribution to overcoming the invisibility of LDDoS, exploring LDDoS-related features, and improving the efficiency of LDDoS detection.

In recent years, feature optimization algorithms based on Evolutionary Algorithms (EAs) are popular [7–13]. However, the existing studies have suffered from the following limitations.

(1) Existing EA-based feature optimization methods usually evolve individuals in a population, leading to the loss of diversity when evolving individuals

(2) The existing methods do not consider the coevolution of similar individuals during evolution, which leads to the reduction of convergence ratio

In order to overcome the shortcomings mentioned above, a structured coevolution feature optimization method for LDDoS detection in SDN-IoT (DIAMOND) is proposed. The main contributions are concluded as below:

(1) By dividing the population into several subpopulations based on the proposed reachable count sorting clustering algorithm and a group structuring method and executing coevolution based on the designed comutation strategy and cocrossover strategy, DIAMOND is proposed

(2) A reachable count sorting clustering algorithm (BONNET) is designed to divide the population into subpopulations with different SDN-IoT network feature information, and each subpopulation is considered as a suboptimal solution set in the solution space

(3) A group structuring method is designed to further structure a subpopulation into structural trees in order to sort the individuals in the subpopulation orderly based on SDN-IoT network information, where multiple structural trees evolve in concert

(4) A comutation strategy is proposed based on the optimal subpopulation guidance direction, amount of information on SDN-IoT network features, and the evolutionary trajectory of the subpopulation, to move the individual towards the optimal solution in an orderly manner

(5) A cocrossover strategy is designed to facilitate information exchange between different structural trees by exchanging individuals between different structural trees and the genes of individuals in the same tree

2. Related Works

This section presents a review of intrusion detection systems for malicious traffic detection in SDN-IoT networks as well as the deployment of malicious network attack detection technologies for feature selection tasks.

Mafarj and Mirjalili [7] proposed a new wrapper feature selection approach based on whale optimization algorithm (WOA). In this work, two binary variants of the WOA algorithm are proposed to search the optimal feature subsets for classification. Emary et al. [8] proposed a new grey wolf optimization (GWO) algorithm. In that approach, the sigmoidal function is responsible for squeezing the continuously updated position; then, these values are stochastically compared with a threshold to find the updated binary grey wolf position. Abdel-Basset et al. [9] proposed a new grey wolf optimizer algorithm integrated with a two-phase mutation to solve the feature selection for classification problems based on the wrapper methods. Mafarja and Mirjalili [10] used two hybridization models to design different feature selection techniques based on WOA. In the first model, simulated annealing (SA) algorithm is embedded in WOA, while it is used to improve the best solution found after each iteration of WOA in the second model.

Roopak et al. [11] have implemented the nondominated sorting algorithm with its adapted jumping gene operator to solve the optimization problem. The proposed method exploits an extreme learning machine as the classifier for feature selection based on six important objectives for an IoT network. Mafarja et al. [12] presented a novel wrapper feature selection approach based on augmented WOA, which was adopted in the context of IoT attacks detection. Transfer functions (sigmoid function and tanh function) are applied in the real-value step vector such that a high probability of change is given to the dimension with a large value. Haddadpajouh et al. [13] proposed a multikernel support vector machine (SVM) for IoT cloud-edge gateway malware hunting, using the GWO technique.

Table 1 describes the advantages and limitations of the related works. As shown in Table 1, the common limitations in these literatures are the high complexity of the models and the tendency of falling into local optimal solutions. This is because the existing methods do not consider the coevolution of similar individuals during evolution, leading to a complex model, which in turn leads to the decrease of the convergence rate. Likewise, existing methods lose diversity

Table 1: Comparison with related works.

Reference	Technique	Advantage	Limitation
[7]	Whale optimization algorithm, binary variants	The two binary variants avoid a large number of local solutions.	More time and resources are consumed.
[8]	Binary grey wolf optimization	It is efficient in exploration and development.	This highly complex method potentially occupies more time and resource consuming.
[9]	Grey wolf optimizer algorithm, two-phase mutation	The two-stage mutation operator contributes to enhanced exploration.	More parameters need to be determined.
[10]	Hybrid whale optimization algorithm, simulated annealing	The algorithm reaches high accuracy by using a fewer number of features.	It has a high complexity and a tendency to fall into local optimal solutions.
[11]	Nondominated sorting algorithm adapted jumping gene operator	The jumping gene operator contributes to improve the detection accuracy.	Falling into local optimal solutions is a possible limitation.
[12]	The augmented whale optimization algorithm	It can search a more comprehensive range.	The convergence speed may be a little slow.
[13]	Multikernel SVM, gray wolf optimization	The multikernel SVM enhances the accuracy of the algorithm.	The multi-kernel SVM method may lead to overfitting.
DIAMOND	A structured coevolution feature optimization method for LDDoS detection in SDN-IoT	The model enables ordered individuals and coevolution of multiple subpopulations, reducing redundant computations.	How to coevolve subpopulations is an issue need to be addressed.

in populations during evolution, due to the fact that they usually evolve individuals in only one population, leaving the model trapped in a local optimum solution. The performance of the algorithm is limited by these problems. Therefore, DIAMOND is designed to solve these problems in this work.

3. DIAMOND: A Structured Coevolution Feature Optimization Method for LDDoS Detection in SDN-IoT

3.1. Motivation. As a result of the rapid increase in the number of mobile-connected devices such as smartphones and tablets, SDN-IoT is attracting more and more attention from industry and academia. Therefore, the security of SDN-IoT becomes a serious issue [14]. Feature selection methods for network attack detection methods have been a hot research topic among researchers in recent years. However, the existing feature selection methods have some limitations such as follows:

(1) Existing EA-based feature optimization methods usually evolve individuals in a population, leading to the loss of diversity when evolving individuals

(2) The existing methods do not consider the coevolution of similar individuals during evolution, which leads to the reduction of convergence ratio

Existing feature selection methods all have the common limitations of high model complexity and the tendency to fall into local optimal solutions. Therefore, in order to solve these problems, this work enables multiple subpopulations to coevolve and makes subpopulations into ordered populations. In this way, multiple subpopulations can tract each other to avoid falling into local optimal solutions. At the same time, individuals within subpopulations are clearly distinguished to reduce the consumption of computational resources.

3.2. Overall Structure. Figure 2 shows the overall structure of DIAMOND in SDN-IoT. DIAMOND is deployed on the SDN controller and collects network traffic by sending collect traffic instructions to the switch. Generally, DIAMOND includes the following steps:

(i) *Step 1*. DIAMOND sends the instruction to collect network traffic information.

(ii) *Step 2*. Once the SDN controller receives the instruction, it sends OFPT_STATS_REQUEST messages to the switches to get the statistics of flow entries.

(iii) *Step 3*. The switches will reply OFPT_STATS_REPLY messages to the SDN controller. These messages contain the stream data in the flow entries of the switches.

(iv) *Step 4*. After the controller receives the OFPT_STATS_REPLY messages, it deencapsulates these messages and sends the collected statistic data to DIAMOND.

(v) *Step 5*. In DIAMOND, the statistic data of flow entries collected from switches are used to calculate the initial SDN-IoT network features. Subsequently, these features will be coded using the binary coding method to generate the initial population that is consisted of some individuals.

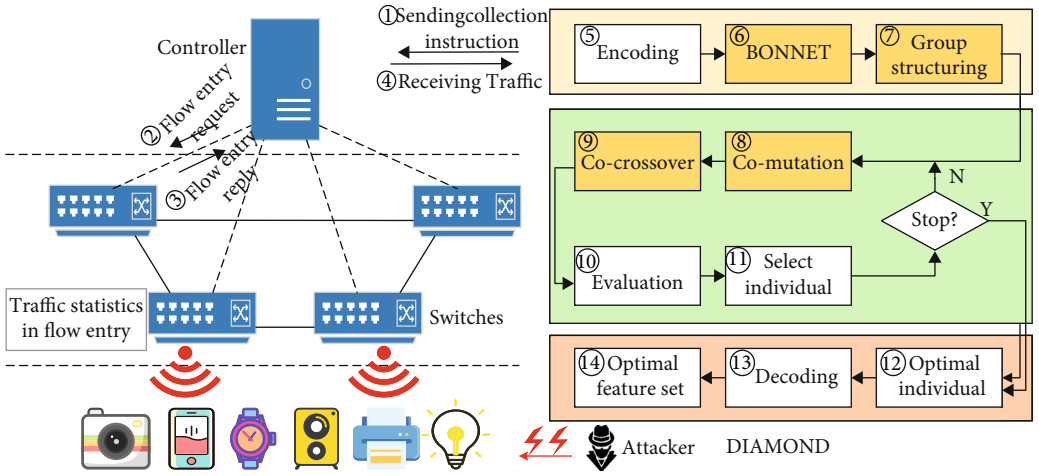

Figure 2: The overall structure of DIAMOND in SDN-IoT.

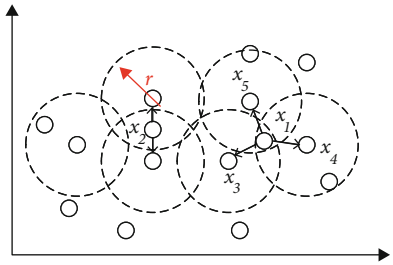

Figure 3: The neighborhood radius of individuals.

(vi) *Step 6*. The population is divided into several subpopulations by BONNET.

(vii) *Step 7*. The structuring factor is calculated based on the fitness factor and the membership factor, which structures the individuals in the subpopulation into a structural tree.

(viii) *Step 8* and *Step 9*. Multiple structural trees evolve collaboratively with each other through the mutation and crossover strategies.

(ix) *Step 10* and *Step 11*. After evaluation and selection, the dominant individuals are retained to the next generation.

(x) *Step 12*, *Step 13*, and *Step 14*. After several iterations, the individual with maximum fitness will be identified as the best individual. Subsequently, the best individual will be decoded into the corresponding feature set.

As described above, we use binary coding to translate the original SDN-IoT network feature set to the individual used in DIAMOND. We assume that the original feature set x_i includes d-dimensional gene sequences. Hence, we set d-dimensional gene sequences as follows:

$$x_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,d}\}. \quad (1)$$

x_i is an individual, which presents a network feature set of SDN-IoT. When gene $x_{i,j} = 1$, it means j-numbered feature is selected. Conversely, when gene $x_{i,j} = 0$, it means j-numbered feature is not selected.

3.3. BONNET: Reachable Count Sorting Clustering Algorithm. In SDN-IoT, the subset SDN-IoT network features are encoded as binary individuals. The initialized individuals are defined as data points that are distributed in the solution space $\mathbb{R}^n$. When individuals share the same or similar SDN-IoT network features, the distribution of these data points in the solution space is relatively aggregated. Then, when individuals distributed in the solution space are clustered into a subpopulation, it means that SDN-IoT network features in individuals are related to the solution closest to the subpopulation. Therefore, we divide the individuals into several subpopulations in order to extract information about SDN-IoT network features in subpopulations and explore the relationship between SDN-IoT network features and optimal solutions. As a result, we design a reachable count sorting clustering algorithm (BONNET) to divide the population into several subpopulations. In BONNET, the following definition is proposed:

(1) A data point (x_i, ϵ) is defined as a reachable point, which is within the hypersphere region with a certain data point x_i. Here, x_i is defined as the center of the sphere, and the length ϵ is recognized as the radius of the neighborhood

(2) A subpopulation centrosome κ is defined as an individual located at the center of the subpopulation K

Further, the distance distribution between individuals in the population is denoted as $S(s) = \{s_i \mid i \in N(N-1)/2\}$. Subsequently, $\epsilon = \mu - 3\sigma$ means neighborhood radius of S,

Input: Population number: N; Individuals: x; Sub-populations number: c
Output: The sub-population: $\mathrm{K}(\tilde{\kappa})$
$Q(x), \mathrm{K}(\tilde{\kappa}), S(s) \longleftarrow \{\}$, $\epsilon = 0$
$\longleftarrow$: the add element operation.
(x_i, ϵ): the reachable points of x_i.
$\mathscr{R}$: the distance function.
$\mathscr{T}$: the function that take out first element.
$\mathscr{F}(x_i^*)$: the set of descending order of x_i^*.
$\mathscr{M}(x_i^*)$: the function that obtain the corresponding individual.
$\mathscr{H}(\mathrm{K}(\tilde{\kappa}), c)$: the function that get the largest c sub-populations in $\mathrm{K}(\tilde{\kappa})$.
1 for $i < N - 1$ do
2 for $i + 1 < j < N$ do
3 $S(s) \longleftarrow \mathscr{R}(x_i, x_j)$
4 end for
5 end for
6 $\mu, \sigma = \bar{S}$, var (S)
7 $\epsilon = \mu - 3\sigma$
8 for $i < N$ do
9 $x_i^* = 0$
10 for each s_j about x_i do
11 if $s_j < \epsilon$ then
12 $x_i^* \mathrel{+}= 1$
13 end if
14 end for
15 $\mathscr{F}(x_i^*) \longleftarrow x_i^*$
16 end for
17 for each x_i^* do
18 $x_i = \mathscr{M}(x_i^*)$
19 $Q(x) \longleftarrow x_i$
20 while $Q(x) \neq \varnothing$ do
21 $x_i = \mathscr{T}(Q(x))$
22 $\widetilde{\kappa_i} = x_i, \{\widetilde{\kappa_i}\} \longleftarrow (x_i, \epsilon)$
23 $\mathrm{K}(\tilde{\kappa}) \longleftarrow \{\widetilde{\kappa_i}\}$
24 $Q(x) = Q(x) - (x_i, \epsilon)$
25 end while
26 while cluster_count$(\mathrm{K}(\tilde{\kappa})) > c$ do
27 $\{\tilde{\kappa}\}_{\min} = \min(\mathrm{K}(\tilde{\kappa}))$, $\{\tilde{\kappa}\}_{\text{main}} = \mathscr{H}(\mathrm{K}(\tilde{\kappa}), c)$
28 $\{\tilde{\kappa}\}_{\min}, \{\tilde{\kappa}\}_{\text{cloest}} = \arg\min \mathscr{R}(\{\tilde{\kappa}\}_{\min}, \{\tilde{\kappa}\}_{\text{main}})$
29 $\{\tilde{\kappa}\}_{\text{cloest}} \longleftarrow \{\tilde{\kappa}\}_{\min}$
30 $\mathrm{K}(\tilde{\kappa}) = \mathrm{K}(\tilde{\kappa}) - \{\tilde{\kappa}\}_{\min}$
31 end while
32 return $\mathrm{K}(\tilde{\kappa})$

Algorithm 1: BONNET.

μ means the mean value of S, and σ stands for the variance of S.

We measure the distribution density of the population in terms of the number of times each individual is utilized as a reachable point. Hence, x_i^* is defined as the number of times that x_i is utilized as a reachable point. Then, $\mathscr{F}(x_i^*)$ means the set of descending order of x_i^*, and $\mathscr{M}(x_i^*)$ stands for obtaining the corresponding individual. Further, the initial queue $Q(x)$ is defined as follows, where x_i^* of the individuals decreases sequentially.

$$x_i = \mathscr{M}(x_i^*), \tag{2}$$

$$Q(x) = \{x_1, x_2, \cdots, x_N | x_i^* > x_{i+1}^*\}. \tag{3}$$

Subsequently, the individual x_i is selected as the subpopulation centrosome $\widetilde{\kappa_i}$ which is removed from $Q(x)$ in turn, and the reachable points within the neighborhood radius ϵ of the individual $\widetilde{\kappa_i}$ are formed as a subpopulation $\{\widetilde{\kappa_i}\}$. Then, the reachable points of $\widetilde{\kappa_i}$ are removed from $Q(x)$. Finally, $\{\widetilde{\kappa_i}\}$ is joined to the subpopulation queue $\mathrm{K}(\tilde{\kappa})$. As shown in Equation (4), c is the number of subpopulations.

$$\mathrm{K}(\mathrm{K}) = \left\{\left\{\widetilde{\kappa_1}\right\}, \left\{\widetilde{\kappa_2}\right\}, \cdots, \left\{\widetilde{\kappa_c}\right\} \mid \left\{\widetilde{\kappa_i}\right\} \cap \left\{\widetilde{\kappa_j}\right\} \cap Q(x) = \emptyset\right\}. \tag{4}$$

An example is shown in Figure 3, where $x_1^* = 3$, $x_2^* = 2$. When x_1 is removed from the initial queue and denoted

the subpopulation centrosome $\tilde{\kappa}_1$, the reachable points x_3, x_4, and x_5 in (x_1, ϵ) are joined to the subpopulation $\{\tilde{\kappa}_1\}$. It is worth noting that there are still outliers and smaller subpopulations around the main subpopulations. Therefore, we merge these outliers and smaller subpopulations into the surrounding main subpopulations. Algorithm 1 describes the detailed steps of BONNET.

3.4. Group Structuring Method. In the subpopulations divided by BONNET, individuals are relatively close together, but

it lacks group coordination during evolution. In order to distinguish the importance of different SDN-IoT network features among common individuals and enable important SDN-IoT network features to provide guidance to individuals in a subpopulation, we propose a group structuring method. During the group structuring, we combine the fitness and membership of the individuals construct the structural tree.

$$\rho_i = \frac{f_i}{\sum_{j=1}^{n} f_j}, \tag{5}$$

$$\gamma_i = \frac{1/s_{j,i}^2}{\sum_{k=1}^{n}\left(1/s_{j,k}^2\right)}. \tag{6}$$

As shown in Equation (5), f_i stands for the fitness of individual x_i, and the fitness factor ρ_i reflects the guidance ability of the individual x_i in the subpopulation. When the individual x_i is close to the optimal solution, ρ_i of x_i has a larger value. As shown in Equation (6), $s_{j,i}$ means the distance from the individual x_i to the subpopulation centrosome $\widetilde{\kappa_j}$. Then, n means the number of individuals in the subpopulation, and the membership factor γ_i means the degree to which the individual belongs to the subpopulation centrosome $\tilde{\kappa}_j$. As shown in Figure 4, γ_i stratifies individuals in the subpopulation $\{\widetilde{\kappa_i}\}$ by distance, and there is some genetic similarity between individuals at the same level. However, individuals in the same layer have different ρ values depending on whether they have the SDN-IoT network features associated with the optimal solution.

Generally, the optimal solution is distributed on one side of the subpopulation rather than inside the subpopulation, and individuals close to the optimal solution are provided with greater fitness. Hence, we design fitness factor minus membership factor as a structural factor.

$$\zeta_i = -\gamma_i + \rho_i. \tag{7}$$

As shown in Figure 5, the individuals at the same level share an identical degree of membership and gene similarity. The closer an individual is to the optimal solution, the stronger the ability to guide. As a result, the structured factor ζ_i integrates individual guidance ability and gene similarity. We build a structural tree $\mathscr{L}_i$ based on the value of ζ_i in

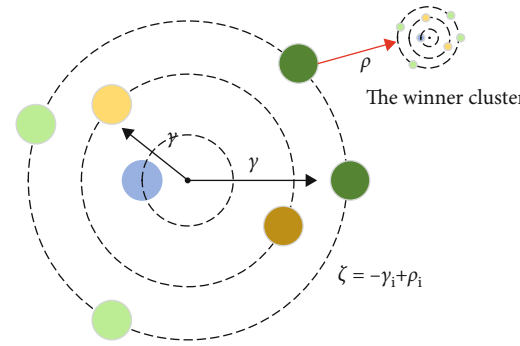

Figure 4: The structural factor of individuals.

descending order, and the individuals with the same score are sibling nodes. As shown in Equation (8), c is the number of structural trees.

$$\mathscr{L} = \{\mathscr{L}_1, \mathscr{L}_2, \cdots, \mathscr{L}_c\}. \tag{8}$$

When the structural tree coevolves, the sibling nodes have equivalent evolutionary guidance ability and similar genes. Therefore, only partial individuals of sibling nodes are randomly selected to calculate the fitness to reduce the consumption of computing resources and accelerate the convergence rate.

Figure 5 shows the diversity of different structural trees. Δf indicates the gradient change of fitness. $\Delta f_3 > \Delta f_2$ indicates that the SDN-IoT network features in the individuals of information $h-1$ and h layers are more different. Due to the width of the structural tree b is greater than the width of the structural tree a, there are more individuals in the same layer, indicating that the gene pool of the structural tree b is rich in network feature information. In the process of the iterative evolution of the population, when the fitness of an individual in the h layer is greater than the average fitness of the node in the $h-1$ layer, the individual will transition to the $h-1$ layer.

To evaluate the convergence and stability of the structural tree, we define Equation (9) for the diversity of the structural tree:

$$\beta = \log \sum_{h=1}^{H} \left| \frac{\overline{f_{h-1}} - \overline{f_h}}{\overline{f_0} - \overline{f_h}} \right|^{\|w\|/\|H-h\|}. \tag{9}$$

As shown in Equation (9), w presents the number of nodes in the h layer, H means the depth of the tree, and $\overline{f_h}$ stands for the average fitness of the nodes in the h layer. When the width of the structural tree and fitness gradient change greatly, the structural tree has stronger diversity.

3.5. Comutation Strategy. In DIAMOND, the balance between global search and local optimization is extremely important. Therefore, we design a comutation strategy to guide the evolution of the population. In this strategy, we define the movement trajectory of the subpopulation centrosome γ, the disturbance factor δ, and the diversity factor β of

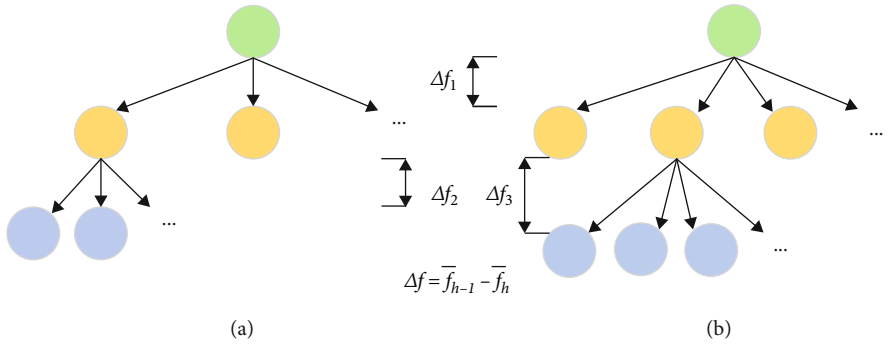

Figure 5: The structural trees of population.

the population structural tree. It is worth noting that the subpopulation centrosome $\tilde{\kappa}$ is characterized as the evolutionary benchmark of the individuals in the subpopulation $\{\tilde{\kappa}\}$. Therefore, the evolutionary trajectory and evolutionary route of the subpopulation centrosome $\tilde{\kappa}$ have important contributions to the evolution of the subpopulation $\{\tilde{\kappa}\}$. The movement trajectory of the subpopulation centrosome $\tilde{\kappa}$ is defined as Equation (10). In Equation (10), γ_{t-1} means the last predicted

position, and $\tilde{\kappa}_t$ stands for the current position, and then, for ϕ, we design it as the weight $\phi = \phi_0/e^{-t}$. Finally, t is the population generation.

$$\gamma_t = \frac{\phi\gamma_{t-1} + (1-\phi)\tilde{\kappa}_t}{1-\phi^t}, \tag{10}$$

$$u_i^{g+1} = (1+\beta)\frac{x_i^g\left(\gamma(\tilde{\kappa}) + \chi(\kappa_{\text{best}})\right)^2}{\left|\gamma(\tilde{\kappa}) + \chi(\kappa_{\text{best}})\right|^2} + \delta\left(x_i^g\right). \tag{11}$$

In Equation (11), u_i^{g+1} means that the mutation individual generated by x_i^g. g denotes the g-th generation of the population. $\gamma(\tilde{\kappa})$ stands for the movement trajectory of the subpopulation centrosome $\tilde{\kappa}$. $\chi(\kappa_{\text{best}})$ is designed as the direction of the winning subpopulation. Furthermore, $\delta(x_i^g)$ presents the disturbance factor generated by the x_i^g. The comutation strategy ensures global search capability while accelerating convergence.

As shown in Figure 6, the movement trajectory $\gamma(\tilde{\kappa})$ of the subpopulation centrosome $\tilde{\kappa}$ and the winning subpopulation $\chi(\kappa_{\text{best}})$ guides the search direction for the individual x_i^g. Further, x_i^g projects the direction that is the sum of direction with the subpopulation centrosome movement trajectory $\gamma(\kappa)$ and the winning subpopulation $\chi(\kappa_{\text{best}})$. The diversity factor β regulates the evolutionary step length of individual x^g. The larger the diversity of the structural tree is, the stronger the overall optimization ability will be, and each individual in the structural tree should be given a larger explore step size. However, the weaker the diversity of the structural tree is, the stronger the local optimization ability will be, and each individual in the structural tree should be given a smaller step size. The disturbance factor $\varrho(x_i^g)$ produced by x_i^g can prevent the population from falling into a locally optimal solution.

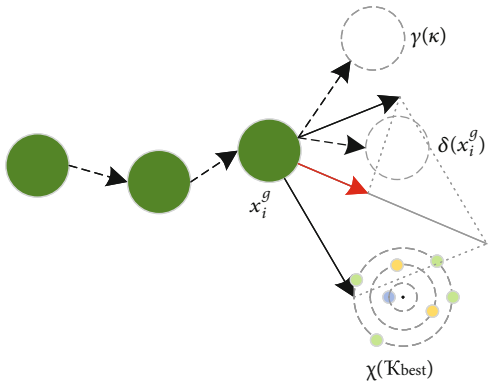

Figure 6: The comutation strategy.

In summary, the mutation individual u_i^{g+1} is directed to the direction of the optimal solution, which contains a large amount of network feature information related to the optimal solution. It is worth noting that the SDN-IoT network features related to the optimal solution in u_i^{g+1}, which carries more information than the SDN-IoT network feature unrelated to the optimal solution. Therefore, in order to reduce redundant features, we retain features closest to the optimal solution according to the network feature information carried by the mutation individual u_i^{g+1}.

$$v_i^{g+1} = \left\{ u_{i,j}^{g+1} \middle| u_{i,j}^{g+1} > \bar{u}_i^{g+1} \right\}, \tag{12}$$

$$v_i^{g+1} = \mathscr{H}\left(v_i^{g+1}, k\right), k = \text{rand}\left(\text{len}\left(v_i^{g+1}\right)\right), \tag{13}$$

$$\begin{cases} u_{i,j}^{g+1} = 1, \text{if } u_{i,j}^{g+1} \text{ in } v_i^{g+1}, \\ u_{i,j}^{g+1} = 0 \text{ otherwise}. \end{cases} \tag{14}$$

As in Equation (12), $\bar{u}_i^{g+1}$ refers to the average amount of information carried by each feature in the mutation individual u_i^{g+1}. v_i^{g+1} is composed of SDN-IoT network features $u_{i,j}^{g+1}$ whose information content is greater than $\bar{u}_i^{g+1}$, and v_i^{g+1} is more closely related to the SDN-IoT network features of the optimal solution. Further, k features with the largest

Input: Network feature: f; Population Number: N; Feature Dimension: d; Iterations Number: G;
Output: The best individual and best fitness: $\triangle$, $F(\triangle)$
1 for $i < N$ do
2 $x_i^g = \text{coding}(f)$
3 Calculate $F(x_i^g)$
4 end for
5 $\triangle = x_i^g$
6 Execute BONNET
7 Execute group structuring
6 for $g \leq G$ do
7 Update β using Eq. 9
8 Update $\gamma(\tilde{\kappa})$ using Eq. 10
9 for $i \leq N$ do
10 Execute mutation using Eq. 11–14
11 end for
12 for $i \leq N$ do
13 Execute crossover using Eq. 15–16
14 if $F(w_i^g) > F(x_i^g)$ then
15 $x_i^g = w_i^g$
16 if $F(w_i^g) > F(\triangle)$ then
17 $\triangle = x_i^g$
18 end if
19 end if
20 end for
21 $g = g + 1$
22 end for
23 return $\triangle$, $F(\triangle)$

Algorithm 2: DIAMOND.

information value in v_i^{g+1} are selected through $\mathscr{H}(v_i^{g+1}, k)$. As in Equation (14), if the network feature $u_{i,j}^{g+1}$ is in v_i^{g+1}, then $u_{i,j}^{g+1}$=1. Otherwise, $u_{i,j}^{g+1}$=0. This method can retain important SDN-IoT network features and filter out redundant SDN-IoT network features.

3.6. Cocrossover Strategy. We propose a cocrossover strategy to facilitate information exchange between different structural trees. The cocrossover strategy includes a crossover strategy between multiple subpopulations and a crossover strategy for structural trees within a subpopulation. As shown in Equations (15) and (16), w^g is a crossover intermediate set that contains all individuals generated by the crossover operation.

(1) As shown in Equation (15), individuals at the same level in multiple structural trees are randomly selected for exchange. For instance, individual x_k and individual x_m are randomly selected from the h -th level individuals of structural tree $\mathscr{L}_i$ and $\mathscr{L}_j$ for exchange, which can increase the information interaction between subpopulations and avoid falling into local optimum

$$w^g \longleftarrow \left\{ x_k, x_m \mid x_k \leftrightarrow x_m, x_k \in \mathscr{L}_i^h, x_m \in \mathscr{L}_j^h \right\} \tag{15}$$

(2) Individuals at the $h-1$ layer and h layer of the structural tree $\mathscr{L}_i$ are randomly selected for single-point crossover. For instance, the k-th gene of the x_i individual is exchanged with the allele of the x_j individual

$$w^g \longleftarrow \left\{ x_i, x_j \mid x_{i,k} \leftrightarrow x_{j,k}, x_i \in \mathscr{L}_i^{h-1}, x_j \in \mathscr{L}_i^h \right\} \tag{16}$$

3.7. Detailed Design of DIAMOND. The IoT devices may include cameras, smartwatches, smartphones, various activity sensors, smart speakers, and so on. These IoT devices are connected to SDN switches, which are the endpoints of the service provider's network [15].

In SDN-IoT, the LDDoS detection method needs to highly adaptively, fast, and accurately detect LDDoS. Hence, in DIAMOND, the AUC of detector performance, detection time (time), accuracy (ACC), and feature dimensions (DIM) are considered as the optimization objective, which is defined as follows:

$$F(x) = \sqrt{(\text{AUC}^2 + \text{ACC}^2) - (\text{time}^2 + \text{DIM}^2)}. \tag{17}$$

Each individual within the population has its binary coded genotype. In Equation (18), g is the number of iterations. N is the population number. d is the feature dimension.

$$x_i^g = (x_{i,1}, x_{i,2}, \cdots, x_{i,d}), x_{i,j} \in \{1, 0\}, i \in [1, N]. \tag{18}$$

Table 2: The flow features.

Feature	Description
Band_ratio	The bandwidth ratio occupied by the flow
Byte_asym	The convective byte asymmetry rate of a flow
Byte_count	The total number of bytes in the flow
Byte_rate	The byte growth rate of a flow
Byte_pkt	The average number of bytes of a packet in a flow
Duration	The duration of flow
eth_type	The Ethernet type of the flow
dst_host_srv_diff_host_ratio	The percentage of connections with different source hosts in the same service and the same target host
src_byte_ratio	The percentage of the number of bytes transferred by the current connection in that of bytes transferred by the same source host connection
srv_diff_host_ratio	The percentage of connection number of different target hosts in that of the same service
Byte_asym_inter	The convective byte asymmetry rate in the cycle
Input	The entering port
dst_ip	The serial number of destination IP
src_ip	The serial number of source IP
pkt_count_inter	The total number of packets of flow in the cycle
Output	The forwarding port
Packet_count	The total number of packets in the flow
pkt_asym	The convective packet asymmetry rate
Packet_rate	The packet growth rate
Priority	The flow priority
Same_dst_count	The total number of same destination IP connections
src_pkt_ratio	The percentage of the number packets in packets of same source host connection
Same_src_count	The total number of same source IP connections
pkt_asym_inter	The convective packet asymmetry rate in the cycle
Same_srv_ratio	The percentage of connections that have the same target hosts and the same services as the current connection
byt_count_inter	The total number of bytes in the flow in the cycle
diff_srv_ratio	The percentage that connects different services in which is the connection of the same target hosts
Byte_pkt_inter	The average number of bytes of a packet in the cycle
Byte_rate_inter	The growth rate of the byte in the cycle
pkt_rate_inter	The growth rate of the packet in the cycle
src_port_count	The total number of same source port connections
dst_port_count	The total number of same destination port connections
ip_proto	The protocol type
tcp_src	The serial number of source TCP
tcp_dst	The serial number of destination TCP
udp_src	The serial number of source UDP
udp_dst	The serial number of destination UDP

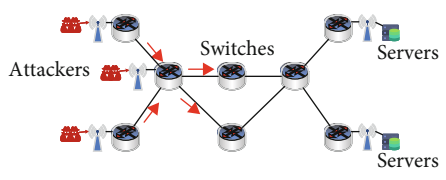

Figure 7: The experimental topology in SDN-IoT.

Table 3: Initial parameter settings in the DIAMOND.

Parm.	Value	Description
G	50	The number of iterations
N	100	The number of populations
c	3	The number of subpopulations
ϕ_0	0.5	The weight of movement trajectory

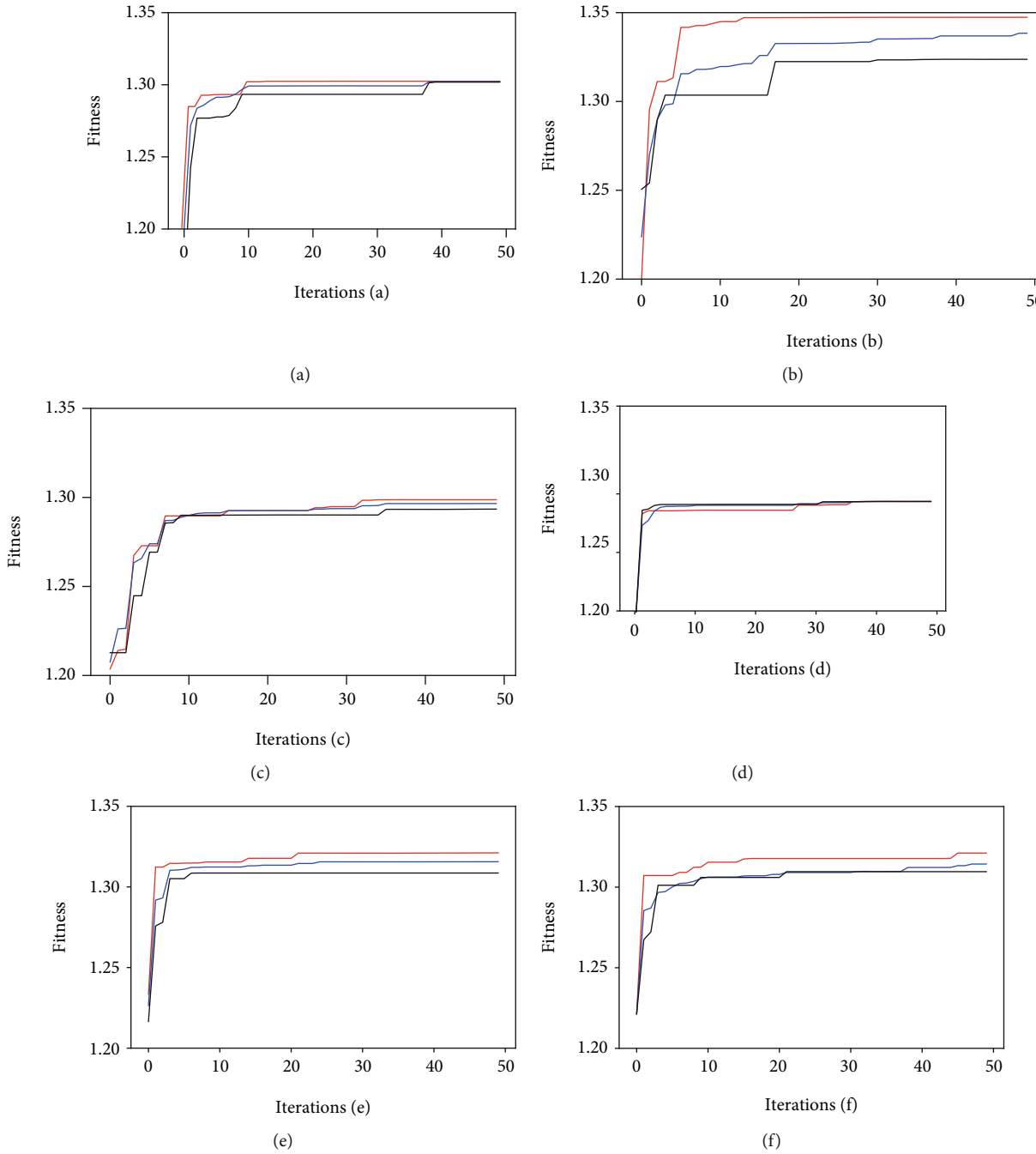

Figure 8: Continued.

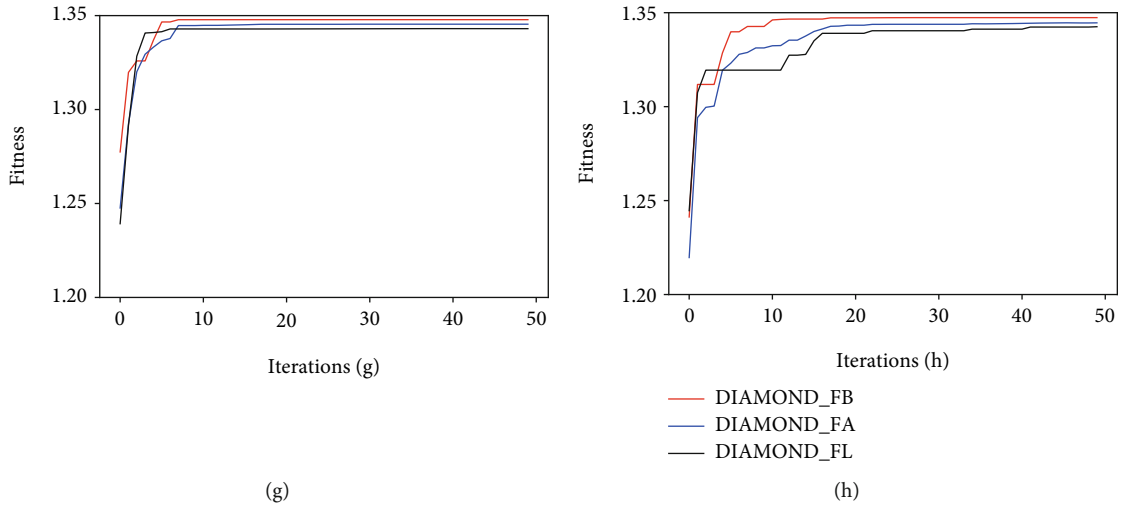

Figure 8: Fitness curve in DIAMOND: (a) SVM, (b) KNN, (c) NB, (d) LR, (e) DT, (f) C4.5, (g) RF, and (h) AB.

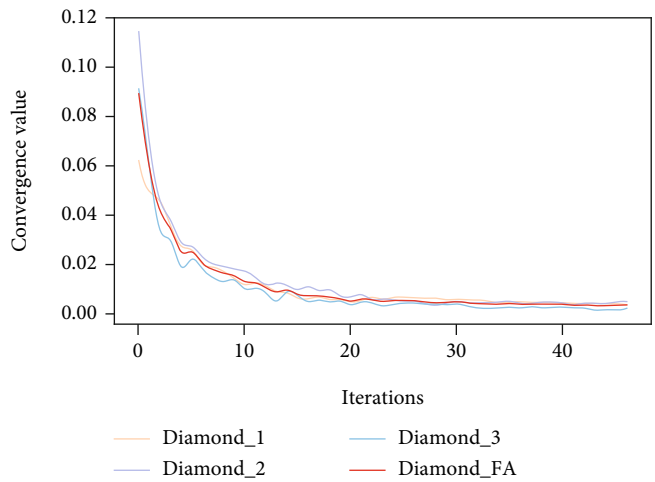

Figure 9: Convergence curve.

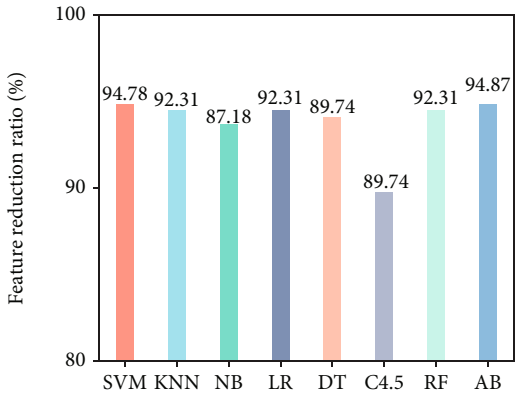

Figure 10: Dimension reduction ratio.

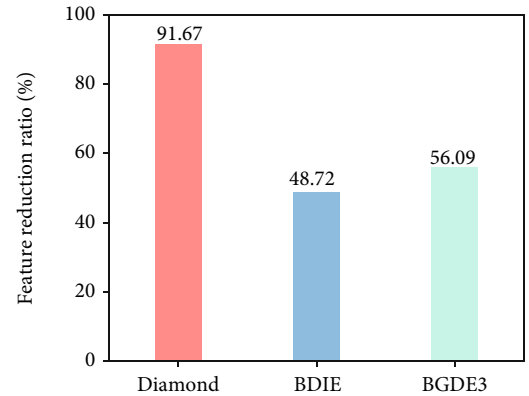

Figure 11: Average dimension reduction ratio.

Table 4: The detection performance comparison.

	Algo.	SVM	KNN	NB	LR	DT	C4.5	RF	AB	AVG	STD
Accuracy	DIAMOND	**91.38**	**94.35**	**93.63**	**89.73**	**90.55**	**91.38**	**93.74**	**93.63**	**92.30**	**1.63**
	BDIE	89.22	90.76	88.30	89.12	90.35	91.17	84.39	89.53	89.11	1.99
	BGDE3	85.73	84.91	88.19	82.03	89.32	76.69	81.93	89.53	85.88	4.38
Precision	DIAMOND	**90.66**	**92.65**	**90.98**	**90.95**	91.35	92.29	**90.99**	**92.16**	**91.50**	**0.70**
	BDIE	90.47	90.83	85.61	90.59	**92.45**	**93.14**	85.07	90.65	89.85	2.76
	BGDE3	91.06	86.96	89.77	88.62	91.88	85.63	89.04	90.52	89.55	2.16
Recall	DIAMOND	**95.59**	**98.47**	**99.32**	**92.19**	**93.21**	**93.55**	**99.49**	**97.79**	**96.20**	**2.75**
	BDIE	91.85	94.23	96.94	91.51	91.51	92.19	89.98	92.19	92.55	1.99
	BGDE3	84.72	88.29	90.83	80.65	90.32	73.85	79.97	92.36	86.72	6.19
F-score	DIAMOND	**93.06**	**95.47**	**94.97**	**91.57**	**92.27**	**92.92**	**95.05**	**94.89**	**93.78**	**1.39**
	BDIE	91.15	92.50	90.92	91.05	91.98	92.66	87.46	91.41	91.14	1.52
	BGDE3	87.77	87.62	90.30	84.44	91.10	79.31	84.36	91.43	88.04	4.09
AUC	DIAMOND	**92.93**	**96.61**	**91.26**	**91.77**	**96.96**	**94.98**	**97.29**	**97.08**	**94.86**	**2.36**
	BDIE	90.94	91.63	90.61	91.05	90.04	90.90	90.65	95.17	91.37	1.50
	BGDE3	91.49	85.79	87.96	85.89	92.07	77.58	89.36	96.50	89.24	5.93
FPR	DIAMOND	15.06	**11.95**	**15.06**	**14.03**	13.51	11.95	**15.06**	**12.73**	**13.67**	**1.26**
	BDIE	14.81	14.55	24.94	14.55	**11.43**	**10.39**	24.16	14.55	16.17	5.08
	BGDE3	**12.73**	20.26	15.84	15.84	12.21	18.96	**15.06**	14.81	15.39	2.80
FNR	DIAMOND	**4.41**	**1.53**	**0.68**	**7.81**	**6.79**	**6.45**	**0.51**	**2.21**	3.80	2.75
	BDIE	8.15	5.77	3.06	8.49	8.49	7.81	10.02	7.81	7.45	1.99
	BGDE3	15.28	11.71	9.17	19.35	9.68	26.15	20.03	7.64	13.29	6.19

After encoding the individuals, DIAMOND executes BONNET and group structuring method using Equations (2)–(8). Then, the comutation strategy is executed through Equations (11)–(14), and the cocrossover strategy is executed through Equations (15) and (16). In addition, by comparing the corresponding advantages and disadvantages of the new generation of the individual with the contemporary population, individuals who are more suitable for survival are selected as the new generations. As shown in Algorithm 2, individuals are encoded in lines 1~4. BONNET and group structuring in lines 6 and 7. The comutation strategy in lines 7~11. The cocrossover strategy and optimal individuals selecting strategy are implemented in lines 12~20.

The flow features are shown in Table 2. These features are extracted and calculated for each flow. Here, a flow is defined as traffic with the same source/destination MAC/IP address and the same source/destination port number.

4. Experimental Design and Result Analysis

4.1. Experiment Environment. To verify the logic of the program and evaluate its performance, DIAMOND is implemented on the RYU controller. In this work, Mininet-wifi implements the communication between IoT devices and access points (APs), using WiFi as the communication protocol between them. As shown in Figure 7, Mininet-wifi is utilized to simulate an SDN-IoT network that consists of eighty IoT endnotes, eight access points, and eight links, as previous work in [16]. The background traffic is to simulate the traffic environment of the IoT in the campus, which includes 40% HTTPS, 10% HTTP, 10% TCP, 10% SSDP, 5% DNS, and 5% NTP [17]. Specifically, the traffic is generated by applications on some IoT devices that require latency and low latency jitter, such as IP telephony and video conferencing. Further, the attack traffic consists of traffic generated by 20 IoT endnotes to launch LDDoS attacks against IoT servers. The attackers, victims, and service requests are all randomly generated. At the same time, the attack traffic accounts for 20% of normal traffic.

Further, in order to verify the detection ability of the optimized feature subset, eight different classifiers including support vector machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes (NB), Logistic Regress (LR), decision tree (DT), C4.5, Random Forest (RF), and AdaBoost (AB) have been applied to evaluate the feature optimization methods. In addition, DIAMOND is compared with other related state-of-the-art algorithms, which are BDIE [18] and BGDE3 [19]. As shown in Table 3, it describes the initial parameter settings in the DIAMOND.

4.2. Fitness Curve. Figure 8 shows the fitness curve of DIAMOND in the first 50 iterations. DIAMOND_FA, DIAMOND_FB, and DIAMOND_FL, respectively, represent the average fitness value, the best fitness value, and the lowest fitness value of DIAMOND. According to Figure 8, it can be

seen that after several iterations, the average fitness DIAMOND_FA steadily increases and reaches high values in all

Table 5: The classification detection time (ms) comparison.

Algo.	SVM	KNN	NB	LR	DT	C4.5	RF	AB
DIAMOND	**106.72**	**26.93**	**0.99**	**4.99**	**1.98**	**1.99**	**13.96**	**146.61**
BDIE	174.50	45.85	1.00	12.96	1.99	**1.99**	15.96	187.50
BGDE3	164.59	65.80	2.00	16.95	2.99	3.99	13.99	162.56

cases. At the early stage of evolution, the difference between the values of DIAMOND_FA, DIAMOND_FB, and DIAMOND_FL is small because they have not yet found the optimal solution set. In the middle of evolution, the difference between the values of DIAMOND_FB and DIAMOND_FL becomes larger because they have calculated different sets of solutions. At the later stage, almost all the values of DIAMOND_FB and DIAMOND_FL converge again, indicating that they have found out the optimal solution set.

4.3. Convergence Performance. Figure 9 exposes the convergence curve of DIAMOND. DIAMOND_1-3, respectively, stand for the convergence value of DIAMOND in 3 runs. DIAMOND_FA is the average convergence value of DIAMOND in 3 runs. DIAMOND_FA is the average convergence value. The convergence values of DIAMOND converge rapidly in the first half but gradually stabilize in the second half. The convergence value of DIAMOND decreases rapidly before 20 generations, while it stabilizes and decreases slowly after 20 generations. This indicates that the strong global search ability of DIAMOND leads to the exploration of an excellent solution set before 20 generations. Therefore, the population can converge to the outstanding individual quickly. Then, after 20 generations of iteration, DIAMOND performs an exact local exploration on the outstanding set of solutions that have been explored.

4.4. Dimension Reduction Ratio. Figure 10 shows the dimension reduction ratio of DIAMOND using eight classifiers, respectively. From Figure 10, it can be seen that the highest feature reduction ratio was 94.78%. Further, we compared the average feature reduction ratio of DIAMOND with BDIE and BGDE3. As shown in Figure 11, the average feature reduction ratio of BDIE, BGDE3, and DIAMOND are 91.67%, 48.72%, and 56.09%, respectively. The dimension reduction ratio indicates that applying DIAMOND in the actual SDN-IoT environment can effectively reduce the computational resource consumption of SDN controllers subject to LDDoS attacks.

4.5. Detection Performance. Table 4 shows the detection performance of different classifiers when using DIAMOND, BDIE, and BGDE3. AVG and STD represent the average detection performance and standard deviation of the algorithm, respectively. If the AVG values of the algorithms are equal, the algorithm with a smaller STD value is better. Besides, bold values in the table indicate the best value. The main purpose of Table 4 is to verify the robustness of the proposed algorithm on different classifiers. Obviously, DIAMOND performs better than BDIE and BGDE3 in most cases. It means that the features selected by the DIAMOND contain the key classification information. For instance, the average accuracy of DIAMOND is 92.30%, which is higher than 89.11% of BEID and 85.88% of BGDE3. In addition, all STD of DIAMOND is quite better than BDIE and BGDE3, which means that DIAMOND is more stable.

4.6. Detection Time. To evaluate the performance of DIAMOND in terms of detection time, the detection time is calculated in this evaluation. Table 5 describes the detection time of eight classifiers when using the BDIE, BGDE3, and DIAMOND. The detection efficiency of DIAMOND is quite better than that of BDIE and BGDE3. For instance, when using SVM, the detection time of DIAMOND is 106.72 ms, which is lower than 174.50 ms of BDIE and 164.59 ms of BGDE3. The detection time of DIAMOND is decreased by 39%, 35% compared to BDIE and BGDE3.

5. Conclusions

In this paper, we proposed DIAMOND, a structured coevolution feature optimization method for LDDoS detection in SDN-IoT. More specifically, a reachable count sorting clustering algorithm, a group structuring method, a comutation strategy, and a cocrossover strategy are proposed. The evaluation demonstrates that DIAMOND can effectively improve the detection accuracy, reduce the size of feature subsets, and achieve shortened detection time compared with the baseline methods.

Data Availability

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

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant No. 2019YFB1803500, the National Natural Science Foundation of China (NSFC) under the Grant Nos. 61902085 and 62102111, the Guizhou Provincial Science and Technology Foundation under the Grant No. [2020]1Y267, the Scientific Research Foundation for Introduced Talents of Guizhou University under the Grant No. (2019)52, and the Development Project of Ordinary University Young Scientific and

Technological Talents of Guizhou Provincial Science under the Grant No. Qian Jiao He KY Zi[2021]136.

References

[1] A. Rahman, U. Sara, D. Kundu et al., "Distb-sdoindustry: enhancing security in industry 4.0 services based on distributed blockchain through software defined networking-iot enabled architecture," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, pp. 674–681, 2020.

[2] M. A. Ferrag, L. Shu, H. Djallel, and K.-K. R. Choo, "Deep learning-based intrusion detection for distributed denial of service attack in Agriculture 4.0," *Electronics*, vol. 10, no. 11, p. 1257, 2021.

[3] Y. Cui, Q. Qian, C. Guo et al., "Towards DDoS detection mechanisms in software-defined networking," *Journal of Network and Computer Applications*, vol. 190, article 103156, 2021.

[4] N. Ravi and S. M. Shalinie, "Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3559–3570, 2020.

[5] Z. Wu, Q. Xu, J. Wang, M. Yue, and L. Liu, "Low-rate DDoS attack detection based on factorization machine in software defined network," *IEEE Access*, vol. 8, pp. 17404–17418, 2020.

[6] J. Shu, L. Zhou, W. Zhang, X. Du, and M. Guizani, "Collaborative intrusion detection for VANETs: a deep learning-based distributed SDN approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4519–4530, 2020.

[7] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Applied Soft Computing*, vol. 62, pp. 441–453, 2018.

[8] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, no. 8, pp. 371–381, 2016.

[9] M. Abdel-Basset, D. el-Shahat, I. el-henawy, V. H. C. de Albuquerque, and S. Mirjalili, "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection," *Expert Systems with Applications*, vol. 139, article 112824, 2020.

[10] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, no. 18, pp. 302–312, 2017.

[11] M. Roopak, G. Tian, and J. Chambers, "Multi-objective-based feature selection for DDoS attack detection in IoT networks," *IET Networks*, vol. 9, no. 3, pp. 120–127, 2020.

[12] M. Mafarja, A. A. Heidari, M. Habib, H. Faris, T. Thaher, and I. Aljarah, "Augmented whale feature selection for IoT attacks: structure, analysis and applications," *Future Generation Computer Systems*, vol. 112, pp. 18–40, 2020.

[13] H. Haddadpajouh, A. Mohtadi, A. Dehghantanaha, H. Karimipour, X. Lin, and K. K. R. Choo, "A multi-kernel and meta-heuristic feature selection approach for IoT malware threat hunting in the edge layer," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4540–4547, 2021.

[14] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "Rdtids: rules and decision tree-based intrusion detection system for internet-of-things networks," *Future internet*, vol. 12, no. 3, p. 44, 2020.

[15] Y. Cui and Q. Qian, "MIND: message classification based controller scheduling method for resisting DDoS attack in software-defined networking," in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*, pp. 486–490, Shanghai, China, 2020.

[16] W. Rafique, X. He, Z. Liu, Y. Sun, and W. Dou, "CFADefense: a security solution to detect and mitigate crossfire attacks in software-defined IoT-edge infrastructure," in *2019 IEEE 21st International Conference on High Performance Computing and Communications, IEEE 17th International Conference on Smart City, IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 500–509, Zhangjiajie, China, 2019.

[17] A. Sivanathan, D. Sherratt, H. H. Gharakheili et al., "Characterizing and classifying IoT traffic in smart cities and campuses," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 559–564, Atlanta, GA, USA, 2017.

[18] T. Li, H. Dong, and J. Sun, "Binary differential evolution based on individual entropy for feature subset optimization," *IEEE access*, vol. 7, pp. 24109–24121, 2019.

[19] A. A. Bidgoli, S. Rahnamayan, and H. Ebrahimpour-Komleh, "Opposition-based multi-objective binary differential evolution for multi-label feature selection," *International Conference on Evolutionary Multi-Criterion Optimization*, vol. 11411, 2019.