Network intrusion detection algorithm based on LightGBM model and improved particle swarm optimization

1st Yican Geng
School of Cyber Science and Engineering
Nanjing University of Science and Technology
Wuxi China
yicangeng@njust.edu.cn

1st Haoyang Hu
School of Cyber Science and Engineering
Nanjing University of Science and Technology
Wuxi China
hhynb@njust.edu.cn

2nd Zhaoxuan Ge
School of Cyber Science and Engineering
Nanjing University of Science and Technology
Wuxi China
GZX788@njust.edu.cn

3rd *Zhichao Lian
School of Cyber Science and Engineering
Nanjing University of Science and Technology
Wuxi China
lzcts@163.com

***Abstract*—In order to solve the problem of insufficient adaptive ability of the network intrusion detection model, the large-scale fast search capability of the particle swarm optimization (PSO) algorithm is introduced into the intrusion detection model. In order to solve the problem that PSO is easy to fall into local optimality, the genetic algorithm (GA) is introduced. An improved particle swarm optimization (GAPSO) algorithm based on genetic algorithm is proposed. This algorithm optimizes the parameters that are difficult to adjust in the lightweight gradient boosting machine (LightGBM) algorithm, so that the PSO algorithm can quickly converge while ensuring the optimization accuracy, and obtain the optimal network intrusion detection model. Experimental results show that GAPSO is more effective than the basic PSO algorithm when dealing with high-dimensional, complex structure optimization problems.**

Keywords—Supervised learning, genetic algorithm, particle swarm optimization algorithm, intrusion detection, parameter optimization

I. Introduction

In the current digital era, the importance of network intrusion detection systems (IDSs) is becoming more and more prominent as the means of cyber-attacks continue to evolve and the network environment becomes increasingly complex. These systems assume the key role of identifying and responding to potential security threats, protecting the security and stability of cyberspace.

Intrusion detection systems are divided into two main categories: signature-based detection and behavior-based detection. Signature-based detection identifies attacks by comparing the characteristics of known attacks, similar to antivirus software detecting viruses. Behavior-based detection, on the other hand, detects potential attacks by analyzing the abnormal behavior of network traffic and system activities, and this method is more suitable for identifying new types of attacks and zero-day vulnerabilities. With the development of technical methods such as statistical analysis, machine learning and deep learning, more and more intrusion detection systems are adopting these techniques.

After our research, we found that the two major challenges of the current traffic analysis technology are the detection efficiency in identifying and classifying and the accuracy of the detection of abnormal traffic, respectively. As a machine learning model-based algorithm, the LightGBM model detects potential attacks by analyzing the anomalous behavior of network traffic and system activities. The classification metrics of the network traffic classification technique based on the machine learning model LightGBM are the statistical features of network traffic rather than utilizing simple and easily disguised port numbers and feature codes, which provides high detection accuracy and efficiency for intrusion detection and therefore has a broad application prospect.

Machine learning based network traffic classification can be categorized into supervised and unsupervised classification according to different application scenarios. Supervised classification is to construct a classification model by learning a dataset with known sample categories, which can achieve high accuracy detection of known traffic types. The representative methods of supervised classification are Categorical features gradient Boosting (CatBoost) algorithm, Support Vector Machine (SVM), K-Nearest Neighbor (KNN) K-Nearest Neighbor (KNN), Decision Tree, and so on.

In this paper, we propose a network intrusion detection algorithm based on Genetic Algorithm (GA) and Genetic Algorithm Particle Swarm Optimization (GAPSO), and the proven Light Gradient Boosting Machine (LightGBM) is used to detect network intrusions. Boosting Machine (LightGBM) for parameter optimization to build network intrusion detection algorithms.

979-8-3315-3209-3/24/$31.00 ©2024 IEEE
DOI 10.1109/CyberSciTech64112.2024.00021

The main work of this paper is:

• Horizontally compare LightGBM algorithm with other intrusion detection methods, and comprehensively consider the effectiveness of multiple intrusion detection algorithms for the current KDD dataset.

• The improved particle swarm optimization algorithm (GAPSO) incorporating genetic algorithm is proposed to improve the convergence speed while guaranteeing the optimization accuracy in the large-scale search process;

• A network intrusion detection algorithm based on LightGBM, a lightweight classification algorithm, is designed, and GAPSO is used to optimize the parameters of LightGBM algorithm to achieve higher detection accuracy.

II. Related work

Genetic and particle swarm algorithms have achieved good results in areas such as parameter optimization. Literature [1] used the optimization characteristics of genetic and particle swarm algorithms to perform global optimization to improve the performance of the algorithms; Literature [2] used an improved quantum genetic algorithm to select the features, which makes full use of the parallel processing and global search capabilities of quantum genetic algorithms, improves the quality of the data classification, reduces the size of the problem, eliminates the redundant attributes, and accelerates the speed of the data processing; Literature [3] applied a quantum particle swarm algorithm is applied to the learning of BP network to improve the BP network for intrusion detection; Literature [4] proposes a network intrusion detection model based on hybrid particle swarm optimization algorithm to select features, which improves the network intrusion detection rate; Literature [5] uses a genetic algorithm to quickly select the network intrusion feature set, and then an improved ant colony algorithm is used to further optimize the feature selection process, but for a large-scale datasets, the computational complexity of the algorithm may be higher, resulting in longer running time; Literature [6] proposes a binary mothball optimization (BPMFO) algorithm fused with particle swarm optimization , but the computational complexity may be higher, and the requirement of computational resources increases accordingly; Literature [7] proposes an improved Drosophila algorithm to optimize the weighted limit learner intrusion detection algorithm, using the Literature [7] proposed an improved Drosophila algorithm to optimize the intrusion detection algorithm of weighted extreme learning machine, and used the adaptive iterative step size of Drosophila algorithm to optimize the weights and biases of the inputs of weighted extreme learning machine to avoid the algorithm from falling into the local optimum; Literature [8] combined the extreme Gradient Boosting (eXtreme Gradient Boosting) algorithm with the improved PSO algorithm for parameter searching, and solved the problem of continuous multivariate optimization; Literature [9] used the PSO algorithm to optimize XGBoost. PSO algorithm to optimize XGBoost to classify the images of new crown pneumonia, and the accuracy is improved. However, when the amount of data is large, the complexity of XGBoost is high, and the overhead in space and time is relatively large; in the literature [10], chaotic mapping and bacterial foraging algorithms are used to improve the gravitational search algorithm, and then the improved gravitational search algorithm is used to search for the optimization of the parameters of the SVM classifiers.

Aiming at the challenges of network intrusion detection with large data volume, high computational overhead and low detection accuracy, we propose a network intrusion detection model based on LightGBM, a lightweight classification algorithm. In order to quickly adjust the parameters, make the model with adaptive training ability and improve the detection effect, we combine the diversity introduction feature of genetic algorithm with particle swarm optimization algorithm, and optimize the parameters of LightGBM algorithm using genetic algorithm by making individual particles in the particle swarm search faster towards the optimal direction, so as to obtain the optimized network intrusion detection algorithm. This combination strategy not only improves the global search ability of the algorithm and avoids falling into the local optimum, but also accelerates the convergence speed of the algorithm and effectively improves the overall detection accuracy and efficiency.

III. PSO algorithm for fusion GA

A. Basic Particle Swarm Optimization Algorithm

PSO algorithm is a swarm intelligent optimization algorithm inspired by the foraging behavior of bird flocks and has the ability of global iterative optimization. PSO algorithm has the advantages of simple structure and good robustness, and is often used to solve the problem of optimal solution. used to solve the problem of optimal solution.

In a multidimensional space, the PSO algorithm assigns each particle xi in the population S a value in each dimension, and each particle has a velocity attribute to update its own value in different dimensions towards a better direction. In the iterative process, the algorithm records the optimal values of the individuals and the population as the update direction of each individual, and the algorithm flow is as follows:

Step 1: Initialize the parameters of the particle population and assign the position and velocity attributes to each particle in the population.

Step 2: Obtain the fitness value of each particle through the fitness function F, and compare the size of the fitness value to obtain the global optimal value and the individual optimal value.

Step 3: Update the velocity and position of each particle within the population through the global optimal value, which are expressed by equations (1) to (2), respectively:

$$v_{i(d+1)} = \omega v_{id} + c_1 r_1 \left(p_{id} - x_{id}\right) + c_2 r_2 \left(p_{gd} - x_{id}\right) \quad (1)$$

$$v_{i(d+1)} = x_{id} + v_{i(d+1)} \quad (2)$$

Where: ω is the inertia weight, which is used to regulate the algorithm's local search ability and global search ability; v_{id} is the velocity of particle i in the d-dimension; x_{id} is the position of particle i in the d-dimension; c_1 and c_2 are the acceleration factors, which usually take the value of 2; r_1 and r_2 are [0, 1] random numbers; p_{id} and p_{gd} denote the individual optimal value and global optimal value of the ith variable in the d-dimension, respectively; $v_{i(d+1)}$ is the velocity of particle i in the $d + 1$ dimension updated by the above variables; $x_{i(d+1)}$ is the position of particle i in the d+1 dimension updated by the historical position x_{id} and velocity $v_{i(d+1)}$.

B. Basic Genetic Algorithm

GA is a heuristic search algorithm that simulates natural selection and genetic inheritance mechanism developed by John Holland et al. in 1975.GA mimics the genetic and evolutionary process of organisms in nature, and solves optimization problems through genetic operations such as selection, crossover and mutation. It is more effective than traditional optimization methods in solving complex optimization problems, especially in searching for global optimal solutions.

In GA, individuals (called chromosomes) in a population are evaluated for their ability to adapt to the environment by means of a fitness function. The genes in the chromosomes determine the characteristics of the individuals. The basic process of genetic algorithm includes initializing the population, evaluating fitness, selection, crossover and mutation. In each generation, individuals are selected based on their fitness and new offspring are produced through crossover and mutation.

Selection: the selection operation is based on the fitness of individuals. The selection probability of an individual is usually proportional to its fitness. The process can be realized by roulette methods or tournament selection.

Crossover: the crossover operation is responsible for exchanging genes in the parent to generate new offspring. It can be represented by the following equation (3):

$$x_{t+1}^{i} = \begin{cases} c \cdot x_t^i + (1-c) \cdot x_t^i, \ if \ r_3 < CR \\ x_t^i, otherwise \end{cases} \quad (3)$$

Where and are the two parent individuals selected, is the crossover rate, which determines the extent of gene exchange, is the crossover probability, which controls the threshold for whether or not a crossover is performed, and is a random number in the interval [0, 1].

Mutation: the mutation operation introduces new genetic variation by randomly changing certain genes in a chromosome and can be described using equation (4) below:

$$x_{t+1}^{i} = \begin{cases} x_t^i + \sigma \cdot \left(x_{best} - x_t^i\right) \ if \ r_4 < MR \\ x_t^i, otherwise \end{cases} \quad (4)$$

where is the optimal individual in the current population, is the strength of the mutation, usually a smaller value, is the mutation probability, the threshold that controls whether or not the mutation is performed, and is another random number in the interval [0, 1].

Parameters in these formulas such as crossover rate c, crossover probability CR, mutation strength σ and mutation probability MR are key elements of genetic algorithms and they determine the exploration and exploitation capabilities of the algorithm. By adjusting these parameters appropriately, genetic algorithms can find a good balance between global search and local fine search.

C. GAPSO

Aiming at the limitations of GA and PSO in their respective local and global searches, this paper proposes a new hybrid optimization algorithm called GAPSO. The algorithm combines the global search capability of GA and the fast convergence of PSO to improve the search efficiency and optimization accuracy.GAPSO utilizes the genetic operation of GA to maintain the diversity of the population, and at the same time accelerates the convergence speed with the help of the speed updating strategy of PSO, which is implemented in the following steps:

Step 1: Initialize the population

Initialize the individuals in the population according to the genetic algorithm framework. Each individual consists of three parameters: maximum depth (max_depth), minimum data amount (min_data_in_leaf), and feature fraction (feature_fraction). These parameters directly affect the construction and performance of the LightGBM model.

Step 2: Evaluate the fitness

Use the fitness function evalModel to evaluate the performance of each individual. This function trains the model and calculates the accuracy on the validation dataset by configuring the parameters of the LightGBM model. Individuals with high fitness indicate that their parameter configuration solves the optimization problem more efficiently.

Step 3: Genetic manipulation

Implement the selection, crossover, and mutation operations in the genetic algorithm. Use tournament selection as a selection mechanism to ensure that superior genes are retained, hybrid crossover to increase the genetic diversity of the population, and Gaussian mutation to explore in a localized range to prevent the algorithm from falling into a local optimum.

Step 4: Particle swarm velocity update

Integrate the particle swarm velocity and position update formulas in the genetic algorithm framework. The position update of each individual is not only affected by its genetic variation, but also adjusted by mimicking the velocity update strategy in the particle swarm. This step is carried out by the following equations (5) to (6):

$$v_{i(d+1)} = \omega \cdot v_i(d) + c_1 \cdot r_1 (p_{best} - x_i) + c_2 \cdot r_2 (g_{best} - x_i) \quad (5)$$

$$x_{i(d+1)} = x_i(d) + v_i(d+1) \quad (6)$$

Step 5: Iterative update

Steps 2 to 4 are repeated until the termination conditions are met, such as the maximum number of iterations is reached or the adaptation is no longer significantly improved. After each iteration, update the individuals in the population in order to implement genetic and PSO operations in the next generation.

By introducing GAPSO, we can effectively combine the advantages of GA and PSO to realize a fast and comprehensive search. Ultimately, the algorithm finds the optimal individual with high fitness, which represents the best configuration of model parameters, thus improving the accuracy of the model on a specific dataset.

IV. GAPSO-BASED LIGHTGBM INTRUSION DETECTION ALGORITHM

A. Basic LightGBM algorithm

LightGBM is an improved implementation in the framework of Gradient Boosting Decision Tree (GBDT) algorithm, which is a fast, distributed, and high-performance GBDT framework based on Histogram decision tree algorithm. Proposed by Microsoft in 2016, it is widely used with the advantages of faster training speed and lower consumption of computational resources. Like other boosting algorithms, this algorithm boosts multiple weak classifiers into strong classifiers with strong classification effects, which can be used for anomalous traffic detection in network intrusion detection. The objective function h(x) is shown in equation (7):

$$h(x) = \sum_{i=1}^{n} L(y_i, y_i^t) + \sum_{i=1}^{t} \Omega(f_i) \quad (7)$$

Where: L is the loss function, Ω is the regular term, and yi is the predicted value. The model controls the accuracy and complexity of the model through the loss function and the regular term. The model fits the loss by negative gradient and the objective function can be obtained by Taylor expansion as shown in equation (8):

$$h_t(x) = \sum_{i=1}^{n} \left[L(y_i, y_i^t) + m_i f_i(x_i) + \frac{1}{2} m_i f_i^2(x_i) \right] + \Omega(f_t) + C \quad (8)$$

Where C is a constant term. The objective function is simplified to obtain equation (9):

$$h_t(x) = \left[m_i f_i(x_i) + \frac{1}{2} m_i f_i^2(x_i) \right] + \Omega(f_t) \quad (9)$$

B. GAPSO-LightGBM intrusion detection algorithm

In the process of machine learning model training, the LightGBM algorithm has faster training speed and higher optimization accuracy than the traditional GBDT algorithm, as well as excellent parallel processing capability due to its optimized implementation under the gradient boosting framework. However, LightGBM may still encounter overfitting and high computational complexity when dealing with complex models, especially during parameter optimization, and its performance is limited by the limitations of traditional optimization methods.

In order to overcome these challenges and to take advantage of LightGBM's capability in handling large-scale datasets, this paper proposes a new optimization method, the algorithm combining GAPSO and LightGBM, i.e., the GAPSO-LightGBM algorithm. The algorithm first preprocesses the network intrusion dataset to convert the raw data into numerical data that can be recognized by the model. The preprocessing includes encoding categorical variables and normalizing the numerical data to ensure that the data is balanced and representative during the training process. The model flow is shown in Fig 1.

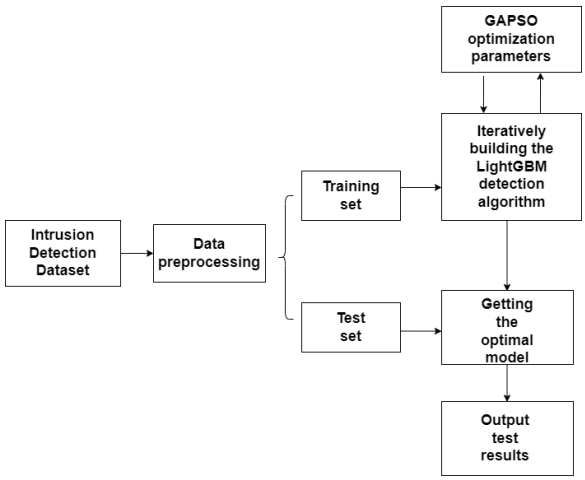

Fig. 1. GAPSO-LightTGBM algorithm model

Taking advantage of GAPSO, the algorithm can quickly locate the potential regions of optimal parameter configurations globally, and through the speed and position update mechanism of particle swarm, it can quickly and finely search in these regions to find the optimal model parameters.GAPSO combines the global searching ability of genetic algorithm and the fast convergence property of particle swarm optimization, which makes the whole searching process both efficient and accurate.

In the GAPSO-LightGBM algorithm, the crossover and mutation operations in the genetic algorithm will be used to maintain the genetic diversity of the population, while

the speed and position update formulas in the particle swarm optimization algorithm will be used to quickly adjust the search direction and step size. This combination not only enhances the algorithm's ability to search in high-dimensional parameter space, but also improves its adaptability and stability in dynamically changing environments.

The LightGBM algorithm contains many parameters, and different values of the parameters will have a certain impact on the classification results. The parameters of LightGBM are optimized using SSAPSO to obtain better detection accuracy and detection speed. The parameters set to be optimized are shown in Table I.

TABLE I. LIGHTGBM OPTIMIZATION PARAMETERS

Parameter	Optimization range
max_depth	[3，10]
min_data_in_leaf	[20，40]
feature_fraction	[0.5，0.9]

Where: max_depth parameter is used to limit the depth of the tree, min_data_in_leaf parameter is used to deal with the overfitting problem of the leaf_wise tree, and feature_fraction parameter is set to use feature sampling to speed up the training speed.

The GAPSO-LightGBM algorithm is divided into four main steps:

Step 1 Data preprocessing:

1) Feature processing: extract features and target variables from the dataset. The features contained in the dataset are mainly numerical, but may also include some categorized data, such as protocol types like TCP, UDP and ICMP. All categorical data need to be converted to numerical identifiers for algorithmic processing.

2) Standardisation: the data are normalized to eliminate the influence between different quantitative features. The standardization formula is shown in equation (10):

$$x^* = \frac{x - \mu}{\sigma} \tag{10}$$

where x denotes the original feature value, μ and σ are the mean and standard deviation of the features, respectively.

3) Normalization: the data are further normalized to scale the eigenvalues into the interval [0, 1]. The normalization formula is shown in equation (11):

$$x' = \frac{x^* - \min(x^*)}{\max(x^*) - \min(x^*)} \tag{11}$$

where denotes the normalized data, and and are the minimum and maximum values of the data, respectively.

4) Divide the data into training set and test set.

Step 2 Model training and parameter initialization:

Substitute the processed training set data into the LightGBM algorithm, initialize the algorithm parameters, and optimize the parameters of the LightGBM model by combining the selection, crossover, and mutation operations of the genetic algorithm and the speed update mechanism of the particle swarm optimization. The set parameters include max_depth, min_data_in_leaf, and feature_fraction.

Step 3 Optimize the iterative process:

1) Define the fitness function: define the fitness function by the prediction accuracy of the LightGBM model, and use the test set to calculate the accuracy.

2) Initialize particles: create a particle swarm, each particle represents a set of parameters of LightGBM.

3) Iterative optimization: Optimize the parameters by the particle swarm algorithm within a set number of iterations. In each generation, the fitness of the particles is evaluated, the optimal particle is selected based on the fitness value, and the position and velocity of the particle swarm are updated.

The algorithm is evaluated using a fitness test dataset and the fitness function is calculated based on the accuracy of the LightGBM model. The combination of the genetic algorithm and the particle swarm continuously iteratively updates the individual positions and velocities to find the optimal combination of parameters. In each iteration, the fitness of all individuals in the current population is evaluated and the individual with the highest fitness is selected as the parent into the next generation.

Step 4 Model validation and parameter adjustment:

The LightGBM classifier is set up using the optimized parameters and trained using the training set data. Then evaluate the classification effect of the model with the test set data to verify the improvement of the model performance. Based on the test results, the classifier parameters are tuned to achieve optimal performance.

In each step, the GAPSO-LightGBM algorithm aims to improve the performance of LightGBM through intelligent optimization techniques so as to increase the prediction accuracy of the model while maintaining high efficiency.

V. EXPERIMENTS AND RESULT ANALYSIS

A. Intrusion detection dataset and environment

1) Intrusion detection dataset

In order to verify the detection effect of the model, this paper chooses the classical intrusion detection dataset KDDCUP99 for experiments, and the information of the training set and test set is shown in Table II. The dataset has 41-dimensional features, which are classified into four attack types and one normal type, which are Denial of Service (DoS) attack, unauthorized remote access (Remote-to-Login, R2L) attack, unauthorized local access (User-to-Root, U2R) attack, and Probeing. PROBE) attacks, and a type of Normal traffic (Normal). The types

of attacks are described in detail as follows:

1) DOS attack: the attacker occupies the computing or memory resources needed to process valid requests, making the system unable to answer normal user requests.

2) R2L attack: The attacker remotely accesses the system without authorization and uses a valid user account.

3) U2R attack: The attacker remotely accesses the network and illegally obtains super-user privileges to use a valid user account.

4) PROBE attack: the attacker tries to obtain computer network related information.

Table II gives the details of KDDCUP99 dataset. In this paper, 5000 data are randomly selected for experiments, of which 3500 are training set and 1500 are testing set.

TABLE II. INFORMATION IN KDDCUP99 TRAINING AND TEST DATASET

Category	Sample size	
	Training set	Test set
Normal	97278	979
DOS	391458	3960
R2L	1126	61
U2R	52	32
PROBE	4107	48

2) Experimental environment

The experimental hardware environment uses AMD Ryzen 9 5900HS CPU + GeForce GTX 3060 + 16GB RAM; the software environment uses Anaconda 2. 6. 0 + Python 3. 10. 5.

B. Analysis of Experimental Results

1) Demonstration of GAPSO Optimized LightGBM Detection Accuracy

In the experiments of this paper, according to the calculation of the model detection sample categories and the actual categories of the samples, the accuracy (Accuracy, Acc), recall (Recall), precision (Precision, Pre) and F1 index (F1_score) are used as the evaluation indexes for detecting the effects of various types of attacks. The formulas for each index are as follows, respectively:

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{12}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{13}$$

$$\mathrm{Pre} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{14}$$

$$\mathrm{F1\_score} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{15}$$

Where: the first letter in TP, TN, FP, FN indicates whether the classifier recognition result is correct or not, correct is indicated by the initial letter T of True, and error is indicated by the initial letter F of False; the second letter indicates the determination result of the classifier; P indicates that the classifier determines a positive sample (Positive Sample), N indicates that the classifier determines a negative sample (Negative Sample), where the attack class samples are positive samples and the normal samples are negative samples; TP (True-Positive) denotes the number of attack class samples correctly recognized by the classifier, TN (True-Negative) denotes the number of normal samples correctly recognized by the classifier, and FN (False-Negative) denotes the number of normal samples correctly recognized by the classifier. the number of attack class samples that the classifier detects as normal samples, and FP (False-Positive) denotes the number of normal samples that the classifier detects as attack class samples.

We upgraded from the most basic LightGBM to PSO-LightGBM and finally implemented GAPSO-LightGBM. Table III gives the running results of several classification algorithms using the KDDCUP99 dataset. From this table, it can be seen that the GAPSO-LightGBM algorithm has higher accuracy, recall, precision and F1 index than the other classification algorithms. The GAPSO-LightGBM algorithm has improved the accuracy, recall, precision and F1 index by 13.40%, 1.42%, 16.18%, and 9.46% respectively compared to that derived by the CatBoost algorithm, and compared to the LightGBM algorithm yields 2.68%, 0.72%, 1.04%, and 0.98% improvement in accuracy, recall, precision, and F1 index, respectively, and a high recall results in a low underreporting rate. It can be seen that the GAPSO-LightGBM algorithm has better feature representation for attack samples and can classify the features more accurately, which is conducive to more efficient and accurate discrimination for intrusion detection.

TABLE III. CLASSIFICATION ALGORITHM DETECTION ACCURACY UNIT：%

algorithm	Acc	Recall	Pre	F1_score
KNN	92.10	91.38	92.08	90.04
CatBoost	86.22	96.24	81.92	88.50
LightGBM	96.94	96.94	97.06	96.98
PSO-LightGBM	97.70	97.14	97.32	97.66
GAPSO-LightGBM	99.62	97.66	98.10	97.96

Then, for the five types of data (one Normal and four Abnormal) in the test dataset of this paper's research, GAPSO-LightGBM algorithm, PSO-LightGBM algorithm, basic LightGBM algorithm, CatBoost algorithm and KNN algorithm are used for the comparison of network intrusion detection, and the results of the detection are shown in Table IV. As shown in Table IV, the accuracy of GAPSO-LightGBM algorithm for Normal detection is up to 99.60%, for DOS is up to 99.70%, for R2L is up to 98.36%, for U2R is up to 96.88%, and for PROBE is up to 97.92%.GAPSO-LightGBM algorithm for Normal, DOS, R2L, U2R, and

PROBE in the dataset is improved by 0.20%, 0.30%, 0.36%, 1.68%, and 3.92%, respectively, compared to the CatBoost algorithm, and the LightGBM algorithm is improved by 2.66%, 2.68%, 2.68%, and 3.28%, respectively, compared to the LightGBM algorithm for Normal, DOS, and R2L in the dataset. and 3.28%, respectively. And the lightness of LightGBM algorithm has been proved in practical applications. Therefore, compared with other algorithms, GAPSO-LightGBM is more suitable for intrusion detection.

TABLE IV. COMPARISON OF NETWORK INTRUSION DETECTION ACCURACY AMONG CLASSIFICATION ALGORITHMS UNIT：%

algorithm	Normal	DOS	R2L	U2R	PROBE
KNN	97.80	93.60	91.60	96.82	88.40
CatBoost	99.40	99.40	98.00	95.20	94.00
LightGBM	96.94	97.02	95.08	96.88	97.92
PSO-LightGBM	97.55	97.78	96.72	96.88	100.00
GAPSO-LightGBM	99.60	99.70	98.36	96.88	97.92

2) Performance testing of GAPSO

In order to verify the optimization performance of GAPSO, this paper uses Griewank function, Rosenbrock function and Styblinski-Tang function to test the performance of GAPSO. Among them, the Griewank function is a classical multi-peak function in the field of optimization; while the Rosenbrock function and the Styblinski-Tang function, as representatives of more complex and higher dimensional optimization problems, are used to test the algorithm's ability to search globally and avoid falling into local optima.

The functional expressions of the three functions is shown in Table V, and Table VI gives their variables such as the range of values of the variables, the number of dimensions and the optimal value of the variables.

TABLE V. FUNCTIONAL EXPRESSIONS

Function	Expression
Griewank	$f(x)=1+\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$
Rosenbrock	$f(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2]$
Styblinski-Tang	$f(x)=\frac{1}{2}\sum_{i=1}^{n}(x_i^4-16x_i^2+5x_i)$

TABLE VI. TEST FUNCTIONS AND PARAMETER INFORMATION

Function	Definition	Dimension	Optimal
	Domain		Value
Griewank	[−600,600]	2	0
Rosenbrock	[−5,10]	10	0
Styblinski-Tang	[−5,5]	10	-391

The GAPSO algorithm and the ordinary PSO algorithm are tested for optimization on the above three functions, the number of iterations is set to 1000, 100 particles are used, and the test results are shown in Table VII.

TABLE VII. BEST POSITION AND OPTIMAL VALUE

Subject		Optimal position	Optimal value
Griewank	GAPSO	[2.96,4.47]	0.01839
	PSO	[13.28,−1.73]	0.04234
Rosenbrock	GAPSO	[−5.000,0.806,0.348, 0.141,0.083,−0.146, −0.008,0.030,0.026, −0.002]	6.535
	PSO	[0.680,0.450,0.207,0 .015,0.023,0.011,0.0 32,0.020,0.019,−0.0 22]	7.182
Styblinski-Tang	GAPSO	[−2.969,−2.877,−2.8 92,−2.898,−2.910,−2 .893,2.746,−2.889,− 2.894,−2.908]	−377.52
	PSO	[−2.904,−2.904,−2.9 04,−2.904,2.747,2.7 47,2.747,−2.904,−2. 904,−2.904]	−349.25

In the test of Griewank function, we can see that the improved genetic particle swarm optimization algorithm shows better performance than the traditional particle swarm optimization algorithm. This indicates that the genetic operation of GAPSO is more effective in exploring the solution space and avoiding falling into local optimums on problems such as the Griewank function, which has a large number of local optimal solutions.

Further, on the complex 10-dimensional Styblinski-Tang function, GAPSO achieves lower function values, showing its advantage in dealing with high-dimensional problems.PSO, although close to the optimal solution in some dimensions, does not have better overall function values than GAPSO, which is due to the fact that PSO will fall into local optima faster or fail to explore other regions of the search space effectively. other regions of the search space.

This result supports that GAPSO is more effective than traditional PSO when dealing with high-dimensional and structurally complex optimization problems. The potential advantages of genetic manipulation in exploring globally optimal solutions are demonstrated, especially in complex

problems that require jumping from multiple local optima.

3) Demonstration of GAPSO optimization effect for LightGBM

In the process of optimizing LightGBM parameters, we compare the optimization effect of basic PSO algorithm and GAPSO algorithm for Light GBM. We introduce the crossover and mutation operations in genetic algorithm on the traditional PSO. We specify the values of crossover rate, mutation rate, and adaptive parameters as shown in the Table VIII:

TABLE VIII. PARAMETERS USED BY GENETIC ALGORITHMS

Variable	Value
Crossover rate	0.5
Variability rate	0.1
Adaptive parameter	0.5

The convergence speed and accuracy of PSO and GAPSO for optimizing Light GBM algorithm are judged through 50 iterations and the results are shown in Fig 2:

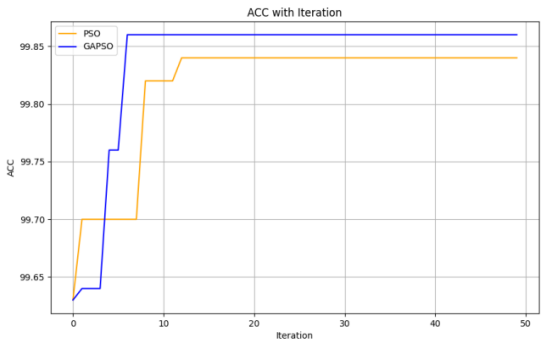

Fig. 2. Convergence curve comparison

lightweight characteristics are suitable for classification applications of intrusion detection. In future research, deep learning related algorithms can be used to further mine data relationships and build more intelligent models.

As can be seen from Fig 2, in terms of convergence speed, GAPSO completes the convergence in about 10 times, compared with PSO algorithm, the convergence speed of GAPSO is faster and the optimization accuracy is higher, and the accuracy rate reaches more than 99.85%. Therefore, the improved GAPSO is better than the basic PSO algorithm in the application of parameter optimization of LightGBM.

VI. CONCLUSION

In order to solve the problem that it is difficult to quickly adjust the parameters of the LightGBM algorithm training model in network intrusion detection, this paper uses the large-scale fast search capability in GA to improve the PSO algorithm, and uses GAPSO to optimize the parameters of the LightGBM algorithm to establish GAPSO-LightGBM. By comparison, the detection accuracy of GAPSO-LightGBM is higher than other algorithms, and its lightweight characteristics are suitable for classification applications of intrusion detection. In future research, deep learning related algorithms can be used to further mine data relationships and build more intelligent models.

ACKNOWLEDGEMENT

This work was supported by the Key R&D Program of Jiangsu(BE2022081).

REFERENCES

[1] Zheng Hongying, Ni Lin, Hou Meiju, Wang Yu. "Comparative Analysis of Intrusion Detection Based on Genetic Evolution and Particle Swarm Optimization Algorithms." Journal of Computer Applications, vol. 2010, no. 6, 2010, pp. 1486-1488.

[2] Liu Jun, Di Wenhui. "Intrusion Detection Feature Selection Based on Improved Quantum Genetic Algorithm." Computer Measurement & Control, vol. 2011, no. 4, 2011, pp. 813-815.

[3] Yuan Hao. "BP Network for Intrusion Detection Based on Quantum Particle Swarm." Sensors and Microsystems, vol. 2010, no. 2, 2010, pp. 108-110.

[4] Yuan, Kaiyin, and Fei Lan. "Detection of Network Intrusion Based on Hybrid Particle Swarm Optimization Algorithm Selecting Features." (No publication information provided).

[5] Yuan Qinqin, Lü Lintao. "Network Intrusion Detection Based on Improved Ant Colony and Genetic Algorithm Combination." Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2017, vol. 29, no. 1, p. 84. DOI: 10.3979/j.issn.1673-825X.2017.01.013.

[6] Xu Hui, Fang Ce, Liu Xiang, Ye Zhiwei. "Application of Improved Moth Flame Optimization Algorithm in Network Intrusion Detection System." Journal of Computer Applications, 2018, no. 11, pp. 3231-3235.

[7] Dang Jianwu, Tan Ling. "An Intrusion Detection Algorithm Based on Improved Fruit Fly Optimization Algorithm and Weighted Extreme Learning Machine." Journal of System Simulation, 2021, vol. 33, no. 2, pp. 331-338.

[8] Song K, Yan F, Ding T, et al. "A Steel Property Optimization Model Based on the XGBoost Algorithm and Improved PSO." Computational Materials Science, 2020, vol. 174, article no. 109472.

[9] Dias Júnior D A, Da Cruz L B, Diniz J O B, et al. "Automatic Method for Classifying COVID-19 Patients Based on Chest X-ray Images, Using Deep Features and PSO-optimized XGBoost." Expert Systems with Applications, 2021, vol. 183, article no. 115452.

[10] Zhang Xiaoyu, Wang Huazhong. "Improved Gravitational Search Algorithm for Industrial Control System Intrusion Detection." Computer Engineering and Design, 2020, vol. 41, no. 1, pp. 33-39.