An Industrial Network Intrusion Detection Algorithm Based on IGWO-GRU

wei yang
Northeastern University
yao shan (1810596@stu.neu.edu.cn)
Northeastern University
jiaxuan wang
Northeastern University
yu yao
Northeastern University

Research Article

Keywords: industrial control network traffic, intrusion detection, GRU, Grey Wolf Optimizer

Posted Date: September 1st, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3302312/v1

Additional Declarations: No competing interests reported.

An Industrial Network Intrusion Detection Algorithm Based on IGWO-GRU

Wei Yang [1] ; Yao Shan [2] ; Jiaxuan Wang [3]; Yu Yao [4]

[1] College of Software Engineering, Northeastern University, Shenyang 110179, China.

[2] College of Computer Science and Engineering, Northeastern University, Shenyang 110179, China.

[3] College of Computer Science and Engineering, Northeastern University, Shenyang 110179, China.

[4] College of Computer Science and Engineering, Northeastern University, Shenyang 110179, China.

Word Count: 7419 words (main body) with 11 figures and 7 tables

Article submitted to: Cluster Computing-The Journal of Networks Software Tools and Applications

Version: 1

Corresponding author: Yao Shan, College of Computer Science and Engineering, Northeastern University, No.195, Chuangxin Road, Hunnan District, Shenyang, 110179, China.

Tel.: +8615734009265, **E-mail:** 1810596@stu.neu.edu.cn

Acknowledgments

The work was supported by a National Key Research and Development Program of China (Grant No: 2021YFB3101700).

Abstract
Nowadays, the industrial control system has become open and interconnected, and informatization also increases the risk of network attacks and damage due to frequent intrusion. Research on industrial intrusion detection is ongoing, but many current methods need to consider the characteristics of industrial control flow. Therefore, this paper proposes an industrial network intrusion detection algorithm based on IGWO-GRU: starting from the timing of industrial control network traffic, select the simple architecture of the gated recurrent unit (GRU) as the network model; in view of the problem of the number of network parameters such as neurons and the learning rate, the Grey Wolf Optimizer (GWO) is integrated with conducting autonomous learning to find the optimal parameters of the model and solve the problem of slow convergence rate caused by a large amount of data volume of the industrial control network traffic. However, due to the slow convergence speed and low optimization accuracy of the GWO algorithm and data imbalance, this paper improves an improved grey Wolf optimization algorithm (IGWO) by improving the nonlinear convergence factor and weight adjustment strategy to increase the convergence rate of the algorithm further and avoid falling into the local optimal solution. With the data set of the natural gas pipeline control system, the intrusion detection system is simulated for classifying abnormal flow attacks. The experimental results show that the IGWO-GRU algorithm has obvious advantages in accuracy, false alarm rate, and false report rate, which improves the safety protection ability of industrial control systems.

Keywords: industrial control network traffic; intrusion detection; GRU; Grey Wolf Optimizer

1 Introduction

1.1 The research background

Industrial Control System (ICS) is widely used in many modern industries, such as sewage treatment, power generation, water conservancy, petrochemical, etc. With the development of computer technology, new technologies and new tools for implementing various attacks of industrial control systems emerge endlessly, and the research on abnormal flow detection of industrial control systems has become a hot topic in the field of industrial control security.

So far, domestic and foreign scholars have put forward many different types of network traffic anomaly detection methods. The anomaly detection method of network traffic can be divided into four categories [1]: classification, information theory, clustering, and statistics. Among them, the anomaly detection method based on classification for network traffic is the most important, with common supervised machine learning models such as decision tree (DT) and support vector machine (SVM). With the development of deep learning ideas, a large number of neural network models use network traffic classification-based methods for detection, such as Back Propagation (BP), Long Short Term Memory (LSTM), etc. First, the specific number and category of classification are determined, and then the model is trained with the labeled network traffic data, and then the new traffic is classified, and the classification result is the detection result.

In order to further improve the efficiency of abnormal traffic detection, the group intelligence optimization algorithm also began to combine with the commonly used algorithms. Swarm Intelligence Algorithm is to construct the stochastic optimization algorithm by simulating the group behavior of natural organisms. Now, it has become a hot topic of interdisciplinary topics such as artificial intelligence. Compared with the traditional calculation method, there are more outstanding advantages of an intelligent optimization algorithm, such as no centralized control, multiple agent mechanism, simple structure, and implicit parallelism. These advantages promote its development in application optimization technology and make use of group advantages to provide new ideas for finding solutions to complex problems in the absence of centralized control and no global model.

1.2 The research work

This paper uses the excellent feature learning ability of deep learning to carry out network traffic classification research. In the construction process of a neural network, the improved grey Wolf optimization algorithm is combined to find the optimal parameters of the model, which improves the convergence speed of the algorithm and the performance of anomaly detection. The details are as follows:

1. For the late iteration of the traditional grey Wolf optimization algorithm, poor population diversity, and slow convergence rate, this paper proposes an adaptive position adjustment strategy based on weight to solve the optimal, excellent, and suboptimal solution and average fitness by establishing a probability distribution

proportional to the fitness. The way of location update is selected by the relationship of individual fitness to average fitness.

2. The Grey Wolf algorithm is easy to fall into the local optimal solution. Therefore, the linear decay of the original parameter α is modified to make a nonlinear decay to increase the proportion of the global search times. The more global search times, the stronger the global search ability of the algorithm, and the less likely it is to fall into the local optimal solution.

3. A new anomalous flow detection algorithm is proposed. For the industrial controlled flow, the anomalous detection model combining the gating cycle unit and the adaptive grey Wolf optimization algorithm is used to classify the anomalous flow. Using the improved grey Wolf optimization algorithm to optimize the neural network parameters and the network structure can improve the detection speed and accuracy.

1.3 The organizational structure

The subsequent sections of this paper are organized as follows. In Section 2, this paper introduces related work. Section 3 introduces the improvement ideas of the optimization algorithm and the algorithm applied to intrusion detection. Section 4 presents the experimental setting. Section 5 evaluates the improvement of the optimization algorithm and its application to the intrusion detection algorithm. Section 6 concludes this paper.

2. Related Work

2.1. Intrusion Detection Algorithm

With the development of machine learning and deep learning technology, a large number of relevant algorithms are used in abnormal traffic detection in industrial control systems. Lee J H et al. used the ID3 algorithm to generate a decision tree to detect [2] abnormal traffic. In the experiment, the ID3 decision tree generation algorithm was used to establish a decision tree for each type of attack behavior in the DARPA dataset and achieved good accuracy in the detection of DoS, R2L, U2R, and scan attacks. However, decision tree-based models are highly prone to the problem of overfitting during training. Shang et al. proposed a clustering algorithm and support vector machine combined abnormal traffic detection method [3]; the method combines supervised support vector machine algorithm (SVM) and unsupervised fuzzy C mean clustering algorithm (FCM), by calculating the distance between industrial control network traffic data and cluster center, part of the data will meet the threshold condition through the support vector machine for further classification. Experimental results show that this method can effectively reduce the training time and improve the classification accuracy rather than the conventional anomaly flow detection method without knowing the classification label of the data prior.

Due to the time series properties in the network traffic data, the recurrent neural network (RNN) is a good choice. Fang et al. [4] proposed an intrusion detection model based on hybrid CNN and RNN models that can accurately identify the types of network traffic and solve high-level persistent threats in power information networks. Goh et al. proposed an unsupervised anomaly detection model based on RNN [5], which can detect the vast majority of intrusion behavior designed by experimenters with a very low false positives rate, applied to the industrial water processing dataset. The algorithm is very competitive, but a large number of parameters increase the training time and the use of resources. Yu et al. [6] raised the problem of needing more ability to improve RNN temporal memory based on LSTM. Xu et al. [7] introduced a novel intrusion detection system consisting of an RNN with a GRU to simplify the memory cell structure of the LSTM and reduce the computation time of the algorithm while maintaining classification accuracy.

The above methods use neural networks to train the models, but these processes easily fall into the local optimal solution during the model training. This study combines optimization algorithms to help train better models.

2.2 Intelligent optimization algorithm

The group intelligent optimization algorithm is to transform the engineering optimization problem into a function optimization problem, establishes the objective function, and finds the optimal solution of the objective function. Common algorithms include the ant colony algorithm (ACO), the particle swarm algorithm (PSO), and other methods inspired by biological groups, such as the Grey Wolf Optimizer (GWO).

2.2.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [8], which was proposed by Eberhart and Kennedy in 1995, is a population-based stochastic optimization technique inspired by the clustering behavior of insects, herds, birds, and fish groups. The core idea is to constantly adjust the position of the particle itself to approach it to the direction of the

optimal solution through the mutual cooperation between the particles and the sharing of information.

Each particle in a particle population represents a possible solution to a problem, realizing the intelligence of the problem solution through information interaction through the simple behavior of individual particles. Since the PSO is simple to operate and has a fast convergence speed, it has been widely used in many fields, such as function optimization, image processing, and geodetic measurement. With the expansion of the application scope, the PSO algorithm has some problems, such as early convergence, dimension disaster, and easily falling into the local extrema.

2.2.2 Grey Wolf Optimization

The Grey Wolf Optimization algorithm (GWO) [9], proposed by Mirjalili et al. in 2014, is a new group intelligent optimization algorithm inspired by the social hierarchy mechanism of grey Wolf populations and the process of hunting their prey in nature. It has the characteristics of strong convergence performance, few parameters, simple principle, and ease of realization, and it is widely used in neural network parameter optimization.

The Grey Wolf is a social animal. The social hierarchy within the grey Wolf population is shown in Fig. 1. The first layer of the pyramid is α, the leader of the entire grey Wolf population, also known as the head wolf. The second layer of the pyramid is the β, which obeys the commands of the α, and so on.

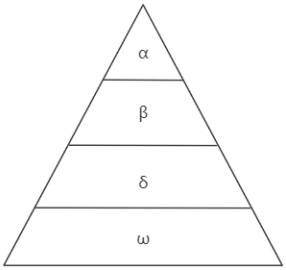

Fig. 1 Social hierarchy of grey Wolf

In the simulation, the three wolves with the best fitness are selected and defined as α, β, and δ, which will guide the other individuals in the grey Wolf population to search in the direction of the optimal solution. Other individuals in the grey wolf population, where the candidate solution is defined as the ω, are position-updated in the direction guided by the α, β, and δ.

2.3 Application of optimization algorithm

To further improve the efficiency of anomaly flow detection, the optimization algorithm begins to combine with commonly used anomaly detection algorithms. For the security of the Internet of Things, Yang et al. proposed an LM-BP neural network model [10] for intrusion detection systems, using its fast optimization speed, generalization ability of the LM algorithm to optimize the weight and threshold of the traditional BP neural network; compared with the traditional BP neural network model, the model has higher detection rate in DOS, R2L, U2L and detection attack and lower false alarm rate.

Different from traditional optimization algorithms, group intelligent optimization algorithm is a probabilistic search algorithm with strong self-study habits, self-organization and other intelligent characteristics, simple structure, fast convergence, good global convergence, and other advantages. Shang et al. proposed an abnormal flow detection algorithm based on single class support vector machine [11] and designed the particle swarm optimization algorithm to optimize the model parameters; it was not only efficient and reliable but also met the requirements of real-time performance; however, it had only the basic vector processing of Modbus functional code and did not model all the obtained data, which affected the reliability of intrusion detection to a certain extent. Chen et al. [12] propose a network intrusion detection method based on a one-dimensional convolutional neural network and the Grey Wolf optimization algorithm. A one-dimensional convolutional neural network is used to extract high-level features from the intrusion detection data. Then a support vector machine is used to classify the extracted high-level features, where the parameters of the support vector are optimized using the grey Wolf optimization algorithm.

The above research status shows that the computing model combining group intelligent optimization algorithm and intrusion detection algorithm has high research value in the field of industrial control system security. Therefore, starting from the characteristics of industrial control network traffic, this paper proposes an industrial network intrusion detection algorithm based on IGWO-GRU, selects GRU as the basic model and uses the improved grey Wolf optimization algorithm to continuously adjust the model weight to improve the convergence speed of the algorithm and avoid falling into the local optimal solution.

3 Methodology

3.1 The method of GWO

Step 1: Initialize the grey wolf population and calculate the parameters.

In Eqs. (1) to (3), t represents the current number of iterations, $\vec{A}$ and $\vec{C}$ represent the coefficient vector, and $\vec{a}$ is the convergence factor. The convergence factor changes linearly from 2 to 0 as the number of iterations rises. The value of $\vec{A}$ controls whether the grey wolf individual is searching in the neighborhood of the current optimal solution or in the global range. The change of value $\vec{A}$ is controlled by the change of $\vec{a}$, which is linearly reduced, so $|\vec{A}| > 1$ and $|\vec{A}| < 1$ each occupy half of the iterations. The module of $\vec{r}_1$ and $\vec{r}_2$ is the random number between [0,1].

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{1}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{2}$$

$$a = 2 - \frac{2t}{T} \tag{3}$$

Step 2: Update the grey Wolf position. If the maximum number of iterations is not reached, repeat steps 1-2 until the maximum number of iterations is reached.

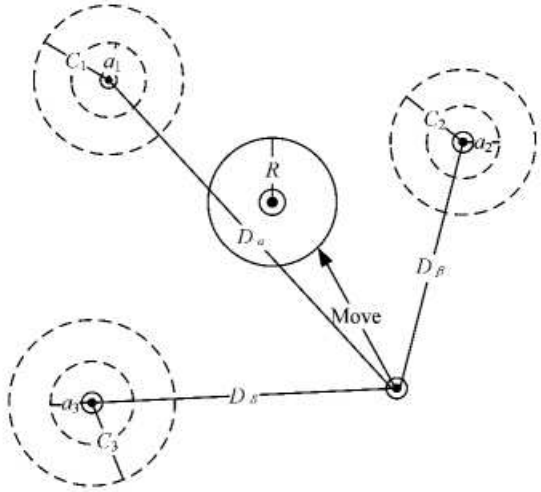

Fig. 2 grey wolf location Update Diagram

The three grey wolves with the best adaptability are selected as α, β, and δ, and infer the exact location of the prey by α, β, and δ. Moreover, update the location information of the remaining individuals in the population to complete the approximation of the prey. The location update method of individual grey wolves is shown in Fig. 2.

$$\begin{cases} \vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - X| \\ \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - X| \\ \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - X| \end{cases} \tag{4}$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - A_1 \cdot (\vec{D}_\alpha) \\ \vec{X}_2 = \vec{X}_\beta - A_2 \cdot (\vec{D}_\beta) \\ \vec{X}_3 = \vec{X}_\delta - A_3 \cdot (\vec{D}_\delta) \end{cases} \tag{5}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{6}$$

In Eq. (4), $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ represent the distance from another individual in the grey wolf population to α, β, and δ. $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ represent the current location information of α, β, and δ. $\vec{C}_1$, $\vec{C}_2$, and $\vec{C}_3$ are random vectors, and $\vec{X}$ represents the current location of the grey wolf individuals. Eq. (5) describes the direction and step length of the individual ω moving towards α, β, and δ. Eq. (6) represents the final position of the individual ω.

3.2 The method of IGWO

A thorough analysis of the principle of the Grey Wolf optimization algorithm shows that the algorithm has the characteristics of fast convergence speed and strong global search ability in the early stage of iteration. However, the value of the parameter a in the Grey Wolf optimization algorithm changes linearly, resulting in half of the iterations being used for the local optimal search, which makes the global search of the algorithm weak and easy to fall into the local optimum. And the Grey Wolf optimization algorithm only uses α, β, and δ to update the location of the population. In the process of location update without considering the optimal, excellent, and suboptimal solution, with the increase of iterations, the late population diversity, the convergence of the algorithm will slow, also more easily into local optimal solution, for the optimization of multiple peak function cannot find the global optimal solution. Considering the above issues, this paper proposes an improved grey Wolf optimization algorithm (IGWO).

3.2.1 Nonlinear convergence factor

Finding the global minimum is a common and challenging task in all minimization algorithms. In group search-based optimization algorithms, ideal solutions converging to the global minimum can be divided into two fundamental stages. In the early stages of optimization, individuals should be scattered as much throughout the search space as possible. In other words, they should try to search in the entire search space rather than the neighboring rows around the local minimum, which can help jump out of the local optimal solution; in the later stages of optimization, individuals must approach the [13] to the global

minimum using the information collected during the global search.

In the grey Wolf optimization algorithm, the two stages of global search and local search are controlled by the value of $|\vec{A}|$, when $|\vec{A}| < 1$ individuals in the neighborhood of the current optimal solution, namely, local search, when $|\vec{A}| > 1$, the individual in the search far from the current optimal solution, the global search. Generally speaking, the higher the exploration of the search space, the lower the possibility of falling into the local optimal solution and thus stagnation, and excessive local search will make the algorithm more likely to fall into the local optimal solution. Therefore, the two phases of global search and local search can be balanced by fine-tuning the values of the parameters $\vec{a}$ and $\vec{A}$, thus accelerating the search speed for the global minimum value. As shown in Eq. (7), if $f_i > f_{avg}$, the original formula is used, and if $f_i \leq f_{avg}$, the original linear change changes to a nonlinear change.

$$a = \begin{cases} 2 - \dfrac{2t^2}{T^2}, f_i \leq f_{avg} \\ 2 - \dfrac{2t}{T}, f_i > f_{avg} \end{cases} \tag{7}$$

Where T is the maximum number of iterations, and t is the current number of iterations. Since the original formula, the value of a is linearly reduced. As shown in Fig. 3, the number of iterations for both the global search and the local search is 50%, which makes the GWO algorithm lacks the searchability of the global optimal solution, and it is easy to fall into the local optimal solution.

In the improved algorithm, 70% of the iterations were used for the global search, and 30% of the iterations were used for the local element searches, as shown in b in Fig. 3. The improved algorithm is used for the global search more frequently, increasing the ability of the global search, and thus it is less likely to fall into the local optimal solution.

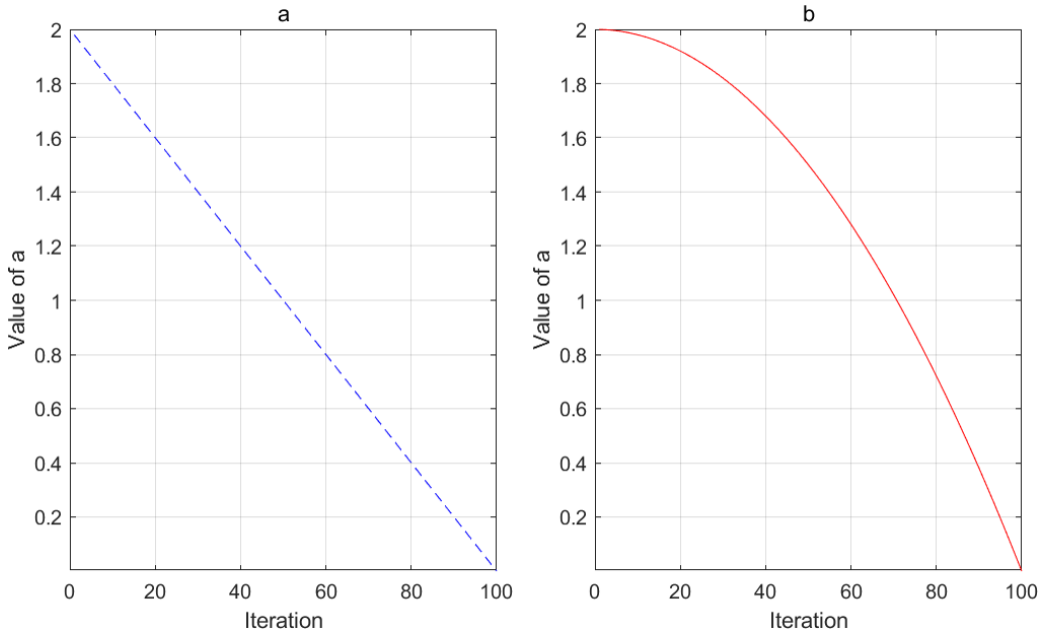

Fig. 3 Variation curve of parameter a

3.2.2 Adjustment strategy for the weights

In the grey Wolf optimization algorithm, the fitness of the individual grey Wolf can describe the quality of the current position. The α, β, and δ stand for the optimal, excellent, and suboptimal solution of the Grey Wolf population, which play an equally important role as the Grey Wolf population approaches the target. But their influence is not reflected in Eq. (6), so when searching within a neighborhood range with low fitness, its convergence efficiency is low and easily into the local optimal solution. In the Grey Wolf optimization algorithm, all the Grey Wolf location updates are moved in the direction where α is located, resulting in the reduction of population diversity in the later stage and making it easy to fall into the local optima. To further increase the convergence rate of the algorithm, While avoiding the problem of poor population diversity in later stages. This paper proposes a new method to update the location of the grey Wolf population: firstly, calculate the average fitness f_{avg} of the grey Wolf population. Then compare the fitness f_i of the current grey wolf individual with the mean fitness f_{avg}. If $f_i >$

f_{avg}, update the location information using the location update strategy of the original Grey Wolf optimization algorithm; if $f_i \le f_{avg}$, update the Grey Wolf location with a new location update method. The improved location update is described in Eqs. (8) to (9):

$$\vec{X}(t+1) = \begin{cases} \dfrac{(\frac{1}{f_\alpha})\vec{X}_1 + (\frac{1}{f_\beta})\vec{X}_2 + (\frac{1}{f_\delta})\vec{X}_3}{\frac{1}{f}}, f_i \le f_{avg} \\ \dfrac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, f_i > f_{avg} \end{cases} \tag{8}$$

$$\frac{1}{f} = \frac{1}{f_\alpha} + \frac{1}{f_\beta} + \frac{1}{f_\delta} \tag{9}$$

3.3 The model of GRU

The gated recurrent unit (GRU) [14] is also a variant of the recurrent neural network, whose main purpose is also to deal with the gradient disappearance and explosion phenomenon occurring during the long sequence of the training process. Since the structures of the GRU and the LSTM are very similar, the GRU can be seen as a variant of the LSTM. In some cases, GRU and LSTM can produce the same results, but compared with LSTM, GRU has a simpler architecture with fewer parameters, so it has a faster training speed and requires fewer resources. Unlike LSTM, which controls which information is output through input, output, and forgetting doors, there are only two updates and reset doors inside. Compared to the LSTM, the GRU has a simpler architecture and fewer parameters, thus having a faster training speed and requiring fewer resources. The specific structure of the GRU is shown in Fig. 4.

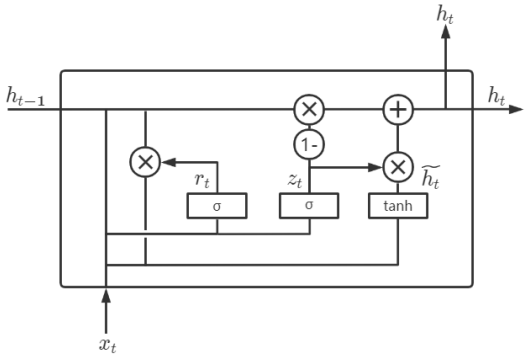

Fig. 4 The structure of GRU

z_t represents the update gate (see Fig. 4), which is used to control the degree to which the state information of the previous moment is introduced into the current state. The larger the value of the update gate is, the higher the degree to which the state information of the previous moment is introduced into the current state. The r_t represents a reset gate to control state information at the previous moment, which is written to the current candidate set $\tilde{h}_t$.

The update gate in the GRU is equivalent to integrating the input gate and the forgetting gate in LSTM, making its structure simpler and having fewer parameters than LSTM. The mathematical description of the GRU is given in Eqs. (10) to (13):

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{10}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{11}$$

$$\tilde{h}_t = tanh(W \cdot [r_t \odot h_{t-1}, x_t]) \tag{12}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{13}$$

where σ is the sigmoid function, which can change the input data into values between [0,1], and can therefore be used as a gating signal.

Eq. (12) defines the state of the candidate hidden layer, where h_{t-1} contains the past information, and r_t is the reset gate, defining how to combine the new input information with the previous memory. The $r_t \odot h_{t-1}$ represents the magnitude of the hidden information at the previous moment in predicting the future. The more the value of r_t tends to be 0, and the more hidden information in the past is discarded. When the value of r_t is 0, only the current input information is retained, and all the past hidden information will be discarded; when the value of r_t is 1, all the past hidden information will be retained and added to the current input information.

Eq. (13) defines the final state of the hidden layer, h_{t-1} in the formula is also the state information of the past, and $\tilde{h}_t$ represents the hidden layer state of the current candidate. The z_t represents the update gate, describing the remaining content of the previous memory saved to the current step. This step is to forget some data in h_{t-1} and add some of the data input from the current node. The value of z_t is between 0 and 1. The closer z_t is to 1, the more past information is inherited, the more long-term dependence remains, and the closer to 0, the more information

in the hidden state is forgotten.

3.4 Intrusion detection algorithm based on GRU

The GRU algorithm has high accuracy and a fast training speed, which is very suitable for detecting the flow data generated by the industrial control system. As shown in Fig. 5, the steps of the GRU-based industrial control network intrusion detection algorithm are as follows:

1. The data were normalized. Because of the order of magnitude difference in the data between the different characteristics of the industrial control flow rate, it is necessary to limit the data to a certain range through the normalization function to avoid the problem of gradient disappearance and gradient explosion. The common normalization method is the min-max normalization, which linearly transforms the raw data to map the results between the [0,1]. The convergence rate of the model increases after normalization and may improve its accuracy. The min-max normalization is defined as shown in Eq. (14).

$$x' = \frac{x - Min}{Max - Min} \tag{14}$$

Each column of the data was processed using the min-max normalization method, with max representing the maximum value in a column and min representing the minimum value in a column.

2. Determine the structure of the GRU network, including the input layer, hidden layer, and output layer; input the normalized training set to the GRU network for training parameters; the GRU network parameters include the network learning rate, the hidden layer, and the number of neurons of the hidden layer; each initial value is determined according to step one; the output layer uses softmax as the activation function, and the output result is a certain type of traffic prediction classification label. The softmax function is defined as shown in Eq. (15), where e^i represents the output value of node i and j is the number of output nodes, which is also the number of the categories classified.

$$S_i = \frac{e^i}{\sum_j e^j} \tag{15}$$

3. The GRU network was trained using the normalized training set, and the cross-entropy was used as a loss function for the multi-classification task to change the weight values of the network by backpropagation. The cross-entropy loss function is defined as shown in Eq. (16):

$$L = \sum_{i=1}^{k} y_i \, log(p_i) \tag{16}$$

where k is the number of species, y_i is the label, and p_i is the output of the neural network prediction.

4. The trained GRU network was verified by using the normalized test set to assess the accuracy of the model classifier.

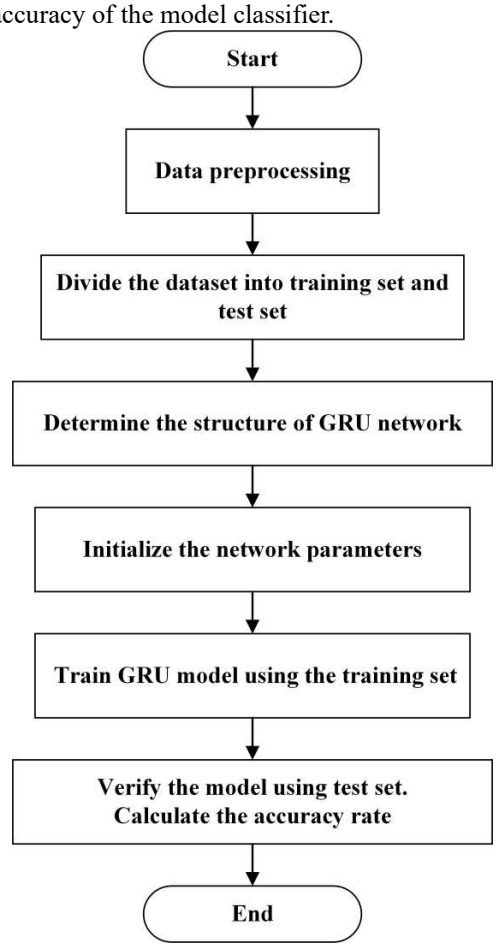

Fig. 5 Flowchart of intrusion detection algorithm based on GRU

3.5 Intrusion detection algorithm based on IGWO-GRU

The industrial control network intrusion detection algorithm model based on GRU includes three parts: input layer, hidden layer, and output layer. Among them, the parameters such as the number of hidden layers, the number of hidden layer neurons, and the learning rate can greatly affect the training effect of the GRU network. Theoretically, the more layers, the more neurons, the more complex the structure of the network, the better the fit to the data. However, in practice, too many hidden layers and the number of hidden layer neurons will not only increase the training difficulty of the model but also appear the

phenomenon of overfitting to reduce the accuracy of the model. If the structure of the network is simple enough, the ability to fit the data will be seriously insufficient, which will also lead to low accuracy problems. Therefore, when training the network, it is necessary to optimize the number of hidden layers and the number of neurons in the hidden layer, and find the appropriate number of hidden layers and the number of hidden layer neurons, so as to achieve the optimal effect of the model. Usually, the parameters such as the number of hidden layers, the number of neurons, and the learning rate are set according to the experience, which is greatly influenced by subjectivity.

Therefore, this paper proposes the IGWO-GRU model to automatically learn the number of hidden layer neurons and the learning rate of the network through the IGWO algorithm. As shown in Fig. 6, the steps of the IGWO-GRU are as follows:

1. Initialize the grey wolf populations and parameters such as $\vec{a}$, $\vec{A}$, and $\vec{C}$. Let the grey Wolf location information include the number of neurons in the hidden layer u and the learning rate l of the network.
2. The number of hidden layer neurons in the grey Wolf location information and the learning rate of the network is brought into the GRU network, and the training set is used to complete the model training.
3. Calculate the fitness of individual grey wolves. The test set uses the accuracy of the model test as the fitness value of the individual grey wolves. The top three individuals with fitness are served as α, β and δ, while the remaining individuals are ω. Then calculated the average fitness of the grey wolf population.
4. If the current fitness of the individual grey wolf is greater than the average fitness, the GWO algorithm is used to update the location information and the values of the parameters α, β and δ. Otherwise, the IGWO algorithm is used to update the location information and the values of parameters α, β, and δ using the IGWO algorithm. The values of u and l will change accordingly when the location information is updated.
5. If the maximum number of iterations is reached, the position information of the IGWO-GRU model and α is output. At this time, the model is the optimal model, and the optimal values of u and l can be obtained from the position information of α. Otherwise, jump to 2 until the iteration is completed, and then output the position information of the model and the α.

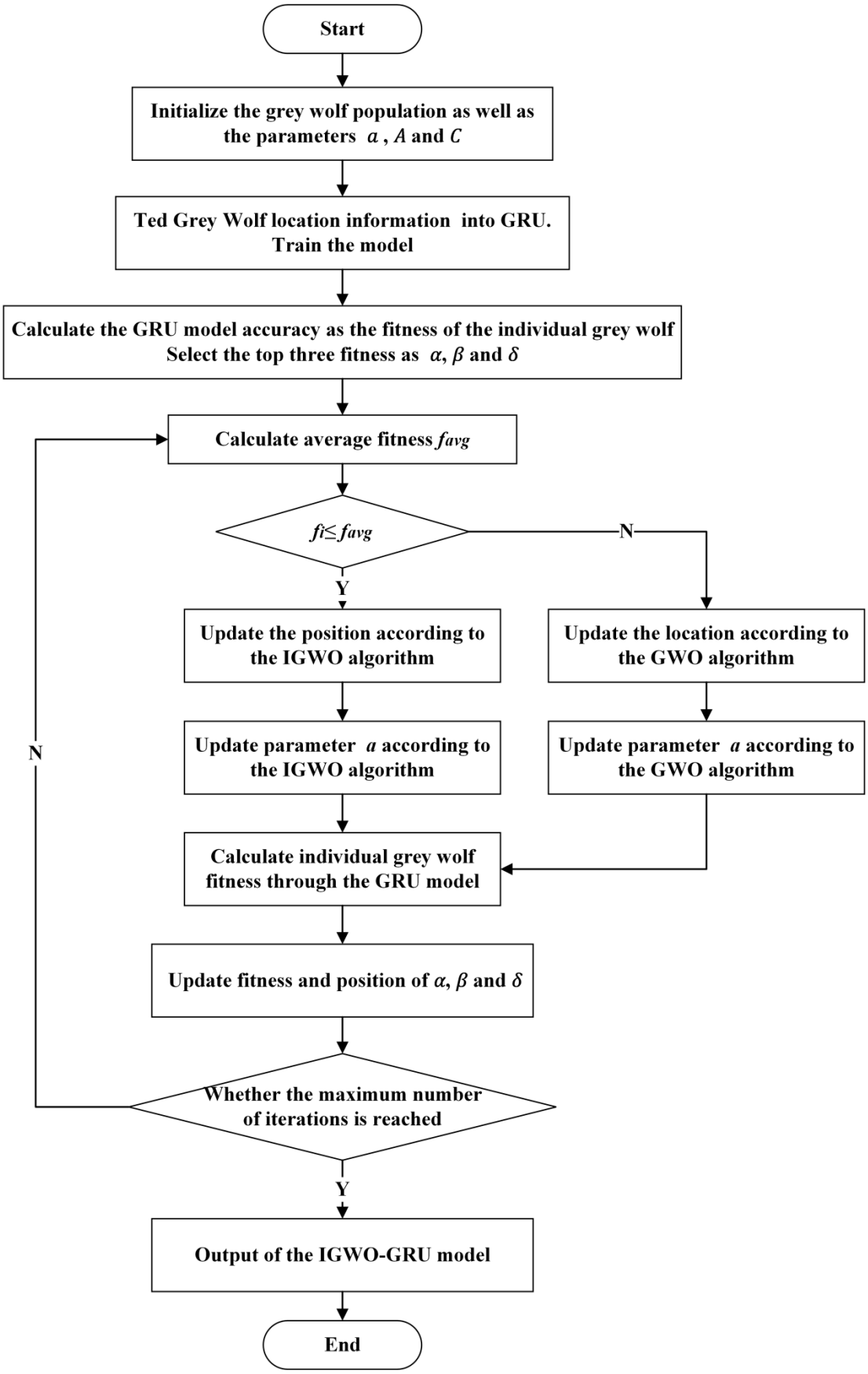

Fig. 6 Flowchart of intrusion detection algorithm based on IGWO-GRU

4 Experiment and Evaluation

4.1 Experimental settings of the optimization algorithm

In order to verify the effectiveness of the improved grey Wolf optimization algorithm (IGWO) proposed in this paper, six test functions (see Table 1) are used to test the test results are compared with the standard grey Wolf optimization algorithm (GWO) [12] and particle swarm optimization algorithm (PSO) [11]. Among the six test functions, F_1, F_2, and F_3 are unimodal functions, which are suitable for benchmarking the algorithm development; the performance of the algorithm in unimodal functions can effectively measure the quality of the optimization algorithm. F_4, F_5, and F_6 are multimodal functions, which can better simulate the situation encountered in reality generation, and can also be used to measure the ability of the algorithm to avoid falling into the local optimal solution. During the course of the experiment, the population number of IGWO, GWO, and PSO was set to 30, and the dimensions of the test function were set to 30, with the learning factor $c_1 = c_2 = 2$ and the inertia weight $\omega = 0.9$. The maximum number of iterations of the three swarm intelligent optimization algorithms was set to 3000. Thirty times were then run independently for each algorithm to obtain the mean and standard deviation of the optimal solution. The following table is the information for the six test functions.

Table 1 Description of benchmark function

Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$F_2(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
$F_3(x) = \sum_{i=1}^{n} i x_i^4 + random(0,1)$	30	[-1.28,1.28]	0
$F_4(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$F_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{1}^{n}\cos(2\pi x_i)\right) + 2 + e$	30	[-32,32]	0
$F_6(x) = \frac{1}{4000}\sum_{1}^{n} x_i^2 - \prod_{1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0

4.2 Performance of IGWO

As can be seen from Table 2, the optimization effect of IGWO and GWO is better than that of PSO, the average optimization effect of the IGWO algorithm is better than that of the standard GWO algorithm, and the standard deviation of IGWO is smaller than GWO, indicating that IGWO algorithm has less volatility and higher stability than GWO algorithm.

As can be seen from Fig. 7, in functions F_1 and F_3, the optimization effect of the IGWO and GWO algorithms is much better than that of the PSO algorithm. In function F_2, although the convergence rate of the PSO algorithm is much slower than that of IGWO and GWO, the best optimization effect found is less than the minimum value of the IGWO algorithm and GWO algorithm. In functions F_1, F_2, and F_3, the IGWO optimization effect is better than the GWO effect; however, in the three functions, IGWO has in early convergence rate than GWO, but with the number of iterations increased GWO

algorithm population diversity, an individual position no longer change, cause algorithm stagnation, while IGWO can keep better in the late population diversity so that can find a better solution. The comprehensive analysis of Table 2 and Fig. 7 shows that the IGWO algorithm is well-optimized on the unimodal function.

Table 2 Results of unimodal benchmark functions.

F	PSO[11]		GWO[12]		IGWO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.
F_1	2.3333333	5.5876848	1.75E-106	3.22E-106	6.11E-137	1.38E-136
F_2	43.146866	28.591404	26.541513	0.8925144	26.1337732	0.7858325
F_3	5.2956747	5.8987453	0.0003133	0.0002535	0.00014661	0.0000987

Table 3 Results of multimodal benchmark functions.

F	PSO[11]		GWO[12]		IGWO	
	Avg.	Std.	Avg.	Std.	Avg.	Std.
F_4	65.989211	24.582439	0	0	0	0
F_5	1.13E-11	4.75E-11	8.027E-15	1.77E-15	7.5495165	0
F_6	0.0106723	0.0103677	0.0019507	0.0059364	0	0

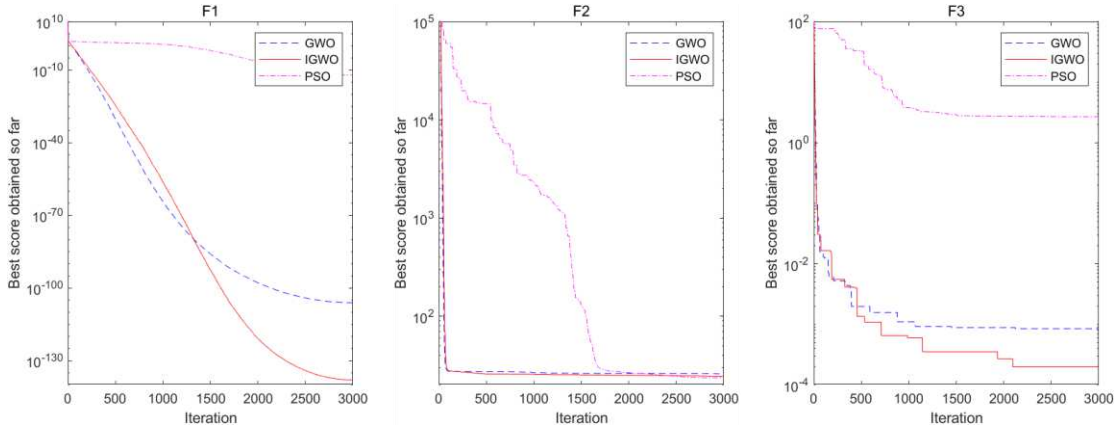

Fig. 7 Convergence graph of unimodal benchmark functions.

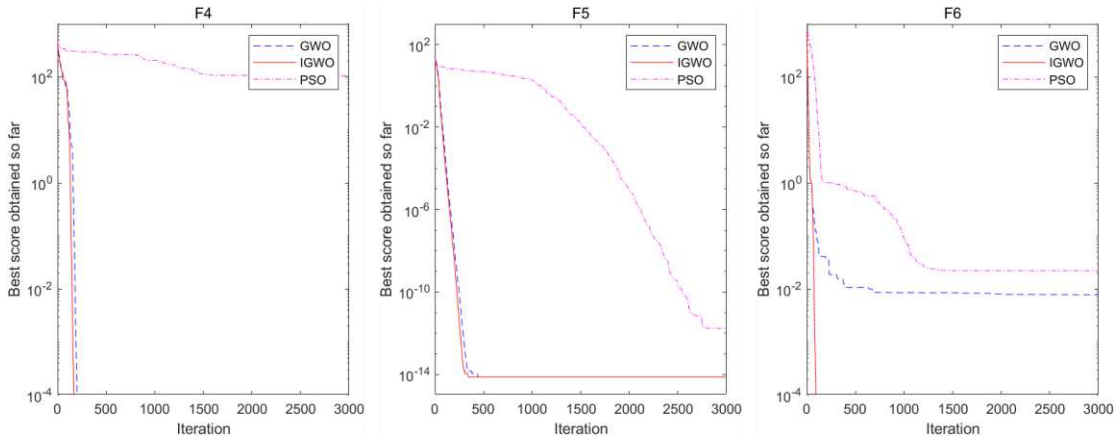

Fig. 8 Convergence graph of multimodal benchmark functions.

As can be seen from Table 3, both IGWO and GWO outperform the PSO algorithm on three multimodal function optimization problems. For F_4 and F_6, the IGWO algorithm reaches the global best advantage, while the GWO algorithm only reaches the global optimum on the function

F_4 , indicating that the IGWO algorithm outperforms the GWO algorithm in jumping out of the local optimum.

As can be seen from Fig. 8, IGWO converges faster than the GWO algorithm on all three multimodal test functions. In function F_4, the IGWO finds the global optimal solution at around 200 iterations, while the GWO algorithm finds the optimal solution at about 260 steps. In the function F_5, IGWO, and GWO all fall into the same local optimal solution, but IGWO converges even faster. In function F_6, the IGWO algorithm finds the global optima at around 100 iterations, while the GWO falls into the local optima, indicating that the IGWO outperforms the GWO in its ability to jump out of the local optimal solution. Through the comprehensive analysis of Table 3 and Fig. 8, the IGWO algorithm has a very good optimization effect on the multipeak function.

By testing on unimodal and multimodal test functions, we show that the proposed IGWO algorithm has better optimization ability than the standard GWO algorithm and a better ability to jump out of the local optima.

4.3 Experimental settings of an intrusion detection algorithm

The industrial control system network data used in the experiment process of this paper is the Natural Gas Pipeline Control System dataset [15] published by Mississippi State University in 2014. This dataset is a general standard dataset in the field of industrial control security, whose traffic data is captured through a network data logger. Compared with the KDD99 data set, the data set is updated and generated by simulating the real industrial control system environment, which can truly reflect the characteristics of the industrial control network traffic, so it can more effectively compare the advantages and disadvantages of various industrial control system abnormal traffic detection schemes.

The dataset contains normal traffic and abnormal traffic generated through 28 attacks, with a total of 97,019 items. The 28 modes of attack can be summarized into seven classes, as shown in Table 4. There are 26 features and one taxonomic label for each data bar in the dataset, and the amplitude varies greatly between the different features in the real data in the dataset, so the data needs to be processed using the normalization function.

Table 4 Description of data

type	label	quantity	description
Normal	0	61156	normal
NMRI	1	2763	Simple malicious response injection attack
CMRI	2	15466	Complex malicious response injection attack
MSCI	3	782	Malicious state command Inject attack
MPCI	4	7637	Malicious parameter command injection attack
MFCI	5	573	Malicious feature command inject attack
DOS	6	1837	denial of service attack
Reconnaissance	7	6805	Reconnaissance attack

All the experiments in this chapter use the Windows platform; the processor is Intel Core i7-7700HQ, 32G memory, graphics card, and 4G video memory is NVIDIA GeForce GTX 1050Ti. The programming language was implemented using Python, and the GRU model was implemented using TensorFlow.

Eighty percent of the data from the experimental dataset was used for model training and 20% for testing. The number of hidden layers of the algorithm is set to 2. The number of hidden layer neurons u has a value range of [8,128]. The range of values of the learning rate l is [0.001,0.1]. Population size is set to 30, with a maximum of 500 iterations.

If the abnormal flow is regarded as positive and normal flow as negative, the following four situations will occur: true positive (TP), false negative (FN), true negative (TN), and false positive (FP). The confusion matrix is shown in Table 5.

Table 5 Confusion matrix

	Predicted as abnormal	Predicted as normal
abnormal	TP	FN
normal	FP	TN

The overall accuracy (ACC), false positive rate (FPR), and false negative rate (FNR) are

generally used as the evaluation criteria of the model. Under a certain accuracy rate, the lower the missing alarm rate and false alarm rate, the better the detection effect of the model is. In general, the false positive rate and the false negative rate cannot be reduced at the same time. The calculation formulas are determined as in Eqs. (17) to (19):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

$$FNR = \frac{FN}{FN + TP} \tag{18}$$

$$FPR = \frac{FP}{FP + TN} \tag{19}$$

4.4 Performance of IGWO-GRU

This experiment first verifies that the basic model GRU is more suitable for the intrusion detection of industrial control network traffic compared with other algorithms BP, DT, and SVM. The experimental results are shown in Table 6, and it can be seen that GRU has the highest accuracy and the lowest underreporting rate, while the false alarm rate is slightly worse than BP. However, in terms of the characteristics of the data set time series, GRU is generally more appropriate.

Table 6 Comparison of experimental results

Algorithm	ACC	FPR	FNR
GRU[14]	94.51%	2.90%	5.56%
BP[10]	91.26%	2.53%	11.61%
DT[2]	88.43%	3.47%	9.33%
SVM[3]	86.79%	4.00%	10.32%

The GRU is selected as the basic model, and the improved optimization algorithm IGWO is used for the model optimization. From Table 7, compared to the GRU algorithm, the IGWO-GRU algorithm improves the accuracy rate by 3.11% and reduces the false positive rate and false negative rate by 2.01% and 4.03%. Due to the use of IGWO, the algorithm has significantly better accuracy, false alarm rate, and omission rate.

Table 7 Comparison of experimental results

Algorithm	ACC	FPR	FNR
IGWO-GRU	97.62%	0.89%	1.53%
GRU[14]	94.51%	2.90%	5.56%

To further illustrate that the abnormal flow detection algorithm based on IGWO-GRU is better than that based on GRU. In this paper, we compare the convergence curves and the accuracy curves of the two algorithms. The convergence curve is shown in Fig. 9, showing from the convergence curve that the IGWO-GRU algorithm converges faster and converges better. The accuracy curve is shown in Fig. 10. It can also be seen from the accuracy curve that the convergence rate of the IGWO-GRU algorithm is faster, and compared with the GRU algorithm, the IGWO-GRU algorithm has higher accuracy, less volatility, and more stability.

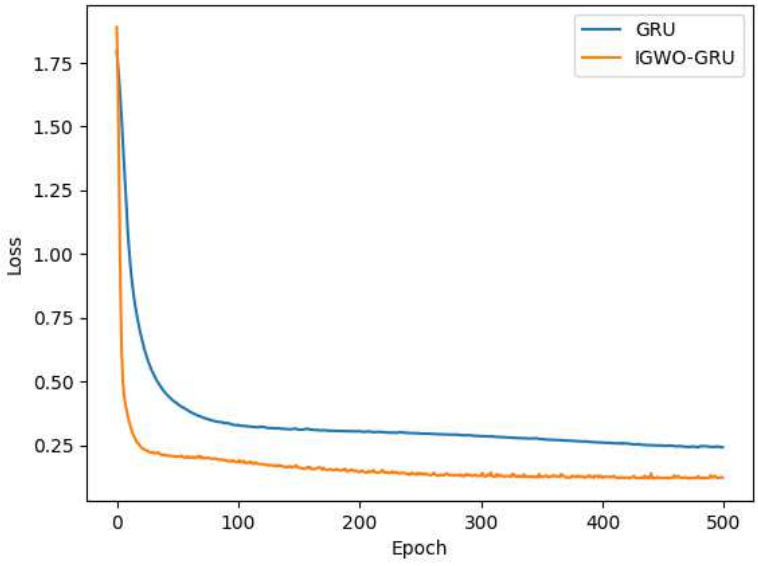

Fig. 9 Convergence curve.

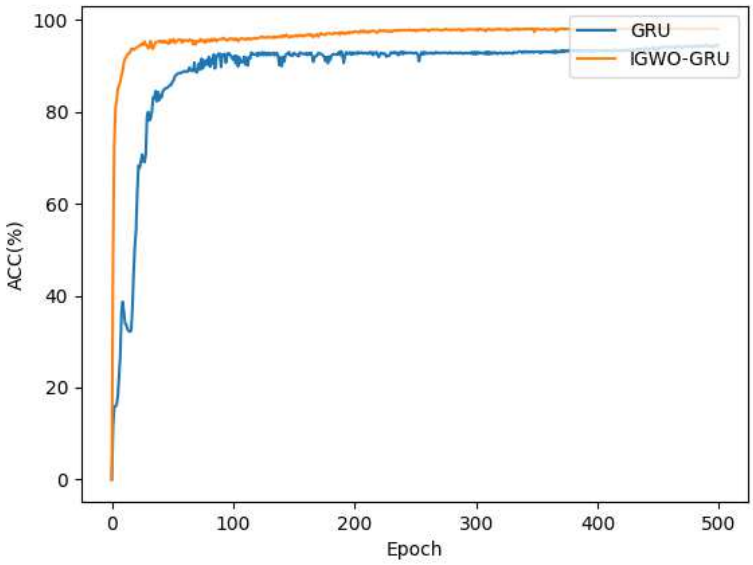

Fig. 10 Accuracy curve.

This paper compares IGWO-GRU with GRU, BP, DT, and SVM. The results are shown in Fig. 11. It can be seen that the IGWO-GRU has the highest accuracy, 97.62%; the lowest accurate algorithm is SVM, whose accuracy is 86.79%. The algorithm with the lowest false alarm rate is IGWO-GRU, with 0.89%, and the highest alarm algorithm is SVM, with a false alarm rate of 4.00%. The algorithm with the lowest omission rate was IGWO-GRU, with a miss rate of 1.53%, and the algorithm with the highest omission rate was BP, with a miss rate of 11.56%.

The accuracy, false alarm rate, and under-report rate of the IGWO-GRU algorithm are optimal among the compared algorithms, indicating that the advantages of the IGWO-GRU algorithm in abnormal traffic detection in industrial control systems are superior.

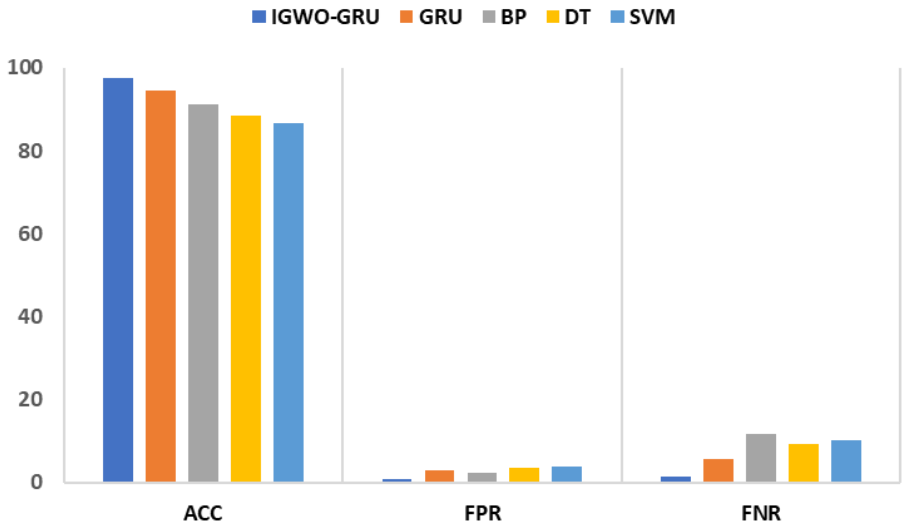

Fig. 11 Bar graph of the models' performance.

5 Conclusion

This paper proposes an industrial control network intrusion detection algorithm based on the IGWO-GRU model, with two improvements for the GWO algorithm. Through comparative experiments, the effectiveness of the proposed algorithm is verified.

1. Proposes the weight-based adaptive adjustment strategy and divides the two methods to update particles, enrich the diversity of the population, and ensure a fast convergence rate;
2. The parameter changing from linearity to nonlinearity can improve the global search capability of the algorithm and avoid falling into local optimum.

References

1. Ahmed M, Mahmood A N, Hu J. A survey of network anomaly detection techniques[J.Jounal of Network and Computer Applications, 2016,60: 19-31.
2. Lee J H, Lee J H, Sohn S G, et al. Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system[C]//2008 10th International conference on advanced communication technology. IEEE, 2008, 2: 1170-1175.
3. Shang W, Cui J, Song C, et al. Research on industrial control anomaly detection based on FCM and SVM[C]//2018 17th IEEE international conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE). IEEE, 2018: 218-222.
4. Fang Y, Li M, Wang P, Jiang X, Zhang X. Intrusion detection model based on hybrid convolutional neural network and recurrent neural network. J Comput Appl 2018;38(10):2903–7,2917.
5. Goh J, Adepu S, Tan M, et al. Anomaly detection in cyber physical systems using recurrent neural networks[C]//2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE). IEEE, 2017: 140-145.
6. [20] Yu B, Wang H, Yan B. Intrusion detection of industrial control system based on long short term memory. Inf Control 2018;47(1):54-9.
7. Xu C, Shen J, Du X, Zhang F. An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access 2018;6:48697-707.
8. Bian J, Wang L, Scherer R, et al. Abnormal detection of electricity consumption of user based on particle swarm optimization and long short term memory with the attention mechanism[J]. IEEE Access, 2021, 9: 47252-47265.
9. Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in engineering software, 2014, 69: 46-61.
10. Yang A, Zhuansun Y, Liu C, et al. Design of intrusion detection system for internet of things based on improved BP neural network[J]. Ieee Access, 2019, 7: 106043-106052.
11. Shang W, Zeng P, Wan M, et al. Intrusion detection algorithm based on OCSVM in industrial control system[J]. Security and Communication Networks, 2016, 9(10): 1040-1049.
12. Chen Chen, Qi Yajiang, Yang Lintao, Wang Guanghua, Ye Xiaoyan, Wei Dan. Network intrusion detection method based on one-dimensional CNN and GWO-SVM[P]. State Key Laboratory of Astronautic Dynamics (China); Xi'an Satellite Control Center (China),2022.
13. Mittal N, Singh U, Sohi B S. Modified grey wolf optimizer for global engineering optimization[J]. Applied Computational Intelligence and Soft Computing, 2016, 2016.
14. Wang Z, Wang Z S, Yi F Z, et al. Attack Traffic Detection Based on LetNet-5 and GRU Hierarchical Deep Neural Network[C]//International Conference on Wireless Algorithms, Systems, and Applications. Springer, Cham, 2021: 327-334.
15. Morris T, Gao W. Industrial control system traffic data sets for intrusion detection research[C]//International conference on critical infrastructure protection. Springer, Berlin, Heidelberg, 2014: 65-78.