ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Research on Network Characterization Security Response Based on SDN Rule Definition of Power IOT Service Traffic

Binyuan Yan[1,*]

[1] Information Center of Guizhou Power Grid Co., LTD., Guiyang, Guizhou, 550002, China

SUMMARY: *In order to accurately assess the SDN network security status of power IOT business traffic, this paper proposes an SDN-oriented network security sensing method. The method extracts network security posture indicators based on the attack characteristics suffered by data, control and application planes. On this basis, a network security posture assessment method based on CS-BPNN is proposed, which uses the cuckoo search algorithm to find the optimal weights and thresholds of the model, and then applies the BP algorithm to adjust the error. On the basis of network security posture assessment, in order to solve the nonlinear problem in security posture prediction, the parameters of the LSTM prediction model are calculated using the gray wolf optimization algorithm to control the direction of the optimization search. The results show that the overall error of the proposed CS-BPNN assessment model is small and closest to 0. The prediction accuracy of the GWO-LSTM model in different tasks reaches more than 95%, which confirms the accuracy and stability of the method proposed in this paper. It provides methodological support for network feature security response and strategy design in the environment of electric power IoT.*

KEYWORDS: *cuckoo search algorithm; gray wolf optimization algorithm; posture assessment; posture prediction; SDN; power IoT service traffic*

1 Introduction

In the current context of vigorously developing new energy, many emerging businesses rely on the Internet of Things of electricity to develop and grow [1]. For example, the electric vehicle service business, the business to electric power Internet of things as a basis, the use of Internet of things technology to realize the people rely on the network to control electric vehicles and charging piles, charging piles and electric vehicles can also feedback real-time information to the people to realize the interconnection and interaction. The stable development of electric power IoT cannot be separated from a strong electric power communication network, verified communication professional in support of strong smart grid and electric power IoT business, the demand for the main 50 types of business, of which 35 types of traditional business, 15 types of new business [2, 3]. These services put forward higher requirements on the scalability, reliability and security of electric power communication networks. The traditional electric power communication network can no longer meet the increasingly changing and diverse electric power IOT business needs. At the same time, power equipment is developing in the direction of intelligence and integration, leading to a sharp increase in terminal data collection information, which is mainly gathered in the control center for processing, posing a huge challenge to the capacity of the communication network.

*binyuanyan64@163.com
https://doi.org/10.65102/is2026699
Published: 30 April, 2026

Software Defined Networking (SDN) is a new network paradigm that separates control decision making from data forwarding, with a new network architecture that is automated, adaptive, manageable, cost-effective and dynamic [4]. This innovative architecture provides intelligent management, dynamic adaptability and other characteristics for power communication networks, which brings a new research area in the field of network technology. Literature [5] proposes a lightweight SDN framework called μSDN, which significantly reduces SDN control overhead through protocol and architectural optimization, bringing it to a practical level in IEEE 802.15.4 networks while maintaining interoperability with IPv6 and existing routing protocols. Literature [6] proposes a Link-Related Risk-based Critical Link Identification Algorithm (LRR-CLIA) for the application of SDN in power communication networks for service scenarios with dual paths of work and backup in software-defined power communication networks. Literature [7] proposes a hierarchical software-defined network framework, which covers the design of microgrid cluster and global grid at two levels, and evaluates and verifies the performance of the framework in terms of delay, reliability, and security through theoretical analysis and practical experiments. Literature [8] proposes a load balancing scheme based on SDN controllers, designs a basic framework for progressive layered transformation in combination with the actual architecture of power communication networks, and further constructs a multi-controller deployment model and gives the corresponding solution algorithms by establishing a response delay model for SDN controllers. Literature [9] studied the reliability service guarantee mechanism of SDN controller in power communication network deployment, and established a mathematical model of controller deployment in line with the characteristics of power communication by analyzing the design of SDN-based power communication network security protection architecture. Literature [10] analyzes the technical limitations of the current electric power communication network and power distribution equipment, and then proposes a scheme for upgrading and transforming the existing electric power communication network based on SDN technology, which provides design ideas for constructing an SDN-enabled intelligent power distribution system. In the research related to the deployment of controllers, researchers through mainly consider, design, and optimize controller deployment algorithms through different performance indicators or in different application scenarios [11]. Literature [12] proposes an adaptive, continuous consistency model for distributed SDN controllers in large-scale deployments, and proposes a deployment solution for content-centric delivery networks (C-CDNs) in the environment of the Internet of Things (IoT) based on SDN technology. Literature [13] explores the Controller Deployment Problem (CPP), and its scaling in multi-controller scenarios (MCPP), focusing on the fact that although it has been relatively easy to adopt SDN in small and medium-sized networks, the deployment and implementation of SDN in large-scale networks is still the focus of current research.

As the scale of power communication networks continues to expand, higher requirements are placed on network scalability, reliability and security. As a single centralized controller can no longer meet the needs of network security and assurance [14]. In order to enhance the security of SDN, literature [15] proposes an SDN-based security risk assessment framework for smart grid communication networks, which quantifies the risk of denial-of-service (DoS) attacks on Intelligent Electronic Devices (IEDs) and IEC 61850 networks by effectively quantifying the risk of denial-of-service (DoS) attacks on IEDs and IEC 61850 networks. Literature [16] developed a fault tree model for assessing system functional unavailability due to data communication failures for power plant safety critical systems, which aims to quantify the impact of data link and network failures on safety signal generation. Literature [17] systematically analyzes how SDN enhances the resilience of smart grids against malicious attacks, identifies additional risks that SDN may introduce and their management strategies, and proposes a methodology for validating and evaluating resilience solutions based on SDN.

Literature [18] proposes a hierarchical Device-to-Device (D2D) communication architecture for public safety applications in smart cities, which reduces energy consumption by reducing the dependence on Long Term Evolution (LTE) communication links through centralized SDN controllers in collaboration with the cloud. Literature [19] proposes a novel smart grid security architecture that incorporates Digital Twin (DT), SDN, Deep Learning (DL), and Blockchain technologies, which firstly designs an authentication mechanism based on Blockchain that is resistant to a variety of known attacks in order to establish a secure communication channel. Literature [20] addresses the problem of frequent cyber-attacks on power grids leading to large-scale blackouts by proposing an SDN architecture with an integrated hybrid Intrusion Detection System (IDS), which is designed to detect and block the injection of malicious messages based on the IEC 61850 Generic Object-Oriented System Event (GOOSE) in a digital substation, as well as being able to locate the point of failure and mitigate it by shutting down specific ports.

For research in the power IoT business environment, literature [21] proposes a random number generator-based hierarchical intrusion detection system (RNGHID) to address security challenges in IoT exacerbated by device interconnections and open environments, especially for the denial-of-service (DoS) attack subclass of delegated entity attacks (DEAs), which are more camouflaged. Literature [22] proposes an SDN platform based on industrial IoT technology, which achieves immediate response to power failures and network restoration through real-time monitoring technology, and provides multifunctional control and optimization using SDN controllers to assist operators in managing demand, resources, and improving system reliability based on real-time data. Literature [23] focuses on the resilience assessment of distribution networks under the threat of natural disasters and network intrusion, and proposes a set of distribution network resilience monitoring index system that integrates the access of IoT Distributed Energy Resources (DERs), which constructs the IoT Trustworthiness Score (ITS) by integrating neural networks with federated learning, and then integrates the ITS by utilizing the fuzzy multi-criteria decision making (F-MCDM) approach and other toughness influencing factors to calculate the primary node toughness (PNR). Literature [24] explores the potential network threats, security vulnerabilities, and the main exploitation strategies commonly used by attackers, and on this basis proposes a taxonomy of network attacks against critical information infrastructure to guide the enhancement of the effectiveness and resilience of network security protection programs. Literature [25] systematically explores the cybersecurity challenges faced by the power system after the introduction of IoT technology, comprehensively overviews various types of cyberattacks against the IoT-enabled power system, and focuses on analyzing the attack entrances and countermeasures.

This paper proposes a set of security response optimization methods for the electric power IoT environment. The study firstly designs a network security posture indicator extraction system covering three planes to comprehensively quantify the security threats faced by the power IoT business traffic. Then, the global optimization capability of cuckoo search algorithm is used to overcome the defect of BP neural network that is easy to fall into local optimization, so as to realize the accurate quantitative assessment of the overall network security posture of power IoT. Then, the parameters of the LSTM model are adaptively optimized by the Gray Wolf optimization algorithm, which effectively improves the model's ability to extract nonlinear features, and realizes the accurate prediction of the risk value of the security posture and the frequency of security events. Finally, the constructed CS-BPNN model is trained and tested to analyze the overall error and convergence of the model, and the GWO-LSTM model is compared and experimented with classical LSTM and other prediction models to comprehensively evaluate the prediction performance of the model in this paper.

2 Network characteristics defined by SDN rules for power IoT business traffic

In this paper, we quantify the network security posture values based on the SDN architecture by extracting the characteristics of possible attacks from the data plane, control level, and application level, respectively.

2.1 Data plane

2.1.1 Port scanning

Port scanning is the basis for attackers to find their targets of interest and perform further attacks. The main purpose is to collect and summarize the resources related to the attack target, and extracting the port scanning features in the flow table of SDN switches is an indispensable part of situational awareness.

The features that can be extracted for the port scanning in the flow table of SDN switches are as follows:

1) Number of destination ports: attackers who launch port scans usually send packets from a source IP address to different destination ports. Therefore, the number of different destination ports increases rapidly when subjected to a port scanning attack, but most of the ports are unavailable. The number of destination ports can be calculated by formula (1):

$$\begin{aligned} a = count(\{obj.dst\_port != ref.dst\_portANDobj.dst\_addr \\ = ref.network\_addr\}) \end{aligned} \tag{1}$$

2) Proportion of Failed Connections: since the attacker will send a large number of streams to check the status of the port, most of the streams will not succeed in establishing a connection. The percentage of unsuccessful connection streams can be calculated using equation (2):

$$u = \begin{aligned} &count(obj.dst\_addr \\ &= ref.dst\_addrAND((obj.pkg\_count \\ &< 3ANDobj.protocol = TCP) \\ &orobj.pkg\_count \\ &< 2ANDobj.protocol = UDP))) \end{aligned} \Big/ \begin{aligned} &count(obj.dst\_addr \\ &= ref.network\_addr) \end{aligned} \tag{2}$$

2.1.2 Switch DDoS Attacks

DDoS attack mainly refers to the attacker consumes the computational resources of the SDN controller through a large number of puppet hosts and prevents providing services to legitimate users.

Detecting DDoS attacks based on traffic statistics features is a common method in SDN. Therefore, we refer to the DDoS attack hexadecimal group as the characteristics of DDoS attack, and the DDoS attack hexadecimal group is as follows:

Average Packet Number of Flow (APf): the median number of packets for each flow is taken after sequentially arranging the number of packets, where X denotes the number of packets of the flow and n denotes the number of flows.

$$APf = \frac{X(n/2) + X((n+1)/2)}{2} \tag{3}$$

Percentage of Paired Flows (PPf): calculates the number of paired flows that have the same communication protocol. and Num_Pair_flows indicates the number of paired flows, and Num_flows indicates the total number of flows. The calculation method is as follows:

$$PPf = \frac{2 * Num\_Pair\_flow}{Num\_flows} \tag{4}$$

Growth rate of a single flow (GSf): the growth rate of a single flow in which the source IP address, and the destination IP address do not interact within the statistical time window W.

$$GSf = (Num\_flows - 2 * Num\_pair\_flows) / W \tag{5}$$

Growth rate of different ports (GDp): the growth rate of different ports of the switch during the statistical time window W.

$$(GDp) = Num\_ports / W \tag{6}$$

2.2 Control planes

2.2.1 Controller DDoS attacks

The attacker has 2 types of attacks to perform DDoS attack on the controller, the DDoS attack on the controller is shown in Figure 1.

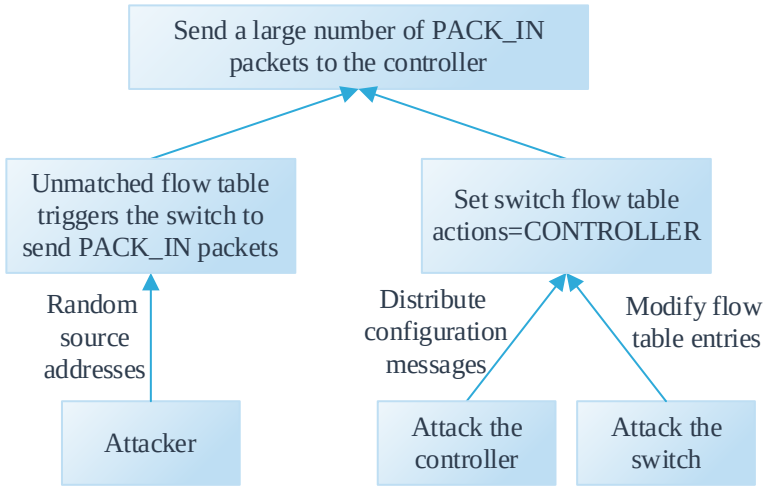


Figure 1: DDoS attack mode of the controller

In the first way, the attacker randomly forges the source IP address so that the packets cannot match the flow table in the switch, triggering the switch to send Pack_in messages to the controller.

In the second way, the attacker attacks the switch and modifies the actions field of each table entry in the flow table, setting it to CONTROLLER, so that the switch sends all packets to the controller.

2.2.2 ARP attacks

The ARP address resolution protocol is used to speed up network communication by finding the corresponding MAC address through the IP address and storing the resolution results in an ARP cache in the form of IP-MAC pairs. the ARP frames are divided into request frames and response frames.

ARP attacks in SDN can be categorized into two situations. The first case is ARP flooding attack; the second case is ARP cache attack. Considering the ARP communication process and attack mode, we extract the following two features related to ARP attack as follows:

Ratio of response frames to request frames: when an ARP attack occurs, the source host sends more ARP frames than it receives. If the ratio of response frames to request frames is less than a threshold value, the probability of an ARP attack is high. The number of request frames and response frames can be obtained by counting ARP packets and analyzing the fields in the packets. To get the ARP packet using pkt_arp=pkt.get_protocol, the request frame is: pkt_arp.opcode= =arp_ARP_REQUEST and the response frame is: pkt_arp.opcode= =arp_ARP_REPLY.

IP-MAC mapping pairs: Under normal conditions, the IP-MAC mapping in the ARP cache is in a 1-1 pattern. If the IP-MAC mapping in the ARP cache is in a 1-N or N-1 pattern, it means that the network is likely to have suffered an ARP cache attack. The historical IP and MAC addresses can be obtained from the OpenFlow match field, ARP_spa is the source IP address, ARP_tpa is the destination IP address, ARP_sha is the source MAC address, and ARP_tha is the destination MAC address. The new IP address and MAC address can be obtained by monitoring Openflow messages.

2.2.3 Controller overload

The controller is the core component of the control plane, through which the controller can centrally control the switches to achieve fast forwarding of data, convenient and secure management of the network, and improve the overall performance of the network. Therefore, determining whether the controller is overloaded or not is an important factor to measure the performance of SDN network. In this paper, controller performance is measured by the following 2 factors: T : Calculating the flow path is the most basic function of the controller, and the length of the calculation time is an important factor in determining whether the controller is overloaded and under attack.

Let: the time for the switch to upload Pack_in messages is $t1$; the time for the controller to send down flow_mod messages is $t2$; the number of Pack_in messages is N , then:

$$T = (t2 - t1) / N \tag{7}$$

Passive flow table down rate R : flow table is the basis for the switch to process incoming network packets, when normal by the controller according to the network information down the flow table; if the controller is overloaded or attacked, will not be able to normal down the flow table, resulting in the data plane can not realize the data forwarding. Therefore, judging the passive flow table issuance rate is an important factor to measure the security of SDN network.

Let: the number of N Pack_in messages uploaded to the controller, the number of flow_mods downloaded by the controller is M , then:

$$R = (N - M) / N \tag{8}$$

2.3 Application Plane

The controller transforms the application policies of SDN into flow table rules and sends them down to the bottom layer switches, which strictly match the flow tables corresponding to the policies. In this paper, the real-time dynamic policy detection mechanism is used to detect the existence of policy conflicts as one of the network security situational awareness indicators, and the algorithm steps are as follows:

1) Find the flowpath flow-path from the switch flow table and network topology information;

2) First set flowpath to be empty, and set the source address and source port number of the first flow table entry in the switch flow table to the source address and source port number of flowpath;

3) Set an empty switch linked list switch_liked_list;

4) Set the variable nexthop = switch_liked_list.next();

5) When nexthop is not empty, loop through the port numbers to get the next switch pointed to by each flow table entry in the flow table;

6) If the fetched switch is not empty, add the fetched switch to flowpath and switch_liked_list;

7) Set the destination address and destination port of the last flow table entry to the destination address and port number of flowpath, at which point the loop ends;

8) Get the complete flowpath flowpath.

It can be seen through the analysis, this algorithm step mainly contains two loops, first get the switch_liked_list this list of nexthop, in the case of nexthop is not null, the loop to get each switch flow table points to the next switch, so the complexity of the algorithm is $O(n2)$.

3 CS-BPNN based network security posture assessment methodology

3.1 CS-BPNN Theoretical Foundations

3.1.1 Cuckoo search algorithm

The cuckoo's behavior of breeding offspring is used to design an optimization algorithm in which each generation is represented by a set of nests, each carrying an egg representing a feasible solution, and the quality of the feasible solutions is improved by generating new feasible solutions from existing feasible solutions and modifying certain characteristics, with the number of solutions remaining fixed in each generation.

The CS algorithm starts with the first generation of cuckoos, determines whether to adjust the next generation based on a fitness function, and generates the location of the next generation using a Levy flight, which consists of a series of random excursions in consecutive random steps. It is expressed in mathematical language as:

$$S_N = \sum_{i=1}^{N} X_i = X_1 + \cdots + X_N \tag{9}$$

It can also be written in the following form:

$$S_N = \sum_{i=1}^{N-1} X_i + X_N = S_{N-1} + X_N \tag{10}$$

where S_N denotes a randomized wander and X_i denotes a step size that obeys a random distribution.

Cuckoo algorithm is using Lévy flight for random wandering.

In the cuckoo algorithm search for the next generation, the role of Levy flight in which can be expressed in the following formula:

$$x_i^{t+1} = x_i^t + \alpha \oplus levy(\lambda) \tag{11}$$

where α is the step size, which usually takes the value of 1 to reduce complexity.The large step size of the random wandering through the Lévy flights makes them behave as if they have more powerful spatial exploration capabilities.

The Lévy distribution can be expressed by the following equation:

$$Levy \sim u = t^{-\lambda}, (1 < \lambda \le 3) \tag{12}$$

In the CS algorithm, the Mantegna algorithm is used to generate random steps.

The Mantegna algorithm generates the random step size by means of a Lévy stable distribution, which requires the distribution parameter α and the number of iteration generations, where $\alpha \in (0.3, 1.99)$. The step size v when using Mantegna's algorithm can be represented by x and y in a Gaussian distribution:

$$v = \frac{x}{|y|^{1/\alpha}} \tag{13}$$

where x and y are randomized design variables with standard deviations.

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)\sin(\Pi\alpha/2)}{\Gamma((1+\alpha)/2)\alpha 2^{(\alpha-1)/2}} \right]^{1/\alpha} \tag{14}$$

$$\sigma_y = 1 \tag{15}$$

where Γ is the gamma function.

3.1.2 BP Neural Networks

BP neural networks play an important role in various fields with their powerful capabilities.

The determination of the number of nodes in the hidden layer depends on many factors, including the number of nodes in the input and output layers, and the complexity of the actual problem. Currently, the number of hidden layer nodes is generally determined by the trial-and-error method: the initial number is determined based on empirical formulas, and then the network is trained to find the number of hidden layer nodes corresponding to the network with the best training effect. There are four empirical formulas below:

$$l = 2n + 1 \tag{16}$$

$$l = \sqrt{n+m} + \alpha \tag{17}$$

$$l = \log_2 n \tag{18}$$

where l, m and n are the number of nodes in the implicit, output and input layers, respectively, and α is a positive integer less than 10.

The number of nodes in the output layer is determined based on the output vector, and the dimension of the output vector determines the number of nodes in the output layer, if the length of the output vector is 3, the number of nodes is 3, and if the length of the vector is 2, the number of nodes is 2.

3.2 Design of Cybersecurity Posture Assessment Methods

3.2.1 Design Ideas

The design ideas are as follows: use the powerful nonlinear mapping ability of BP neural network to solve the problem of uncertainty in the relationship between posture data and posture value in the posture assessment, discover the law from the posture data and reason out the posture value, so as to make the network security posture assessment system more flexible. The CS algorithm is used to find the optimal combination of weights and thresholds to make the CS-BPNN model assessment results more accurate.

3.2.2 Design of a posture assessment model

Combining the characteristics of network security posture assessment, the CS-BPNN model is designed as shown in Figure 2.

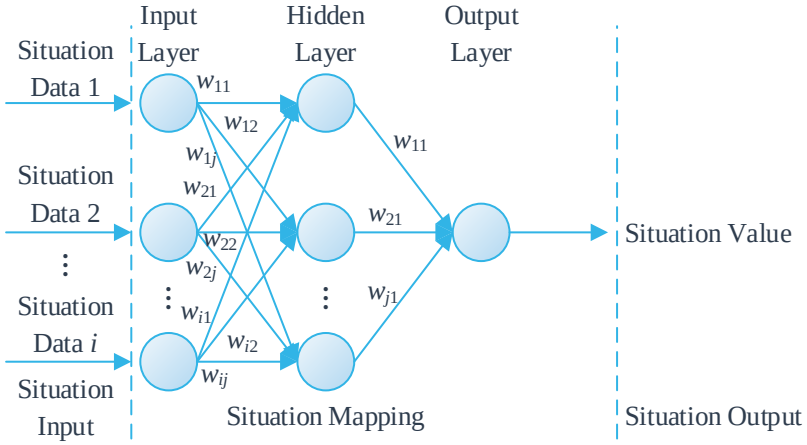


Figure 2: CS-BPNN network security situation evaluation model

The model consists of posture data input, posture mapping and posture output parts, and the functions of each part are designed as follows:

1) Posture data input: the posture related data of each node in the network is collected in time periods, and the data collected in each time period is used as a set of data in the posture input part.

2) Posture data mapping: this part consists of three layers, the input layer that accepts posture data, the implied layer that performs information processing and the output layer that outputs posture values.

The method of setting the number of nodes in the implicit layer generally adopts the trial-and-error method: firstly, use the empirical formula to set fewer implicit nodes, and then increase an equal number of implicit nodes each time, and then compare the size of the training

error and select the number of nodes corresponding to the smallest error under the premise of using the same sample set. The initial number of hidden nodes l is:

$$l = \sqrt{n+m} + \alpha \tag{20}$$

Setting up the mapping part of the CS-BPNN evaluation model, there are connection weights w_{ij} from the input layer to the implicit layer, and a threshold θ_j for the implicit layer; connection weights v_{jk} from the implicit layer to the output layer, and a threshold r_k for the output layer; and the output value of the j th implicit layer node is y_j, and the output value of the k th output layer node is y_k. Then:

$$y_j = f\left(\sum_{i=1}^{n} w_{ij} \times x_i - \theta_j\right) \tag{21}$$

$$y_k = f\left(\sum_{j=1}^{n} v_{jk} \times y_j + r_k\right) \tag{22}$$

$f(x)$ is the Sigmoid function, which is the standard BP neural network transfer function.

3) Posture data output: receive the value passed by the posture mapping part, which is the posture value of the network in a certain time period.

3.2.3 Cybersecurity posture assessment

The purpose of network security posture assessment is to complete the mapping from the posture data set to the posture result set, which mainly includes sensing, acquiring and evaluating the calculation of the posture data, and giving the judgmental results on the network security status.

The design of network security posture assessment process is shown in Figure 3.

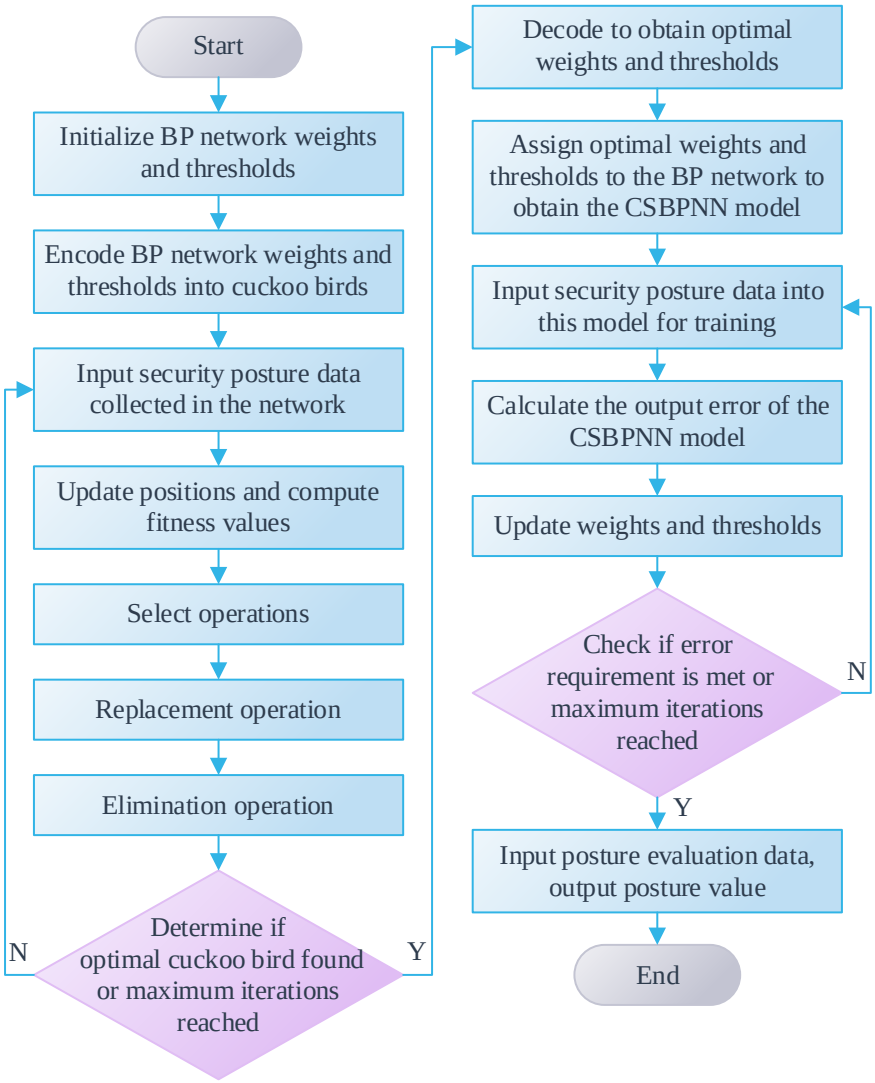


Figure 3: Situational evaluation process

The steps of network security posture assessment based on CS-BPNN are designed as follows:

Step 1: Collect the posture-related data from each node in the network, eliminate incomplete data, and obtain network parameters such as flows and packets for processing to generate input data for the CS-BPNN model;

Step 2: Randomly generate n cuckoos $X_0 = (x_1^{(0)}, x_2^{(0)}, ..., x_n^{(0)})^T$. The n cuckoos are coded, and the posture data obtained in the previous step are substituted into the fitness function to obtain the fitness values of individual cuckoos. Select the cuckoo $x_i^{(0)}$ with the optimal fitness value;

Step 3: Retain the optimal cuckoo individual $x_i^{(0)}$ from the previous generation, and update the cuckoo's position according to the position update formula (23) $X_t = (x_1^{(t)}, x_2^{(t)}, ..., x_n^{(t)})^T$. The position update formula for cuckoo individuals is:

$$x_i^{t+1} = x_i^t + \alpha \oplus Levy(\lambda) \tag{23}$$

In Eq. (23), $\alpha = O(L/10)$ is the step size and direction of Levy's flight, L is the range

of the search space, $\oplus$ is the point-to-point product, x_i^t is the position of the i th cuckoo in the t th generation, and $Levy(\lambda)$ is the Levy distribution, which can be expressed as:

$$Levy(\lambda) \sim u = t^{-\lambda}, (1 < \lambda < 3) \tag{24}$$

Calculate the fitness value of this generation of cuckoos and find the optimal cuckoo individual $x_i^{(t)}$;

Step 4: Randomly generate a small number m in the range of $[0,1]$, and compare the size of m with the probability of discovery p. If $m < p$, then update the cuckoo's position according to equation (23), calculate and compare the fitness values of the new cuckoo individual and the original cuckoo individual, and keep the cuckoo individual with the larger fitness value to obtain the updated cuckoo $X^{t+1} = \left(x_1^{(t+1)}, x_2^{(t+1)}, \cdots,)x_n^{(t+1)}\right)^T$; if $m \geq p$, keep the original cuckoo;

Step 5: Judge whether the optimal cuckoo individual satisfies the conditions or whether the number of iterations reaches the requirement, if yes, then decode the optimal cuckoo to obtain the optimal weights and thresholds, and assign them to the CS-BPNN model. Instead, step 3 is performed;

Step 6: Take the posture input data as the input of the CS-BPNN model, take the posture values as the output of the CS-BPNN model, and train the CS-BPNN model with a sufficient number of samples to complete the mapping of the posture data to the posture values;

Step 7: Input the posture indicator data into the CS-BPNN model with evaluation capability, and obtain the posture value of the network after mapping.

3.3 Experimental Analysis of Cybersecurity Posture Assessment Methods

3.3.1 Experimental data and processing

The research in this paper uses the data obtained from the real SDN converged network environment, for the acquisition of the data source information in the experiment is mainly obtained through the following ways, namely: sFlow-based information acquisition, SNMP-based traditional switch information acquisition, Snort-based alarm log information, SDN switch information based on the REST API of SDN. Based on these ways, the acquired information extracts the important data fields needed in this paper.

In this paper, based on the above acquired key index elements, we extract the eigenvalues and construct the dataset by collecting the data generated in the actual network and extracting the eigenvalues according to the quantitative calculation formula. As the raw data acquired through the SDN controller needs to be quantized, the data is mapped into the range of 0 to 1, and different indicators are mapped into the same interval range so that they have equal importance in calculating the weights.

$$x_i = \frac{x_i' - x_{\min}}{x_{\max} - x_i'} \tag{25}$$

where x_i denotes the value of the posture indicator after normalization, x_i' denotes the value of the posture indicator, $x_{\min}$ denotes the minimum value in the dataset, and $x_{\max}$ denotes the maximum value in the dataset.

In this paper, according to the different impacts caused by the changes in the actual network

environment, multiple experts score, so as to obtain the real value of the network posture. In order to avoid the influence of subjective factors in the expert scoring process, the scoring of multiple experts is processed in this paper. In the evaluation process, the concentration and dispersion of scoring by multiple experts are the indicators used to assess the consistency and credibility of expert scoring. Concentration refers to the average of the scores scored by multiple experts, which reflects the degree of concentration of expert scoring, and calculates the degree of concentration of expert scoring. Dispersion refers to the variability between the scores of multiple experts, reflecting the degree of decentralization of expert scoring, and is used to calculate the degree of dispersion of expert scoring.

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} x_i, i = 1, 2, \cdots, n \tag{26}$$

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{X})^2}, i = 1, 2, \cdots, n \tag{27}$$

where n represents the total number of experts and x_i represents the score of a particular expert.

In order to facilitate the analysis of the network posture condition, this paper uses the value of 0~1 to quantitatively analyze the posture assessment value for the purpose of quantitative assessment, and divides the posture level of SDN environment into five levels, and the level division is shown in Table 1.

Table 1: Network posture hierarchy

Posture level	Range	Index standard	Safety risk	Business impact
Superior	(0.8,1.0]	The bandwidth is sufficient, the response time is fast, the load balancing	Low	Nothing
Good	(0.6,0.8]	The bandwidth is sufficient, the response time is faster and the load is more balanced	General	Slight
Medium	(0.4,0.6]	The bandwidth basically meets the requirements, the response time is longer, the load inequality is balanced	Medium	General
Difference	(0.2,0.4]	The bandwidth is insufficient, the response time is long, the load is severely uneven	Height	Severity
Aberration	[0.0,0.2]	Network service is not available	Very high	Catastrophic

3.3.2 Analysis of experimental results

In this paper, the mean square error is used as the fitness value of the CS algorithm, and the change of this value can react to a certain extent to evaluate the convergence of the model on the training set. In this paper, two different algorithms for optimizing BP neural networks are plotted separately to find the parameter optimum in the process of the change curve of the fitness value of the particle swarm individuals and cuckoo individuals, and the fitness graphs of the different algorithms are shown in Fig. 4.

It can be found that: the particle swarm individual in the PSO algorithm converges to the

optimum after 36 iterations, with a value of 0.0401, and the cuckoo individual in the CS algorithm converges to the optimum after 24 iterations, with a value of 0.0251, which shows that the CS algorithm achieves the expected effect and is feasible.

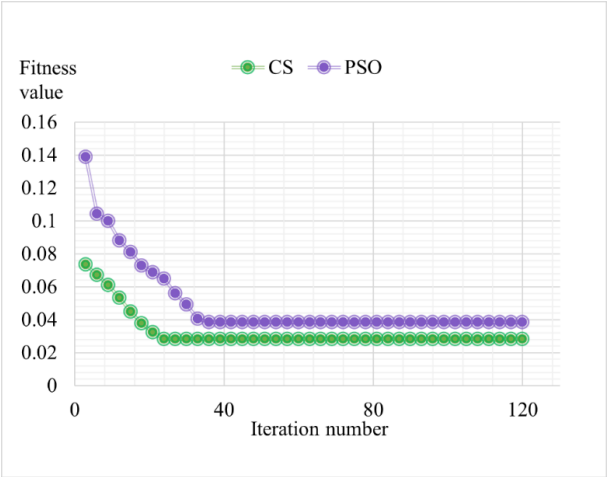
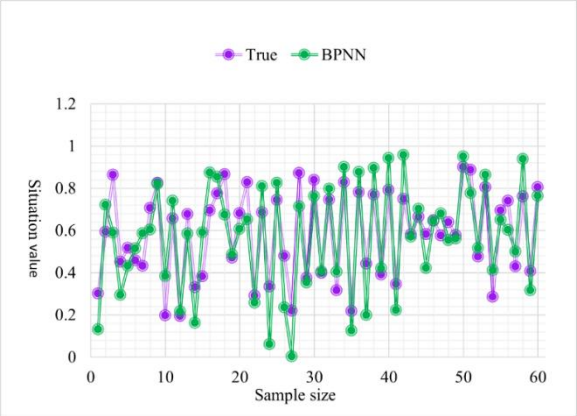


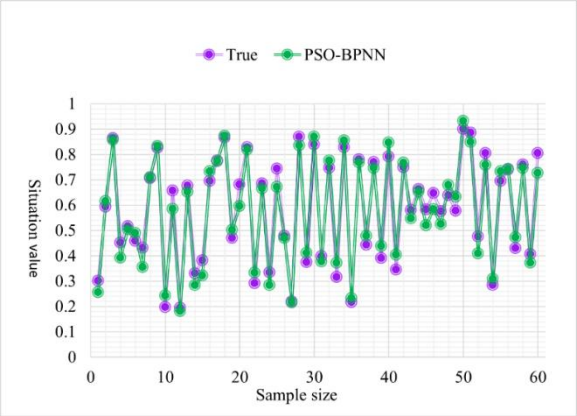
Figure 4: The fitness diagram of different algorithms

In order to better verify the effectiveness of the algorithms in this paper in network posture assessment, the algorithms proposed in this paper are compared with the BPNN and PSO-BPNN algorithms. The comparison between the evaluation results obtained from the three evaluation models and the real posture values through the dataset constructed in this paper is shown in Figure 5. Figures (a)~(c) show the evaluation results of BPNN, PSO-BPNN and CS-BPNN models, respectively.

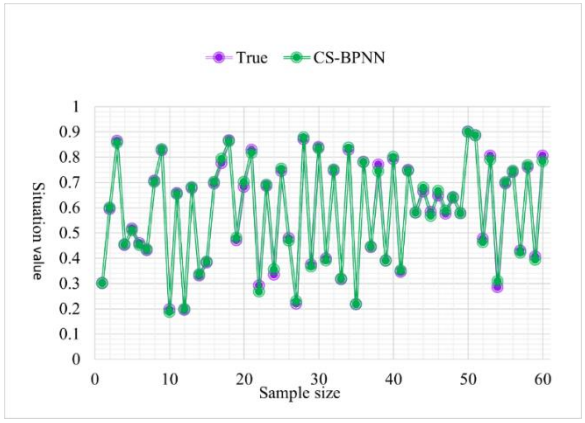
The CS-BPNN network posture assessment model proposed in this paper is closest to the trend of the real value, and its curve is basically consistent with the real value curve. In contrast, the other three assessment models show different degrees of deviation.The BPNN assessment model has the worst curve fit with the true value, showing that the assessment accuracy of this model is relatively low. The PSO-BPNN model, on the other hand, improves relative to the BPNN model, but is still not as good as the CS-BPNN model. Therefore, it can be concluded that the CS-BPNN network posture assessment model proposed in this paper has better accuracy and can assess the state of the network more accurately.



(a)The results of the BPNN model



(b)The results of the PSO-BPNN model



(c)The results of the CS-BPNN model

Figure 5: The comparison between the results and the real situation values

Figure 6 shows the trend of the absolute error value during the training process. The figure can clearly show the fluctuation of the error value during the training process. It can be seen that: the CS-BPNN posture assessment model proposed in this paper has the smallest error in the training process. Compared with other evaluation models, the error curve of the CS-BPNN posture evaluation model is the closest to 0, which can also corroborate that the evaluation model proposed in this paper has smaller error and higher accuracy.

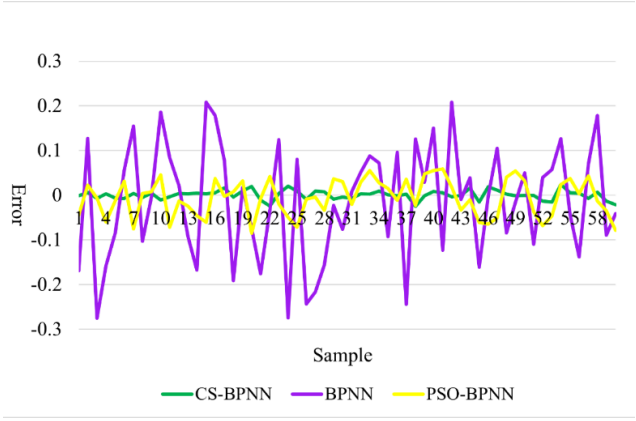


Figure 6: The change in the absolute error value

4 A GWO-LSTM based approach for SDN network posture prediction

4.1 LSTM model

LSTM, or Long Short-Term Memory Model, is a kind of recurrent neural network (RNN) with improved hidden layer neuron structure.

The three gate structures of forgetting gate, input gate, and output gate added in the LSTM hidden layer neurons can control the transmission of information more efficiently and effectively reduce the accumulation of non-primary information in the transmission system, thus effectively making up for the problems existing in RNN.

LSTM is able to learn the nonlinear mapping between data very well, in the process of LSTM model training, the input of each LSTM neuron cell contains three parts of the last

moment unit state C_{t-1} and the last moment LSTM output h_{t-1} and the current moment input X_t, at the same time, the input gates, the forgetting gates and the output gates within the neuron cell complete the information The input gate, forgetting gate and output gate in the neural unit fulfill the functions of information selection and information conversion.

The forgetting gate is used to select whether the information of the previous moment is retained or forgotten, 1 means retained and 0 means discarded, w_{fx}, w_{fh} and b_f are the corresponding weights and bias parameters of the input gate, respectively, h_{t-1} denotes the information of the output of the previous moment, and X_t is the input of the current moment:

$$f_t = \sigma(w_{fx} h_{t-1} + w_{fh} X_t + b_f) \in \{0,1\} \tag{28}$$

The input gate is responsible for computing the generation of the current input reservation information i_t and the new state information C_t', and w_{ix}, w_{ih}, b_i, w_{cx}, w_{ch}, and b_c are the corresponding weights and bias parameters of the oblivion gate:

$$\begin{cases} i_t = \sigma(w_{ih} h_{t-1} + w_{ix} X_t + b_i) \\ C_t' = \tanh(w_{ch} h_{t-1} + w_{cx} X_t + b_c) \end{cases} \tag{29}$$

The current hidden layer neuron state of LSTM is updated as $C_t = f_t * C_{t-1} + i_t * C_t'$;

The output gate represents the output information h_t at the current moment, which is the result of the joint action of the state of the previous moment, the current input and the state of the implicit layer, and w_{cx}, w_{oh}, b_o are the corresponding weights and bias parameters of the output gate:

$$\begin{cases} o_t = \sigma(w_{oh} h_{t-1} + w_{ox} X_t + b_o) \\ h_t = o_t * \tanh C_t \end{cases} \tag{30}$$

In the transfer process, after the information is output through the LSTM hidden layer neurons, the sigmod function is generally used in the output layer for convergence, i.e., the output layer is calculated as:

$$Y_t = \sigma\left(w_{yh} h_t\right) \tag{31}$$

Among them, w_{yh} is the output layer weight that participates in training and iterative update along with w_{cx}, w_{oh}, b_o, w_{fx}, w_{fh}, b_f, w_{ix}, w_{ih}, b_i, w_{cx}, w_{ch}, b_c, until the iteration reaches the maximum critical condition K and stops the training. The current optimal parameters are saved to construct the LSTM network model.

4.2 GWO Algorithm

As a novel swarm intelligence optimization algorithm, GWO performs optimization by mimicking the social hierarchy and hunting behavior of gray wolves. Gray wolves in a gray wolf pack are strictly stratified according to their social hierarchy, and are divided into four main tiers, from high to low, namely Alpha wolves, Beta wolves, Delta wolves, and ordinary wolves.Alpha wolves are the leaders who are responsible for determining the direction of the

pack, and Beta wolves are the youthful decision makers who help the Alpha wolves in other activities, such as decision making.Delta wolves, the part of the pack, are composed of the sentinels, scouts, hunters, elders, and caretakers who provide awareness to the pack. Hunters, Elders, and Caretakers make up the Surveyors who provide awareness to the pack. The Common wolf, the lowest social level, on the other hand, is dependent on the other major parts of the wolves in the first three social classes. Gray wolves have the ability to remember where their victims are around them.

The GWO algorithm effectively incorporates the characteristics of the pack system of gray wolves and builds a model with α, β, δ, and ω as hierarchical representatives. The algorithm allows the potential solution to the corresponding solving problem to be represented by the position of the gray wolf in the pack, while the degree of approximation of the solution can be illustrated by the rank of the gray wolf. Where α represents the wolf that is currently in the optimal position, followed by β and δ which represent the second and third optimal position wolves, respectively, and the remaining wolves update their positions based on the α, β, and δ wolves. In order to obtain the position of individual wolves in the corresponding wolf pack, the gray wolf algorithm establishes a position fitness function related to the solution parameters, and obtains the social class of each wolf through comparison, and iteratively searches for optimization to obtain the optimal solution of the algorithm while controlling the search direction. The algorithm is described as follows:

On the D-dimensional search space, the position of each wolf is a vector $X_i(t) = \left(X_{i,1}(t), X_{i,2}(t), ..., X_{i,D}(t) \right)$, where $i = 1, 2, 3, ..., N$ denotes that the pack consists of N wolves. When the gray wolf X_i performs a position update, it first calculates its distance from the best three wolves $X_{best}(t) = \left\{ X_\alpha(t), X_\beta(t), X_\delta(t) \right\}$ at the current position, which is given by the formula:

$$\begin{cases} D_\alpha = C_1 X_\alpha - X_i(t) \\ D_\beta = C_2 X_\beta - X_i(t) \\ D_\delta = C_3 X_\delta - X_i(t) \end{cases} \tag{32}$$

where $X_i(t)$ denotes the position of the ith gray wolf at moment t.

Then, X_i updates its position by the formula $X(t+1) = \left(X_1'(t) + X_2'(t) + X_3'(t) \right) / 3$, which represents the direction of approaching to the optimal three gray wolves, i.e., the optimal solution, in the wolf pack. Where $X(t+1)$ denotes the final movement result of the gray wolf, $X_1'(t)$, $X_2'(t)$, and $X_3'(t)$ are the vectors of the gray wolf X_i to be moved towards α, β, and δ, respectively, which are calculated as follows:

$$\begin{cases} X_1'(t) = X_\alpha - A_1 D_\alpha \\ X_2'(t) = X_\beta - A_2 D_\beta \\ X_3'(t) = X_\delta - A_3 D_\delta \end{cases} \tag{33}$$

a is the convergence factor, which is introduced to better constrain the parameter optimization. Eq. It can be seen that as the number of iterations increases, the value of the convergence factor decreases linearly from 2 to 0, resulting in a decrease in the value of $|A_k|$ from greater than 1 to a state less than 1, thus controlling the optimization direction.

4.3 Security posture prediction with LSTM based on GWO optimization

In order to make the SDN network security posture prediction more efficient, the GWO algorithm is used to optimize the LSTM prediction model to obtain the improved model of LSTM, i.e., the security posture prediction model based on GWO-LSTM, and the model structure is shown in Fig. 7.

The GWO-LSTM prediction model firstly needs to determine the optimized objective parameters in the LSTM network and the individual fitness calculation function of the Gray Wolf optimization algorithm.

According to the structure of the LSTM network, the spatial size of each parameter can be obtained as $\{w_{fh}, w_{ih}, w_{oh}, w_{ch}\} \in R^a$, $\{w_{fx}, w_{ix}, w_{ox}, w_{cx}\} \in R^b$, $\{b_f, b_i, b_o, b_c\} \in R^d$, $w_{yh} \in R^g$. where $a = hidden\_num \times hidden\_num$, $b = hidden\_num \times input\_num$, $d = hidden\_num \times 1$, and $g = hidden\_num \times output\_num$. Where the corresponding $input\_num$, $hidden\_num$ and $output\_num$ are the number of neurons in the input, hidden and output layers, respectively, and the same as the LSTM model $input\_num$ corresponds to the prediction step, and the predicted output corresponds to the sample sequence data intercepted at the $output\_num$ prediction step. The location $pos(i)$ of the Gray Wolf optimization algorithm should contain all the above parameters to be optimized, i.e., its parameters can be characterized as the following equation (34), where, $i = 1, 2, ..., N$, N is the number of gray wolves.

$$pos(i) \in R^{4a+4b+4d+g} \tag{34}$$

In the process of solving optimization, the selection of individual fitness function is especially important for the swarm optimization algorithm, and the appropriateness of the fitness function has a direct impact on the convergence speed of the algorithm and whether the optimal solution can be found. In order to better control the optimization direction of the LSTM prediction model and achieve the purpose of iterative convergence, the fitness function should correspond to the model optimization objective - the training mean square error MSE. therefore, in this paper, the fitness function of the gray wolf optimization algorithm is set as:

$$fitness(l) = \left(Y' - Y\right)^T \left(Y' - Y\right) \tag{35}$$

where Y' is the LSTM model output value, Y is the corresponding label value of training sample X, and l is the current iteration number. Obviously, when $fitness$ reaches the minimum, the LSTM prediction model has the optimal MSE, and the optimal LSTM neural network prediction model can be obtained by saving the network optimization parameters at this time.

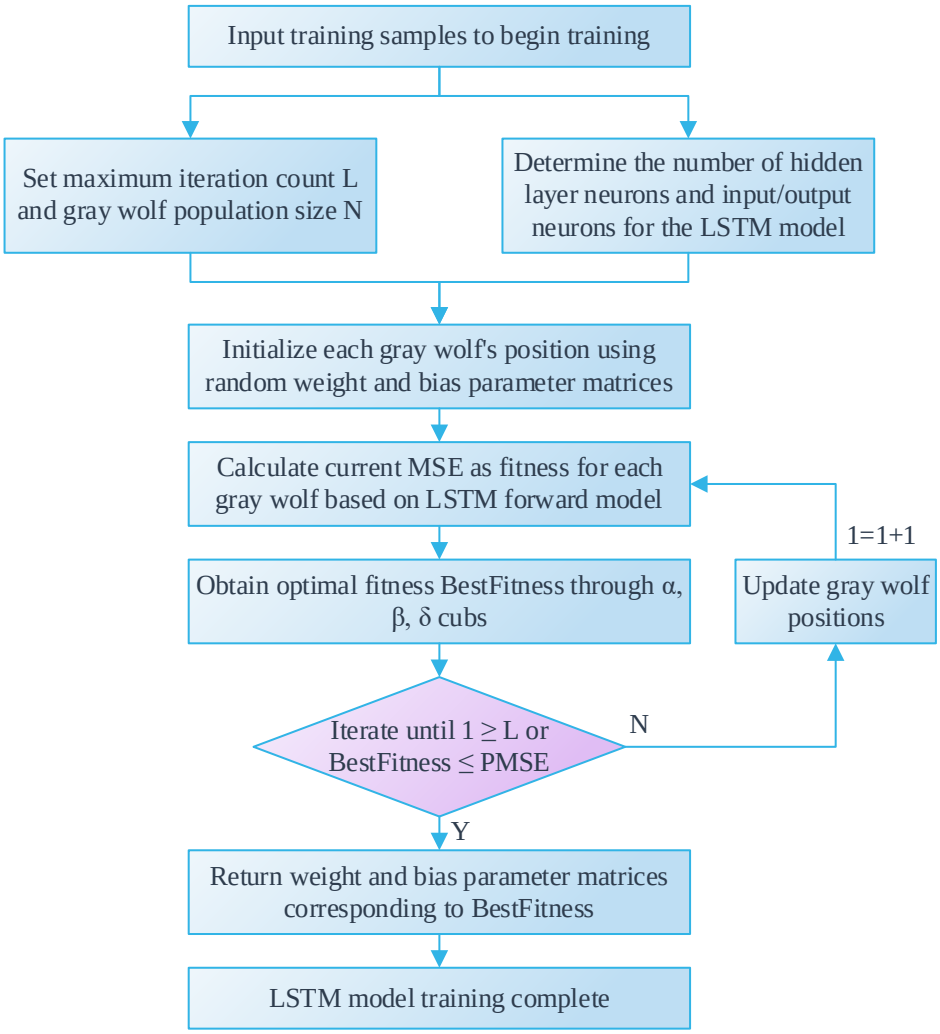


Figure 7: GWO optimizes the LSTM model process

4.4 Experimental validation

This section tests the security posture prediction model for SDN data plane. It is mainly categorized into the prediction of security posture risk value and the prediction of security event frequency. The feasibility, validity, stability and accuracy of the combined model prediction are verified.

The assembled desktop used in this experiment, the graphics card is Nvidia GeForce RTX 2060, and the system is Windows 10 64-bit with TensorFlow GPU version, and the experiment is conducted with Python 3.6.

4.4.1 Criteria for evaluation of experimental results

The mean square error MSE is selected as the fitness function of nonlinear dynamic particle swarm algorithm in the model and also as the loss function of the neural network.The RMSE is the result of the open square of MSE, where the measure is the measure of the predicted data, and therefore instead of MSE is used as the evaluation criterion of the experimental results. The mean absolute error MAE and mean absolute percentage error MAPE are also selected as the other two evaluation criteria, in which the measure of MAE is also the measure of the prediction data, while MAPE calculates the percentage of error, so MAPE can be used to calculate the model prediction accuracy.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \tag{36}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2} = \sqrt{MSE} \tag{37}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|f_i - y_i\right| \tag{38}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{f_i - y_i}{y_i}\right| = \frac{MAE}{\bar{y}} \tag{39}$$

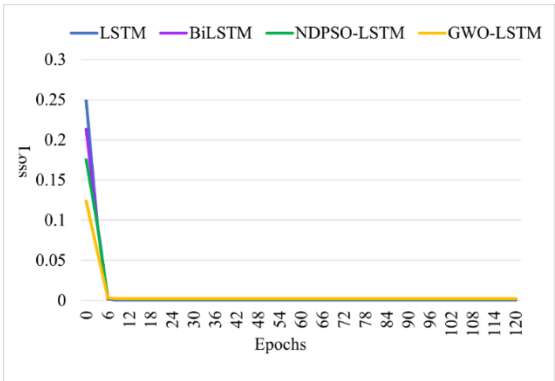
where f_i is the predicted value of the security posture indicator, y_i is the actual value of the security posture indicator, $\bar{y}$ is the average value of the actual value, and n is the scale size of the test data. The results of the above formulas are all errors, so the smaller the calculation result, the higher the prediction accuracy.

4.4.2 Analysis of the results of the value-at-risk prediction of the security posture

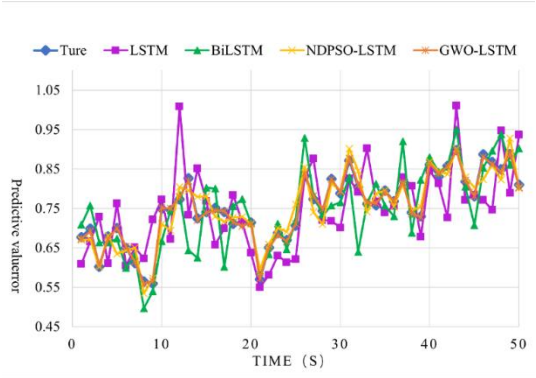
Based on the results of security posture assessment, the risk value curve of security posture assessment can be generated. Four experimental models, LSTM, BiLSTM, NDPSO-LSTM and GWO-LSTM, are selected for the experiment, and the experimental dataset is utilized for training and testing. The prediction results are shown in Fig. 8.

Each generation of training process in this experiment covers about 1000 sets of training data. Figures (a) to (c) show the loss function iteration effect, normalized predicted value vs. true value, and prediction error comparison, respectively. From Fig. (a), it can be seen that during the actual training of the model, whether it is the two models with empirically set structural parameters or the two combined models utilizing the two algorithms, the function finding speed of the four models is relatively fast, and the model can hardly have any growth in prediction effect after about 6 generations of training in the set training process of 120 generations. The BiLSTM model in Fig. (b) has a larger prediction error than the LSTM model, and the prediction value of the former is relatively more conservative, while the prediction style of the latter is more aggressive. Both the LSTM model optimized by NDPSO and the LSTM model optimized by GWO have better prediction results, with the GWO-LSTM visually closest to the true value curve.

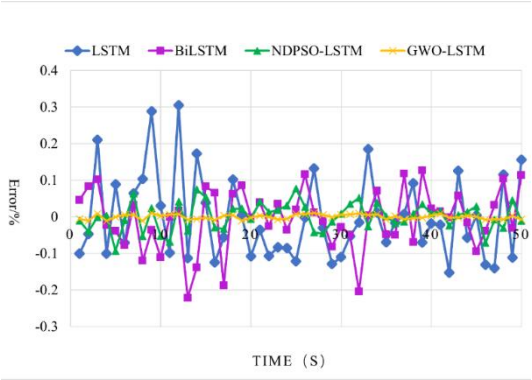
In Fig. (c), the absolute value of the prediction error does not exceed 0.4% among the 50 sets of results of the prediction experiments. The security posture risk value represents the size of the danger of the network environment, the larger the value is the more dangerous, then the prediction value bias will make the security management personnel biased towards overestimating the danger of the network environment, and it will be safer for the whole system.



(a)Loss function iteration effect



(b)The predicted value is compared to the real value



(c) Prediction error contrast

Figure 8: Predictive result

Table 2 shows the comparison of the prediction results showing that the LSTM model possesses 95.75% prediction accuracy, the NDPSO-LSTM model prediction accuracy is 95.95%, and the latter prediction accuracy is improved by 0.20 percentage points compared to the former, the BiLSTM model possesses 95.90% prediction accuracy, and the GWO-LSTM model prediction accuracy is 96.79%, and the GWO-LSTM The prediction accuracy of GWO-LSTM is improved by 0.84 percentage points over NDPSO-LSTM.

Empirically, the BiLSTM model predicts better than the LSTM model when the same structural parameters are set; the model predicting the number of hidden layer nodes with GWO is better than the model predicting the number of hidden layer nodes with NDPSO. In summary, the GWO-LSTM combined prediction model has the highest prediction accuracy in the experiment.

Table 2: Comparison shows

Model	MSE	RMSE	MAE	MAPE(%)
LSTM	0.000092	0.009592	0.00782	4.2461
BiLSTM	0.000083	0.009110	0.00744	4.0983
NDPSO-LSTM	0.000087	0.009327	0.00773	4.0535
GWO-LSTM	0.000055	0.007416	0.00594	3.2146

4.4.3 Analysis of results of security incident frequency prediction

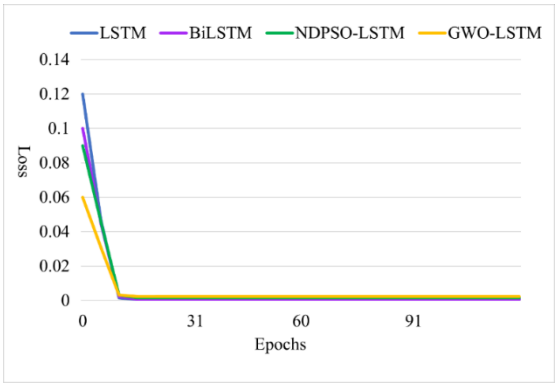
Among the metrics of security posture assessment, the frequency of security events is an important parameter to calculate. At the same time, the frequency of security events also

represents the current threat level of different security events. Therefore, if this frequency indicator can be predicted more directly, the number of network security events in the future period can be obtained indirectly through calculation. Therefore, the frequency of security events calculated with 150 minutes as the statistical time period is selected for the prediction experiment, and the test results are shown in Figure 9.

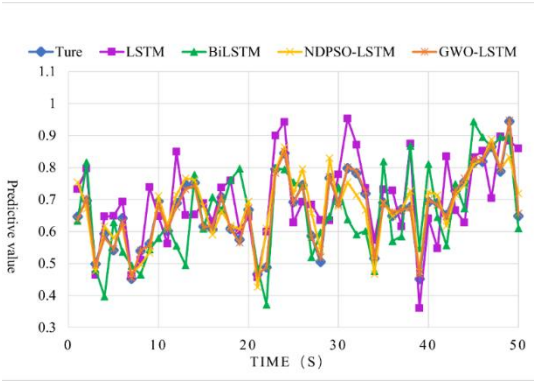
Each generation of training process in this experiment covers about 1000 groups of training data. Figures (a) to (c) show the loss function iteration effect, normalized prediction value vs. true value and prediction error comparison, respectively. From Fig. (a), it can be seen that the function optimization speed of the four models is relatively fast, and in the training process of the set 120 generations, the loss function value of the combined model decreases faster in the first 9 generations of training, and the loss function value of the combined model decreases slowly but with a lower value in the latter half of training, which also reflects the idea of the GWO algorithm proposed in this paper that the search for optimal solutions is faster in the early stage and more refined in the later stage. The idea of the GWO algorithm in this paper is also reflected here.

In Fig. (b), the prediction effect of the GWO-LSTM model is closer to the real value, which meets the expectation of the experiment.

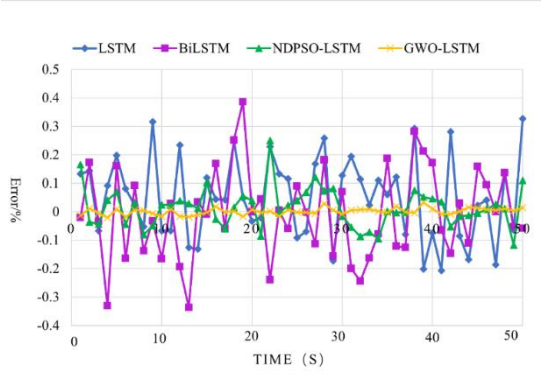
In Fig. (c), the absolute value of the prediction error does not exceed 0.5% in 50 groups of experimental results. Especially for the GWO-LSTM model, the error is in the range of [-0.05%,0.05%], and the predicted frequency of security events is usually on the large side, which is rather a slight long term for the system security administrators, and keeping a cautious attitude towards the predicted values can make the system more secure.



(a)Loss function iteration effect



(b)The predicted value is compared to the real value



(c) Prediction error contrast

Figure 9: Predictive result

Table 3 shows the results of security event frequency prediction, and the prediction accuracies of the four models are 94.92%, 95.86%, 95.18%, and 96.53%, respectively. This shows that when adjusting the neural network structure parameters of this experiment, the model accuracy can be higher by calculating the hidden layer node values according to the GWO method, and GWO-LSTM is the optimal model in this experiment.

Table 3: Safety event frequency prediction results

Model	MSE	RMSE	MAE	MAPE(%)
LSTM	0.82752	0.90968	0.72202	5.0831
BiLSTM	0.57594	0.75891	0.59269	4.1448
NDPSO-LSTM	0.70572	0.84007	0.68068	4.8189
GWO-LSTM	0.48231	0.69449	0.50011	3.4668

5 Conclusion

This study addresses the problems of incompatibility and management complexity among different types of network devices in modern networks. Based on the SDN technology framework, it explores the SDN rule customization and network feature-driven security response mechanism for electric power business traffic. Through feature extraction and modeling of electric power logistics terminal traffic behavior, the sharing and optimization of network resources are realized, and the accuracy of the network security response mechanism is improved.

In the security posture assessment session, a network posture assessment model based on CS-BPNN is proposed. The BP neural network parameters are optimized using the CS algorithm, and the power IoT business traffic network security posture indicators are input into the network posture assessment model. The experimental results of the posture assessment show that the CS-BPNN posture assessment algorithm proposed in this paper is better than other comparative algorithms in terms of comparison of assessment results, individual adaptability, and error, etc. The algorithm in this paper converges to the optimal value of 0.0251 in only 24 iterations, and the overall error curve is the closest to 0, which proves that the CS-BPNN algorithm proposed in this paper has a better accuracy in the network security posture assessment.

Aiming at the problem of security posture prediction for power IoT, a GWO-LSTM security posture prediction model for SDN data is proposed. The Gray Wolf optimization algorithm is introduced to improve the prediction accuracy and stability of the LSTM prediction model, and enhance the adaptability of the model to different sample data. The model in this paper is compared with other models such as NDPSO-LSTM to verify the effectiveness and stability of the GWO-LSTM model. The prediction accuracy of the GWO-LSTM model reaches 96.79% and 96.53% on different problems of predicting the risk value of cybersecurity posture and the frequency of security events, respectively, which meets the performance requirements of real-world applications.

Funding

Fund Project: Research on Digital Grid Asset Characteristic Recognition Technology and Automated Disposal Technology (project number: 066700KK52230001).

About the Author

Binyuan Yan (1989.9-), male, Miao ethnicity, native of Guiyang, Guizhou Province, Senior Engineer, Bachelor's degree, Research focus: Information Security, Information Security Protection Systems.

References

[1] Bedi, G., Venayagamoorthy, G. K., Singh, R., Brooks, R. R., & Wang, K. C. (2018). Review of Internet of Things (IoT) in electric power and energy systems. IEEE Internet of things Journal, 5(2), 847-870.

[2] Wang, Q., & Wang, Y. G. (2018, October). Research on power Internet of Things architecture for smart grid demand. In 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2) (pp. 1-9). IEEE.

[3] Paukstadt, U., & Becker, J. (2021). Uncovering the business value of the internet of things in the energy domain–a review of smart energy business models. Electronic Markets, 31(1), 51-66.

[4] Karakus, M., & Durresi, A. (2017). A survey: Control plane scalability issues and approaches in software-defined networking (SDN). Computer Networks, 112, 279-293.

[5] Baddeley, M., Nejabati, R., Oikonomou, G., Sooriyabandara, M., & Simeonidou, D. (2018, June). Evolving SDN for low-power IoT networks. In 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft) (pp. 71-79). IEEE.

[6] Fan, B., Tan, H., & Li, Y. (2023). Critical link identification algorithm for power communication networks in SDN architecture. International Journal of Critical Infrastructure Protection, 40, 100584.

[7] Lu, Z., Sun, C., Cheng, J., Li, Y., Li, Y., & Wen, X. (2017). SDN-Enabled Communication Network Framework for Energy Internet. Journal of Computer Networks and Communications, 2017(1), 8213854.

[8] Wang, X. Z., Gao, J. Z., & Zhang, X. M. (2023, April). Load balancing strategy of power communication network based on SDN controller. In Journal of Physics: Conference Series (Vol. 2476, No. 1, p. 012073). IOP Publishing.

[9] An, G. C., Wang, X. Z., & Wang, Q. (2023, April). Reliability Analysis of Next Generation Power Communication Network Based on SDN. In Journal of Physics: Conference Series (Vol. 2476, No. 1, p. 012080). IOP Publishing.

[10] Zhang, S., Song, Y., Zhen, C., & Chen, Y. (2018, June). Intelligent power service and improvement analysis of communication network based on SDN. In IOP Conference Series: Materials Science and Engineering (Vol. 366, No. 1, p. 012052). IOP Publishing.

[11] Chai, R., Yang, X., Du, C., & Chen, Q. (2021). Network cost optimization-based capacitated controller deployment for SDN. Computer Networks, 197, 108326.

[12] Bannour, F., Souihi, S., & Mellouk, A. (2020). Adaptive distributed SDN controllers: Application to content-centric delivery networks. Future Generation Computer Systems, 113, 78-93.

[13] Sapkota, B., Dawadi, B. R., & Joshi, S. R. (2024). Controller placement problem during SDN deployment in the ISP/Telco networks: A survey. Engineering Reports, 6(2), e12801.

[14] Hou, R., Ren, G., Zhou, C., Yue, H., Liu, H., & Liu, J. (2020). Analysis and research on network security and privacy security in ubiquitous electricity Internet of Things. Computer communications, 158, 64-72.

[15] Maziku, H., Shetty, S., & Nicol, D. M. (2019). Security risk assessment for SDN-enabled smart grids. Computer Communications, 133, 1-11.

[16] Lee, S. H., Son, K. S., Jung, W., & Kang, H. G. (2017). Risk assessment of safety data link and network communication in digital safety feature control system of nuclear power plant. Annals of Nuclear Energy, 108, 394-405.

[17] Dong, X., Lin, H., Tan, R., Iyer, R. K., & Kalbarczyk, Z. (2015, April). Software-defined networking for smart grid resilience: Opportunities and challenges. In Proceedings of the 1st ACM workshop on cyber-physical system security (pp. 61-68).

[18] Usman, M., Gebremariam, A. A., Raza, U., & Granelli, F. (2015). A software-defined device-to-device communication architecture for public safety applications in 5G networks. IEEE Access, 3, 1649-1654.

[19] Kumar, P., Kumar, R., Aljuhani, A., Javeed, D., Jolfaei, A., & Islam, A. N. (2023). Digital twin-driven SDN for smart grid: A deep learning integrated blockchain for cybersecurity. Solar Energy, 263, 111921.

[20] Girdhar, M., Hong, J., Su, W., Herath, A., & Liu, C. C. (2024, July). SDN-Based dynamic cybersecurity framework of IEC-61850 communications in smart grid. In 2024 IEEE Power & Energy Society General Meeting (PESGM) (pp. 1-5). IEEE.

[21] Kumar, A., Dhabliya, D., Agarwal, P., Aneja, N., Dadheech, P., Jamal, S. S., & Antwi, O. A. (2022). Cyber-Internet Security Framework to Conquer Energy-Related Attacks on the Internet of Things with Machine Learning Techniques. Computational intelligence and neuroscience, 2022(1), 8803586.

[22] Al-Rubaye, S., Kadhum, E., Ni, Q., & Anpalagan, A. (2017). Industrial internet of things driven by SDN platform for smart grid resiliency. IEEE Internet of Things Journal, 6(1), 267-277.

[23] Sarker, P. S., Sadanandan, S. K., & Srivastava, A. K. (2022). Resiliency metrics for monitoring and analysis of cyber-power distribution system with IoTs. IEEE Internet of Things Journal, 10(9), 7469-7479.

[24] Djenna, A., Harous, S., & Saidouni, D. E. (2021). Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure. Applied sciences, 11(10), 4580.

[25] Sarjan, H., Ameli, A., & Ghafouri, M. (2022). Cyber-security of industrial internet of things in electric power systems. IEEE Access, 10, 92390-92409.