Enhanced Network Security Hybrid Cloud Workflow Scheduling Using Levy-Optimized Slime Mould Algorithm

Jingang Guan
College of Artificial Intelligence, Yantai Institute of Science and Technology, Yantai 265600, China
E-mail: guanjingang721215@163.com
*Corresponding author

Keywords: SMA, network security, hybrid cloud, workflow scheduling, swarm intelligence optimization

Received: October 14, 2024

Mainstream network security models based on PSO and other algorithms have not been optimized for hybrid cloud environments, resulting in limitations in their application. To ensure network security in hybrid cloud environments, the study first introduces the Levy distribution, uses the Levy distribution to generate random numbers and redesign the position update formula. Then, the penalty parameters of SVM and the radial basis kernel function parameters of SVM are used as individuals to improve SMA. After iteration, the optimal position is the optimal solution of the two parameters. After determining the optimal parameters, an improved slime mould algorithm optimization algorithm is designed. Finally, the algorithm is applied to the encoded hybrid cloud workflow scheduling to propose a method based on the improved slime mould algorithm. The test results showed that when the number of tasks was 1000, the proposed algorithm took 13.7 hours to run, which was better than the traditional 18.2 hours and more than 50 hours of the particle swarm algorithm (P<0.05). Except for Guess passward attack, the detection rate of the proposed algorithm exceeded 91%, while that of other algorithms was between 77% and 91%. For Guess passward attacks, the detection rate of the traditional slime mould algorithm was 8.93%, but the proposed algorithm increased to 11.54% (P<0.05). After applying this method, the task completion time in the hybrid cloud environment reduced by 61.9%, the resource utilization rate increased by 27.5%, and the incidence of network security events reduced by 87.9%. Based on the testing results, the proposed algorithm has considerable detection performance for network security issues in hybrid cloud environments. It can correctly identify the types of network attacks in network security monitoring and improves its application effectiveness in network security.

Povzetek: Raziskava predstavlja izboljšan algoritem za razporejanje delovnih tokov v hibridnem oblaku, ki z Levy-optimiziranim Slime Mould algoritmom poveča varnost omrežij ter učinkovitost in zaznavo napadov.

1 Introduction

Hybrid cloud is a special cloud computing architecture that combines both public and private clouds, including scalability, high resource scale, and security [1]. It refers to a cloud computing model that combines two or more cloud computing deployment models (public cloud and private cloud). In this environment, enterprise organizations or institutions can deploy different types of applications, data, and services on public clouds and private cloud, and make dynamic adjustments and management if necessary. The hybrid cloud environment can combine the high availability, flexibility and cost-effectiveness of the public cloud, as well as the data security of the private cloud. However, in workflow scheduling in a hybrid cloud environment, due to excessive computing resources and complex timing issues, it faces more complex network security issues than workflow scheduling in other environments [2]. The current main network security models lack customization and optimization for hybrid cloud environments, so there are limitations in their application in this environment. Therefore, a network security hybrid cloud workflow scheduling algorithm based on Slime Mould Algorithm (SMA) is proposed. SMA is a swarm intelligent optimization algorithm that simulates the shrinkage process of slime moulds when searching for food [3]. Compared with traditional Particle Swarm Optimization (PSO) algorithms, this algorithm has higher adaptability, efficiency, and parallelism [4]. Firstly, to address the issue of uniform distribution falling into local optima, Levy distribution is introduced to generate random numbers and redesign the position update formula. Then, to determine the slow defects of SVM in the invasion detection, the penalty parameters and radial basis kernel function parameters of SVM are taken as the individuals to improve the SMA. After iteration, the optimal myxobacteria location is the optimal solution of these two parameters. After confirming the optimal parameters, an improved myxobacteria optimization algorithm is designed. Finally, the proposed algorithm is applied to the encoded hybrid cloud workflow scheduling to solve the too large matrix solution space. The algorithm is built to increase data security in a hybrid cloud environment and fill in the gap in network security issues.

This paper is divided into six sections. The second section is related works, which reviews recent literature and elaborates on the significance of this paper. The third section describes the principle and derivation process of the proposed scheme. The fourth section demonstrates the performance of the proposed method through experiments. The fifth section discusses and compares the experimental results. The sixth section is the results, which summarize the entire paper.

This study makes substantial contributions to the research in this field in the following areas. First, the Levy flight model effectively addresses the local optimization in SMA, resulting in improved efficiency. Second, Support Vector Machine (SVM) integrated with Enhanced SMA for network security tasks. Lastly, the application of SMA in network security protection in hybrid cloud environments provides potential directions for the overall development of this field.

2 Related works

Cloud computing is an important evolution direction of Internet and data technology. Many scholars have also explored the security issues of cloud computing. Abroshan believed that current cloud computing required an encryption scheme that had a small impact on performance to improve the security of cloud data without impeding computing efficiency. An encryption algorithm based on the improved Blowfish algorithm and elliptic curve algorithm was proposed [5]. The evaluation results showed that the throughput, execution time, and memory consumption parameters of this algorithm were superior to traditional algorithms. To use the network intrusion detection model in cloud-based systems and identify known and unknown attacks, Vashishtha L K et al. introduced a hybrid intrusion detection model for cloud-based systems that could detect all types of attacks using signature-based detection and exception-based detection. The detection rates of the model on UNSW-NB 15, CICIDS and NSL-KDD datasets were 92.7%, 85.1% and 99.8%, respectively, outperforming some existing models [6]. The contemporary intrusion detection system deployed in the traditional Internet or network environment did not have scalability and adaptability. Therefore, RMB proposed an intrusion detection method based on cloud hybrid machine learning method. The results showed that this method reduced the confidence depth and mean square error of network detection error, which was more suitable for cloud environment [7]. Thabit et al. believed that existing encryption algorithms still had limitations. A new encryption technology was constructed [8]. The first layer of this technology was based on Shannon diffusion and confusion theory. The second layer combined the genetic structure of the central principles of molecular biology to simulate coding processes. Compared with existing technologies, this encryption algorithm had significant advantages in data security, password size, and execution time, and could effectively improve the security of cloud computing applications. However, this research was mainly based on a single cloud environment, lacking optimization for hybrid cloud environments. To effectively improve the security of cloud computing applications, David Vako defined a set of requirements for cognitive assistants based on existing cloud service providers and assistant solutions. These requirements were used to guide the development process. Compared with the established reference architecture, the cognitive assistant output had better security [9]. Due to the dynamics of the cloud environment, the complexity of resource virtualization, and the diversity of user requests, it is imperative to develop effective technologies to evaluate and analyze the performance of the cloud center. Debbi H et al. used probabilistic model testing as an effective framework for providing resources in the evaluation and performance analysis cloud. This framework could effectively measure and analyze the cloud performance [10].

As a high-performance swarm intelligence optimization algorithm, SMA has been applied by some scholars. To improve the prediction accuracy of wind power, Lian et al. proposed a prediction model based on wavelet denoising and improved SMA optimized by SVM [11]. The wavelet denoising algorithm was taken to denoise wind power data. Then, the SVM was used as the prediction model. The test results showed that the proposed prediction model had high prediction accuracy. Jz et al. proposed a new technology based on SMA to optimize unknown variables in proton exchange membrane fuel cells. The results showed that this technology provided the best results in model recognition compared with other technologies [12]. Nassef and Handam applied SMA to the methane reforming hydrogen production process and used the algorithm to find the optimal parameter set for improving hydrogen production [13]. The optimized SMA was compared with particle swarm optimization and evolution strategy. The statistical results showed that the optimized SMA had the optimal mean value and standard deviation. Yildiz et al. compared various swarm optimization algorithms, including SMA, in the shape design of vehicle supports [14]. Based on the comparison results, a new optimizer combining swarm optimization schemes such as SMA was proposed. This study provided new optimization ideas for future vehicle design.

The keywords, methods, and results of the above studies are summarized in Table 1.

Table 1: Summary table of literature reviews.

Author	Keywords	Method	Results
Abroshan H [5]	Cloud computing, encryption, and Blowfish algorithm	A Hybrid Encryption Solution to Improve Cloud Computing Security using Symmetric and Asymmetric Cryptography Algorithm	The throughput, execution time and memory consumption parameters are better than the conventional algorithm
Vashishtha L K et al. [6]	Cloud-based system, intrusion detection, and signature	HIDM: A hybrid intrusion detection model for cloud-based systems	The model detected 92.7% on UNSW-NB 15,85.1% on the CICIDS dataset and 99.8% on the NSL-KDD dataset, outperforming some existing models
RM B et al. [7]	Hybrid cloud, machine learning, and intrusion detection	Hybrid machine intrusion detection in cloud based on learning approach: A metaheuristic assisted model	This method reduces the detection error and mean square error in deep confidence networks, and is more suitable for the cloud environment
Thabit F et al. [8]	Encryption technology, confused theory, and hybrid clouds	A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions	The encryption algorithm has obvious advantages in data security, password size and execution time, and can effectively improve the security of cloud computing applications
Dávid Vako et al. [9]	Cloud computing, security and cognitive assistant	Development and Implementation of a Cognitive Cloud Assistant for Optimal Cloud Service Provider Selection	The output of this cognitive assistant has superior security compared to the established reference architecture
Debbi H et al. [10]	Cloud environment, probabilistic model, and resource allocation	Modeling and Performance Analysis of Resource Provisioning in Cloud Computing using Probabilistic Model Checking	Effectively measure and analyze cloud performance
Lian L et al. [11]	Misci algorithm, wavelet denoising, and SVM algorithm	Wind power prediction based on wavelet denoising and improved slime mould algorithm optimized by support vector machine	The prediction model has a high prediction accuracy
Jz A et al. [12]	Myxobacteria algorithm, proton exchange membrane, and model identification	Balanced version of Slime Mould Algorithm: A study on PEM fuel cell system parameters identification	This technique gives the best results relative to other models in model identification techniques
Nassef A M et al. [13]	Myxobacteria algorithm, hydrogen production process, and parameter set	Parameter Estimation-Based on Slime Mould Algorithm of Photocatalytic Methane Reforming Process for Hydrogen Production	The algorithm can be used to find the optimal parameter set for improving hydrogen production
Yildiz B S et al. [14]	Misci algorithm and vehicle design optimizer	Conceptual comparison of the ecogeography-based algorithm, equilibrium algorithm, marine predators' algorithm, and slime mould algorithm for optimal product design	A new optimizer is proposed to provide a new optimization idea for the future vehicle design

In summary, current research has achieved some results in the field of network security in hybrid cloud environments, but there are still some shortcomings. Firstly, when dealing with network security threats, existing intelligent algorithms are prone to targeted attacks by exploiting algorithm logic vulnerabilities, lacking sufficient security protection measures. Secondly, the decision-making process of intelligent algorithms is often not transparent enough, which makes it difficult for users to understand and evaluate their behavior and decision-making basis in workflow scheduling. This may lead to users' concerns about network security and privacy protection. Therefore, a hybrid cloud workflow scheduling algorithm is proposed to ensure the network security in hybrid cloud environments. The introduced SMA provides new ideas to solve these problems. As a swarm intelligent optimization algorithm based on the behavior of natural myxobacteria, SMA has few parameters, strong optimization ability, and flexible fitness adjustment. In hybrid cloud workflow scheduling, SMA can effectively explore and utilize resources in complex and changing network environments by simulating the foraging behavior of myxobacteria. In addition, SMA's multiple exploration mechanisms give it strong global optimization capabilities, which help improve the system's ability to identify and respond to potential threats in network security hybrid cloud workflow scheduling, while maintaining high scheduling efficiency and resource utilization. Therefore, applying the SMA algorithm to network security hybrid cloud workflow scheduling is expected to make up for the shortcomings of existing intelligent algorithms to some extent and promote the development of this field.

3 Network security hybrid cloud workflow scheduling based on optimized SMA

3.1 Construction of optimized SMA based on levy flight

Workflow Scheduling Problem is to allocate appropriate execution sequences and resources for each task in a specific workflow model to minimize the workflow completion time or maximize the workflow throughput. How to scientifically and reasonably allocate resources for different types and quantities of tasks to achieve the best performance indicators has become the focus of workflow scheduling research. This technology has the potential to be applied in network security. The flowchart of a traditional SMA is shown in Figure 1. The SMA is a biomimetic algorithm that simulates the behavior of slime mould populations when searching for food as an optimization algorithm. This algorithm adjusts the weight based on the positive and negative feedback generated by slime moulds during the food search process, and calculates the propagation wave by simulating a biological oscillator to obtain the optimal result [15]. Due to its unique oscillation mode, the SMA has been widely used in engineering and other fields, and can achieve better optimization results than similar algorithms [16]. However, like other optimization algorithms, the SMA algorithm still has local optima. In the SMA, due to individuals only considering the current state and moving locally during the local search phase, it is possible for individuals to fall into local optima and cannot jump out [17]. To solve this problem, corresponding optimization methods are taken to make the algorithm have better convergence and robustness.

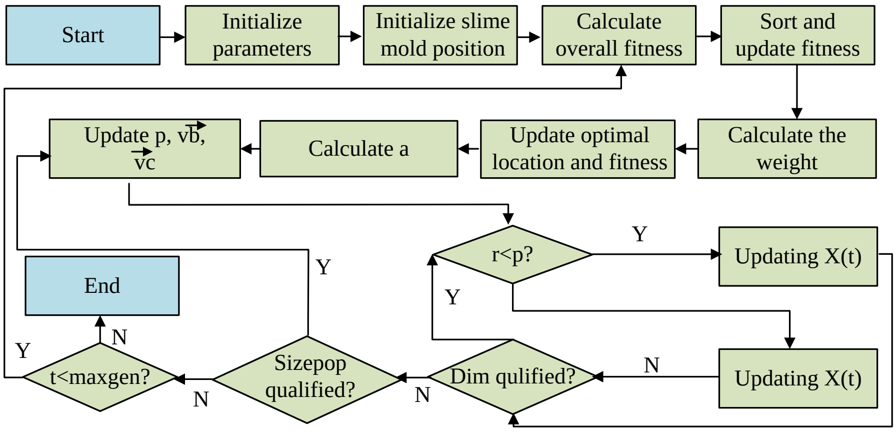


Figure 1: Flow chart of SMA.

To further optimize SMA and effectively reduce its probability of falling into local optimization, an optimized SMA is proposed. Based on traditional SMA, the algorithm adds a Levy flight model. This is a random walk particle model based on Levy stable distribution, which can improve the convergence ability of the optimization algorithm. Its principle is to increase the algorithm's ability to cross over local optima by introducing random walks to jump out of local optima and increase the convergence to global optima. The distribution of this model follows a power-law formula, as shown in Formula (1) [18].

$$f(x) \sim |x|^{-1-\alpha} \tag{1}$$

In Formula (1), $f(x)$ is the Levy stable distribution. α represents a random number within (0,2]. Different from other common distributions, Levy stable distribution has a relatively heavy tail, showing a long tail phenomenon. This means that it can describe data sets with outliers and extreme events. Formula (2) is the calculation process of random numbers.

$$Levy(\alpha) \sim \frac{\phi u}{|v|^{1/\alpha}} \tag{2}$$

In Formula (2), u and v are parameters that conform to the standard Chint distribution. ϕ is obtained from α and the standard gamma function. Assuming that the standard gamma function is G, Formula (3) is the calculation process.

$$\phi = \left[\frac{G(1+\alpha)\sin(\pi \frac{\alpha}{2})}{G\left(\left(\frac{1+\beta}{2}\right)2^{\frac{(\alpha-1)}{2}}\alpha\right)} \right] \tag{3}$$

Figure 2 is the iteration trajectory of the Levy flight model. In Figure 2, the figure shows the motion of one iteration in 250 iterations. The trajectory of each operation shows significant differences. Based on this feature, the Levy flight model can theoretically help swarm intelligence optimization algorithms further avoid the local optima.

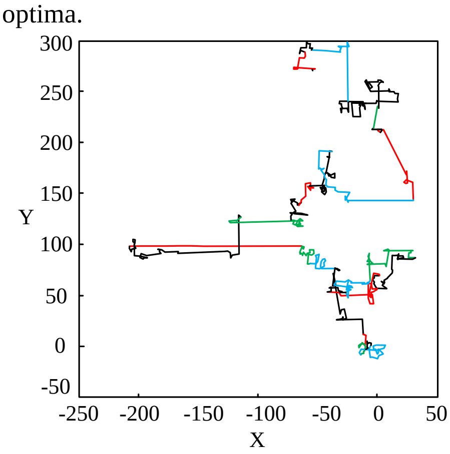


Figure 2: Iterative trajectory of levy flight.

When SMA seeks the best path, it first needs to use a contraction formula to simulate the scenario where slime moulds approach food. It is assumed that $\vec{x}(t+1)$ is the location of the slime mould population in the $(t+1)$ generation, $\vec{x}_b$ is the single point location with the highest odor concentration currently. $\vec{x}_A$ and $\vec{x}_B$ are randomly selected slime mould locations, and $\vec{W}$ represents the slime mould weight. Formula (4) is a contraction process.

$$\vec{x}(t+1)=\begin{cases}\vec{x}_b+\vec{v}b(\vec{W}\vec{x}_A(t)-\vec{x}_B(t)),\, r<p\\ \vec{v}c\bullet\vec{x}(t)\,,r\geq p\end{cases}\tag{4}$$

$\vec{v}b$ and $\vec{v}c$ in Formula (4) are parameters, where the range of $\vec{v}b$ values is $\vec{v}c$ linearly reduced from 1 to 0. r is a judgment condition, while p is a classification criterion. Formula (5) is the calculation of p.

$$p=\tanh\left|S(i)-DF\right|\tag{5}$$

In Formula (5), $S(i)$ is the fitness of slime mould at the current position. DF represents the best fitness value given by the maximum number of iterations. The tanh is obtained from $(\frac{e^x+e^{-x}}{e^x-e^{-x}})^{-1}$. Formula (6) is the value of $\vec{v}b$.

$$\vec{v}b=[-a,a]\tag{6}$$

In Formula (6), a is a number obtained based on the maximum number of iterations set by the SMA, as shown in Formula (7).

$$a=\arctan h(-(\frac{t}{t\max})+1)\tag{7}$$

In Formula (7), $t\max$ is the maximum number of iterations of the algorithm. t is the current number of iterations. In addition, it is necessary to define the slime mould weight in the slime mould shrinkage formula. Formula (8) is the mathematical definition.

$$\vec{w}(smellindex(i))=\begin{cases}1+r\log(1+\frac{BF-s(i)}{BF-wF})\,,specificcondition\\ 1-r\log(1+\frac{BF-s(i)}{BF-wF})\,,othercondition\end{cases}\tag{8}$$

In Formula (8), BF represents the optimal fitness of the iteration up to now, while wF represents the worst fitness at this time. $specificcondition$ is the formula condition, representing the population with the slime mould fitness in the first half of the current position. $othercondition$ represents the population whose fitness is not in the top half. The base of the log is 10. Under the joint action of the above factors, SMA can form a search vector from any angle, making it easier for the algorithm to find the optimal solution. After detecting food, slime moulds will try to go to places with higher food concentration, that is, higher weight. According to this feature, Formula (9) is a mathematical expression for the location update of slime moulds.

$$\vec{x}=\begin{cases}\vec{x}_b(t)+\vec{v}b(\vec{w}\vec{x}_A(t)-\vec{x}_B(t))\,,r<p\\ \vec{v}c\vec{x}(t)\,,r\geq p\end{cases}\tag{9}$$

After constructing the contraction mode of SMA, it can apply the Levy flight model to SMA to improve its convergence performance. During the search process, Levy flight can maximize the resource search efficiency, thereby increasing the efficiency of the algorithm. The Levy flight model is used to replace the traditional uniform distribution, so the location update rules of the SMA are changed in Formula (10).

$$\vec{x}(t+1)=\vec{x}_b(t)+\vec{v}b(\vec{w}\vec{x}_A(t)-\vec{x}_B(t))\otimes\varepsilon\otimes Levy\tag{10}$$

In Formula (10), ε is a parameter whose value is fixed to 0.01. $\otimes$ is hadamard accumulation. In the optimized position update formula, Levy flight determines its random compensation. Under the new location update rule, Formula (11) is the new location update formula.

$$\vec{x}=\begin{cases}\vec{x}_b(t)+\vec{v}b(\vec{w}\vec{x}_A(t)-\vec{x}_B(t))\otimes\varepsilon\otimes Levy\,,r<p\\ \vec{v}c\vec{x}(t)\,,r\geq p\end{cases}\tag{11}$$

The SMA is a swarm intelligence optimization algorithm. To apply it to network security tasks in a hybrid cloud environment, certain optimizations for network security projects need to be considered. In the field of network security, commonly used methods in both public and private cloud environments include random forest and SVM. In this study, SVM is chosen as the network security monitoring module. SVM is a widely used machine learning algorithm in network security. It maps data to a high-dimensional space and constructs an optimal hyper plane for binary or multi-class classification tasks. There are several reasons for selecting SVM over other methods in this study. Firstly, SVM has better generalization performance compared with algorithms such as random forest, enabling it to output better results in cases of over fitting. Secondly, SVM is less affected by data dimensionality and can handle both high-dimensional and low-dimensional data, making it better for detecting intrusion data. In this study, a traditional SVM model is introduced and improved to meet the research objectives.

To ensure that SVM can effectively handle data with different dimensions, an appropriate kernel function needs to be defined. Polynomial and Gaussian kernels are excellent choices for handling nonlinear and normalized data. However, in the field of network security, the security model needs to constantly check for linear data. Therefore, a linear kernel is more suitable. Thus, a linear kernel is provided, as shown in Formula (12).

$$j(x_1, x_2) = o + x_1 * x_2 \tag{12}$$

In Formula (12), o is a designed constant. x_1 and x_2 are variables. After setting the SVM kernel function, its penalty parameter is used as an individual input into the SMA algorithm, and iterative optimization is carried out to obtain the optimal parameters. Taking into account the entire algorithm process, the proposed SMA flowchart is shown in Figure 3.

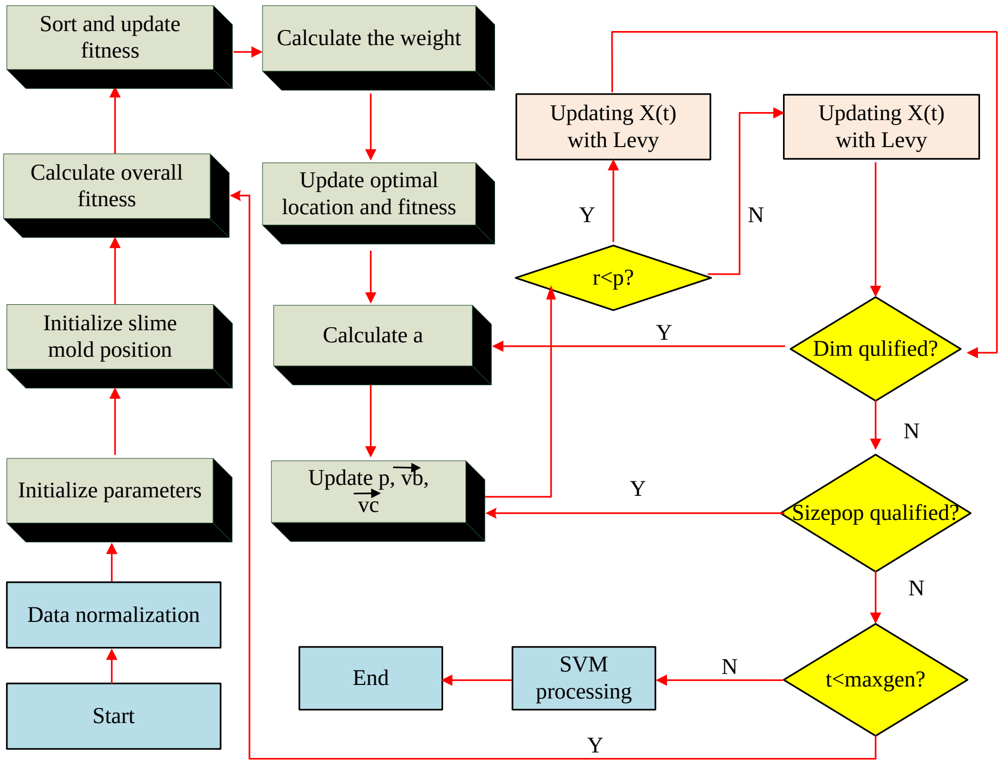


Figure 3: Flow chart of SMA combined with levy flight.

In Figure 3, first, the population size, initial position, and maximum iteration value of the slime mould are initialized. Each individual parameter, fitness, and algorithm weight are calculated and iterated. In the running process of the algorithm, two random slime mould positions are generated first. Then, the iterative action is completed by continuously calculating the fitness value and arranging them according to the size. Finally, the algorithm updates the best location and the best fitness through the weight value. The elements that determine whether an algorithm iterates to achieve the optimal result include size, dimension, and number of iterations. When the iteration result meets the conditions, the algorithm outputs the final result. The Levy algorithm is only used to update the location formula in SMA, so it does not affect the complexity of the algorithm.

The Levy distribution is introduced into the SMA algorithm to generate random numbers. Then, the position update formula is redesigned accordingly, which mainly affects the exploration behavior and search efficiency of the algorithm, rather than directly changing its temporal and spatial complexity. The property of the Levy distribution lies in combining large pace jumps and fine search with small steps, which helps the algorithm to explore the potential solutions more efficiently in the search space. The position update formula adjustment only uses these randomly generated Levy distribution values to guide the search direction, without adding additional storage requirements or calculation steps. Therefore, the space complexity of the algorithm is not improved. Moreover, although the change of exploration behavior may affect the number of iterations required for the algorithm to reach the convergence state, it belongs to the algorithm performance adjustment, rather than the essential change of time complexity. Therefore, this improved method will not affect the complexity of the SMA algorithm.

Figure 4 displays a detailed pseudocode for the interaction between the SMA optimization and Levy flight.

```
Improved SAM algorithm
Initialize:parameters    popsize,Max_iteraition;
Initialize:positions of slime mould X;(i=1,2,….,n);
Counter t=0
While t≤Max_iteraition do
    Calculate the fitness of all slime mould;
    update bestFitness X
    Calculate the W by Eq.(3.5);
    For each individual
        update p,vb,vc;
        update positions by Eq.(3.12);
    End For
    t=t+1;
End While
Return bestFitness X;
```

Figure 4: An improved pseudocode of the SMA algorithm.

In the network security scenario, compared with its classifier, SVM has higher adaptability, which can be well handled by both high-dimensional data and low-dimensional data. It is a very suitable machine learning method to detect intrusion data with small samples, nonlinear, and high latitude characteristics. However, in the invasion detection, how to select the parameters of SVM to achieve better results still needs to be solved. When SVM does not have effective selection parameters, the intrusion detection accuracy will be greatly affected, thus affecting the detection results and providing wrong information.

Faced with parameter selection of SVM in invasion detection, the penalty parameters and radial basis kernel function parameters of VM are taken as individuals of myxobacteria, and the invasion detection rate is used as the objective function to obtain the optimal SVM parameters through the mutual cooperation between myxobacteria. The stepwise algorithm is designed as follows:

Step1: Set the parameters of the model algorithm, the size of myxobacteria species, the maximum number of iterations, and the initial position of myxobacteria;

Step2: Produce a random set of initial myxobacteria. Each slime mould site includes penalty parameters and radial basis kernel function parameters;

Step3: Calculate the fitness value of all individuals, sort it according to the size of the value, and record it. The individual extreme value and the group extreme value are determined according to its value;

Step4: Calculate the weight through the myxobacteria weight formula;

Step5: Update the optimal location and optimal fitness of myxobacteria to produce new myxobacteria;

Step6: Update the individual location and other parameters according to the new location update rule formula;

Step7: If the number of iterations reaches the threshold set by Step1, the value obtained by the iteration is the optimal position of myxobacteria. If the iteration conditions are not met, the process returns to Step3;

Step8: Output the optimal individual fitness value and the optimal location of myxobacteria;

Step9: After the end of the cycle, the optimal parameter penalty parameters and radial basis kernel function parameters of SVM are extracted. Finally, the data is detected.

3.2 Network security workflow and coding scheme in a hybrid cloud environment

Workflow scheduling and network security are closely related. Workflow scheduling refers to a series of scheduling strategies developed in a distributed computing environment to optimize task execution processes, aiming to improve task execution efficiency and system resource utilization. Network security refers to the activity of protecting computer networks from illegal intrusion, espionage, and sabotage. Network security tasks can also be the goal of workflow scheduling. Efficient scheduling method can improve the efficiency of network security model to identify and eliminate threats, so as to improve the security of the system. Workflow models in a hybrid cloud environment mainly rely on virtual machine models and task models, and the workflow model is also based on these two models [19]. The main reason for adopting the virtual machine model is that in network security scheduling tasks involving workflow, service providers typically use virtual machines to represent computing resources, and each virtual machine has uniqueness. Virtual machines are divided into two categories in coding, including public cloud virtual machines and private cloud virtual machines [20-22]. Task model is used to represent task scheduling, which includes task number, number of operation instructions, space size for storing tasks, and depth of tasks in workflow in detail [23-24]. Generally speaking, in Figure 5, the depth of task 1 is 1, and the task 6 is 3. The depth of other tasks is 2.

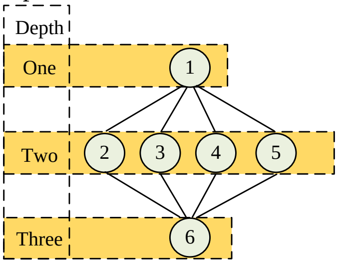


Figure 5: Depth of the task model.

Workflow scheduling is actually a combinatorial optimization problem, and swarm intelligence optimization algorithms generally use continuous numerical optimization. Due to this optimization, the iteration of swarm intelligence algorithms only calculates real numbers, so a reasonable coding and conversion scheme must be adopted. A binary matrix is used to represent the decoding scheme here. Binary encoding has a shorter length, which can improve the search efficiency while reducing storage space. This makes the logical computations of the algorithm more convenient and reduces implementation difficulty. The binary matrix is represented by Formula (13).

$$Y = \begin{bmatrix} Y_{1,1} & Y_{1,2} & \cdots & Y_{1,n} \\ Y_{2,1} & Y_{2,2} & \cdots & Y_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ Y_{n,1} & Y_{n,2} & \cdots & Y_{n,m} \end{bmatrix} \tag{13}$$

In Formula (12), Y is a binary matrix, m is a virtual machine number, and n means tasks number. When $Y_{a,b} = 1$, task a is assigned to virtual machine b. In addition, the situation indicates that task a is not assigned to the virtual machine b. In addition, due to the exclusive nature of each task after decomposition, the solution Y also needs to satisfy Formula (14).

$$\sum_{b=0}^{b=m} Y_{a,b} = 1 \tag{14}$$

Task exclusivity refers to the nature of a task that can only run on one virtual machine. After meeting this constraint, the preliminary work of coding conversion is completed. Because the workflow scheduling model in the hybrid cloud environment mainly deals with discrete optimization problems, it is also necessary to discretize the matrix solution. S-type function is selected for discretization, which can limit the function value obtained by SMA to the identifiable interval. In addition, the workflow scheduling model needs to use a transformation function to confirm the value of the scheduling solution matrix. Formula (15) is a conversion function.

$$V(y) = \begin{cases} 1, s(y) > 0.5 \\ 0, otherwise \end{cases} \tag{15}$$

In Formula (15), $V(y)$ represents the conversion function, and $s(y)$ represents the value of an S-type function. So far, the coding scheme has integrated conventional workflow scheduling issues and SMA. However, it should be noted that a large portion of workflow task scheduling for network security occurs in a hybrid cloud environment. Therefore, coding is also required for this environment. According to a hybrid cloud environment, a task can only be executed on at most one virtual machine. It can be considered that the number of tasks participating in scheduling is equal to the number of virtual machines. At this time, public cloud resources are infinite. Under these conditions, the matrix representing the task scheduling scheme must be a 0-1 binary matrix. When the task amount is n, this matrix is a $n*n$ matrix. On this basis, a three-chromosome encoding method is proposed, as shown in Formula (16).

$$Scheduler = (task\_order, vm\_order, vm\_type) \tag{16}$$

In Formula (16), $Scheduler$ represents a scheduling scheme. $task\_order$ is the scheduling order of the task, and each value in this array represents the number of different tasks. vm_order represents the mapping relationship between each task and its corresponding virtual machine, and the values in this array represent the number of the virtual machine. vm_type represents the model of each virtual machine. The encoding method of binary matrices can bring high temporal and spatial complexity. Using this one-to-one three-chromosome encoding method can effectively reduce complexity. Figure 6 shows the schematic diagram and task scheduling scheme of trisomy coding. Figure 6 (a) is an individual solution of the coding scheme in a hybrid cloud environment. Figure 6 (b) is a schematic diagram of task scheduling. Each task corresponds to a virtual machine one by one, and different virtual machines in a workflow may belong to the same type.

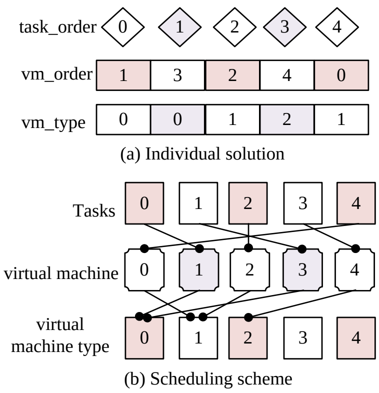


Figure 6: Coding scheme and workflow task scheduling.

In Figure 6, because the workflow task scheduling scheme is expressed by a matrix, it inevitably faces excessive solution space problems. Therefore, after coding the workflow scheduling model, the SMA is embedded in workflow scheduling. First, an initial population that conforms to the workflow execution sequence is initialized. Then, their fitness values are calculated and the best value is determined. After continuous contraction and iteration, the global optimal mutation and optimal individual are found. The added SMA can effectively solve the excessive matrix solution space in workflow scheduling models through swarm intelligence optimization, making scheduling work more efficient. Currently, network security is a key issue faced by many fields and industries, and they adopt different workflow models based on their respective work content and processes. For network security-oriented hybrid cloud workflow scheduling models, the ability to run normally under multiple hybrid cloud workflow models is also required. These four most important workflow models are used to adapt the proposed workflow scheduling model, as shown in Figure 7.

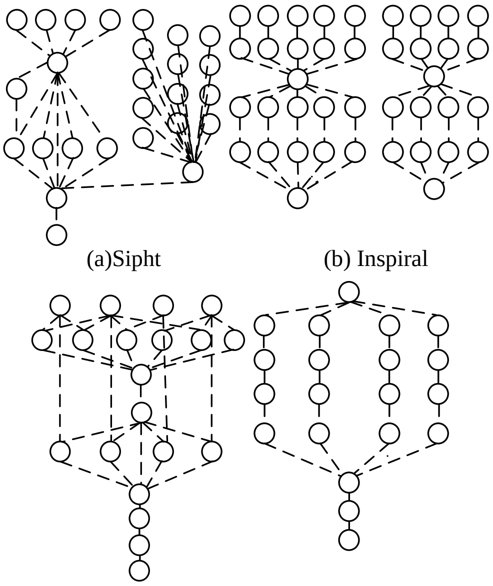


Figure 7: Different workflow structures.

In Figure 7, they are Sipht inphysics, Inspiral and Montage in astronomy, and Epigenomic in biology.

4 Testing of network security hybrid cloud workflow scheduling algorithm for SMA

The proposed algorithm has two main improvements, namely, improvements for SMA and improvements in coding for network security workflow scheduling in a hybrid cloud environment. Therefore, this test mainly tests the performance of the improved SMA algorithm and the performance of the hybrid cloud workflow scheduling model, It mainly includes: (1) using four benchmark test functions to test the effectiveness of the improved SMA algorithm; (2) Optimizing dataset scheduling for four workflows: Sipht model, Inspiral model, Montage model and Epigenomic model, and testing the completion time and completion cost of the improved SMA algorithm in different datasets and in different numbers of tasks. When the completion time and cost value are smaller, the algorithm performance is better; (3) Testing the convergence curves of the improved SMA algorithm on four workflow datasets: Sipht model, Inspiral model, Montage model, and Epigenomic model, and analyzing the convergence of the improved SMA algorithm; (4) Explaining the correlation between this study and network security scenarios, and evaluating the actual protection capability of the research method against network attacks. All experiments are implemented in MATLAB R2018b with OS Windows 10 and CPU of 64-bit Intel (R) Core (TM) i5-8300H and 32GB memory. In the proposed improved SMA and original SMA, the parameters are set as follows. The population is 50, the maximum number of iterations is 1,000, and the initial fitness weight of each individual is 1. To avoid the influence of the initial position, the initial position of SMA and improved SMA is fixed in the experiment. The experiment of each function is repeated 30 times. Because network security workflow scheduling in a hybrid cloud environment mostly relies on virtual machines, experiments need to first configure virtual machines. Table 2 shows the configuration of virtual machines. This test includes several representative virtual machine types. The bandwidth cost of each virtual machine is 0.02 Gbps, while the storage cost is 0.001per second. In actual testing, different workflow scheduling models are applied to each virtual machine and their average performance is calculated.

Table 2: Virtual machine configuration.

Type	Vcpu amount	Process capability (MIPS)	Memory (GB)	Cost ($/hour)
C6g.medium	1	4,400	2	0.034
C6g.Large	4	17,600	8	0.136
a1. large	2	8,800	4	0.051
c5d.xlarge	16	70,400	32	0.768
t3a.xlarge	4	17,600	16	0.1504

Firstly, to test the differences between the proposed improved SMA and traditional SMA, four benchmark testing functions are used to compare their performance. Two of the functions are unimodal and two are multimodal. Figure 8 shows the test results of unimodal function. Figure 8 (a) shows the test results under unimodal function f1, and Figure 8 (b) shows the test results under unimodal function f2. The average fitness, that is, standard deviation convergence rate, is used to evaluate algorithm performance under different functions. For objective evaluation, in addition to the proposed algorithm, traditional SMA, PSO, and Grasshopper Optimization Algorithm (GOA) are also been used in testing as comparators. GOA simulates behaviors such as foraging and clustering, combining random search and local search strategies to continuously search for the global optimal solution. PSO is the most basic swarm intelligence optimization algorithm that utilizes velocity and position adjustment methods between particles to find the global optimal solution. From an image perspective, the standard deviation convergence of the proposed algorithm is the fastest under different functions, and there is a significant gap between this algorithm and others. Under the unimodal function, the standard deviation of GOA and PSO decreases very slowly, and the variance after convergence is also much higher than that of SMA. Under function f1, both the proposed algorithm and traditional SMA have a minimum standard deviation of less than 10-300. However, the proposed algorithm has already met this standard at the 400th iteration, while traditional SMA is later than the 400th iteration. Under function f2, the standard deviation of all algorithms decreases more slowly. When the number of iterations reaches 500, the standard deviation of the proposed algorithm is lower than that of the other algorithms (P<0.05).

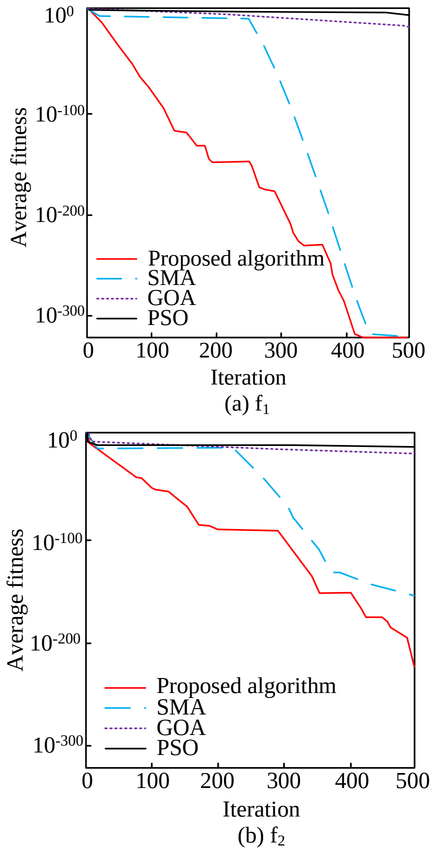


Figure 8: Test results under unimodal function.

The above algorithm performance in multimodal functions has also been tested in Figure 9. Figure 9 (a) shows the performance of each algorithm under the multimodal function mf1, and Figure 9 (b) shows the performance of each algorithm under the multimodal function mf2. Under multimodal functions, the standard deviation reduction ability of all functions is stronger than that under unimodal functions. However, from the image, the decline speed and amplitude of the traditional SMA and the proposed algorithm are much higher (P<0.05). In addition, the performance of traditional SMA and the proposed algorithm in multimodal functions is closer to that in unimodal functions. This is because SMA is easier to handle multimodal functions, resulting in faster optimal solutions and higher difficulty in optimization, which may be difficult for Levy flight to further optimize.

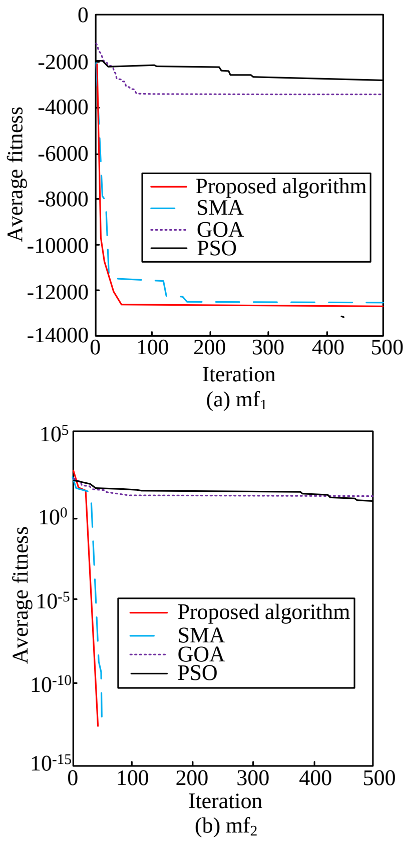


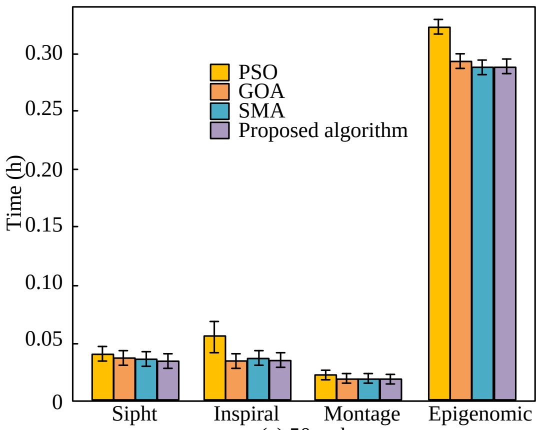
Figure 9: Test results under multimodal function.

Because the target of network security hybrid cloud task scheduling is network attacks, its operation and response speed are extremely important. Therefore, this test records the running time of each algorithm used for comparison under the same environment. Figure 10 shows the results. Figure 10 (a), Figure 10 (b), and Figure 10 (c) show running speed when the number of tasks is 50, 100, and 1,000. Table 3 shows the detailed numerical comparison case for each workflow model.

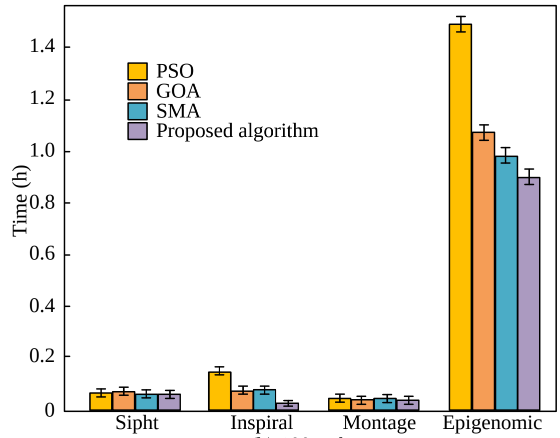
Table 3: The detailed numerical comparison of each workflow model.

Each workflow model		PSO	GOA	SMA	Proposed algorithm
Sipht	50 tasks	0.043	0.04	0.04	0.041
	100 tasks	0.042	0.049	0.043	0.042
	1,000 tasks	1.81	0.93	0.92	0.89
Inspiral	50 tasks	0.062	0.039	0.041	0.042
	100 tasks	0.131	0.073	0.072	0.021
	1,000 tasks	4.58	2.23	2.22	1.53
Montage	50 tasks	0.031	0.03	0.03	0.03
	100 tasks	0.028	0.027	0.031	0.028
	1,000 tasks	0.83	0.79	0.84	0.78
Epigenomic	50 tasks	0.323	0.284	0.279	0.278
	100 tasks	1.52	1.07	0.98	0.852
	1,000 tasks	51.2	21	182	13.7

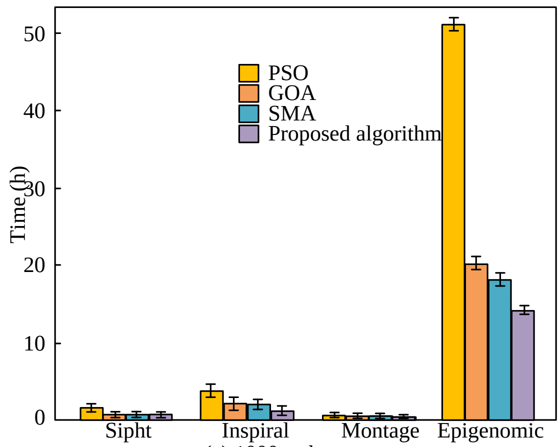
To enhance the comprehensiveness of the testing, different hybrid cloud workflow models are added as the second independent variable in the testing of this link. By comparison, PSO runs at the slowest speed in most cases, and its differences from other algorithms increase as the number of tasks increases in the Epigenomic model. In these three models except Epigenomic, the gap among these algorithms is relatively insignificant. In the Epigenomic model, the advantages of the proposed algorithm become more apparent as the number of tasks increases. When the number of tasks is 1,000, the running time of the proposed algorithm is 13.7 hours. Traditional SMA is second only to the proposed algorithm and takes 18.2 hours. The operation time of PSO under this condition exceeds 50 hours. The running time difference of the improved algorithm is significantly different, (P<0.05).



(a) 50 tasks



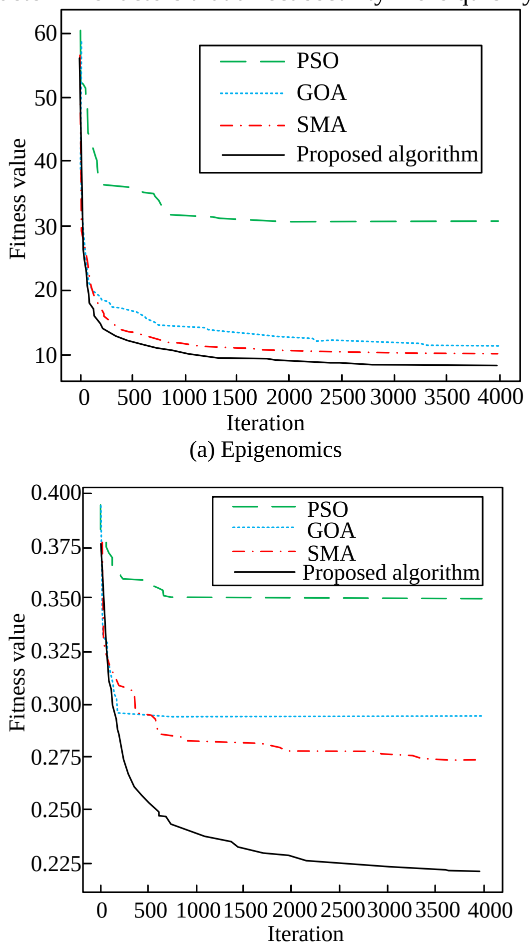
(b) 100 tasks



(c) 1000 tasks

Figure 10: Algorithm speed in different environments.

In the previous testing steps, the convergence of the proposed improved SMA is analyzed. However, to confirm the performance of the proposed algorithm as a whole in network security hybrid cloud workflow scheduling, it needs a conduct convergence testing in task scheduling scenarios. Figure 11 shows the test results. Figure 11 (a) shows the convergence under Epigenomics workflow, and Figure 11 (b) shows the convergence under Montage. Under Epigenomics workflow, the fitness value of PSO converges to about 30, which is much higher than other three algorithms. The performances of GOA, SMA, and the proposed algorithm are similar under the Epigenomics workflow model, but the proposed algorithm has the best fitness after convergence, as low as 4.1. This value is not only the best among the centralized algorithms used in the experiment, but also sufficient to complete scheduling tasks in actual work according to the actual data of the corresponding workflow model. Under the Montage workflow, the convergence differences between these four algorithms are significant. The proposed algorithm is the best, and its fitness is as low as 0.198 after convergence ($P<0.05$). In network security work, algorithms with faster convergence speed help to detect and determine factors that affect security more quickly.



(a) Epigenomics

(b) Montage

Figure 11: Convergence of the algorithm in task scheduling under different model.

After testing the proposed optimization strategy of the SMA, the network security performance of the proposed workflow scheduling model in a hybrid cloud environment is evaluated. The proposed algorithm addresses the task scheduling in the field of network security in a hybrid cloud. Therefore, the actual defense capability against network attacks needs to be assessed as a performance metric. The KDD Cup 99 dataset is used as a data source for this experiment. The intrusion detection dataset consists of three parts: all dataset, 10% dataset, and dataset with intrusion mark. 10% of the dataset contains the training dataset (with attack markers) and the test dataset (without attack markers), with a total of about 500,000 records. The test dataset contains more intrusion behaviors (38), the trained dataset is less (23), and the main intrusion categories are Probe, Dos and R2L. Training samples and test samples are selected from 10% dataset and dataset containing intrusion records. 8,000 pieces of normal behavior and 8,000 data with intrusion behavior are extracted from 10% dataset as training set, and 4,000 pieces of normal behavior and 4,000 intrusion behavior data are extracted from the latter dataset as test set. After iterative training using KDD network attack dataset, each algorithm is used in network attack simulation testing in Table 4. Table 2 shows the detection rate of several algorithms under four common attack methods. These attack methods mainly include Normal, Satan, Smurf, and Guess password. Under attacks other than Guess password, the detection rates of the proposed algorithms all exceed 91%, while the detection efficiency of other algorithms typically ranges from 77% to 91%. The detection of Guess password attacks is difficult, and traditional PSO and GOA cannot effectively monitor them. The traditional SMA has the detection ability for Guess Pass, with a detection rate of 8.93. The proposed algorithm further enhances the detection rate of SMA against this attack method, reaching 11.54%, and a P-value of less than 0.05, with statistically significant differences. The detection rate of the proposed algorithm is significantly lower than that of other Guess password attack types. This may be due to the fact that the Guess password attack features do not fully match the characteristics of the detection model used. To improve the detection rate of such attacks, the behavior patterns and characteristics of Guess password attacks are deeply analyzed, and more targeted features are extracted for model training to improve the model's generalization ability and detection performance.

Identifying the types of network attacks correctly is a prerequisite for ensuring network security. The proposed method can effectively improve the security of cloud computing applications and enhance the potential of network security task scheduling. This improves the efficiency of data processing in network intrusion detection and significantly improves the accuracy of network attacks. This further improves the ability of the improved method to correctly identify the types of network attacks in network security monitoring, and improves its application effectiveness in network security.

Table 4: The response of algorithms to network attacks.

Attack type	Detection rate (%)				
	PSO	GOA	SMA	Proposed algorithm	TC3PoP
Normal	78.99	83.72	88.57	91.93	90.08
satan	80.07	87.31	91.55	96.57	92.17
Smurf	77.69	81.14	87.93	90.69	89.00
Guess passward	0	0	8.93	11.54	9.14

Finally, to verify the real-world applicability of the network security hybrid cloud workflow scheduling strategy based on the improved SMA algorithm, a specific experimental environment is constructed. The environment combines private clouds, public clouds, and local data centers, simulating complex hybrid cloud scenarios. Inside the hybrid cloud, an advanced network security protection system, including firewalls, intrusion detection systems, and data encryption measures, is configured to ensure the security of the experimental environment. Meanwhile, a workflow scheduling system based on the improved SMA algorithm is developed and seamlessly integrated into a hybrid cloud environment. The research method is applied to the benchmark test cases in the hybrid cloud environment to record the task completion time and resource utilization during the task execution, monitor the incidence of network security events in the hybrid cloud environment, and evaluate the impact of the scheduling method on network security. The results before and after application are shown in Table 5.

Table 5: Results obtained before and after the application of the proposed method.

Evaluation indicators		Task completion time (s)	Resource utilization rate (%)	Security event rate (%)
Before the application	1	25.3	76.2	30.2
	2	26.7	69.7	28,7
	3	24.8	65.4	25,4
	Average	25.6	70.4	30.2
Post application	1	11.1	86.9	2.8
	2	9.8	90.4	4.9
	3	8.3	92.1	3.2
	Average	9.7	89.8	3.6

From Table 5, after applying the proposed method, the task completion time, resource utilization, and network security event rate in task execution have all been significantly improved in the hybrid cloud environment. After applying the research method, the average task completion time reduces by 61.9%, the resource utilization rate increases by 27.5%, and the incidence of network security events decreases by 87.9%. The results show that the research method can effectively improve the scheduling efficiency of network security hybrid cloud workflow, reduce the occurrence probability of security events, and have strong applicability in the real world.

5 Discussion

According to the test results of the unimodal function, the proposed algorithm converges the fastest under different functions, and has a significant gap compared with other algorithms (P<0.05). Under the unimodal function, the convergence rate and variance of GOA and PSO are much lower than the proposed algorithm and traditional SMA (P <0.05). With the specific function f1, the standard deviation of the proposed algorithm at the 400th iteration is already below 10-300, which is earlier than the traditional SMA. Under the function f2, the standard variance of the proposed algorithm remains lower after 500 iterations (P<0.05). The proposed algorithm shows the same features when tests in the multimodal function environment, indicating that it outperforms some mainstream swarm optimization algorithms. This conclusion is consistent with the results of obtained by Huiling Chen et al [25] in the SMA. This is because SMA can handle different scenarios due to its diverse convergence mechanisms. In this experiment, SMA outperforms the other group optimization algorithms. However, the convergence ability of the proposed algorithm is better than the conventional SMA algorithm. Therefore, its optimization path is effective. Under 1,000 tasks, the proposed algorithm takes the shortest running time of 13.7 hours, which is significantly better than the 18.2 hours of the traditional SMA and the 50 hours of the PSO, and the running time difference between the improved SMA algorithm and the comparison algorithm is statistically significant (P<0.05). This is the same conclusion that Ophir Nave and [26] apply SMA to workflow scheduling tasks. This is because in the proposed algorithm, Levy flight can maximize resource search efficiency, enhance the exploration ability of search individuals in the global space, and thus solve local optimal problems.

In the field of SMA algorithm optimization, some previous studies have used Levy flight to optimize SMA. Another team presented an SMA debate based on oppositional learning. It utilizes adversarial learning to improve convergence speed and employs Levy flight to optimize the search capability for SMA [27]. The results of this study show that after adopting the method, the task completion time of hybrid cloud environment shortens by 61.9%, the resource utilization rate increases by 27.5%, and the incidence of network security events reduces by 87.9%. This is similar to the conclusion reached by Ling Zheng et al [28] in their study on SMA. This is because the improved SMA algorithm can effectively solve the large solution space problem of matrix optimization in the workflow scheduling model through swarm intelligence optimization, making the scheduling process more efficient. This improves the efficiency of the algorithm without compromising its complexity. Furthermore, based on the SVM, the proposed algorithm is specifically tailored to network security workflow scheduling in a hybrid cloud environment. The results show that under most attacks, the detection rate of the proposed algorithm exceeds 91%, which is better than other algorithms. For Guess passward attack, the detection effect of the traditional algorithm is not good, but the proposed algorithm significantly improves the detection rate to 11.54% ($P<0.05$), which has a statistical difference. The Levy flight model further optimizes the SMA and applies it to hybrid cloud workflow scheduling encoding. This study solves the large solution space in the matrix, improves data security in the hybrid cloud environment, and fills the gap in the field of network security.

6 Conclusions

A network security monitoring algorithm based on hybrid cloud workflow scheduling and SMA is proposed to address the data being vulnerable to network attacks in the current hybrid cloud environment. This algorithm optimizes traditional SMA and applies it to workflow scheduling for network security in a hybrid cloud environment. The improved SMA algorithm proposed in this study effectively improves the probability of jumping out of the local optimum in the search process, and can find the optimal solution faster, and the convergence rate is faster than other comparison algorithms. Moreover, the WorkflowSim simulation platform is extended with four different datasets, and the effectiveness and feasibility of the proposed algorithm are verified from the convergence, cost, and fitness value. These help to improve the system's ability to identify and respond to potential threats in the network security hybrid cloud workflow scheduling, while maintaining high scheduling efficiency and resource utilization.

Although this study demonstrates the potential for optimization, there are still some limitations that need to be addressed. First of all, the study improves the Levy flight only for the uniform distribution in the formula, without considering other methods to improve the probability of escaping local optima. Next, Improvements can be made from other aspects to expand the application. Then, in intrusion detection, classifier parameter optimization always exists. How to find the optimal parameter to improve the detection rate and false alarm rate of intrusion detection needs to be further explored. Finally, in the scheduling model, the scheduling mechanism is simplified. Therefore, there may be deviations from the expected results when applying the model to practical processes. In future work, more research should be conducted on the scheduling mechanism of cloud computing platforms to form a scheduling model that is more in line with reality.

References

[1] E Balamurugan, Abolfazl Mehbodniy, Elham Kariri, Kusum Yadav, Anil Kumar, and Mohd Anul Haq. Network optimization using defender system in cloud computing security-based intrusion detection system withgame theory deep neural network (IDSGT-DNN). Pattern Recognition Letters, 156:142-151, 2022. https://doi.org/10.1016/j.patrec.2022.02.013

[2] Sandesh Achar. An overview of environmental scalability and security in hybrid cloud infrastructure designs. Asia Pacific Journal of Energy and Environment, 8(2):39-46, 2021. https://doi.org/10.18034/apjee.v8i2.650

[3] Juan Zhao, Zhengming Gao, and Wu Sun. The improved slime mould algorithm with levy flight. Journal of Physics: Conference Series, 1617(1):1-8, 2020. https://doi.org/10.1088/1742-6596/1617/1/012033

[4] Dildar Gures, Sujin Bureerat, Sadiq M. Sait, and Ali Rıza Yildiz. Comparison of the arithmetic optimization algorithm, the slime mold optimization algorithm, the marine predator's algorithm, the salp swarm algorithm for real-world engineering applications. Materials Testing, 63(5):1-5, 2021. https://doi.org/10.1515/mt-2020-0076

[5] Hossein Abroshan. A hybrid encryption solution to improve cloud computing security using symmetric and asymmetric cryptography algorithms. International Journal of Advanced Computer Science and Applications, 12(6):31-37, 2021. https://doi.org/10.14569/IJACSA.2021.0120604

[6] Lalit Kumar Vashishtha, Akhil Pratap Singh, and Kakali Chatterjee. HIDM: A hybrid intrusion detection model for cloud-based systems. Wireless Personal Communications, 128(4):2637-2666, 2023. https://doi.org/10.1007/s11277-022-10063-y

[7] Murali Mohan V, Balajee RM, Hiren K Mewada, Rajakumar BR, Binu D. Hybrid machine learning approach-based intrusion detection in cloud: A metaheuristic assisted model. Multiagent and Grid Systems, 18(1):21-43, 2022. https://doi.org/10.3233/MGS-220360

[8] Fursan Thabit, Sharaf Alhomdy, and Sudhir Jagtap. A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions. International Journal of Intelligent Networks, 2:18-33, 2021. https://doi.org/10.1016/j.ijin.2021.03.001

[9] Dávid Vako, Kapa M, and Norbert dáml. Development and implementation of a cognitive cloud assistant for optimal cloud service provider selection. Acta Electrotechnica et Informatica, 24(2):19-24, 2024. https://doi.org/10.2478/aei-2024-0007

[10] Debbi H. Modeling and performance analysis of resource provisioning in cloud computing using probabilistic model checking. Informatica: An International Journal of Computing and Informatics, 45(4):529-541, 2021. https://doi.org/10.31449/inf.v45i4.3308

[11] Lian Lian, and Kan He. Wind power prediction based on wavelet denoising and improved slime mold algorithm optimized support vector machine. Wind Engineering, 46(3):866-885, 2022. https://doi.org/10.1177/0309524X211056822

[12] Junlong Zheng, Yujie Xie, Xiaoping Huang, Zhongxing Wei, and Bahman Taheri. Balanced version of slime mold algorithm: A study on pem fuel cell system parameters identification. Energy Reports, 7(1):3199-3209, 2021. https://doi.org/10.1016/j.egyr.2021.05.052

[13] Ahmed M. Nassef, and Ahmed Handam. Parameter estimation-based slime mold algorithm of photocatalytic methane reforming process for hydrogen production. Sustainability, 14(5):1-5, 2022. https://doi.org/10.3390/su14052970

[14] Betül Sultan Yildiz, Vivek Patel, Nantiwat Pholdee, Sadiq M. Sait, Sujin Bureerat, and Ali Rıza Yildiz. Conceptual comparison of the ecogeography-based algorithm, equilibrium algorithm, marine predators' algorithm and slime mold algorithm for optimal product design. Materials Testing, 63(4):336-340, 2021. https://doi.org/10.1515/mt-2020-0049

[15] Juan Zhao, Zhengming Gao, and Wu Sun. The improved slime mould algorithm with levy flight. Journal of Physics Conference Series, 1617(1):012033, 2020. https://doi.org/10.1088/1742-6596/1617/1/012033

[16] Verner Püvi, Robert J. Millar, Eero Saarijrvi, Ken Hayami, Tahitoa Arbelot, and Matti Lehtonen. Slime mold inspired distribution network initial solution. Energies, 13(23):1-17, 2020. https://doi.org/10.3390/en13236278

[17] Papri Ghosh, Ritam Dutta, and V. Muthulakshmi. Slime mold algorithm based hybridized artificial neural network model for efficient automatic voltage regulation control. Journal of Nano- and Electronic Physics, 13(3):03038-1-03038-6, 2021. https://doi.org/10.21272/jnep.13(3).03038

[18] Meijiao Liu, Yanhui Li, Ang Li, Qi Huo, Ning Zhang, and Nan Qu. A slime mold-ant colony fusion algorithm for solving traveling salesman problem. IEEE Access, 8:202508-202521, 2020. https://doi.org/10.1109/ACCESS.2020.3035584

[19] Divya Agarwal, and Pushpendra S. Bharti. Implementing modified swarm intelligence algorithm based on Slime moulds for path planning and obstacle avoidance problem in mobile robots. Applied Soft Computing, 107(4):1-15, 2021. https://doi.org/10.1016/j.asoc.2021.107372

[20] Abhishek Pandey, and Soumya Banerjee. Test data generation and selection using levy flight-based firefly algorithm. International Journal of Software Innovation, 9(2):18-34, 2021. https://doi.org/10.4018/IJSI.2021040102

[21] Fursan Thabit, Associate Prof Sharaf Alhomdy, Abdulrazzaq H.A. Al-Ahdal, and Prof Dr Sudhir Jagtap. A new lightweight cryptographic algorithm for enhancing data security in cloud computing. Global Transitions Proceedings, 2(1):91-99, 2021. https://doi.org/10.1016/j.gltp.2021.01.013

[22] Mohammed Altaee, and Mafaz Alanezi. Enhancing cloud computing security by paillier homomorphic encryption. International Journal of Electrical and Computer Engineering, 11(2):1771-1779, 2021. https://doi.org/10.11591/ijece.v11i2.pp1771-1779

[23] Bader Alouffi, Muhammad Hasnain, Abdullah Alharbi, and Wael Alosaimi. A systematic literature review on cloud computing security: Threats and mitigation strategies. IEEE Access, 9:57792-57807, 2021. https://doi.org/10.1109/ACCESS.2021.3073203

[24] Morteza Mollajafari, and Mohammad Hassan Shojaeefard. Shojaeefard. TC3PoP: A time-cost compromised workflow scheduling heuristic customized for cloud environments. Cluster Computing, 24(3):2639-2656, 2021. https://doi.org/10.1007/s10586-021-03285-5

[25] Huiling Chen, Chenyang Li, Majdi Mafarja, Ali Asghar Heidari, Yi Chen, and Zhennao Cai. Slime mould algorithm: a comprehensive review of recent variants and applications. International Journal of Systems Science, 54(1):204-235, 2022. https://doi.org/10.1080/00207721.2022.2153635

[26] Ophir Nave. Modification of semi-analytical method applied system of ODE. Modern Applied Science, 14(6):75, 2020. https://doi.org/10.5539/mas.v14n6p75

[27] Laith Abualigah, Ali Diabat, and Mohamed Abd Elaziz. Improved slime mould algorithm by opposition-based learning and levy flight distribution for global optimization and advances in real-world engineering problems. Journal of Ambient Intelligence and Humanized Computing, 14:1163-1202, 2023. https://doi.org/10.1007/s12652-021-03372-w

[28] Ling Zheng, Yan Tian, Hu Wang, Chengzhi Hong, and Bijun Li. Path planning of autonomous mobile robots based on an improved slime mould algorithm. Drones, 257(7):1-19, 2023. https://doi.org/10.3390/drones7040257