Intelligent Warning of Oil Depot Fire Based on Optimized Quantum Particle Swarm Optimization Algorithm in the Oil Depot Fire Information System

Na Liu[1], Yeting Yuan[2]
[1]Department of Information Engineering, Cangzhou Technical College, Cangzhou, 061000, China
[2]Academic Affairs Office, Cangzhou Technical College, Cangzhou, 061000, China
E-mail: yuanyeting2025@163.com

Keywords: Oil depot, Fire information system, Task uninstallation, Quantum particle swarm optimization algorithm, Fire warning

Received: May 26, 2025

There are issues with increased energy consumption of terminal equipment in the oil depot information system, as well as issues that are not conducive to the intelligent fire warning of this system. In this regard, edge computing is introduced, multi platform task uninstallation algorithm is designed, and a mathematical model is built. Quantum particle swarm optimization algorithm is used for optimization solution to determine the intelligent uninstallation strategy for multi platform tasks. An intelligent fire warning algorithm based on quantum particle swarm optimization and back propagation neural network is constructed to judge the fire situation. In the simulation results, quantum particle swarm optimization algorithm has significant advantages in multi platform task uninstallation. Compared with particle swarm optimization, quantum particle swarm optimization can reduce energy consumption by up to 17.1%. Compared with completely local algorithms, this research algorithm has saved 13.5%, 24.3%, and 38.3% of energy consumption, indicating the effectiveness of this research method algorithm. The mean squared error of the back propagation neural network optimized by quantum particle swarm optimization algorithm and the back propagation reached the expected error value in 106 iterations and 180 iterations respectively. The former has better convergence and global search ability than those of the latter. The back propagation neural network model optimized by quantum particle swarm optimization algorithm can effectively identify open fire, smoldering fire, and non-fire situations, and there is no false or missed reporting. This indicates that the research method can be beneficial for the intelligent fire warning of oil depot fire information system and promote the operational safety of oil depot.

Povzetek: Študija uvede robno računalništvo z večplatformnim razbremenjevanjem nalog, optimiziranim z kvantnim PSO za manjšo porabo energije, ter kvantno-PSO optimizirano BP-nevronsko mrežo za inteligentno požarno opozarjanje v informacijskem sistemu naftnega skladišča.

1 Introduction

The rapid Internet of Things (IoT) technology development can effectively improve the production efficiency of enterprises in industry. At present, China's oil depot (OD) enterprises are actively introducing advanced foreign technology, and some OD have established wireless storage and transportation platforms and started operating. However, the application of the IoT will lead to a sharp increase in the number of network edge device. In addition, the original information system equipment will produce huge amounts of data. If not handled properly, it is likely to result in increased energy consumption of terminal equipment, slower data operation, and unable to achieve intelligent prediction[1]. OD is a national oil reserve and supply base, whose main role is to coordinate the production and processing of crude oil, as well as the supply and transportation of finished oil. It plays an important role in the stability and rapid development of country's social economy. At present, there are two problems in OD information system in China. Firstly, the coverage of tank farm is wide, and the cost of wiring and laying in the early stage of data transmission is relatively high [2]. With the upgrade of OD and the increase of equipment, it is necessary to rewire, making it difficult to carry out renovation, and the cost of repairing faults is also high. The second issue is that OD information systems generally add some new functions, such as video surveillance systems, on top of DCS and SCADA systems. The information standards used by different manufacturers also vary. This will result in sub functional data not being available, leading to the phenomenon of

information silos, which can lead to the inability to effectively utilize resources and thus constrain the level of intelligence. By combining IoT technology with OD information systems, data tasks generated by sensors and other devices during operation can be collected in real-time at the perception layer. At the network layer, wireless transmission is achieved using technologies such as NB-IOT [3]. It can also be combined with edge computing and cloud computing to unify data standards, complete task computing, and feedback operation passwords in real time, thus improving work efficiency. The application layer mainly realizes OD data visualization, real-time monitoring and control of OD, early warning of faults and fires, automatic driving of Tank truck and other functions. This research intends to start with the quantum particle swarm optimization algorithm (QPSO) and carry out the research on OD information system technology. In task uninstallation (TU), a multi-platform intelligent uninstallation strategy for computing tasks was constructed with the goal of optimizing energy consumption. This achieves a reasonable distribution of computing tasks generated by terminal devices, thereby reducing the energy consumption of system operation and improving work efficiency. In the intelligent warning algorithm, the QPSO is introduced into back propagation neural network (BPNN) to optimize it. The model was trained using data from national standard experimental fires, and the correctness of the model was verified through experiments. By performing fire warning on the data transmitted from OD fire protection system, the safety performance of OD has been improved. This study mainly consists of four parts. Firstly, it is a literature review, which mainly summarizes the relevant research on fire prediction and QPSO. Secondly, the research method mainly focuses on intelligent uninstallation of multi-platform tasks based on QPSO and intelligent fire warning based on optimized QPSO. Then, simulation analysis is conducted on the research method. Finally, the conclusions drawn from the study and future research directions are summarized.

2 Related works

To ensure OD safety, many scholars have conducted corresponding research using QPSO, and scholars in other fields have also conducted a large amount of research using QPSO and achieved good research results. To strengthen the prevention and control of OD fire and explosion accidents, Xie et al. proposed a cloud hierarchical analysis method and group cloud decision-making method based on fuzzy cloud membership functions. They designed a fuzzy risk quantitative assessment model. This method can accurately identify weak links in OD fire and explosion accident safety system, and promote risk prevention, assessment, and control [4]. Aliser et al. proposed a segmentation technology based on deep network architecture for the problem of flame detection in images or videos to achieve an early fire warning system. By integrating the segmentation network structure of the attention gate module, the flame area can be identified more accurately. After conducting experimental research using a dataset containing 500 images and applying the five-fold cross-validation criterion, the success of the deep network architecture was evaluated through the dice, Tversky, and focal Tversky loss functions. The results show excellent performance under the mean dice and Jaccard similarity criteria [5]. Amjad et al. proposed a dynamic fire and smoke detection model in view of the importance of fire and smoke detection mechanisms. The visualization results confirm that DFDM effectively reduces background noise compared to the baseline model and provides a wider field of view, demonstrating its robustness in complex fire and smoke detection scenarios [6]. Hiremath et al. proposed a classifier based on the random forest algorithm for the forest fire early warning problem in the Odisha area, aiming to predict the occurrence of fires one day in advance. The results show that when the class imbalance ratio increases to 1:9, all indicators reach the maximum value, indicating that the model performs well in dealing with class imbalance [7]. Kale et al. proposed a method for predicting the spread of surface fires using the open-source WRF-SFIRE model to address the forest fire spread prediction in the northern Sikkim region of the Himalayas in India. The verification results show that the predicted burned-out area (1.72 km^2) is in good agreement with the Higash-to-space base data information (1.07 km^2), and the shape matching benchmark of the original burned-out area and the predicted area is good [8]. Liu P et al. proposed a multi-sensor fire detection method EIF-LSTM based on long short-term memory network for false alarm of fires. By extracting and fusing the time series features of multiple sensors, advanced detection performance was achieved, and the robustness and accuracy of the detection were enhanced [9]. Hosseinmardi A M et al. proposed a new formula based on network flow and the MLSTP genetic algorithm for the maximum leaf spanning tree problem. They solved 37 benchmark cases and successfully applied them to the fire detection system in Arasbaran Forest, Iran, significantly improving the early detection and alarm capabilities of fires [10].

Liu X et al. designed a fire source identification method with optimization constraint particle swarm optimization (PSO) to predict the overall tunnel temperature distribution to strengthen fire source identification in tunnel fires. This method has a higher accuracy in identifying fire sources than other current methods, and can be applied to fire source identification in public engineering [11]. PSO is difficult to find the optimal solution when solving bilevel programming. Zeng M et al. designed an improved hybrid cuckoo search QPSO based on simulated annealing criterion. The mechanism results in the cuckoo algorithm can solve the problem of particles easily falling into local optima. The optimization algorithm has better bilevel programming ability than other conventional algorithms [12]. Tu et al. proposed an improved method to address the premature convergence in QPSO search process. The optimal particles were selected using new selection techniques

and the algorithm's solvability was maintained using new mutation methods. A dynamic parameter strategy was used to balance search and exploration, and it was applied to electromagnetic design. The improved method can avoid the problem of premature convergence and has certain effectiveness in electromagnetic design [13]. Yu et al. proposed an improved QPSO to address the difficulties in selecting bands in hyperspectral images and the insufficient exploration ability of PSO, allowing particles to jump out of local optima. Applying the improved algorithm to band selection in hyperspectral images can improve classification accuracy and overcome the disadvantage of falling into local optima [14].

In the above research, QPSO has good predictive performance in oil depot fire warning, and QPSO has also been applied to various recognition problems with good application results. Considering the large amount of data in OD system, this study will focus on optimizing the energy consumption of information system equipment and developing an intelligent fire warning system to improve OD fire warning accuracy. The innovation and contribution of the research method lie in establishing a quantum edge intelligent disaster prevention and control paradigm, and proposing a quantum optimized BPNN with physical information embedding. The heat conduction equation is integrated as a hard constraint into the network architecture, and a quantum tunneling mechanism is introduced to optimize weights, thereby reducing the false alarm rate of smoke burning fires. Meanwhile, a quantum entanglement TU model is constructed, which achieves joint optimization of energy consumption and delay through superposition state decision-making to reduce system energy consumption in industrial sites. A quantum chain disaster prediction equation is established, which simulates catastrophic coupling effects between storage tanks using parameter quantum circuits and provides early warning of secondary explosions.

3 Energy consumption optimization and intelligent fire prediction of od fire information system equipment

The research mainly focuses on two issues: energy consumption optimization of the equipment in the OD fire information system and intelligent fire prediction. The first problem is that in the OD information system, due to the increase in energy consumption of a large number of devices, the computing speed of the system decreases and the intelligent prediction ability weakens. To address this challenge, the research introduced edge computing, designed a multi-platform TU algorithm, and established the corresponding mathematical model. The second issue concerns how to make efficient use of the large amount of data generated daily in the OD information system. Therefore, the intelligent fire early warning algorithm based on the BPNN optimized by QPSO is studied. This algorithm has the ability to predict the possibility of fire by analyzing the collected temperature, smoke and CO data, thereby issuing early warnings in a timely manner and improving the safety of OD.

3.1 Intelligent uninstallation of multi-platform tasks based on QPSO

A multi platform TU computing system is constructed, which is divided into a three-layer network model, including m edge servers, k cloud servers, and n terminal devices. In this model, it is assumed that the set of local terminal devices is $N=(1,2,...n)$, and $i \in N$ is used to describe the terminal device i. The collection of edge servers is $M=(1,2,...m)$, and $j \in M$ means edge servers [15]. The collection of cloud computing servers is $K=(1,2,...k)$, and $\circ \in K$ is cloud computing server. Assuming that terminal devices in this system need to complete energy sensitive tasks and ensure industrial task computing efficiency, and meet the maximum tolerance delay for completing tasks, equation (1) is the task request expression for terminal device i.

$$W_i \overset{\Delta}{=} (I_i, C_i, Z_i) \tag{1}$$

In equation (1), I_i is used to describe the amount of data required to input for calculating terminal device tasks. C_i means the number of calculated resources that must be allocated to complete a task on a terminal device. Z_i represents the maximum allowable delay at the end of terminal device operation. Equation (1) is a classic triple model of computing tasks, which is widely applied in the TU problems of mobile computing, cloud computing, and edge computing [16]. In the model, it is assumed that terminal tasks can not only be completed locally, but also from the edge server or cloud server to complete the overall task. This system scheme has chosen a quasi static scheme, which means that these system properties will not change during TU [17]. Meanwhile, due to the indivisibility of terminal equipment tasks, each task can only be completed on one platform. The information transmission between terminal devices and base stations will not be affected and can access the base stations simultaneously. Figure 1 shows the system model.

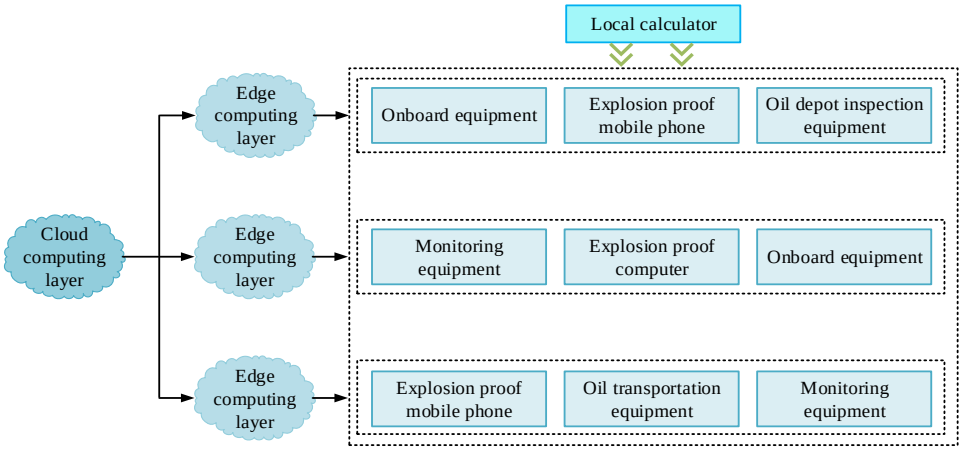


Figure 1: Multi-platform computing system model diagram

According to Figure 1, in a multi-platform computing system, it mainly includes cloud, edge and local computing layers. In a multi-platform computing TU system, the tasks of terminal devices can be completed in three ways: local execution, MEC uninstallation execution, and cloud computing uninstallation execution. Users choosing different platforms to execute tasks will generate different energy consumption. This study aims to establish an energy consumption optimization model, which defines energy consumption as task execution energy consumption and data transmission energy consumption, and calculates the TU energy consumption of the three platforms separately. In the constraint model, TU is constrained from three aspects: resource allocation, task allowable delay, and transmission rate. A multi-platform TU method based on energy consumption optimization for industrial applications is studied. The total energy consumption in a multi-platform TU system is defined as the maximum energy consumption required to be used during the task completion process. Equation (2) is a function modeling expression.

$$E = \min_{i \in N} E_i \tag{2}$$

In equation (2), E_i represents the energy consumption required to complete the task of the terminal device. Terminal device tasks in computing mainly include local uninstallation, MEC uninstallation, and cloud computing uninstallation. Equation (3) is the expression for constructing the model.

$$E_i = x_i E_i^0 + x_{i,j} E_{i,j} + x_{i,O} E_{i,O} \tag{3}$$

In equation (3), x_i represents the local execution variable, and when the terminal task is executing locally, $x_i = 1$. Conversely, $x_i = 0$. E_i^0 is used to describe the energy consumption generated by local execution of terminal device tasks. $x_{i,j}$ represents a MEC server uninstallation variable. During the calculation process from terminal TU to MEC, then $x_{i,j} = 1$. Conversely, $x_{i,j} = 0$. $E_{i,j}$ represents the energy consumption generated by terminal device tasks in MEC server calculations. $x_{i,0}$ represents the cloud server uninstallation variable. When computing from terminal TU to cloud server, $x_{i,O} = 1$. Conversely, $x_{i,O} = 0$. $E_{i,O}$ represents the energy consumption generated by terminal device tasks in cloud server computing [18]. Equations (2) and (3) jointly model the total energy consumption of the system. Minimizing the total energy consumption of the system is one of the core optimization objectives for energy sensitive systems such as the IoT and mobile devices [19]. Equation (4) is the energy consumption modeling expression for the local execution of terminal device tasks.

$$E_i^0 = \delta_i C_i Q_i^2 \tag{4}$$

In equation (4), δ_i, C_i, and Q_i respectively represent the processor energy consumption coefficient ($J \cdot s^2$), computing resources required to complete tasks (CPU cycles), and computing power of terminal device ($Hz \cdot s^{-1}$). The local execution energy consumption of equation (4) is related to the dynamic power consumption model of the processor, such as the specific architecture. To simplify the calculation, the study assumes that the voltage is fixed and adopts the quadratic model [20]. Equation (5) is a modeling expression for the energy consumption from terminal device TU to MEC server.

$$E_{i,j} = E_{i,j}^t + E_{i,j}^c \tag{5}$$

In equation (5), $E_{i,j}^t$ represents the energy consumed by terminal device i in sending task data to MEC server base station j, which is used to describe the energy consumed by the task of terminal device i in MEC server execution. Equation (6) is the energy modeling expression for $E_{i,j}^t$.

$$E_{i,j}^t = p_{i,j} T_{i,j}^t \tag{6}$$

In equation (6), $p_{i,j}$ and $T_{i,j}^t$ respectively represent power and time delay of terminal device i sending task data to MEC server base station j. Equation (5) and equation (6) together constitute the energy consumption of MEC uninstallation, mainly including the energy consumption of uploading task data to the MEC server and the energy consumption of the MEC server in processing computing tasks [21]. This study has

proposed a TU algorithm aimed at minimizing system energy consumption, which is a single optimization method. In practical use, it may encounter the problem of increasing task completion delay. Therefore, to achieve energy optimization through the objective function, it is necessary to combine the maximum task allowable delay with other constraint factors and optimize the solution. This mainly includes task execution maximum tolerance delay constraints, TU constraints, power allocation constraints, transmission rate constraints, and computing resource allocation constraints. Among them, equation (7) is the maximum tolerance delay constraint for task execution.

$$0 \le T_{i,j,O} \le T_i^{\max} \tag{7}$$

In equation (7), $T_i^{\max}$ represents the need to meet the maximum task tolerance delay in TU. Equation (7) is an added delay constraint. To ensure that real-time tasks must be completed before the deadline, this equation guarantees that the total computing time of the selected platform does not exceed the maximum tolerated delay [22]. Equation (8) is the expression for TU constraint.

$$\begin{cases} \sum_{i \in N} x_{i,j} \le \min\{B_j, S_j\} \\ \sum_{i \in N} x_{i,o} \le \min\{B_o, S_o\} \end{cases} \tag{8}$$

In equation (8), B_j represents the maximum accessible tasks number for base station j. Equation (8) serves as a TU constraint, mainly to limit the capacity of resources. The wireless resources and server computing resources that each base station or edge server can access are limited. It is not possible to split all tasks during execution. Therefore, only one of these three platforms can be selected for uninstallation [23]. In the power allocation constraint, the terminal device's transmission power needs to be between 0 and the maximum transmission power in equation (9).

$$0 \le p_{i,j,O} \le p_i^{\max} \tag{9}$$

In equation (9), $p_i^{\max}$ represents the maximum transmission power. Equation (9) mainly stems from the limitations of physical devices. The transmission power of the equipment is restricted by hardware circuits and regulations, such as power amplifier limitations and heat generation, etc [24]. In the transmission rate constraint, the transmission rate of terminal device should be greater than the minimum transmission rate in equation (10).

$$R_{i,j,O} \ge R_i^{\min} \tag{10}$$

In equation (10), $R_i^{\min}$ represents the minimum transmission rate. The transmission rate constraint of equation (10) is mainly limited by service quality or existing channel conditions. The device needs to ensure that it can obtain the minimum rate required for task transmission, which depends on the environmental signal-to-noise ratio, channel bandwidth, and modulation and coding schemes [25]. When calculating resource allocation constraints, due to the relatively limited computing resources of MEC servers and cloud servers, there are restrictions on the allocated resources for tasks uninstalled to MEC servers and cloud servers, as shown in equation (11).

$$\begin{cases} 0 \le F_{i,j} \le F_j \\ 0 \le F_{i,O} \le F_O \end{cases} \tag{11}$$

The constraint on computing resource allocation in equation (11) is due to the limited total CPU computing power of MEC edge servers or cloud servers. Therefore, the sum of the resources allocated to all processed computing tasks cannot exceed the total computing power value [26]. This study aims to establish TU energy consumption models for multiple platforms based on energy consumption optimization and uninstallation energy models. Minimizing the total energy consumption of task execution, taking into account multiple factors such as maximum task tolerance delay, TU, and transmission rate, a corresponding optimization model is established in equation (12).

$$\begin{aligned} \min_{X_{i,j,o}, p_{i,j,o}, F_{i.j.o}} E = {} & x_i \delta_i C_i Q_i^2 \\ & + x_{i,j}\left(p_{i,j} T_{i,j}^t + E_{i,j}^c\right) \\ & + x_{i,O} E_{i,O} \end{aligned} \tag{12}$$

Equation (12) is the main core optimization objective, which is to achieve the minimum energy efficiency under all service requirements. Essentially, it is a complex NP-Hard MINLP problem. An optimal algorithm is applied to model and derive TU strategies across multiple platforms, focusing on minimizing energy consumption. The transformation of practical problems into energy consumption optimization models brings non convex challenges, making it difficult to solve directly through traditional equations. The enumeration method may waste computing resources, prompting heuristic algorithms to obtain more effective solutions.

One of the algorithms is QPSO. It is based on PSO, a technique inspired by the cooperative foraging behavior of birds. Unlike genetic algorithms, PSO does not involve crossover or mutation but conducts random searches in the solution space. In PSO, particles represent potential solutions and are initialized randomly, with each particle updating its position based on its fitness relative to others.

However, PSO can become trapped in local optima due to its fixed velocity and particle trajectory. QPSO mitigates this issue by introducing quantum uncertainty, allowing particles to move unpredictably throughout the solution space. This approach, characterized by limited parameter selection and fast convergence, enhances the ability to find global optima beyond what traditional PSO can achieve. Figure 2 shows the process of QPSO.

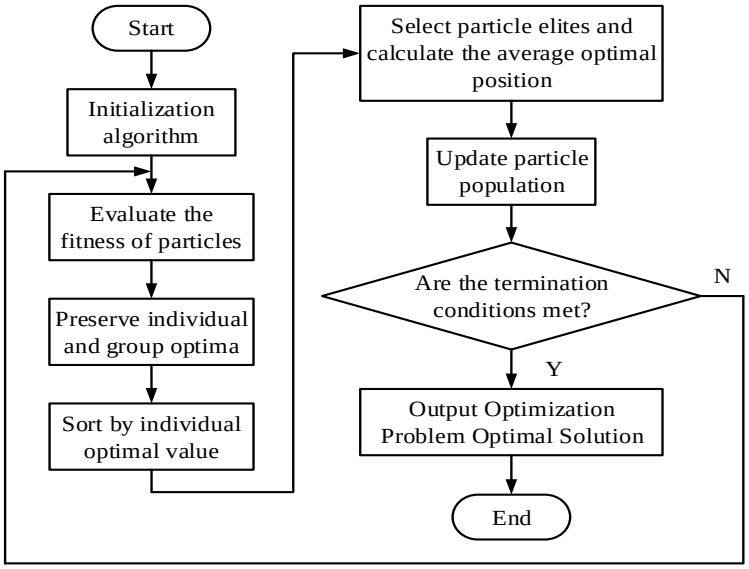


Figure 2: QPSO algorithm flowchart

According to Figure 2, in QPSO, it is necessary to first initialize the algorithm and evaluate particle fitness, while retaining individual and group optimal values. Afterwards, sorting is carried out based on individual optimal values, and particle elites are selected to calculate the average optimal position. The particle population is updated in real-time to determine if it meets the termination criteria. If satisfied, the optimal solution can be output and the process ends. If not, it is necessary to re-evaluate particle fitness and continue cycling until the conditions are met. The innovation in Figure 2 lies in modeling the particle state as a quantum superposition state. The position update follows the wave function collapse law. The particle does not need to move along a continuous trajectory but covers the entire solution space with probability density, enabling the algorithm to have the quantum tunneling ability to cross the local optimal "barrier". In the state of quantum particles, particles no longer have definite positions and velocities. The state of each particle is described by a quantized wave function, which contains the point in space to find the probability density of the particle. Its normalized probability density function is described as equation (13).

$$|\psi(x)|^2 = 1/L_i(t) * \exp(-2|x - p_i(t)|/L_i(t)) \quad (13)$$

In equation (13), $L_{i(t)}$ represents the characteristic length scale, and this variable controls the dispersion of particle positions. $p_i(t)$ represents the local attraction point of particle i at time t, which is usually defined as the individual historical optimal position $p_{\text{best}}(t)$ and the group historical optimal position $g_{\text{best}}(t)$ of the particle. The random weighted average of this point is shown in equation (14).

$$p_i(t) = \phi_i(t) * p_{\text{best}}(t) + (1 - \phi_i(t)) * g_{best}(t) \quad (14)$$

In equation (14), $\phi_i(t)$ represents the uniform distribution of the local attraction points of the particle in the interval [0, 1]. For the particle position update method, QPSO is obtained by folding the wave function. In quantum mechanics, observation causes the wave function to collapse to a definite state. In QPSO, each iteration is equivalent to a "measurement" of particle position. For equation (13), the position update equation of particle *i* at the *t+1*st generation can be derived through the Monte Carlo method, as detailed in equation (15).

$$x_i(t+1) = p_i(t) \pm (L_i(t/2) * \ln(1/u_i(t)) \quad (15)$$

In equation (15), $u_i(t)$ represents a random number uniformly distributed within the interval (0, 1). $\pm$ represents taking the plus or minus sign with equal probability, reflecting the randomness of quantum measurement. Particles can directly "tunnel" to positions far from $p_i(t)$, endowing the algorithm with stronger global exploration capabilities. The characteristic length scale $L_i(t)$ of QPSO is usually related to the state of the entire particle swarm to guide convergence. The most common definition is based on the deviation between the average value $m_{\text{best}}(t)$ of the optimal positions of all individual particles and the current particle position, as detailed in equation (16).

$$\begin{cases} L_i(t) = 2 * \beta * |m_{\text{best}}(t) - x_i(t)| \\ m_{\text{best}}(t) = (1/M) * \sum_{j=1}^{M} p_{j,best}(t) \end{cases} \quad (16)$$

In equation (16), M represents the size of the group. β stands for the contract-expansion coefficient, which is the most crucial control parameter of QPSO, directly determining the convergence behavior of the algorithm and the balance of global/local search capabilities. When β is greater than 1, $L_i(t)$ increases, the search space is broad, and global search is emphasized. When β is less than 1, the search is concentrated near $p_i(t)$, emphasizing the local search. In QPSO, the research aims to linearly reduce β over time to achieve early exploration and later utilization.

3.2 Intelligent fire warning based on optimized QPSO

After solving the TU problem of multiple platforms in OD information system, how to effectively utilize the large amount of data generated by OD information system on a daily basis is an urgent problem that needs to be solved. By mining the hidden relationships between data, people can realize statistics and prediction. Research was conducted on fire warning algorithm based on QPSO, which optimizes neural networks. The temperature, smoke, and CO data transmitted from OD fire protection system are processed using neural network algorithms to predict the likelihood of a fire and provide early warning before it, thereby improving OD safety [27]. Due to the significant losses caused by fires in people's production and daily life, people have conducted in-depth research on the processing methods of fire signals. Through mathematical analysis and processing of the general laws hidden in a fire, predictions can be made about the development of the fire. When a fire occurs, the range of fire signal processing is determined by processing and analyzing these signals detected by sensors to determine whether a fire has occurred. If the fire alarm signals processing is modeled using a function, a variable random signal is composed of fire and non-fire signals in equation (17).

$$x(t)=\begin{cases} x_f(t)+x_m(t) \\ x_m(t) \end{cases} \tag{17}$$

In equation (17), $x_f(t)$ represents the characteristic parameter signal that will be generated when a fire occurs. $x_m(t)$ is a signal change caused by environmental interference, which is a non-fire signal. Figure 3 shows fire detection.

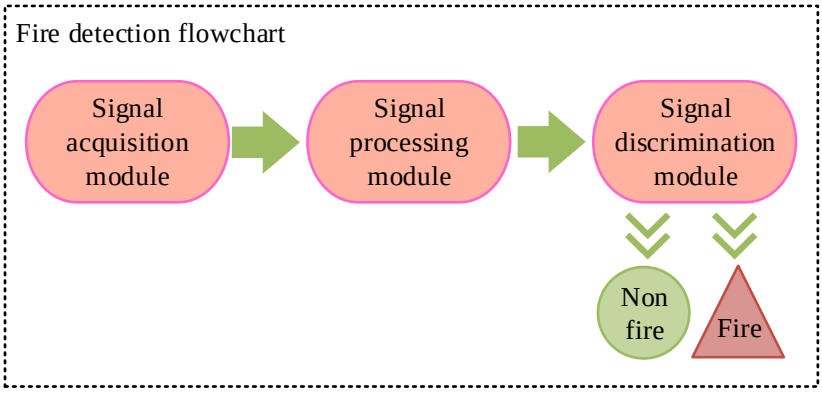


Figure 3: Fire detection flowchart

According to Figure 3, fire detection comprises three main modules: signal acquisition, signal processing, and signal judgment, which distinguish between fires and non-fires. Fire signals are non-stationary random signals, and various factors should be carefully considered when setting thresholds. In fire warning systems, machine learning is employed alongside multi-sensor data to enhance the system's resilience to interference and improve early warning accuracy.

Neural networks, inspired by the human brain, are effective in analyzing information and providing real-time feedback while learning autonomously [28]. Artificial neural networks, particularly BPNN, have shown great efficacy in fire warning applications. In this intelligent fire warning algorithm, the input layer utilizes temperature, smoke, and CO signals. After signal preprocessing, the network is trained to establish the relationship between these signals and fire probability, facilitating accurate fire prediction. Figure 4 shows the neural network fire prediction process.

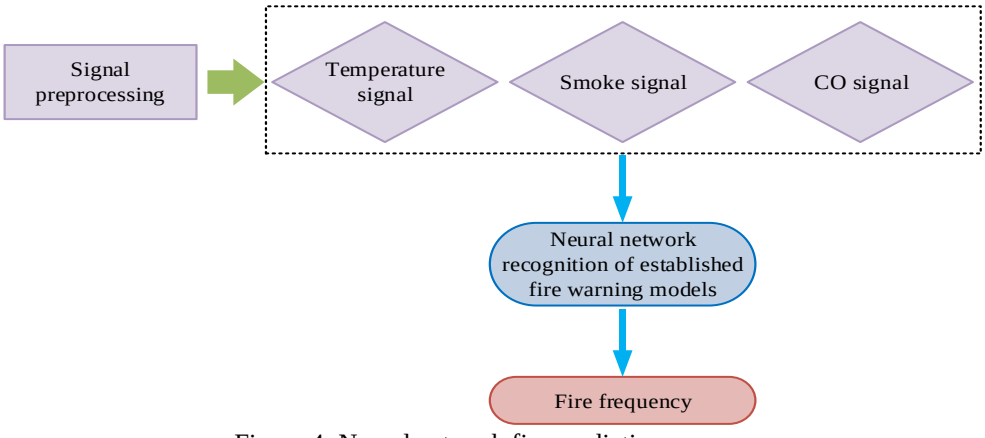


Figure 4: Neural network fire prediction process

In the initial ignition stage of a fire, identifiable characteristics are minimal, with low temperatures and little smoke, making human detection challenging. However, sensors can monitor environmental changes and use intelligent algorithms for early warnings, potentially preventing fire escalation. As the fire smolders, flames are absent, yet thick smoke and significant environmental changes occur. During active combustion, large flames and intense heat raise temperatures rapidly. In the extinguishing stage, the fire weakens, temperatures decline, and smoke is produced.

These four typical fire stages can vary significantly depending on the combustible materials involved, resulting in different fire behaviors. For instance, some substances burn slowly with minimal temperature changes that may go undetected by sensors, leading to false alarms. Others may release volatile substances, produce smoke, and cause similar detection issues. Therefore, fire detection is complex and influenced by multiple factors.

Most ordinary detection (OD) fire protection systems rely on intuitive or system-driven detection methods, which often lack robustness, leading to false alarms and missed detections, endangering safety. Neural network algorithm is used to analyze the feature changes of fire signals, especially temperature, smoke, and CO data. Multi parameter fusion can significantly improve the accuracy and resilience of the system. This method minimizes false alarms and missed detections in complex environments to the greatest extent possible.

To implement this solution, sensor data must be preprocessed and normalized for effective recognition by the neural network. This process enhances network training efficiency and enables rapid and precise fire predictions for early warning, thus improving the safety of OD systems.

The OD fire protection system is to enhance production safety by normalizing data gathered from temperature, smoke, and CO sensors. Integrating QPSO with BPNNs facilitates multi-parameter fusion analysis, improving prediction accuracy and reducing false alarms and missed detections, thereby safeguarding personnel and ensuring operational stability. The initial weight and threshold selection in BPNN are random. Therefore, this method may experience slow convergence and local minimum problems [29]. QPSO's powerful global search capability can effectively optimize these parameters and address these shortcomings. Figure 5 shows the framework of the entire algorithm.

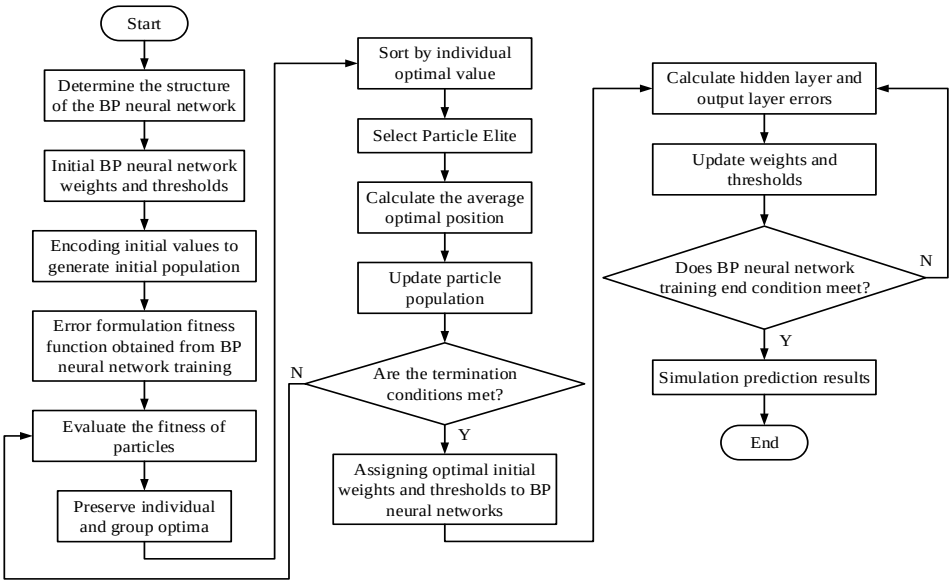


Figure 5: Overall framework of QPSO-BPNN algorithm

In Figure 5, the study integrates physical laws and quantum computing into the fire perception system, forming a triple driving mechanism of "data-physics-quantum". This study innovatively embeds the heat conduction equation as a hard constraint layer into the input, forcing the neural network to learn fire characteristic expressions that conform to thermodynamic laws. This research mainly used QPSO to optimize BPNN to achieve the functionality of OD fire protection system fire warning. The operation process of this algorithm mainly consists of four parts: data preprocessing, neural network structure determination, QPSO optimization of weights and thresholds, and optimized neural network prediction. Each part cooperates with each other to improve prediction accuracy and output fire prediction values. In data preprocessing, this study will use data from temperature, smoke, and CO sensors as inputs to the neural network. The influence between different sensors is eliminated by homogenization, and then the data is normalized. In the neural network structure determination, it is necessary to determine network input layer and output layer, the number of hidden layer nodes, and activation function. Using the Softmax activation function, its output value is

directly interpreted as the predicted probability of each fire category (for example: non-fire, smoldering fire, open flame). The Softmax function ensures that the sum of the predicted probabilities of all categories strictly equals 1, thereby providing an effective and normalized probability distribution. Through research and analysis, both the input and output layers' nodes are 3. The former includes temperature data, smoke concentration data, and CO concentration data. The latter are the probabilities of non-fire, smoldering fire, and open fire, respectively. The hidden layer node can be selected by equation (18).

$$n = \sqrt{n_1 + n_2} + n_3 \tag{18}$$

In equation (18), n, n_1, and n_2 represent the number of nodes in hidden, input, and output layers, respectively. n_3 is a constant between $(0,10)$. Equation (18) is merely an enlightening method for initial topological exploration, and its physical basis is proportional compression for dimensionality reduction in the feature space. In the final deployment model, this equation will be replaced by sensitivity analysis to ensure an exact match between network capacity and data complexity. According to this equation, the hidden layer nodes are 8. Since Sigmoid function is widely used in neural network, it is regarded as an activation function of neural network. The structure of BPNN can be obtained in Figure 6.

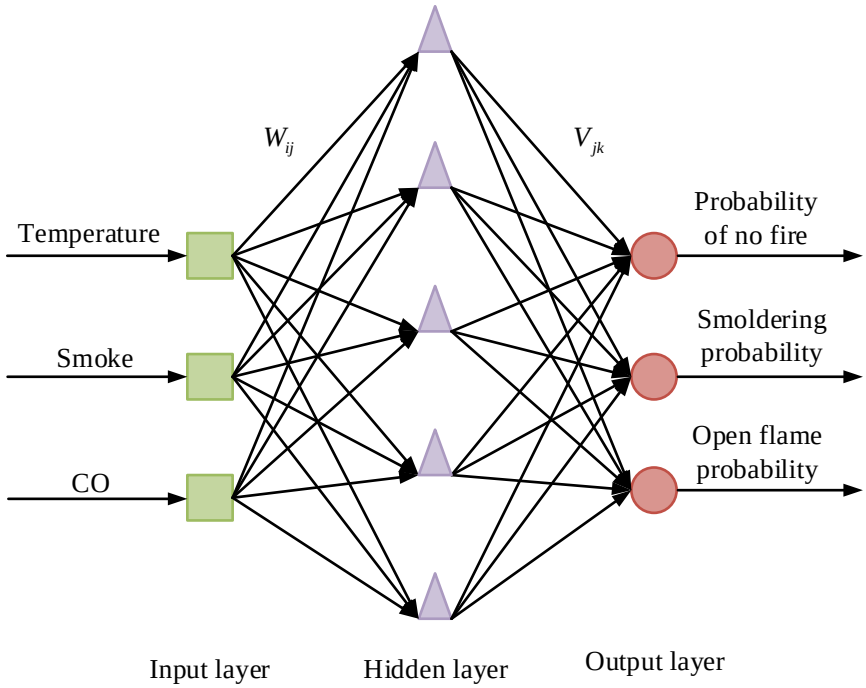


Figure 6: Neural network structure diagram

Figure 6 shows the in-depth customized transformation based on the characteristics of the OD fire scene. Firstly, a multimodal feature separation processing channel is constructed. The temperature branch adopts a temporal convolutional network to capture thermodynamic transient responses. The smoke branch integrates attention gating to enhance feature saliency discrimination. The CO concentration channel sets up a differential amplification link to improve the detection ability of weak signals. After analyzing the BPNN structure, QPSO algorithm is used to optimize its weights and thresholds, to compensate for its shortcomings. The learning performance of BPNN is good, but due to the random selection of weights and thresholds, there are drawbacks such as slow convergence and easy falling into local minima. QPSO algorithm has strong global search ability and achieved good results by optimizing parameters such as weights and thresholds, thereby reducing the computational complexity of neural network and improving accuracy. In the application of QPSO, the initial threshold and weights randomly generated in neural network are first encoded and corresponding to the particles to generate a particle swarm. Then, the optimal algorithm is used to solve a single optimal solution and decoded as a weight threshold. The average error generated by neural network training samples is used to determine individual's fitness value, and then particles are updated.

The fundamental reason for choosing the mean square error (MSE) loss function and the Sigmoid output layer architecture for three-category fire prediction lies in the fact that this combination perfectly fits the multi-state coexistence characteristics of OD fires and the features of industrial data. Sigmoid solves the state coexistence problem by outputting probabilities through independent nodes, avoiding the mutual exclusion constraint distortion of Softmax. The quadratic surface optimization characteristics of MSE loss are superior to cross-entropy on gradient smoothness, multi-peak convergence ability, and outlier robustness.

Finally, the initial weights and thresholds of network were corrected. Where equation (19) is the fitness

function.

$$f(x)=\sum_{i=1}^{w}(y_1-y_2)^2 \tag{19}$$

In equation (19), w represents the number of samples, y_1 and y_2 represent the actual and expected outputs of network, respectively. On this basis, particle state is evaluated by evaluating their fitness. Finally, BPNN is optimized with the optimal solution as weight and threshold. It is optimized using QPSO algorithm to obtain the optimal weights and thresholds, which were used as inputs to BPNN. When training it, further optimization and adjustment can be made based on the feedback information.

Through the above analysis, QPSO-BPNN is a key technical combination. QPSO is an extended PSO algorithm. It utilizes the uncertainty principle in quantum mechanics to optimize the search process of particles, thereby enhancing the global search ability and avoiding falling into local optima. Combined with BPNN, QPSO plays an important role in optimizing the weights and thresholds of neural networks. This optimization can not only accelerate the convergence speed, but also improve the accuracy, thereby enabling more effective prediction of fire situations.

4 Multi-platform TU and fire warning simulation analysis of QPSO

QPSO is simulated and analyzed on multiple platforms TU and fire warning, and MATLAB is used as the simulation software to set the relevant parameters of the simulation environment. The output probabilities of QPSO-BPNN open fire, smoldering fire, and non-fire are fitted in MATLAB. Fire prediction is carried out to output the probabilities of open fire, smoldering fire, and non-fire.

4.1 Simulation analysis of QPSO in multi-platform TU

In the experiment, the research focused on testing three benchmark problems to verify the effectiveness of QPSO-BPNN in the prediction of OD fires. These benchmark issues mainly focus on different scenarios and conditions in fire detection, specifically including the three states of open flame, smoky flame and non-fire. The research constructs a fully connected feedforward neural network model for fire detection. This network consists of an input layer, a hidden layer configured with different numbers of nodes (using the ReLU activation function), and an output layer corresponding to the detection category (using the Softmax activation function). The optimization process adopts stochastic gradient descent with nesterov momentum (SGD), with the learning rate set at a fixed 5e-4 and the momentum factor at 0.9. The training adopts a small-batch strategy, with the batch size fixed at 128. To prevent overfitting, the study adds L2 weight regularization to the loss function, sets the regularization coefficient to 1e-4, and implements an early stop strategy. If the loss of the validation set does not decrease within 10 consecutive training cycles, the training will be terminated (the upper limit of the maximum training cycle is 200). The network weights are initialized using the He initialization method optimized for the ReLU activation function. Training is conducted on a randomly divided training set (70%), with the validation set (15%) used for early stop and the test set (15%) for the final performance reporting.

To construct the training set and test set, a total of 3,600 training samples are selected from the experiments. In addition, 500 test samples are designed to verify the predictive ability of the model. These samples cover different sensor readings and environmental variables that may occur during a fire, enabling the model to be trained in various situations and thereby enhancing its generalization ability and accuracy. During the experimental process of the model, the research processed the input data through standardized preprocessing techniques to eliminate the influence between the readings of different sensors. Meanwhile, in order to evaluate the model effect, not only the training error is focused on, but also the recognition accuracy on the test set is evaluated in detail.

The experiment is conducted in the MATLAB R2023a environment, mainly relying on the following toolboxes: The Neural Network Toolbox is used to construct and train BPNN models, and the Global Optimization Toolbox implements quantum behavior QPSO algorithms to tune network parameters. Performance evaluation and index calculation are carried out using Statistics and Machine Learning Toolbox. All experiments are run on workstations equipped with NVIDIA GeForce RTX 3090 Gpus, taking the Parallel Computing Toolbox to accelerate the training process. The research has clarified the performance indicators for comparison, including convergence speed, solution quality, resource consumption, robustness and scalability. These indicators provide a multi-dimensional evaluation criterion to ensure that the research can comprehensively reflect the performance of different algorithms. Secondly, the study selects a set of representative benchmark problems to cover diverse complexities and characteristics. The selected benchmark problems not only include simple optimization problems, but also cover medium-complexity and large-scale complex problems, ensuring the comprehensiveness of algorithm comparison. In terms of the final experimental design, the research maintains the consistency of the experimental environment, ensures that all algorithms run under the same hardware and software conditions, and conducts multiple experiments to obtain reliable datasets. Meanwhile, detailed parameter tuning is carried out for each algorithm to ensure the fairness of the comparison process. The simulation environment includes a multi platform task processing system composed of multiple cloud servers, edge servers, and terminal devices. Table 1 shows the relevant parameter settings for simulation analysis.

Table 1: Multi-platform uninstallation simulation parameters

Number	Simulation parameters	Value	Number	Simulation parameters	Value
(1)	Number of end device users	30	(6)	MEC server parallel computing maximum number of users	8
(2)	Terminal device computing power	0.5~1G Hz/s	(7)	Number of cloud servers	5
(3)	Number of MEC base stations	10	(8)	Cloud server computing power	50~100GHz/s
(4)	Number of MEC servers	10	(9)	Maximum number of users for cloud server parallel computing	30
(5)	MEC server computing power	5~10G Hz/s	/	/	/

To eliminate the configuration bias, the experiment adopts a hierarchical random sampling strategy to ensure the universality of the results. All range-type parameters in Table 1 are independently sampled according to a uniform distribution at each run. For each algorithm comparison point, 100 independent experiments are conducted. Each experiment regenerates the complete parameter set, and the Halton sequence is adopted to ensure uniform coverage of the parameter space. The specific form of random seed management is shown in Table 2.

Table 2: Random seed management form

```
# The core pseudo-code of the experiment
seeds = np.arange(100)    # Fixed seed range[0,99]
results = []
for seed in seeds:
    np.random.seed(seed)
    config = generate_config(Table1_ranges)    # Generate parameters according to the range in Table 1
    energy = QPSO_optimizer(config)
    results.append(energy)
final_energy = np.mean(results)    # Report average value
C.I = bootstrap_ci(results)         # 95% confidence interval
```

To ensure fairness, system-level hyperparameter tuning is implemented on the PSO baseline model. The search space is defined as follows: The inertia weight values are [0.4, 0.9], the individual learning factor is [0.5, 2.5], the social learning factor is [0.5, 2.5], and the population size is [0.5, 2.5]. By adopting the Bayesian optimization framework and taking the average fitness of 100 independent runs as the objective function, the Pareto optimal solution set is obtained after 50 rounds of iterations. The final optimal parameter values are as follows: The inertia weight is set at 0.72, and the individual learning factor is 1.85. The social learning factor is 1.78. The population size is 60.

The main objective of the research is to minimize the total energy consumption of the system while ensuring fairness in device level energy consumption through constraints. The research conducts a sensitivity analysis on the number of hidden layer nodes of the model and determines the number of hidden layer nodes. The specific results are shown in Table 3.

Table 3: Sensitivity analysis of the number of nodes in the model's hidden layer

Number of nodes	6	8	10	12	15
Smoldering fire recall rate (%)	91.2±0.8	93.5±0.6	95.8±0.5	95.2±0.7	95.7±0.6
Open flame F1 score (%)	98.9±0.3	99.1±0.2	99.3±0.2	99.4±0.1	99.2±0.3
Model inference delay (ms)	4.3±0.2	5.1±0.3	6.4±0.4	7.9±0.5	10.2±0.6
Parameter scale	8.2K	12.7K	18.6K	25.3K	36.8K
Energy consumption (mJ)	0.82	1.05	1.37	1.78	2.31

Based on the sensitivity analysis results in Table 3, the increase in the number of nodes in the hidden layer had a differentiated impact on performance and resource consumption. The initial stage of increasing the number of nodes had a significant impact on the improvement of accuracy. However, after exceeding 10 nodes, the marginal improvement in accuracy was limited, while the model complexity and resource consumption continued to increase significantly. Therefore, choosing 10 nodes achieves a better balance between accuracy and efficiency.

Firstly, a multi-platform TU was conducted using classical PSO and QPSO, and the optimal fitness convergence during the uninstallation optimization process was obtained in Figure 7.

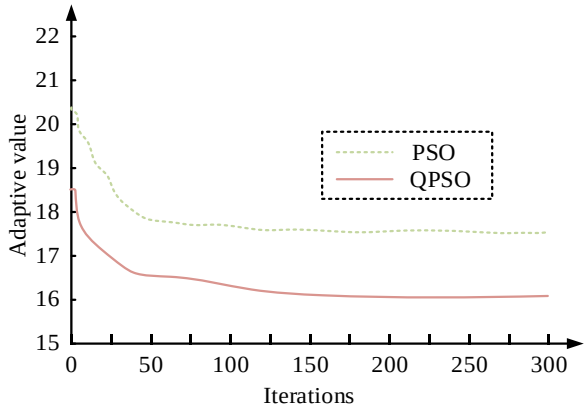
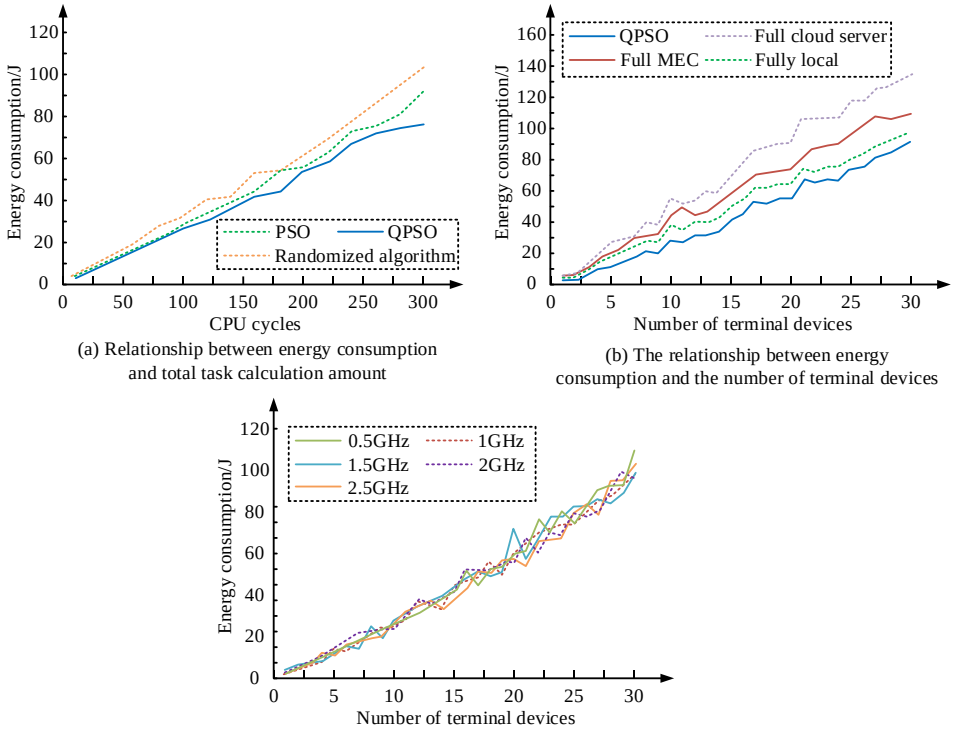


Figure 7: Convergence of fitness values for PSO and QPSO

From Figure 7, compared with PSO, QPSO has faster convergence rate and results. The initial fitness values of PSO and QPSO were 20.2 and 18.4, respectively, and the optimal fitness values of two algorithms were 17.52 and 16.09, respectively. PSO and QPSO achieved optimal fitness values at 231 and 130 iterations, respectively, indicating that QPSO has significant advantages in multi-platform TU. Afterwards, the relationship between energy consumption and total task computation, energy consumption and number of terminal devices, and energy consumption and number of terminal devices under different local computing capabilities is analyzed in Figure 8.



(a) Relationship between energy consumption and total task calculation amount

(b) The relationship between energy consumption and the number of terminal devices

(c) The relationship between energy consumption and the number of terminal devices under different local computing capabilities

Figure 8: The relationship between energy consumption and total task computation, as well as the number of terminal devices

Figures 8 (a), 8 (b), and 8 (c) respectively represent the relationship between energy consumption and total task computation, energy consumption and number of terminal devices, and energy consumption and number of terminal devices under different local computing capabilities. In Figure 8 (a), the order of energy consumption from high to low and stability from low to high is random algorithm, PSO, and QPSO. In comparison, the latter two algorithms can achieve energy optimization under delay constraints, and QPSO is more effective in reducing energy consumption as the computational load increases. Compared with PSO, QPSO could reduce energy consumption by up to 17.1%. In Figure 8 (b), when terminal devices were 20, the energy consumption of multi-platform TU was 55.8 J. The energy consumption for fully local was 64.5 J, for fully MEC was 73.72, and for fully cloud servers was 90.5. Compared with fully local algorithms, the research algorithm saved 13.5%, 24.3%, and 38.3% of energy consumption, respectively. This indicates the effectiveness, and provides ideas for upgrading OD information systems. It can abandon the original centralized server computing mode and introduce MEC and cloud servers to intelligently offload tasks across multiple platforms. This can not only enhance system's risk resistance, but also effectively reduce energy consumption and improve efficiency. In Figure 8 (c), the multi-platform uninstallation algorithm using this research method can steadily reduce energy consumption, improve terminal devices efficiency in completing tasks, and achieve multi-platform TU. The practical significance of industrial applications is that in upgrading

OD information system, there is no need to significantly upgrade the original terminal equipment configuration. Instead, it is necessary to arrange MEC servers and cloud servers, cooperate with this proposed algorithm, and achieve the effect of reducing energy consumption, reducing costs, and upgrading and optimizing. To further verify the feasibility of the proposed method, it is compared with multiple baseline models, including fire detection methods based on the rule-based system, support vector machine (SVM), random forest (RF), one-dimensional convolutional neural network (1D-CNN), and long short-term memory network (LSTM). The comparative experiment results of different models are shown in Table 4.

Table 4: The results of comparative experiments of different models

Scene/Method	Rule system	SVM	RF	1D-CNN	LSTM	QPSO-BPNN
Open flame identification	94.2±1.9	96.8±1.5	97.5±1.3	98.3±1.0	98.9±0.8	99.6±0.8
95% confidence interval	[92.3, 96.1]	[95.3, 98.3]	[96.2, 98.8]	[97.3, 99.3]	[98.1, 99.7]	[98.8, 100.0]
Smoldering fire identification	70.1	78.3	85.4	86.7	90.2	95.8
95% confidence interval	[66.7, 73.5]	[75.4, 81.2]	[82.9, 87.9]	[84.3, 89.1]	[88.1, 92.3]	[94.1, 97.5]
Welding interference false alarm	63.5	32.1	18.7	8.9	5.4	0.3
95% confidence interval	[60.0, 67.0]	[28.8, 35.4]	[16.0, 21.4]	[6.9, 10.9]	[3.8, 7.0]	[0.0, 0.9]
Steam interference false alarm	48.9	25.6	12.3	7.1	3.8	0.1
95% confidence interval	[45.3, 52.5]	[22.5, 28.7]	[10.0, 14.6]	[5.3, 8.9]	[2.5, 5.1]	[0.0, 0.5]
Robustness in high-humidity environments	65.3	82.7	88.5	83.9	91.2	93.6
95% confidence interval	[61.9, 68.7]	[80.0, 85.4]	[86.3, 90.7]	[81.3, 86.5]	[89.2, 93.2]	[91.8, 95.4]

Based on the statistical significance test results in Table 4, the QPSO-BPNN model demonstrated statistically significant superiority in key performance indicators. This model outperformed all baseline methods with an accuracy of 95.8% in the smoldering fire identification task. Its 95% confidence interval [94.1, 97.5] was higher than that of the model LSTM [88.1, 92.3], with no interval overlap and Cohen's d effect size reaching 1.28, indicating that the 5.6% performance improvement was of substantial significance. In the industrial interference scenario, QPSO-BPNN compressed the welding interference false alarm to 0.3%, and the upper limit of the confidence interval at 0.9% was significantly lower than the lower limit of LSTM at 3.8%. The upper limit of the 0.1% range for the steam interference false alarm rate, which was 0.5%, also completely deviated from the confidence range of all baselines, forming a statistical dissociation. Although the interval [98.8, 100.0] of QPSO-BPNN in the open flame recognition scenario partially overlapped with that of LSTM [98.1, 99.7], its 99.6% point estimate still remained leading. In the high-humidity environment adaptability performance, the 93.6% robustness index had limited overlap with the 91.2% interval of LSTM, and further sample expansion verification was necessary. These quantitative evidences confirm the breakthrough progress of this method in reducing the false alarm rate of industry and improving the reliability of smoldering fire detection. To further verify the performance of the model, the study quantifies the full classification performance and false positives of the model. The specific results are shown in Table 5.

Table 5: Model full classification performance and false alarm quantification

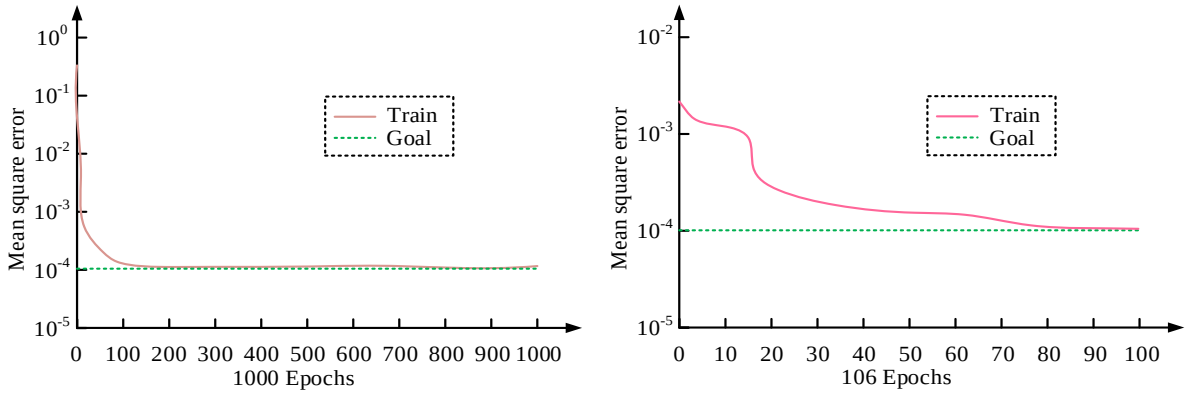
Indicator	Smoldering	Open flame	Non-fire	Macro avg
Precision /%	95.8 ±0.5	99.4 ±0.2	98.1 ±0.3	97.8 ±0.3
Recall /%	95.2 ±0.6	99.3 ±0.3	98.9 ±0.4	97.8 ±0.4
F1 score /%	95.5 ±0.4	99.4 ±0.2	98.5 ±0.3	97.8 ±0.3
Supported sample size	500	500	500	1,500
False positive	21±2	6±1	/	/
False negative	24±3	7±2	/	/

The results in Table 5 show that the proposed model has the best open flame detection performance, with an F1 score reaching 99.4%, followed by the precision and recall of open flames. The F1 score of the smoldering flame was 95.5%. Although it has a relatively high performance, it is lower than that of the open flame test, indicating that it is more susceptible to interference. In terms of false negative risk, the false negative rate of smoldering fire was 24±3 cases. The false positive rate was 21±2 cases, both higher than that of open flames. The results indicate that detecting smoldering fires may delay the fire response. A low false alarm rate for non-fire categories indicates a lower false alarm rate in the system, which helps reduce redundant operations. In safety-critical systems, although the overall recall of fires is relatively high, the false negative of smoldering fires needs to be prioritized for optimization to meet strict safety standards.

3.2Optimization of QPSO Performance and Simulation Analysis of fire warning

The research adopts a strict industrial-grade experimental design, deploying 12 Omega K-type thermocouples (with an accuracy of ±0.75% and a sampling frequency of 10Hz) to monitor the synchronous changes of the temperature field. The smoke detection uses Honeywell XNXTM laser particle counter (with a range of 0-20mg/m^3, calibrated weekly by NIST traceable standard smoke). The CO sensor adopts Alphasense CO-AF electrochemical sensor (with a minimum detection limit of 5ppm, and the drift compensation algorithm is automatically activated every 8 hours). The experiment is conducted in three 20×20m² standard oil tank simulation chambers for 18 months, covering an environmental temperature range of -15℃ to 45℃. Before data collection, environmental background noise baseline calibration is implemented (including dust interference spectrum characteristics ≥78 dB and oil transfer pump vibration noise 15-300 Hz). For each type of fire scene (open flame/smoldering/non-fire), the test is repeated 50 times. Three typical industrial interference sources, namely welding arc, steam injection, and diesel engine exhaust, are introduced to generate countermeasure samples (accounting for 25% of the total data). The humidity control maintains a 30-90% RH gradient through the Vaisala HMP155 probe. All sensors are synchronized in real-time through the OPC UA protocol, with a timestamp accuracy of ±5ms. Eventually, an industrial fire dataset containing 412,800 time-domain samples is constructed. Its noise characteristics are confirmed by Welch power spectrum analysis to comply with the ASTM E681 explosive environment certification standard.

Meanwhile, the research uses representative and authoritative fire data to train the QPSO-BPNN algorithm. The sample data selects typical interference signals in Chinese standard open flames, standard smoldering fires, and kitchen environments. A total of 3,600 samples are selected from each of the three scenarios to form 36 sets of sample data, among which 31 sets of samples are used as the training set and 5 sets as the test set. MATLAB is used for simulation analysis. First, the experiment compares and analyzes the mean squared error of QPSO-BPNN and BP training process in Figure 9.



(a) Mean squared error of BP Neural Network (b) Mean squared error Diagram of QPSO-BP Neural Network

Figure 9: QPSO-BP and Mean squared error during BP training

In Figure 9, the mean squared error of QPSO-BPNN and BP reached the expected error value after 106 iterations and 180 iterations respectively, and the former's convergence rate and global search ability are stronger. QPSO algorithm is used to optimize BP network's weights and thresholds, which can effectively improve algorithm convergence and prevent it from falling into local minima. Afterwards, the outputs of QPSO-BPNN open fire, smoldering fire, and non-fire probabilities were fitted in MATLAB in Figure 10.

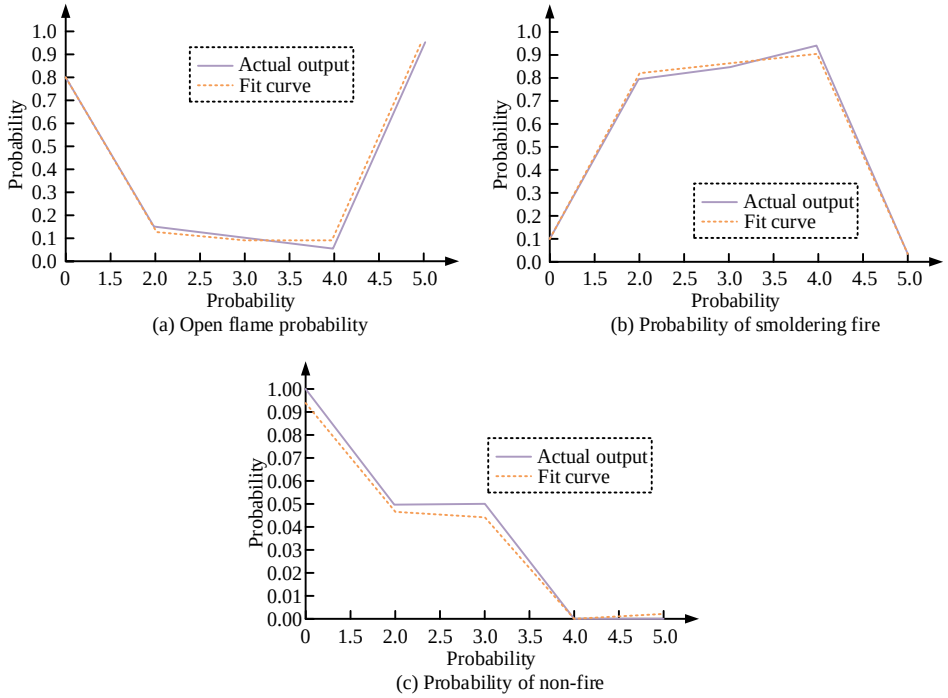


Figure 10: Fitting of probability output of open fire, smoldering fire, and non-fire

According to Figure 10, the output probabilities' numerical curves of open fire, smoldering fire, and non-fire are basically consistent with the actual fire probability, while the maximum error of non-fire probability is only 0.005. This indicates that neural network model can effectively identify and predict fire signals, thereby improving OD fire protection system's safety and accuracy. To further validate this research, a fire experiment was designed using a Raspberry Pi development board paired with temperature, smoke, CO sensors, and Python language. It was programmed and equipped with a multi-sensor data acquisition device. The obtained data was normalized and used as validation samples to test whether the trained model can effectively predict fires.

Fire data collected by multi-sensor data collection equipment will be classified into three states: open fire, non-fire, and smoldering. Five sets of data will be selected from each state to form 15 sets of validation samples to test QPSO-BPNN model's effectiveness. The selected samples were preprocessed and then imported into a model in MATLAB for fire prediction, and the probabilities of open fire, smoldering fire, and non-fire were output. Figure 11 shows the results of an open flame.

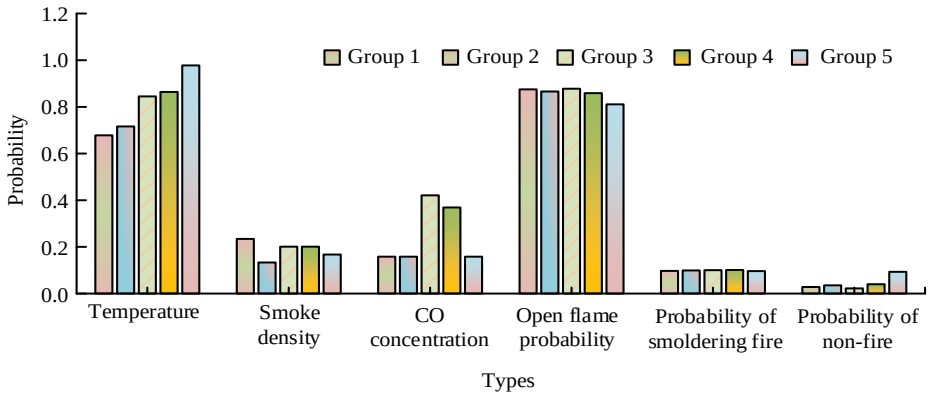


Figure 11: Predictive output of open flame data

From Figure 11, the experimental data of temperature, smoke, and CO are normalized. The output probability obtained by inputting QPSO-BPNN is matched with the corresponding test fire situation. This indicates that QPSO-BPNN model can effectively identify open flame situations and there are no false or missed reporting. Figure 12 shows the output result of smoldering fire.

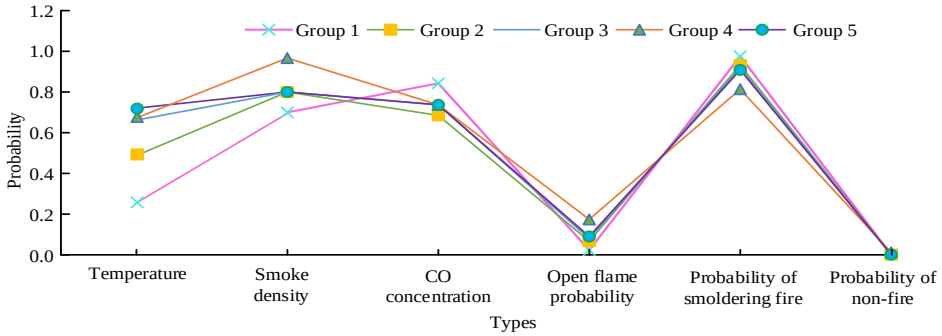


Figure 12: Predictive output of smoldering fire

From Figure 12, the experimental data of temperature, smoke, and CO are normalized. The output probability obtained by inputting QPSO-BPNN is matched with the corresponding experimental fire situation. This indicates that the QPSO-BPNN model can effectively identify smoldering fire situations without any false or missed reporting.

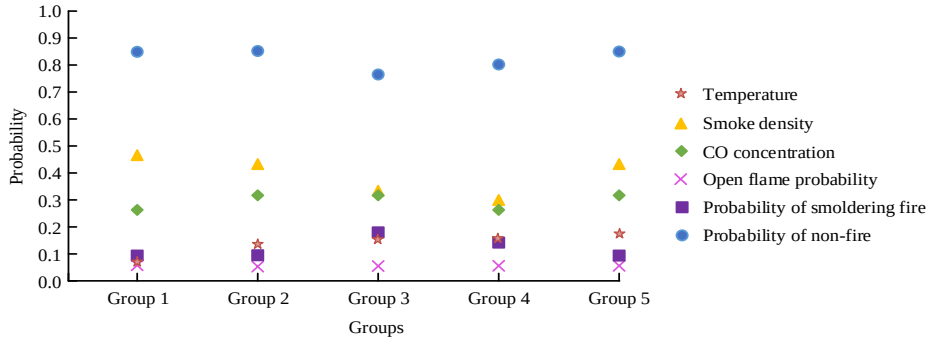


Figure 13: Predictive output without fire

From Figure 13, the experimental data of temperature, smoke, and CO are normalized. The output probability obtained by inputting QPSO-BPNN is matched with the corresponding test fire situation. This indicates that QPSO-BPNN model can effectively identify the situation of non-fire, and there are no false or missed reporting. By judging and predicting the fire situation, accurate judgment and prediction of the fire situation have been achieved, thereby improving the accuracy of oil depot fire warning. The comparison results of the computational complexity of the proposed model are shown in Table 6.

Table 6: Analysis of model computational complexity

Evaluation index	PSO	QPSO	Relative change
Average time of a single iteration	286±12ms	318±15ms	0.112
Floating-point operation (FLOP) in each iteration	7.2×10^9	8.5×10^9	0.181
The number of iterations required for convergence	152±8	87±6	-0.428
Total training time	43.5±2.1s	27.7±1.8s	-0.363
Peak memory usage	1.8GB	2.1GB	0.167
Iterative efficiency index	0.53	1.18	1.226

In Table 6, the QPSO demonstrated a significant computational efficiency advantage over standard PSO. Although the average time consumption of a single iteration increased by 11.2% to 318ms, mainly due to the matrix feature decomposition operation load during the evolution of the quantum wave function, its unique quantum tunneling property significantly reduced the number of iterations required for convergence by 42.8%, and only 87 iterations were needed to achieve the optimization goal. This mechanism effectively avoids the local optimum predicament that traditional algorithms are prone to fall into, ultimately compressing the global training time of the 18.6K parameter model to 27.7 seconds, saving 36.3% of the time cost compared with standard PSO. This efficiency leap is particularly significant in the high-dimensional parameter space, with an iterative efficiency index reaching 1.18, equivalent to a 122.6% increase in the effective convergence step size obtained per unit time. Quantitative computing confirms that the quantum mechanism significantly reduces the total training cost with a controllable single-iteration increment, providing algorithm-level support for the rapid deployment of edge devices.

In response to the core engineering requirements of the fire early warning system in the production environment, this study conducts technical argumentation and trade-off

analysis on the system deployment and real-time performance. In terms of technical demonstration, the system adopts a hierarchical processing architecture to ensure controllable end-to-end latency. At the terminal device layer (deployed on NVIDIA Jetson Nano), the model inference delay is controlled within 11.3ms after being accelerated by TensorRT. When transmitting alarm signals through the LoRaWAN protocol, the transmission delay was 162ms at a maximum distance of 5km. The threshold response time of the cloud-based decision engine was less than 2ms. As a result, the total delay of the worst-case alarm was 175.3ms, which was better than the 1-second limit stipulated in the national standard GB 50116-2023. To cope with sudden loads, the system is designed with a dual-buffered queue. When the concurrent data stream exceeds 50 frames per second, the downsampling mode is automatically activated to stabilize the delay within 200ms. In terms of trade-off analysis, the specific results are shown in Table 7.

Table 7: Measurement of system-level performance trade-off parameters

Optimization dimension	Alarm delay (ms)	Energy consumption (mJ/ Inference)	Fault recovery rate (%)
Edge computing power	11.3	9.8	99.97
	48.7	3.2	99.93
Transmission protocol	162	0.15/pack	98.1
	28	4.7/pack	99.6
Model complexity	6.4	1.37	99.98
	10.2	2.31	99.97
Fault suppression	+8.1ms	+1.9mJ	>99.99
	/	/	95.4

The results in Table 7 showed that although the high-end embedded platform provided an extreme delay of 11.3ms, which was 4.3 times faster than that of the research method mode, the energy consumption cost was as high as 9.8mJ per inference. The LoRaWAN achieved an ultra-low power consumption of 0.15mJ per packet with a delay of 162ms, but it needed to tolerate a network packet loss rate of 2.9%. The current 10-node BPNN achieved a macro average F1 score of 97.8% with a delay of 6.4ms and an energy consumption of 1.37mJ. While the 15-node model improved the recall by 0.5%, the delay increase reached 59% and the energy consumption rose to 2.31mJ. Dual-modal redundancy check had an additional delay of 8.1ms and an energy consumption cost of 1.9mJ, but it increased the system failure recovery rate from 95.4% to no less than 99.99%. Based on the above considerations, the QPSO-BPNN+LoRaWAN combination solution is selected. Under an average daily energy consumption of 86J, it ensures that 99.5% of alarm delays are less than 200ms and fire recall rates are greater than 99%.

5 Conclusion

The intelligent TU strategy of platform was determined based on QPSO. An intelligent fire warning algorithm based on QPSO optimized BPNN was constructed to achieve intelligent prediction of fires. The initial fitness values of PSO and QPSO are 20.2 and 18.4, respectively. The optimal fitness values of two algorithms are 17.52 and 16.09, respectively. PSO and QPSO achieved optimal fitness values at 231 and 130 iterations, respectively, indicating that QPSO has significant advantages in multi-platform TU. Compared with completely local algorithms, the research algorithm has saved 13.5%, 24.3%, and 38.3% of energy consumption, indicating its effectiveness. QPSO algorithm is used to optimize BPNN's weights and thresholds, which can effectively improve its convergence and prevent it from falling into local minima. From the curve plots of open fire, smoldering fire, and non-fire and the actual fire probability, the numerical curve is basically consistent, while the maximum error of non-fire probability is only 0.005. This indicates that QPSO-BPNN model can effectively identify the situation of non-fire, and there are no false or missed reporting. Although breakthroughs have been made in the field of OD fire early warning research, the following key limitations and deepening directions still exist. The current QPSO optimization process relies on the quantum behavior simulation of classical computers and has not yet been integrated with real quantum hardware. The computational complexity of quantum simulators grows exponentially, resulting in the online optimization delay of TU still failing to meet the requirements of millisecond-level industrial control. In fire prediction scenarios, the parameter quantum circuits in quantum neural networks are limited by the simulator bit width, making it difficult to handle sensor fusion features exceeding 100 dimensions. Future research will focus on quantum classical fusion hardware carriers and system level autonomous collaboration mechanisms, developing room temperature quantum coprocessors based on photonic integrated circuits, and achieving task offloading optimization delay compression to the level of 300ms. A quantum blockchain disaster consensus network can be constructed to ensure cross tank instruction transmission through Shor algorithm signature and BB84 protocol encryption.

References

[1] Zhao Y., Song X., Wang F., Cui D., Multiobjective optimal dispatch of microgrid based on analytic hierarchy process and quantum particle swarm optimization, Global Energy Interconnect., 3, 6, 562-570, December 2020. https://doi.org/10.1016/j.gloei.2021.01.008

[2] Wang J., Zhao X., Yu Q., Zhao C., Inverse modeling of thermal decomposition of flame-retardant PET fiber with model-free coupled with particle swarm optimization algorithm, ACS Omega, 6, 21, 13626-13636, May 2021. https://doi.org/10.1021/acsomega.1c00599

[3] Rashad O., Attallah O., Morsi I., A smart PLC-SCADA framework for monitoring petroleum products terminals in industry 4.0 via machine learning, Meas. Control, 55, 7-8, 830-848, May 2022.https://doi.org/10.1177/00202940221103305

[4] Xie S., Dong S., Chen Y., Peng Y., Li X., A novel risk evaluation method for fire and explosion accidents in oil depots using bow-tie analysis and risk matrix analysis method based on cloud model theory, Reliab. Eng. Syst. Saf., 215, Nov., 107791.1-107791.18, May 2021.https://doi.org/10.1016/j.ress.2021.107791

[5] Aliser A., Duranay Z B., Fire/Flame Detection with Attention-Based Deep Semantic Segmentation. Iranian Journal of Science and Technology, Transactions of Electrical Engineering,48, 2, 705-717, February 2024.https://doi.org/10.1007/s40998-024-00697-y

[6] Amjad A., Huroon A M., Chang H T., Tai L C., Dynamic fire and smoke detection module with enhanced feature integration and attention mechanisms. Pattern Analysis and Applications, 28, 2, 1-17, April 2025.https://doi.org/10.1007/s10044-025-01461-6

[7] Hiremath H., Kannan S R., Integrated Anomaly Detection and Early Warning System for Forest Fires in the Odisha Region. Atmosphere, 15, 11, 1284-1284, October 2024.https://doi.org/10.3390/atmos15111284

[8] Kale M P., Meher S S., Chavan M., Kumar V., Sultan M A., Dongre P l., et al. Operational Forest-Fire Spread Forecasting Using the WRF-SFIRE Model. Remote Sensing, 16, 13, 2480-2495, July 2024.https://doi.org/10.3390/rs16132480

[9] Liu P., Xiang P., Lu D., A new multi-sensor fire detection method based on LSTM networks with environmental information fusion. Neural Computing and Applications, 35, 36, 25275-25289, June 2023.https://doi.org/10.1007/s00521-023-08709-4

[10] Hosseinmardi A M., Farvaresh H., Nakhai Kamalabadi I., A new formulation and algorithm for the maximum leaf spanning tree problem with an application in forest fire detection. Engineering Optimization, 55, 7, 1226-1242, June 2023. https://doi.org/10.1080/0305215X.2022.2067151

[11] Liu X., Sun B., Xu Z.D., Liu X., Xu D., Identification of multiple fire sources in the utility tunnel based on a constrained particle swarm optimization algorithm, Fire Technol., 58, 5, 2825-2845, September 2022.https://doi.org/10.1007/s10694-022-01284-5

[12] Zeng M., Quan K., Improved hybrid cuckoo search-based quantum-behaved particle swarm optimization algorithm for bi-level programming, J. Comput. Appl., 40, 7, 1908-1912, March 2020.

[13] Tu S., Rehman O.U., Rehman S.U., Ullah S., Waqas M., Zhu R., A Novel Quantum Inspired Particle Swarm Optimization Algorithm for Electromagnetic Applications, IEEE Access, 8, 1, 21909-21916, January 2020.https://doi.org/10.1109/ACCESS.2020.2968980

[14] Yu L., Han Y., Mu L., Improved quantum evolutionary particle swarm optimization for band selection of hyperspectral image, Remote Sens. Lett., 11, 9, 866-875, September 2020.https://doi.org/10.1080/2150704X.2020.1782501

[15] Zhou H., Bai H., Cai Z., Cai L., Gu J., Tang Z., Container Quota Optimization Algorithm Based on GRNN and LSTM, Acta Electronica Sinica, 50, 2, 366-373, September 2022.

[16] Wang Y., Han X., Jin S., MAP based modeling method and performance study of a task offloading scheme with time-correlated traffic and VM repair in MEC systems. Wireless Networks, 29, 1, 47-68, April 2023.https://doi.org/10.1007/s11276-022-03099-2

[17] Guowei Z., Su Y., Guoqing Z., Pengyue F., Smart firefighting construction in China: Status, problems, and reflections, Fire Mater., 44, 4, 479-486, June 2020.https://doi.org/10.1002/fam.2800

[18] Wu D., Li Y., Du C., Ding H., Li Y., Fast velocity trajectory planning and control algorithm of intelligent 4WD electric vehicle for energy saving using time-based MPC, IET Intel. Transport Syst., 13, 1, 153-159, January 2019.https://doi.org/10.1049/iet-its.2018.5103

[19] Cong Y., Xue K., Wang C., Sun W., Sun S., Hu F., Latency-energy joint optimization for task offloading and resource allocation in MEC-assisted vehicular networks. IEEE Transactions on Vehicular Technology, 72, 12, 16369-16391, July 2023.https://doi.org/10.1109/TVT.2023.3289236

[20] Hua W., Zhou Z., Huang L., Location privacy-aware offloading for MEC-enabled IoT: Optimality and heuristics. IEEE Internet of Things Journal, 10, 21, 19270-19281, August 2023.https://doi.org/10.1109/JIOT.2023.3281609

[21] Bahrami B., Khayyambashi M R., Mirjalili S., Multiobjective placement of edge servers in MEC environment using a hybrid algorithm based on NSGA-II and MOPSO. IEEE Internet of Things Journal, 11, 8, 29819-29837, September 2024.https://doi.org/10.1109/JIOT.2024.3409569

[22] Raza S., Optimizing Energy and Delay in Task Offloading for Connected Vehicles with Proximal Policy Optimization. IETI Transactions on Data Analysis and Forecasting, 2, 2, 29-40, November 2024.https://doi.org/10.3991/itdaf.v2i2.50949

[23] Li H., Liu J., Yang L., Lu L., Sun H., An improved arithmetic optimization algorithm for task offloading in mobile edge computing. Cluster Computing, 27, 2, 1667-1682, June 2024.https://doi.org/10.1007/s10586-023-04048-0

[24] Asghari A., Azgomi H., Zoraghchian A A., Barzegarinezhad A., Energy-aware server placement in mobile edge computing using trees social relations optimization algorithm. The Journal of Supercomputing, 80, 5, 6382-6410, October 2024.https://doi.org/10.1007/s11227-023-05692-4

[25] Banerjee S., Biswash S K., Performance-Driven Resource Allocation Strategy in NDN-Based Mobile Edge Computing (MEC) Networks. Arabian Journal for Science and Engineering, 50, 10, 7741-7764, October 2025.https://doi.org/10.1007/s13369-024-09455-y

[26] Asghari A., Azgomi H., Zoraghchian A A., Barzegarinezhad A., Energy-aware server placement in mobile edge computing using trees social relations optimization algorithm. The Journal of Supercomputing, 80, 5, 6382-6410, October 2024.https://doi.org/10.1007/s11227-023-05692-4

[27] Chen L.Q., Feng Y.X., Zhen H.L., Li Q.V., Lu J.Y., Risk assessment of large crude oil depot based on interpretative structural model and system dynamics, INT J OIL GAS COAL T., 25, 2, 161-183, January 2020.https://doi.org/10.1504/IJOGCT.2020.109445

[28] Mahdiani M.R., Khamehchi E., Hajirezaie S., Hemmati-Saraparden A., Modeling viscosity of crude oil using k-nearest neighbor algorithm, Advances in Geo-Energy Research, 4, 4, 435-447, December 2020.https://doi.org/10.46690/ager.2020.04.08

[29] Bikmukhametov T., Jschke J., Oil Production Monitoring using Gradient Boosting Machine Learning Algorithm, IFAC-PapersOnLine, 52, 1, 514-519, January 2019. https://doi.org/10.1016/j.ifacol.2019.06.114