
A Prediction Optimization Method with Federated Learning and Neural Architecture Search for Distributed Renewable Energy Sources

Jun Su^{1,2}, Chaolong Tang¹ and Zhiquan Liu¹

¹*School of Electrical Engineering and Automation, Xiamen University of Technology, Xiamen, Fujian Province, China, 361012*

²*Xiamen Key Laboratory of Frontier Electric Power Equipment and Intelligent Control, Xiamen, Fujian Province, China, 361012*

E-mail: junsu1989@163.com

**Corresponding Author*

Received 05 October 2024; Accepted 23 March 2025

Abstract

This paper proposes a prediction optimization method with federated learning and neural architecture search for distributed renewable energy sources, which is suitable for intelligent management of multi node distributed energy networks. The federated learning technology is used to enable each energy node to independently train local neural network models without sharing raw data, ensuring data privacy and security, and achieving collaborative optimization among nodes through the aggregation of global model parameters. By combining neural architecture search (NAS) technology, the system can automatically design the optimal neural network architecture and dynamically adapt to the energy demands and environmental changes of different nodes. This method improves the efficiency and robustness of distributed renewable energy systems through real-time scheduling optimization and load forecasting, and is suitable for various renewable energy generation

Distributed Generation & Alternative Energy Journal, Vol. 40_2, 213–238.

doi: 10.13052/dgaej2156-3306.4021

© 2025 River Publishers

scenarios such as wind power and photovoltaics. Finally, the experiment results have shown that the method proposed in this paper improves the intelligence of energy scheduling and the accuracy of prediction, ensuring data security and the adaptive optimization capability of the system.

Keywords: Distributed energy system, predictive optimization, federated learning, neural architecture search, adaptive control.

1 Introduction

With the continuous growth of global energy demand and the increasing emphasis on environmental protection, renewable energy, especially distributed energy systems such as wind power and photovoltaics, have become an important development direction in the energy field [1, 2]. However, the efficient management and optimization scheduling of distributed energy systems face many challenges, such as data privacy protection, coordinated optimization of heterogeneous nodes, and dynamic adaptability in complex environments. Traditional centralized scheduling methods rely on centralized data processing, which poses risks of data privacy breaches and computational resource bottlenecks, making it difficult to meet the needs of modern distributed energy systems [3].

Based on the summary and analysis of existing research results, traditional distributed energy scheduling methods mainly rely on centralized data processing and fixed model architectures, which have the following limitations [4–6]: (1) Data privacy risks: Centralized data processing methods require uploading data from each node to a central server, which poses a risk of data privacy leakage [7]. (2) Computing resource bottleneck: Centralized methods require high computing resources from central servers, making it difficult to meet the demands of large-scale distributed energy systems [8]. (3) Poor model adaptability: The fixed model architecture is difficult to adapt to the diverse needs of different nodes and dynamically changing environments, resulting in a decrease in prediction accuracy and scheduling efficiency [9, 10]. (4) Poor model generalization ability: Most existing scheduling and prediction methods are based on pre-defined fixed models, which often struggle to adapt to the diversity and dynamic changes of distributed energy nodes [11]. Due to significant differences in energy demand and environmental conditions among nodes, a single model is difficult to achieve good performance on all nodes, resulting in unsatisfactory scheduling and prediction results [12]. (5) Lack of collaborative optimization

mechanism: In distributed systems, each node usually schedules and predicts independently, lacking effective collaborative optimization mechanisms [13, 14]. This decentralized optimization approach can easily lead to low overall system efficiency, inability to fully utilize the collaborative effects of each node, resulting in resource waste and low energy utilization. (6) The design of neural network architecture is complex: most existing methods rely on human experience and trial and error methods when designing neural network architecture, which is time-consuming, laborious, and difficult to ensure that the designed architecture is optimal [15]. Especially in complex distributed renewable energy systems, manually designed neural network architectures often fail to meet practical needs, resulting in unstable model performance.

In addition, the existing methods lack effective fault tolerance and adaptive mechanisms when facing system failures or abnormal situations, which can easily lead to a significant decline in system performance [16]. Especially in distributed systems, node failures and network instability often affect the overall stability and reliability of the system. Meanwhile, traditional scheduling methods perform poorly in resource utilization and cannot fully utilize the computing and storage resources of distributed energy nodes [17]. This not only affects the overall efficiency of the system, but also increases energy consumption and operating costs [18].

To solve the above problems, this paper proposes a prediction optimization method with federated learning and neural architecture search for distributed renewable energy sources, which is suitable for intelligent management of multi node distributed energy networks. The system uses federated learning technology to enable each energy node to independently train local neural network models without sharing raw data, ensuring data privacy and security, and achieving collaborative optimization among nodes through the aggregation of global model parameters. By combining neural architecture search technology, the system can automatically design the optimal neural network architecture and dynamically adapt to the energy demands and environmental changes of different nodes. This method improves the efficiency and robustness of distributed renewable energy systems through real-time scheduling optimization and load forecasting, and is suitable for various renewable energy scenarios such as wind power and photovoltaics.

The main contribution of this paper can be summarized as,

- (1) A distributed renewable energy scheduling and prediction optimization method and system based on federated learning and neural architecture

search is proposed, it can improve the scheduling efficiency and prediction accuracy of distributed renewable energy systems, while ensuring data privacy and security.

- (2) The optimal neural network architecture combining NAS technology is designed to dynamically adapt to the needs and environmental changes of different distributed energy nodes, greatly improving the system's adaptability and robustness.
- (3) The combination of federated learning and NAS is discussed to achieve multi-nodes collaborative optimization, enabling the system to effectively utilize computing resources in distributed systems, improve overall scheduling efficiency.

The rest of this paper is organized as follows. In Section 2, the characteristics of distributed renewable energy sources is analyzed. In Section 3, the proposed prediction optimization with federated learning and neural architecture search for distributed energy sources is presented. In Section 4, the combination and collaboration of federated learning and neural architecture search is discussed. Subsequently, the proposed method is validated in Section 5. Finally, the conclusion is drawn in Section 6.

2 Characteristics of Distributed Renewable Energy Sources

As shown in Figure 1, the typical renewable energy systems mainly include wind farms and photovoltaic power stations, which have significant distribution characteristics and are usually distributed in areas with wide geographical locations and abundant resources. Wind farms are usually located in coastal or mountainous areas with abundant wind resources, while photovoltaic power plants are mostly distributed in deserts, deserts, and plateaus with abundant sunshine. This distribution characteristic determines that the power generation capacity of the renewable energy system is closely related to factors such as geographical location and meteorological conditions.

In the renewable energy system, the output of wind power and photovoltaic power generation has a high degree of randomness and volatility. The power generation of wind farms is influenced by meteorological factors such as wind speed and direction, while the power generation of photovoltaic power plants is mainly affected by factors such as solar radiation intensity and temperature. In order to achieve more accurate load forecasting and scheduling optimization, it is necessary to establish mathematical models

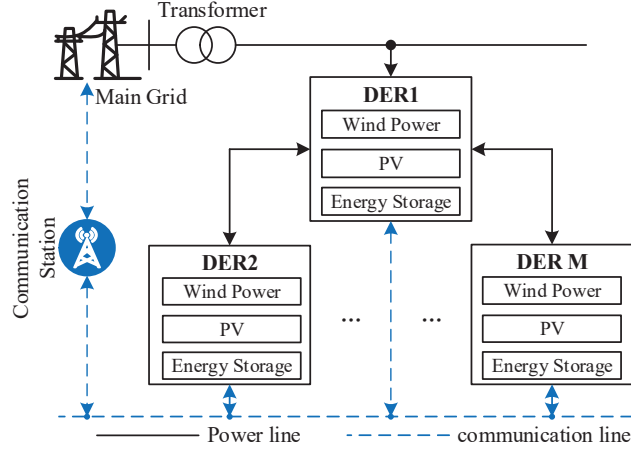


Figure 1 Distributed renewable energy system model.

to describe the relationship between these influencing factors and power generation.

Assuming that the relationship between the power generation P_w and wind speed v of a wind farm can be expressed by the following formula:

$$P_w = \frac{1}{2} \rho A v^3 C_p(\lambda, \beta) \tag{1}$$

where ρ is the air density, A is the swept area of the wind turbine, $C_p(\lambda, \beta)$ is the power coefficient, λ is the blade speed ratio, and β is the blade angle.

Similarly, the relationship between the power generation P_{pv} and the solar radiation intensity I can be expressed,

$$P_{pv} = \eta AI \tag{2}$$

where η is the conversion efficiency of the photovoltaic module, A is the area of the photovoltaic module, and I is the solar radiation intensity.

In the practical applications, the power generation of wind and photovoltaic power is not only affected by the above factors, but also by other environmental factors such as temperature, humidity, air pressure, etc. Therefore, it is necessary to extract the characteristics of these influencing factors through big data analysis and machine learning methods, and establish a multivariate prediction model.

Meanwhile, the autonomy and intelligence level of nodes are key to achieving efficient operation. The node is equipped with advanced sensors

and data acquisition devices, which can collect and process data in real-time locally, monitor operating status, load demand, and environmental changes. These data are analyzed and processed through intelligent algorithms to achieve precise control over energy production and consumption. In order to adapt to different load demands and market changes, the nodes adopt advanced control strategies and optimization algorithms, which can flexibly switch between multiple operating modes. For example, through predictive models and optimization algorithms, nodes can prioritize the use of energy storage systems during peak load periods and store energy during low load periods.

3 Prediction Optimization with Federated Learning and Neural Architecture Search for Distributed Energy Sources

3.1 Application Model of Federated Learning in Distributed Systems

In the scheduling and prediction process of distributed systems in the field of renewable energy, they face multiple challenges such as data decentralization, privacy protection, and efficient utilization of computing resources. Federated learning can effectively solve these problems by training local models on various distributed nodes and summarizing the model parameters of each node without exchanging raw data.

The application model of federated learning in distributed systems first requires defining the local training process of each node. Assuming there are N nodes in the system, each node i has a local dataset D_i . The training objective of the local model is to minimize the loss function $L_i(\theta)$, where θ is the model parameter. The local loss function can be expressed as:

$$L_i(\theta) = \frac{1}{|D_i|} \sum_{(x_j, y_j) \in D_i} \ell(f(x_j; \theta), y_j) \quad (3)$$

where $\ell()$ is the loss function, $f(x_j; \theta)$ is the predicted output of the model, and y_j is the true label. Each node performs gradient descent update on the local dataset,

$$\theta_i^{t+1} = \theta_i^t - \eta \nabla L_i(\theta_i^t) \quad (4)$$

where η is the learning rate, $\nabla L_i(\theta_i^t)$ is the gradient of loss function L_i with respect to parameter θ_i .

In each round of federated learning, each node sends locally updated model parameters to the central server, which aggregates these parameters. One of the commonly used aggregation methods is weighted average, and the aggregated global model parameter θ^{i+1} can be expressed as:

$$\theta^{t+1} = \sum_{i=1}^N \frac{|D_i|}{\sum_{j=1}^N |D_j|} \theta_i^{t+1} \quad (5)$$

Through the above process, federated learning can improve the generalization ability of the model by utilizing global information without exchanging raw data, while protecting data privacy.

In addition, this paper also considers the personalized needs of each node. In renewable energy dispatch and load forecasting, different nodes may have different characteristics and demands. In order to implement a personalized model, the present invention introduces a personalized layer. After the global model training is completed, each node can further fine tune the personalization layer on the local dataset to adapt to local characteristics. The update formula for parameter f_i of the personalization layer is:

$$\phi_i^{t+1} = \phi_i^t - \eta \nabla L_i(\phi_i^t) \quad (6)$$

In this way, each node not only shares the advantages of the global model, but also can make personalized adjustments based on local data, further improving prediction accuracy.

3.2 Construction and Optimization of Neural Architecture Search (NAS) Model

This paper establishes an intelligent and efficient neural architecture search model to meet the complex needs of distributed renewable energy scheduling and load forecasting. The NAS model can dynamically adjust the network architecture in a changing environment by automating the search of neural network structures, thereby achieving efficient model training and dynamic architecture optimization while ensuring data privacy.

The construction of NAS model mainly includes three parts: search space, search strategy, and performance estimation.

- (1) Search space: The search space defines all possible neural network architectures. In the present invention, the search space includes a combination of various neural network layers (such as convolutional layers, fully connected layers, pooling layers, etc.) and their hyperparameters

(such as convolution kernel size, stride, activation function, etc.). In order to reduce the complexity of the search space, the present invention adopts a modular design, dividing the neural network into several basic modules, each module containing several optional operations.

- (2) Search strategy: The search strategy is used to select the optimal neural network architecture in the search space. Common search strategies include evolutionary algorithms, reinforcement learning, and Bayesian optimization. In the present invention, a distributed search strategy based on reinforcement learning is adopted, taking into account the characteristics of federated learning. Specifically, the policy gradient method in reinforcement learning was adopted to generate candidate architectures through policy networks, and parallel evaluation was performed using the client in federated learning to accelerate the search process.
- (3) Performance estimation: Performance estimation is used to evaluate the performance of candidate architectures. Traditional performance estimation methods require training each candidate architecture, which is impractical in situations where computing resources are limited. To solve this problem, the present invention adopts a performance estimation method based on a proxy model. Specifically, training a proxy model to predict the performance of candidate architectures reduces the number of actual training iterations.

To describe the optimization process of the NAS model, the following formula is introduced:

Firstly, define the parameters of the policy network as θ , the candidate architecture as a , and its performance as $R(a)$. The goal of the strategy gradient method is to maximize the expected performance of candidate architectures:

$$\max_{\theta} \mathbb{E}_{a \sim \pi(\theta)} [R(a)] \quad (7)$$

Where, $\pi(\theta)$ represents the probability distribution of candidate architectures generated by the policy network. By using the gradient ascent method, the parameters of the policy network can be updated:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \mathbb{E}_{a \sim \pi(\theta)} [R(a)] \quad (8)$$

Under the federated learning framework, the client evaluates the performance of candidate architectures in parallel and returns the results to the central server. The central server updates the parameters of the policy network based on feedback results, gradually approaching the optimal architecture.

The NAS model proposed in this paper has significant advantages in the field of distributed renewable energy scheduling and load forecasting. Firstly, by automating the search of neural network architecture, the workload of manual design is reduced and the adaptability of the model is improved. Secondly, by combining federated learning technology, efficient model training and dynamic architecture optimization can be achieved while ensuring data privacy. Finally, the performance estimation method based on the proxy model significantly reduces the consumption of computing resources, thereby improving search efficiency.

4 Combination and Collaboration of Federated Learning and Neural Architecture Search

4.1 Integration Mechanism of Federated Learning and NAS

This paper combines federated learning with Neural Architecture Search to achieve the goal of distributed renewable energy scheduling and predictive optimization. This scheme utilizes local data at each node in a distributed system and adopts NAS algorithm to automatically optimize the neural network architecture, thereby significantly improving scheduling efficiency and load forecasting accuracy.

Federated learning is a distributed machine learning method that allows multiple nodes to collaboratively train a global model without sharing local data. Specifically, each node trains the model on local data and sends the model parameters (rather than the data itself) to the central server. The central server aggregates model parameters from various nodes, updates the global model, and then distributes the updated model parameters back to each node. Through this iterative process, federated learning achieves global model optimization while ensuring data privacy. Neural architecture search is an automated method for designing neural network structures, which searches various candidate architectures in the search space to find the optimal neural network structure. NAS typically optimizes network architecture through methods such as reinforcement learning, evolutionary algorithms, or gradient descent. The core of NAS lies in its automation and efficiency, which can find the best performing model among a large number of possible network architectures. The combination mechanism of federated learning and NAS is as follows:

- (1) Local model training: Each distributed node first undergoes preliminary model training on its local data, using the current neural network

architecture for parameter optimization. Assuming that the local data of the i -th node is D_i and its model parameters are θ_i , the objective function for local training can be expressed as:

$$\min_{\theta_i} \mathcal{L}(D_i; \theta_i) \quad (9)$$

Where \mathcal{L} represents the loss function.

- (2) Model parameter aggregation: Each node sends its trained model parameters to the central server. The central server aggregates these parameters to generate global model parameter θ . The commonly used aggregation method is weighted average:

$$\theta = \sum_{i=1}^N \frac{|D_i|}{\sum_{j=1}^N |D_j|} \theta_i \quad (10)$$

Where $|D_i|$ represents the data volume of the i -th node, and N is the total number of nodes.

- (3) NAS optimization: Based on the global model parameters, the central server performs neural architecture search. Specifically, NAS searches for various candidate architectures in the space to find the architecture that performs best on global data. Assuming the search space is \mathcal{A} , the objective function of NAS is:

$$\alpha^* = \arg \max_{\alpha \in \mathcal{A}} \mathcal{P}(\alpha; \theta) \quad (11)$$

Where α represents the candidate architecture, and \mathcal{P} represents the performance evaluation function of the architecture.

- (4) Architecture update and distribution: The central server sends back the optimized optimal architecture α^* and its corresponding parameters θ^* to each node. After receiving the new architecture and parameters, each node continues to train and optimize on local data.

As shown in Figure 2, by combining federated learning with NAS, not only can the local data of each node in the distributed system be fully utilized to improve the generalization ability and prediction accuracy of the model, but also the neural network architecture can be automatically optimized through NAS to enhance the efficiency and performance of the model. This scheme achieves the optimal solution for distributed renewable energy scheduling and prediction while ensuring data privacy, and has broad application prospects and practical value.

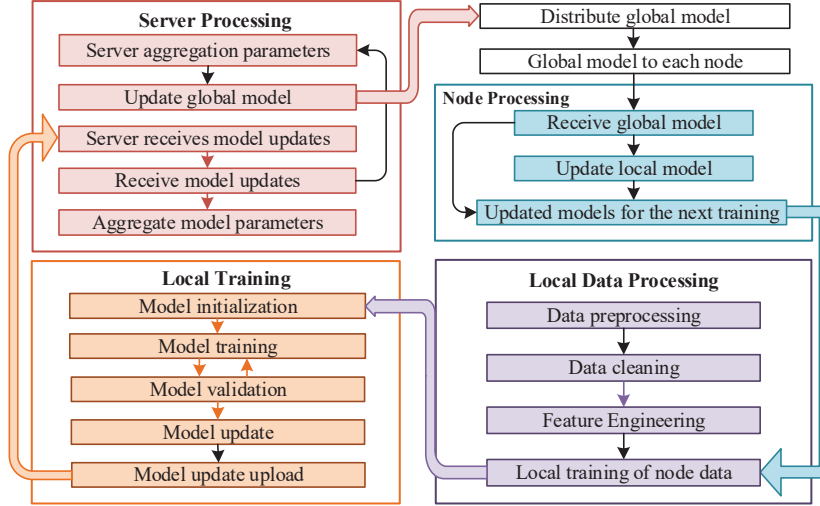


Figure 2 Combination mechanism and collaborative method based on federated learning and neural architecture search.

4.2 Multi-node Collaborative Optimization Scheme

This paper proposes an efficient multi node collaborative optimization scheme based on federated learning and neural architecture search, aiming to achieve high efficiency and accuracy in distributed renewable energy scheduling and prediction, while ensuring data privacy protection and dynamic adaptability.

Firstly, utilizing federated learning techniques to collaboratively train data distributed across different nodes without the need to centralize the data onto a central server. Each node trains a model locally and sends its parameters to the central server for aggregation, forming a global model. Federated learning achieves global model optimization while ensuring data privacy through periodic parameter exchange and updates. Specifically, assuming there are N nodes, each node i has a local dataset D_i , the update process of global model parameter θ can be expressed as:

$$\theta_{t+1} = \theta_t - \eta \sum_{i=1}^N \frac{|D_i|}{|D|} \nabla \mathcal{L}_i(\theta_t) \tag{12}$$

Where η is the learning rate, $\mathcal{L}_i(\theta_t)$ is the loss function of node i , $|D_i|$ is the dataset size of node i , and $|D|$ is the total size of all node datasets.

NAS improves model performance and efficiency by automatically searching for the best neural network architecture. In the specific implementation, reinforcement learning (RL) method is used for architecture search, a controller network is defined to generate candidate architectures, and their performance is evaluated by training sub networks. The optimization objective and the constraints of the controller network can be expressed as:

$$\begin{cases} J(\theta_c) = \arg \max_{\theta_1, \theta_2, \dots, \theta_N} \sum_{n=1}^N J_{R_n}(\theta_c) \\ \text{s.t. } J_{C_n}(\theta_n) \leq 0 \quad n = 1, 2, \dots, N \\ D_{\text{KL}}(\theta_n, \theta_{n,t}) \leq \delta \quad n = 1, 2, \dots, N \end{cases} \quad (13)$$

where $\theta = \{\theta_1, \pi_2, \dots, \theta_N\}$ represents the set of control strategies for all intelligent agents; $D_{\text{KL}}(\cdot)$ is KL divergence function, used to measure the previous strategy $\pi_{n,t}$ and the updated strategy $\pi_{n,t+1}$, the distance between $t+1$ is constrained by the upper limit δ to ensure that the continuous strategies are similar to each other.

Where θ_c is the parameter of the controller network, $\pi_{\theta_c}(a|s)$ is the strategy generated by the controller network for architecture a , and $R(a)$ is the performance evaluation result of architecture a .

By combining federated learning with neural architecture search, the following steps are performed in each training cycle:

- (1) Each node trains the model on local data and updates parameters;
- (2) The local parameters are sent to the central server for aggregation to form a global model;
- (3) Using NAS technology to perform architecture search on the basis of the global model, optimize the neural network structure;
- (4) Distribute the optimized model parameters and architecture to each node for the next round of training.

This multi node collaborative optimization scheme has the following significant advantages:

- (1) Through federated learning technology, data privacy protection is achieved, avoiding security risks caused by data concentration;
- (2) Through neural architecture search technology, the neural network structure is dynamically optimized to improve the adaptability and performance of the model;

- (3) Through distributed training, the computational efficiency has been improved, adapting to the needs of distributed renewable energy scheduling and prediction in smart grids.

In summary, the multi node collaborative optimization scheme based on federated learning and neural architecture search proposed in this paper provides an efficient, accurate, data privacy protected, and dynamically adaptable solution for the development of smart grids.

5 Experimental Verification

In order to verify the application effect of the method proposed in this paper in distributed renewable energy systems, testing and validation were conducted based on specific implementation examples. The testing environment includes hardware configuration and software tools to ensure the comprehensiveness and accuracy of the testing.

5.1 Parameter Settings

The simulation model includes main components such as data collection and preprocessing module, federated learning model training and aggregation module, and neural architecture search optimization module, which can truly reflect the operating characteristics and control behavior of distributed renewable energy systems. The parameter settings of the simulation model refer to the technical specifications and operational data of distributed renewable energy systems, ensuring that the simulation results have practical significance. Simulate various scheduling strategies and load forecasting algorithms on the simulation model, and reorganize them, including different node numbers, load scenarios, energy types, etc., to test the adaptability and robustness of the system.

This study was conducted in a hardware environment equipped with an Intel Xeon Gold 6248 processor (2.5 GHz), 256 GB DDR4 memory, and 1 TB NVMe SSD storage. The operating system used was Ubuntu 20.04 LTS, and the simulation platform was MATLAB/Simulink R2022a. The simulation model includes data collection and preprocessing modules, federated learning model training and aggregation modules, and neural architecture search optimization modules. In the data collection and preprocessing module, data is collected at a sampling frequency of 1 Hz and cleaned using standardization and mean imputation methods. The federated learning model training and

Table 1 Parameters of the models

Name	Parameter	Values
Wind turbine unit	Rated power	2 MW
	Cut-in wind speed	3 m/s
	Rated wind speed	12 m/s
	Cut-out wind speed	25 m/s
PV module	Rated power	300 Wp
	Module efficiency	18%
	Temperature Coefficient	-0.45%/°C
Energy storage	Capacity	1 MWh
	Charge-discharge Efficiency	90%
Load characteristic	Peak load	5 MW
	Valley load	1 MW
Communication delay	Average delay	50 ms
	Maximum Delay	200 ms

Table 2 Parameters of the algorithm

Name	Parameter	Values
Federated Learning	Local training rounds	10
	Global aggregation frequency	30 mins
	Learning rate	0.01
NAS	search space size	1000
	Search algorithm	Reinforcement
Optimization	Objective	Minimize energy consumption
	constraint condition	System stability

aggregation module sets the local training rounds to 10 and the global aggregation rounds to 20, with a learning rate of 0.01. The optimization algorithm uses Adam [13]. The search space of the NAS optimization module includes 20 different network architectures, with evaluation metrics of accuracy and energy consumption. The search strategy adopts a reinforcement learning based architecture search, which is evaluated every 5 rounds of training.

When using MATLAB/Simulink platform for simulation testing, the parameter settings of the simulation model refer to the technical specifications and operating data of distributed renewable energy systems, ensuring that the simulation results have practical significance. The setting of simulation parameters is shown in Table 1. Where, the distributed renewable energy system dataset is set as actual operational data, covering wind and photovoltaic power generation data from 2020 to 2023. The specific parameter settings and algorithm settings are shown in Table 2.

5.2 Result Analysis

From Figure 3, it can be seen that the distributed scheduling algorithm based on federated learning and NAS exhibits high scheduling efficiency in different load scenarios. Specifically, during peak, mid peak, and low peak periods, the scheduling efficiency of the distributed scheduling algorithm based on federated learning and NAS is 80.65%, 89.80%, and 97.50%, respectively. In contrast, the scheduling efficiency of traditional centralized scheduling algorithms is 68.45%, 78.89%, and 90.12%, respectively, while the scheduling efficiency of distributed scheduling algorithms without NAS optimization is 73.25%, 84.50%, and 93.10%. This indicates that the distributed scheduling algorithm based on federated learning and NAS is significantly superior to the other two algorithms in all load scenarios. The distributed scheduling algorithm based on federated learning and NAS has a scheduling efficiency that is 12.20%, 10.91%, and 7.38% higher than traditional centralized scheduling algorithms during peak, medium, and low peak periods, respectively; 7.40%, 5.30%, and 4.40% higher than the distributed scheduling algorithm without NAS optimization. These data indicate that the method proposed in this paper has significantly improved scheduling efficiency in different load scenarios, especially during peak and mid peak periods. These results fully demonstrate the effectiveness and superiority of distributed scheduling algorithms based on federated learning and NAS in improving scheduling efficiency, and have strong adaptability and higher robustness in scenarios with large load fluctuations.

Figure 4 shows the accuracy of load forecasting using different methods. From the Figure 4, it can be seen that the load forecasting method based on

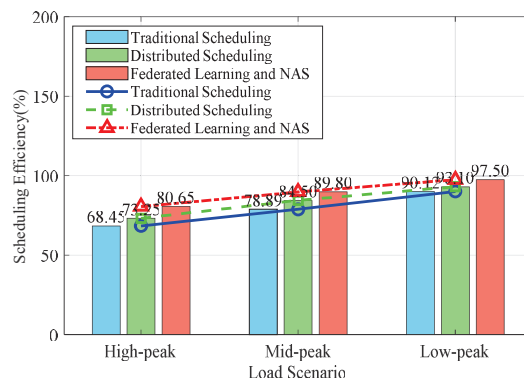


Figure 3 Comparison of distributed energy dispatch efficiency.

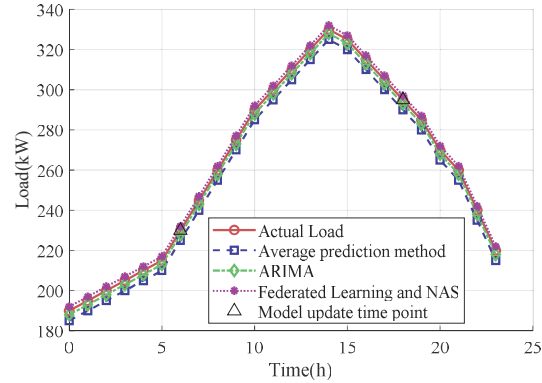


Figure 4 Comparative analysis of load forecasting accuracy.

federated learning and NAS proposed in this paper exhibits high prediction accuracy in 24-hour electricity load changes. In contrast, the traditional historical average prediction method and Autoregressive Integrated Moving Average Model (ARIMA) algorithm have certain deviations in their prediction results during the morning and evening peak periods, making it difficult to accurately capture the peak load changes. The actual load curve shows that the load remains at a low level from 0–5 am, starts to rise after 6 am, and reaches its peak during 12–14 pm, followed by a secondary peak during 18–19 pm. After 20:00 at night, the load gradually decreases, which conforms to the daily electricity consumption pattern. The prediction method based on federated learning and NAS closely fits the actual load, especially during peak load periods and rapid changes. The deviation between the predicted results and the actual values is small, demonstrating strong dynamic response and adaptive optimization capabilities. In contrast, the prediction curve of the historical average prediction method shows a certain degree of lag during peak periods, especially during the 12–14 noon period, when the predicted load is lower than the actual load, indicating its limitations in dealing with complex nonlinear load changes and difficulty in effectively capturing the dynamic changes of peak load. Although the ARIMA algorithm has good fitting ability for the overall load trend, the accuracy of the prediction results decreases during the rapid changes in load, especially from 6–7 and 18–19, showing limited ability to handle nonlinear load fluctuations. It is worth noting that the load forecasting method based on federated learning and NAS can quickly adapt to load changes after model updates at 6–7 and 18–19, and this characteristic is particularly significant during peak load periods.

This method not only improves the prediction accuracy of complex load patterns, but also significantly reduces the prediction error during high load periods. This result validates the superiority of the prediction model combining federated learning and NAS in distributed systems, especially in dealing with load forecasting tasks in multi node and heterogeneous environments, which can effectively improve the system's adaptability and robustness. This analysis results indicate that federated learning and NAS technology can not only improve the accuracy of load forecasting, but also have strong adaptive optimization capabilities and dynamic scheduling performance, which is of great significance for further improving the operational efficiency of smart grids and distributed energy systems.

Figure 4 compares the computational, communication, and storage energy consumption of centralized scheduling systems, distributed systems without NAS, and distributed systems based on federated learning and NAS. Through the stacked bar chart, it can be intuitively seen that the distributed system based on federated learning and NAS has significantly lower energy consumption indicators than the other two systems. The total energy consumption of the distributed system based on federated learning and NAS is 740.74 kWh, which saves 260.26 kWh compared to the centralized scheduling system and 110.11 kWh compared to the distributed system without NAS. This indicates that the method proposed in this paper has significant advantages in overall energy management and can effectively reduce the energy consumption of the system. Further analysis of specific energy consumption data shows that the distributed system based on federated learning and NAS has a computational energy consumption of 400.40 kWh, saving 100.10 kWh and 50.05 kWh respectively compared to centralized scheduling systems and distributed systems without NAS; In terms of communication energy consumption, it is 220.22 kWh, saving 80.08 kWh and 30.03 kWh; In terms of storage energy consumption, it is 120.12 kWh, saving 80.08 kWh and 30.03 kWh. These data indicate that the method proposed by the present invention performs well in various aspects of computing, communication, and storage energy consumption, demonstrating its efficiency and robustness in practical applications.

From Figure 5, it can be seen that there is a significant difference in training time among centralized training methods, distributed scheduling methods without NAS, and distributed scheduling methods based on federated learning and NAS at different system scales. As the system size (number of nodes) increases, the training time of various methods shows a significant nonlinear growth, but there are significant differences in the growth rate of

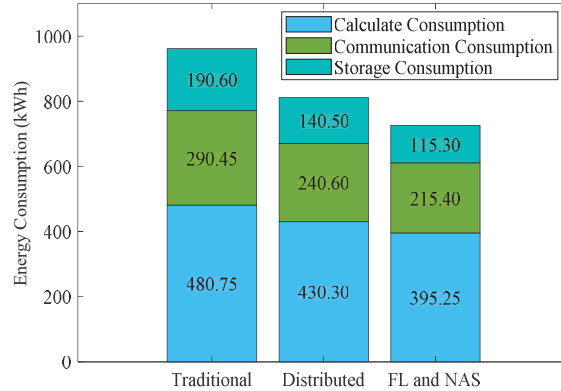


Figure 5 Comparison of system energy consumption.

different methods. The centralized method concentrates all calculations on a single node for processing. As the node size expands, the computational pressure increases sharply, leading to a nonlinear surge in training time. This method performs well in small-scale systems, but shows significant inefficiency in large-scale systems, making it difficult to meet the needs of large-scale distributed systems. The distributed scheduling method without NAS shows better scalability compared to centralized training methods as the system size increases. Distributed scheduling can alleviate the computational bottleneck caused by centralized methods to some extent, but due to the lack of NAS optimization, the training time is still relatively long. Especially when the system scale is large, as the number of nodes increases, the training time increases significantly, indicating the shortcomings of this method in large-scale applications. Finally, the distributed scheduling method based on federated learning and NAS demonstrated significant advantages. Compared to the previous two methods, this method has shorter training time at various system scales, especially with a significantly smaller increase in training time compared to the other two methods under large-scale node counts. This is mainly due to the combination of federated learning and NAS, which optimizes the network structure and distributed computing mode of federated learning through NAS, effectively reducing computational load and improving resource utilization. This method not only improves the scalability of the system, but also significantly shortens the training time, especially suitable for large-scale distributed systems.

In summary, Figure 5 shows that centralized training methods have certain application value in small-scale systems, but exhibit significant

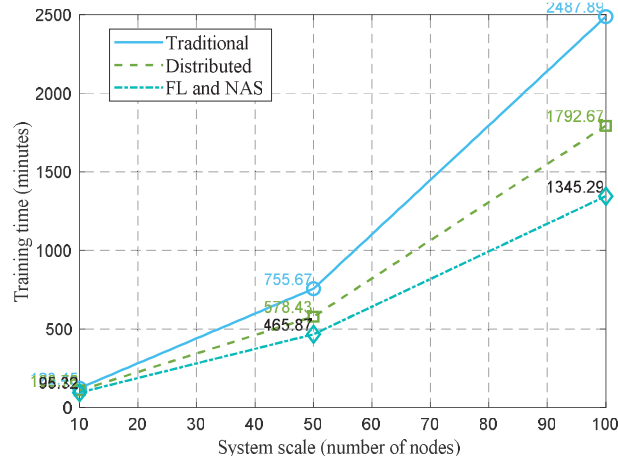


Figure 6 Relationship between training time and system scale.

bottlenecks in large-scale systems; The distributed scheduling method without NAS has improved compared to centralized methods, but there are still efficiency issues in large-scale node numbers. The distributed scheduling method based on federated learning and NAS has shown significant advantages in large-scale systems by optimizing model structure and distributed collaborative learning, and has high practical value and broad application prospects. This result provides important reference for model training optimization of large-scale distributed systems, and also verifies the effectiveness of the combination of federated learning and NAS technology in improving system performance.

From Figure 6, it can be seen that with the increase of load intensity, there are significant differences in the performance of traditional scheduling methods, distributed scheduling methods without combining NAS and federated learning, and distributed scheduling methods based on federated learning and NAS in terms of node failure rate and scheduling fluctuation range. According to Figure 7(a), from the trend of node failure rate, traditional scheduling methods show a significant nonlinear upward trend in failure rate as the load intensity gradually increases. This indicates that the stability of the system significantly decreases and the risk of node failure increases significantly when facing complex high load environments, making it difficult to cope with high load pressures. The distributed scheduling method that does not combine NAS and federated learning performs better than traditional methods in terms of node failure rate. Although its failure

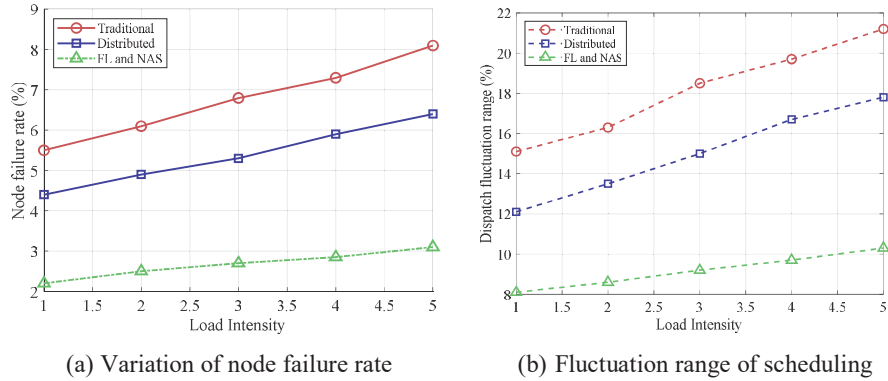


Figure 7 Comparison of the system stability.

rate is lower than traditional methods, it still significantly increases with the increase of load intensity, indicating its vulnerability at higher loads. In contrast, distributed scheduling methods based on federated learning and NAS have shown significant advantages. During the process of increasing the load intensity from level 1 to level 5, the failure rate of this method remained at a relatively low level, with a maximum failure rate of only 3.1%. This indicates that by introducing federated learning and NAS technology, the system can maintain high stability and fault tolerance in the face of complex load changes, significantly reducing node failure rates and demonstrating superior robustness. Based on Figure 7(b), in the trend of fluctuation range in scheduling, the fluctuation range of traditional scheduling methods increases significantly with the increase of load intensity. This indicates that traditional methods significantly reduce the stability and increase volatility of scheduling systems when faced with high load intensity, making it difficult to provide stable scheduling performance. Although the scheduling fluctuation range of the distributed scheduling method without combining NAS and federated learning is slightly lower than that of traditional methods, the fluctuation range still increases from 12.1% to 17.8% as the load intensity gradually increases. This indicates that the method still exhibits significant volatility under high load intensity, which affects the scheduling efficiency and stability of the system. Compared with it, the distributed scheduling method based on federated learning and NAS shows significant stability as the load intensity gradually increases, with the scheduling fluctuation range only increasing from 8.1% to 10.3%, and the fluctuation amplitude is the smallest. This

result indicates that through the distributed collaborative learning of federated learning and the structural optimization of NAS, the system can maintain relatively stable scheduling performance under different load intensities, with a small fluctuation range, and better scheduling efficiency and reliability than other methods.

6 Conclusion

This paper proposes a distributed renewable energy scheduling and prediction optimization method and system based on federated learning and neural architecture search. Through intelligent and adaptive technical means, it improves the scheduling efficiency and prediction accuracy of distributed renewable energy systems, while ensuring data privacy and security. The main conclusions of this paper are as follows:

- (1) By introducing federated learning technology into a distributed renewable energy scheduling system, it has been achieved that each energy node can independently train local neural network models without sharing raw data. Ensured the collaborative optimization capability of the system and effectively addressed the issue of data privacy leakage in centralized scheduling systems.
- (2) By combining NAS technology, the optimal neural network architecture can be automatically designed to dynamically adapt to the needs and environmental changes of different distributed energy nodes, greatly improving the system's adaptability and robustness.
- (3) The combination of federated learning and NAS achieves multi node collaborative optimization, enabling the system to effectively utilize computing resources in distributed systems, improve overall scheduling efficiency, and exhibit stronger adaptability and system robustness in scenarios with large load fluctuations, significantly better than traditional scheduling methods.

Acknowledgements

This research was funded by Xiamen University of Technology scientific research project Research on key technology of "source-network-load" electric-carbon coupling optimized operation in active distribution networks, grant number YKJ22020R

References

- [1] P. Shamsi, H. Xie, A. Longe, and J.-Y. Joo, “Economic dispatch for an agent-based community microgrid,” *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2317–2324, Sep. 2016.
- [2] Yadav, S., Kumar, P., and Kumar, A. (2024). Optimal Design of PV/WT/Battery Based Microgrid for Rural Areas in Leh Using Dragonfly Algorithm. *Distributed Generation & Alternative Energy Journal*, 39(02), 221–262.
- [3] N. Liu, M. Cheng, X. Yu, J. Zhong, and J. Lei, “Energy-sharing provider for PV prosumer clusters: A hybrid approach using stochastic programming and Stackelberg game,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6740–6750, Aug. 2018.
- [4] F. An et al., “Selective Virtual Synthetic Vector Embedding for Full-Range Current Harmonic Suppression of the DC Collector,” *IEEE Trans. Power Electro.*, vol. 38, no. 2, pp. 2577–2588, Feb. 2023.
- [5] F. An, B. Zhao, B. Cui and R. Bai, “Multi-Functional DC Collector for Future All-DC Offshore Wind Power System: Concept, Scheme, and Implement,” *IEEE Trans. Ind. Electron.*, vol. 69, no. 8, pp. 8134–8145, Aug. 2022.
- [6] Jianjie, C., Bo, Z., Fang, Z., Juan, H., and Li, Z. (2024). Identification of High and Low Voltage Ride-Through Control Parameters for Electromechanical Transient Modeling of Photovoltaic Inverter. *Distributed Generation & Alternative Energy Journal*, 39(02), 195–220.
- [7] C. Yuan, M. S. Illindala, and A. S. Khalsa, “Modified Viterbi Algorithm based distribution system restoration strategy for grid resiliency,” *IEEE Trans. Power Del.*, vol. 32, no. 1, pp. 310–319, Feb. 2017.
- [8] F. An et al., “Asymmetric Topology Design and Quasi-Zero-Loss Switching Composite Modulation for IGCT-Based High-Capacity DC Transformer,” in *IEEE Trans. Power Electro.*, vol. 38, no. 4, pp. 4745–4759, April 2023
- [9] B. Zeng, J. Zhang, X. Yang, J. Wang, J. Dong, and Y. Zhang, “Integrate planning for transition to low-carbon distribution system with renewable energy generation and demand response,” *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1153–1165, May 2014.
- [10] H. Kanchev, F. Colas, V. Lazarov, and B. Francois, “Emission reduction and economical optimization of an urban microgrid operation including dispatched PV-based active generators,” *IEEE Trans. Sustain. Energy*, vol. 5, no. 4, pp. 1397–1405, Oct. 2014.

- [11] D. Wang et al., “A demand response and battery storage coordination algorithm for providing microgrid Tie-Line smoothing services,” *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, pp. 476–486, Apr. 2014.
- [12] Ma, G., Hu, S., Wang, Y., Pang, N., and Yu, J. (2025). Energy Storage Configuration Evaluation Method for Renewable Energy Consumption Based on Power Grid Development Planning and Resource Output Forecast Analysis. *Distributed Generation & Alternative Energy Journal*, 39(06), 1125–1152.
- [13] C. A. Hill, M. C. Such, D. Chen, J. Gonzalez, and W. M. Grady, “Battery energy storage for enabling integration of distributed solar power generation,” *IEEE Trans. Smart Grid*, vol. 3, no. 2, pp. 850–857, Jun. 2012.
- [14] Huang, Z., Zhang, L., Huang, W., Li, S., and Chen, Y. (2025). Research on Overvoltage Monitoring Technology for Distributed New Energy Intelligent Stations. *Distributed Generation & Alternative Energy Journal*, 39(06), 1209–1228.
- [15] F. An et al., “DC Cascaded Energy Storage System Based on DC Collector with Gradient Descent Method,” *IEEE Trans. Ind. Electron.*, vol. 71, no. 2, pp. 1594–1605, Feb. 2024.
- [16] L. Wu, M. Shahidehpour, and T. Li, “Stochastic security-constrained unit commitment,” *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 800–811, May 2007.
- [17] Chen, G., Hui, W., Huan, Y., Bingchen, L., and Xingxing, Z. (2025). Research on Optimization of Distribution Network Connection Mode Based on Graph Neural Network and Genetic Algorithm. *Distributed Generation & Alternative Energy Journal*, 39(06), 1179–1208.
- [18] C. Zhang, Y. Xu, Z. Li, and Z. Y. Dong, “Robustly coordinated operation of a multi-energy microgrid with flexible electric and thermal loads,” *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2765–2775, May 2019.

Biographies



Jun Su obtained a Bachelor's degree in Electrical Engineering at Staffordshire University in 2012, followed by a Master's degree in Electrical Energy Systems from Cardiff University in 2014. From October 2017 to December 2020, pursued PhD in Electrical Engineering at Auckland University of Technology, New Zealand. Since July 2021, began teaching at the School of Electrical Engineering and Automation at Xiamen University of Technology. Primary research interests include electric vehicles charging strategy, renewable energy generation systems, intelligent distribution networks.



Chaolong Tang graduated from Xiamen University of Technology in 2022. He is currently studying for a master's degree at Xiamen University of Technology. The main research directions include photovoltaic power prediction.



Zhiquan Liu graduated from Xiamen University of Technology in 2022. He is currently studying for a master's degree at Xiamen University of Technology. The main research directions include photovoltaic power prediction.

