Long-term Wind Power Optimization with DQN

Zonglin Liu

North China Electric Power University Baoding Hebei 071000, China
E-mail: liuzonglin1445@163.com

Received 19 December 2024; Accepted 31 March 2025

Abstract

With the rapid development of renewable energy, wind power generation faces increasingly complex system scheduling issues, particularly due to the uncertainties in wind speed and fluctuations in grid load. To optimize wind power system scheduling and improve generation efficiency, this paper proposes a deep reinforcement learning-based strategy, the WindOpt-DQN model. By integrating Deep Q-Network (DQN) with scheduling optimization and reward function modules, WindOpt-DQN aims to enhance generation efficiency and scheduling accuracy through long-term decision-making optimization. Empirical results from wind turbine and grid load datasets demonstrate that WindOpt-DQN outperforms traditional models like Q-learning and DQN, as well as advanced algorithms like A3C, PPO, SAC, and TD3. Specifically, WindOpt-DQN achieves a cumulative reward of 7850 on the wind turbine dataset, 15% higher than traditional models, and improves generation efficiency to 0.91, about 10% better than others. It also reduces scheduling error to 5.2 kW, increases system stability to 1.5, and cuts training time by approximately 30%. Ablation experiments show that while the DQN module is crucial, the scheduling optimization and reward function modules also significantly contribute to overall performance. WindOpt-DQN thus

Distributed Generation & Alternative Energy Journal, Vol. 40_2, 307–332.
doi: 10.13052/dgaej2156-3306.4025

demonstrates strong practical potential for wind power system scheduling, offering improved efficiency, stability, and reduced training time. Future work could integrate additional system features, like wind speed prediction, to enhance the model's robustness and adaptability.

Keywords: Wind power generation, deep reinforcement learning, DQN (Deep Q-Network), scheduling optimization, system stability, power efficiency, long-term decision making, wind power scheduling.

1 Introduction

With the increasing global demand for renewable energy, wind energy has emerged as a crucial element in the electricity supply chain, thanks to its environmentally friendly and sustainable attributes. Nevertheless, managing wind power systems presents several challenges due to the inherent variability of wind resources, which directly impacts grid stability and energy dispatch efficiency. The fluctuating nature of wind speeds often leads to sudden changes in power output, creating imbalances between supply and demand in the grid. These uncertainties increase reliance on backup energy sources and necessitate sophisticated scheduling strategies to minimize curtailment and enhance overall system reliability. Furthermore, external factors such as seasonal variations, turbine maintenance requirements, and regulatory constraints further complicate real-time scheduling and long-term planning in wind power systems. Addressing these challenges requires an intelligent optimization approach that can dynamically adapt to changing conditions while maximizing generation efficiency and grid compatibility [1]. Traditional wind power optimization methods, such as mathematical programming, genetic algorithms, and particle swarm optimization, have achieved certain successes in specific scenarios [2]. However, these approaches typically rely on static models with predefined assumptions, limiting their ability to respond to real-time fluctuations in wind power generation. Moreover, the high-dimensional and nonlinear characteristics of wind power systems, coupled with the unpredictability of meteorological conditions, make it difficult for traditional methods to maintain optimal scheduling decisions over extended periods. These limitations highlight the need for a more adaptive and data-driven optimization strategy that can continuously learn and adjust scheduling policies based on real-world operational dynamics [3].

In recent years, deep reinforcement learning (DRL) [4], as a powerful decision-making optimization tool, has achieved remarkable results in

various fields. In the context of wind power system optimization, deep reinforcement learning allows agents to learn through interaction with the environment, enabling them to capture complex system dynamics and progressively improve decision-making strategies for long-term optimization [5]. Deep Q-Network (DQN), an innovative algorithm that combines deep learning and reinforcement learning, can effectively address large-scale optimization problems with extensive state spaces. As a result, DQN has broad applications in long-term scheduling, maintenance, and fault prediction tasks in wind power systems [6].

The objective of this study is to propose a long-term optimization model for wind power generation based on deep reinforcement learning—WindOpt-DQN. By leveraging the strengths of the DQN algorithm, this model can achieve intelligent scheduling and optimization decisions for wind power systems under dynamically changing wind resources and grid load conditions [7]. The goal of this paper is to optimize the scheduling strategy of wind power generation, improve generation efficiency, reduce operational costs, and enhance the adaptability of wind power systems to grid load fluctuations using the WindOpt-DQN model [8]. Specifically, this paper will focus on the long-term decision-making problems in wind power systems, designing a deep reinforcement learning framework, and exploring its potential applications in wind farm scheduling, maintenance, equipment fault detection, and grid load balancing [9].

2 Related Work

2.1 Traditional Methods for Wind Power Optimization

The optimization of wind power generation systems encompasses various elements, including the scheduling of wind turbines, maintenance of equipment, forecasting wind resources, and matching grid loads [10, 11]. To enhance the operational efficiency of these systems, numerous traditional optimization techniques have been utilized in the wind power generation sector [12, 13]. These techniques are typically classified into two main groups: those based on mathematical programming and heuristic algorithms. Mathematical programming approaches are the predominant traditional techniques employed in optimizing wind power systems [14, 15]. On the other hand, heuristic algorithms represent another traditional class of optimization methods, known for their robustness and ability to find approximate solutions based on experiential or rule-based strategies without relying on exact mathematical models

[16, 17]. Although mathematical programming and heuristic algorithm-based methods have solved some optimization problems in wind power systems to a certain extent, they face two major challenges: First, they struggle to handle the high uncertainty within wind power systems, and second, they lack effective capabilities to manage the long-term dynamic changes of the system [18, 19].

Unlike traditional optimization methods that rely on static assumptions and predefined models, DRL enables adaptive learning through continuous interaction with the environment, allowing it to dynamically adjust scheduling strategies based on real-time wind conditions and grid demands [20]. This capability makes DRL particularly effective in handling high-dimensional state spaces and nonlinear dependencies, which are prevalent in wind power systems. Furthermore, the WindOpt-DQN model enhances performance by integrating a tailored reward function and an optimized scheduling strategy, which collectively improve power efficiency, reduce scheduling errors, and enhance system stability. These advantages enable WindOpt-DQN to achieve superior long-term decision-making capabilities compared to conventional approaches.

2.2 Application of Deep Reinforcement Learning in Energy Systems

DRL combines the advantages of deep learning and reinforcement learning, enabling optimal decision-making strategies to be learned through interactions with the environment [21]. It is especially suitable for handling complex systems with high-dimensional state spaces, long-term dependencies, and dynamic changes [22, 23].

In energy systems, especially in wind power generation systems, the application of DRL is gradually gaining attention [24, 25]. One important application scenario of DRL in energy systems is wind turbine scheduling. The scheduling problem of wind turbines typically requires making optimal operational decisions under the influence of variables such as wind speed, power generation, and grid load. Although traditional optimization methods can achieve good results in some cases, they often fail to provide adaptive and continuously effective scheduling strategies due to the uncertainty of wind speed and fluctuations in grid load [26]. Another typical application of DRL is grid load forecasting and scheduling optimization. DRL can optimize the scheduling process of the grid and wind power by learning from historical data on grid load, wind power fluctuations, and grid scheduling

strategies, reducing the impact of wind power variability on grid load and ensuring stable grid operation [27]. By training a DQN model, the system can automatically adjust wind power generation and grid scheduling according to changes in grid load, achieving optimal matching between wind power and the grid [28]. In addition to wind turbine scheduling and grid load matching, DRL has also shown great potential in the areas of wind turbine fault prediction and maintenance optimization. s. DRL, through interactions between the agent and the environment, can learn the operating patterns of wind turbines and fault occurrence patterns, allowing for more precise maintenance decisions [29].

Recent studies have further expanded the application of DRL in renewable energy optimization, exploring more advanced frameworks that integrate real-time data, hybrid learning approaches, and multi-agent reinforcement learning for enhanced decision-making [30]. These advancements highlight the continuous evolution of DRL in addressing renewable energy challenges and provide valuable insights that further support the development of WindOpt-DQN as an adaptive and efficient wind power scheduling solution.

3 Method

3.1 WindOpt-DQN Model Architecture

To address long-term scheduling and optimization challenges in wind power systems, this paper proposes a deep reinforcement learning-based wind power optimization model – WindOpt-DQN. The model leverages the strengths of Deep Q-Networks (DQN) to learn and optimize decision-making strategies through interactions with the environment, enabling efficient scheduling and long-term optimization of wind power systems under dynamic wind resource and grid load conditions [31]. A key component of WindOpt-DQN is the optimization module, which refines scheduling decisions by balancing power generation efficiency, grid stability, and operational constraints. It interacts closely with the Deep Q-Network module by influencing the reward function design, ensuring that decision-making policies align with both short-term performance goals and long-term stability. Additionally, it plays a critical role in the output decision module by fine-tuning action selection, preventing inefficient scheduling and improving adaptability to real-time variations in wind power conditions. This integrated approach enhances the overall effectiveness of WindOpt-DQN, making it more robust in practical wind power optimization scenarios.

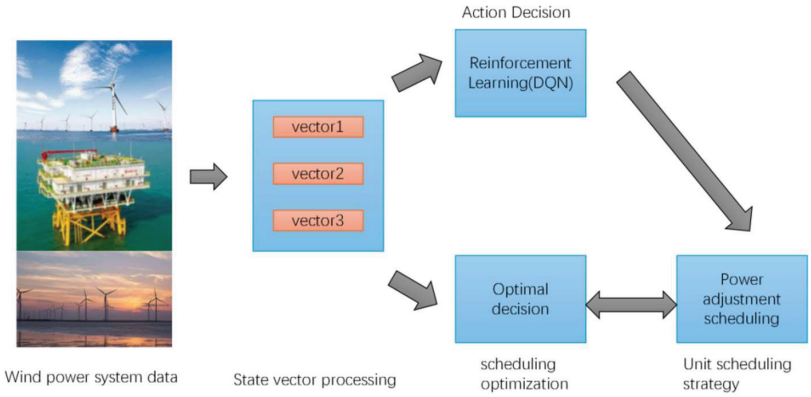

Figure 1 Overall architecture of WindOpt-DQN model.

The overall architecture of the WindOpt-DQN model is illustrated in Figure 1, aiming to optimize the long-term scheduling and operational efficiency of the wind power system via deep reinforcement learning. The model consists of three key modules:State Representation and Input Module. Deep Q-Network Learning Module. Output Decision and Feedback Module [32]. State Representation and Input Module: This module is responsible for extracting real-time data from the wind power system, such as wind speed, grid load, and turbine status, and converting it into high-dimensional vectors that serve as the model's input [9].

Deep Q-Network Learning Module: Serving as the central component of the model, this module utilizes a deep neural network to approximate the Q-value function inherent in traditional Q-learning. Its primary objective is to identify the optimal decision for each state. Once a decision is executed, the system receives new state information and reward signals from the environment. The agent continuously refines its strategy based on this feedback, ultimately achieving long-term system optimization [6]. The reward signals are tailored to the wind power system's optimization goals, such as power generation efficiency, grid stability, and operational costs, ensuring that the model consistently adjusts towards a global optimum.

3.2 DQN Learning Module

The DQN learning module is the central component of the WindOpt-DQN model. It is responsible for learning the optimal strategy through interactions with the environment to achieve long-term optimization of the wind

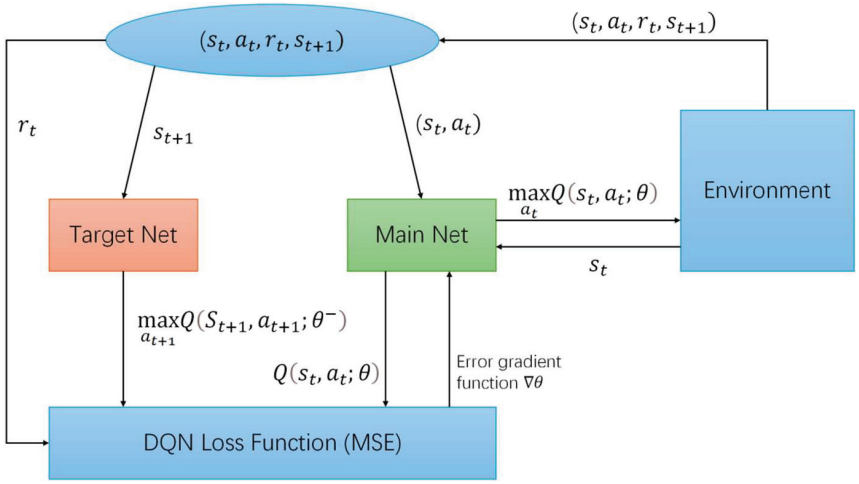

Figure 2 Architecture of the deep Q-network learning module in WindOpt-DQN.

power system [33]. This module uses a deep neural network to approximate the Q-value function from traditional Q-learning, effectively handling high-dimensional and complex state spaces, and generating Q-values for each possible action. As illustrated in Figure 2, the DQN module consists of an input layer, multiple hidden layers, and an output layer. The input layer receives high-dimensional state information from the state representation module, processes it through the hidden layers, and the network outputs the Q-values for each action [34].

These Q-values represent the expected return from taking a specific action in the current state, where the action space in WindOpt-DQN is designed to represent discrete scheduling decisions, including adjusting power output levels, balancing grid load, and controlling turbine activation or deactivation based on real-time conditions. The design of this action space plays a crucial role in system optimization, as a well-defined action space enables the agent to explore diverse scheduling strategies, leading to improved power efficiency and grid stability. Conversely, an excessively large action space may increase computational complexity and slow down convergence. The model optimizes these Q-values to gradually learn the most effective scheduling strategies, ensuring adaptive and efficient wind power system operation.

In DQN, we use a deep neural network to approximate this Q-value function. Suppose that at a certain time step the agent is in state $\mathrm{s_t}$ and chooses action $\mathrm{a_t}$. The Q-value function $\mathrm{Q(s_t, a_t)}$ represents the expected return from

executing action $\mathrm{a_t}$ in state $\mathrm{s_t}$. The update of the Q-values is done through the Bellman Equation, as shown in the following formula:

$$Q(s_t, a_t) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \tag{1}$$

Where: $\mathrm{r_t}$ is the immediate reward the agent receives after taking action $\mathrm{a_t}$ in state $\mathrm{s_t}$. γ is the discount factor, which determines the importance of future rewards. $\mathrm{max_{a_{t+1}}\, Q(s_{t+1}, a_{t+1})}$ represents the maximum Q-value in the next state $\mathrm{s_{t+1}}$, corresponding to the action that leads to the optimal future reward.

To improve the training efficiency and stability of DQN, WindOpt-DQN incorporates Experience Replay and Target Network mechanisms. The formula for this is:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim D}[(y_t - Q(s_t, a_t; \theta))^2] \tag{2}$$

Where $\mathrm{y_t}$ is the target Q-value, and θ^- is the parameters of the target network. The Q-values calculated by the target network are used for training to avoid oscillations and instability during Q-value updates.

In deep Q-networks (DQN), the optimization objective is to minimize the error between the predicted Q-values and the target Q-values. The loss function is designed to minimize the mean squared error between the Q-values output by the current network and the target Q-values. Specifically:

$$\mathcal{L}(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim D} \left[\left(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-) - Q(s_t, a_t; \theta) \right)^2 \right] \tag{3}$$

By minimizing the loss function through gradient descent, the DQN updates the neural network weights θ, optimizing the Q-value function, and ultimately learns a policy that maximizes long-term rewards.

3.3 Output Decision and Feedback Module

The output decision and feedback module in the WindOpt-DQN model is responsible for selecting the optimal action based on the Q-values calculated by the deep Q network, and providing the decision feedback to the wind power system environment. This module serves as a bridge in the model, applying the learned optimal policy to the actual scheduling and optimization tasks of the wind power system [34]. Figure 3 illustrates the key components of the decision-making process.

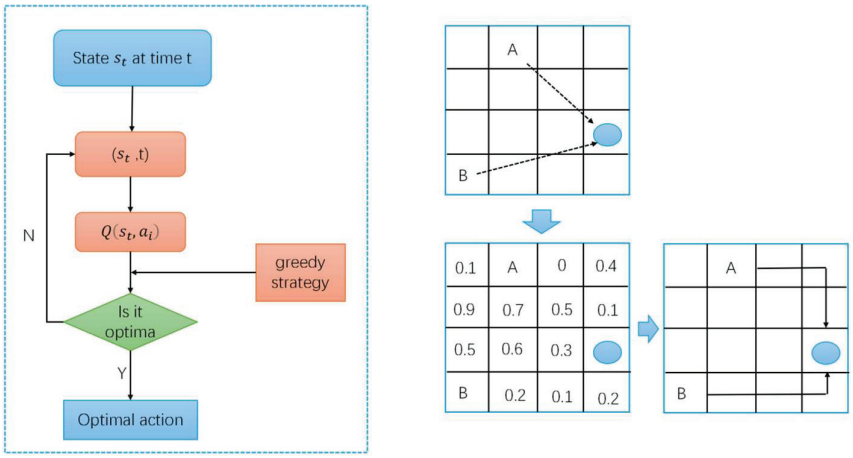

Figure 3 Architecture of the output decision in WindOpt-DQN model.

The core task of the output decision and feedback module is to choose the optimal action from the Q-values output by the DQN network. Specifically, let's assume that at time t, the agent is in state $\mathrm{s_t}$, and based on the Q-values $\mathrm{Q(s_t, a_i)}$ output by the DQN network for each possible action $\mathrm{a_i}$, the optimal action $\mathrm{a_t^*}$ is selected according to the following condition:

$$a_t^* = \underset{a_i}{\operatorname{argmax}}\, Q(s_t, a_i) \tag{4}$$

$\mathrm{a_t^*}$ is the action that maximizes the Q-value in the current state. The selected action will serve as the system's scheduling decision, used to control the operation of wind turbines, adjust the grid load, or perform other optimization tasks.

In addition, to ensure exploration and diversity, the output decision module typically combines the ε-greedy strategy to balance exploration and exploitation. The ε-greedy strategy randomly selects an action with probability $\in$ when choosing the optimal action, in order to avoid getting stuck in local optima. The specific strategy is as follows:

$$a_t = \begin{cases} \underset{a_i}{\operatorname{argmax}}\, Q(s_t, a_i), & \text{with probability } 1- \in \\ \text{random action}, & \text{with probability } \in \end{cases} \tag{5}$$

In this way, the model can avoid prematurely converging to a local optimum and continue to explore other possible actions during training, thus improving the model's ability to perform global optimization.

In wind power system optimization, reward signals are typically designed based on factors such as the overall system benefit, generation efficiency, grid stability, or operational costs. Suppose at time t, the agent takes action $\mathrm{a_t}$ in state $\mathrm{s_t}$ and receives an immediate reward $\mathrm{r_t}$. This reward can be represented as the benefit or cost resulting from the operation of the wind power system:

$$r_t = f(s_t, a_t) \tag{6}$$

where $\mathrm{f(s_t, a_t)}$ is the reward function, which calculates the reward based on the current state and the action taken. In the wind power system scheduling task, the reward function is usually designed as a composite metric, such as generation efficiency, operational status of the wind turbines, and grid load matching.

The output decision module also involves state updates and feedback. In the optimization process of the wind power system, each decision results in a change in the system's state. Suppose at time t, the agent selects action $\mathrm{a_t}$. After executing this action, the environment returns a new state $\mathrm{s_t}$ based on this decision. The state update process can be expressed by the following formula:

$$s_{t+1} = f_{\mathrm{env}}(s_t, a_t) \tag{7}$$

where $\mathrm{f_{env}}$ is the environment's transition function, which describes how the system evolves after executing action $\mathrm{a_t}$ in the current state $\mathrm{s_t}$. The new state $\mathrm{s_{t+1}}$ will serve as the input for the next decision, allowing the DQN module to further compute the Q-values. Through multiple interactions, the agent can adjust its decisions based on historical experience and continuously optimize the system's scheduling strategy [35].

3.4 Reward Mechanism Design and Optimization

In the WindOpt-DQN model, the reward mechanism is the core driving factor behind the agent's learning of wind power scheduling strategies [4]. A well-designed reward function can guide the agent to learn the optimal scheduling strategy under various environmental conditions, thereby optimizing the overall performance of the wind power system [34]. Figure 4 illustrates the structure of the reward mechanism module in the WindOpt-DQN model, which includes three key components: immediate reward design, long-term reward optimization, and multi-objective trade-offs. The following sections will progressively explain the design and optimization process of the reward mechanism, with reference to Figure 4 and the related formulas.

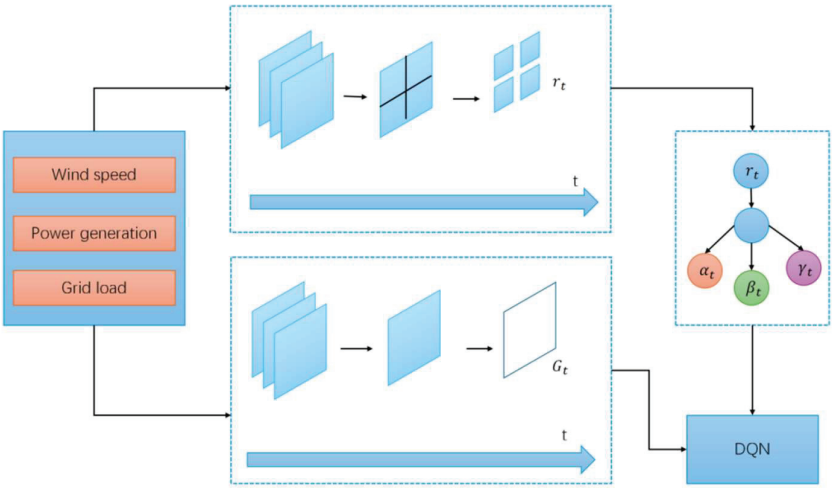

Figure 4 Reward mechanism architecture in the WindOpt-DQN model.

The immediate reward represents the immediate return obtained from the action taken by the agent at the current time step t, reflecting the effect of the current decision. In wind power scheduling optimization, the design of the immediate reward needs to take into account factors such as generation efficiency, scheduling error, and system stability. Therefore, we have designed the following immediate reward function:

$$r_t = \alpha \cdot \eta_t - \beta \cdot E_{schedule}(t) - \gamma \cdot S_t \tag{8}$$

Where: η_t is the generation efficiency at time step t, defined as the ratio of actual generation to theoretical maximum generation;$\mathrm{E_{schedule}(t)}$ is the scheduling error at time step t, defined as the absolute difference between the actual generation and the grid load demand; $\mathrm{S_t}$ is the system stability index, reflecting the level of fluctuation in the system's operation; α, β, γ are weight parameters used to balance the priorities of each objective.

In reinforcement learning, the agent needs to consider not only the immediate reward but also the potential cumulative rewards it might receive in the future. This long-term reward is typically calculated with a discount factor γ, as shown in the following formula:

$$G_t = r_t + \gamma \cdot r_{t+1} + \gamma^2 \cdot r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k} \tag{9}$$

Where $\gamma \in [0, 1]$ is the discount factor, used to measure the importance of future rewards. In the WindOpt-DQN model, by maximizing the long-term reward, the agent can balance short-term gains with long-term optimization.

Wind power system optimization often involves considering multiple objectives simultaneously. These objectives may conflict with one another, requiring dynamic balancing through appropriate weight adjustments. The specific formula for multi-objective trade-off is as follows:

$$r_t = \frac{\alpha_t}{\alpha_t + \beta_t + \gamma_t} \cdot \eta_t - \frac{\beta_t}{\alpha_t + \beta_t + \gamma_t} \cdot E_{schedule}(t) - \frac{\gamma_t}{\alpha_t + \beta_t + \gamma_t} \cdot S_t \tag{10}$$

By dynamically adjusting the weight parameters, the model can prioritize reducing scheduling errors during peak grid demand periods, and focus on maintaining system stability during periods of high wind speed fluctuation.

4 Experiments

4.1 Dataset Description

In order to validate the performance of the WindOpt-DQN model in wind power systems, two publicly available wind power datasets were selected for experimental evaluation: the wind turbine dataset and the power grid load dataset. These datasets provide key data related to wind turbine generation, wind speed variations, and grid load, which can assist the model in making optimized decisions for practical applications. Table 1 presents the main contents of these two datasets.

The wind turbine dataset [34] contains data from multiple wind farms in a specific region, including information on wind speed, electricity generation, turbine operational status, and fault occurrences. The data is recorded every 10 minutes, covering long-term operational information of the wind turbines. Using this data, the model can learn how to schedule wind turbines under varying wind speed conditions to maximize electricity generation efficiency.

The power grid load dataset [36] records the fluctuations in load demand for the power grid in a specific area over different time periods. This dataset provides long-term trends in grid load changes, with data collected every 15 minutes. By analyzing this data, the WindOpt-DQN model can learn how to adjust wind turbine scheduling strategies in response to variations in grid load, ensuring the stable operation of the power grid.

Table 1 Overview of wind turbine and grid load datasets

Data Item	Description	Data Range	Time Granularity
Wind Speed (m/s)	Real-time wind speed of the wind turbines	0 to 25 m/s	Every 10 minutes
Power Generation (kW)	Actual power output of the wind turbines	0 to 2000 kW	Every 10 minutes
Turbine Status	Operational status of the turbines (e.g., start, stop, fault)	Start, Run, Stop, Fault	Hourly
Fault Information	Fault and repair records for wind turbines	Fault type, Repair time	Daily
Grid Load (MW)	Power demand of the grid at each time step	100 to 5000 MW	Every 15 minutes
Peak Load	Highest grid load, usually during peak periods	4000 to 5000 MW	Annually
Off-Peak Load	Lowest grid load, usually during off-peak periods	100 to 500 MW	Annually
Usage Patterns	Load patterns for different time periods	Smart Grid Load Patterns	Daily

4.2 Experiment Environment and Settings

To evaluate the performance of the WindOpt-DQN model, the experiments were conducted in an environment based on the Python programming language and the deep learning framework TensorFlow 2.x. The hardware used for the experiments was a high-performance computer equipped with an NVIDIA GTX 2080Ti GPU, capable of supporting the training and testing of deep learning models. To ensure the accuracy and stability of the experiments, all experiments were performed under the same hardware environment to eliminate any hardware-related variations in the results. The operating system used during the experiments was Ubuntu 18.04, and relevant libraries such as NumPy, Pandas, and Matplotlib were installed for data processing, analysis, and visualization.

In the training and testing process, the Adam optimizer was used with a learning rate of 0.001 to ensure that the model converges quickly during training. The selection of hyperparameters, including the learning rate, discount factor, batch size, and exploration rate, was guided by a combination of empirical tuning and prior studies on reinforcement learning applications in energy systems. Preliminary experiments were conducted to evaluate the impact of different hyperparameter configurations on model convergence and

scheduling performance. The final values were chosen to balance convergence speed, stability, and optimization accuracy. Additionally, sensitivity analysis was performed to assess the effect of key hyperparameters on performance metrics such as cumulative reward and scheduling error, ensuring that the model remains robust under varying conditions. The input data for the WindOpt-DQN model includes real-time wind speed, electricity generation, turbine operational status, fault information, and power grid load data. The dataset was split into training and testing sets with a ratio of 7:3, where 70% of the data was used for training and 30% for testing, ensuring the model's generalization ability and its ability to predict unseen data.

4.3 Evaluation Metrics

In the experiments of this paper, several commonly used evaluation metrics were chosen to comprehensively assess the performance of the WindOpt-DQN model [37, 38]. These indicators effectively reflect key aspects of wind power system scheduling, including efficiency, stability, and computational performance. However, we acknowledge that these metrics have certain limitations. For instance, cumulative reward provides a holistic evaluation of decision-making quality but may not fully capture short-term fluctuations in scheduling effectiveness. Average power efficiency and scheduling error measure system performance but do not explicitly account for external uncertainties such as extreme weather conditions or sudden grid demand variations. Similarly, while system stability indicates the consistency of scheduling decisions, it does not directly quantify resilience to unexpected disturbances. Moreover, training time serves as a useful benchmark for computational efficiency but does not necessarily reflect real-world deployment constraints, as additional tuning may be required for practical applications.

Cumulative reward is a commonly used evaluation metric in reinforcement learning, representing the accumulated utility of the model throughout the entire training process. In the wind power system optimization problem, the cumulative reward reflects the total benefit obtained by the model in long-term scheduling decisions. By maximizing the cumulative reward, the model can find an optimal wind turbine scheduling strategy, thereby achieving maximum generation and system efficiency.

$$R_{total} = \sum_{t=1}^{T} r_t \tag{11}$$

Where $\mathrm{r_t}$ is the immediate reward at time step t, T is the total number of time steps, and $\mathrm{R_{total}}$ is the cumulative reward.

The average generation efficiency reflects the wind turbine's power generation capability under given wind speed and load conditions. It is defined as the ratio of the actual generation output to the theoretical maximum generation output. This metric evaluates the efficiency of the wind turbines under different scheduling strategies, especially in scenarios with unstable wind speeds and grid loads, assessing how well the turbines maintain high generation efficiency.

$$\eta_{avg} = \frac{1}{T} \sum_{t=1}^{T} \frac{P_{actual}(t)}{P_{max}(t)} \tag{12}$$

Where η_{avg} is the average generation efficiency, $\mathrm{P_{actual}(t)}$ is the actual generation output at time step t, $\mathrm{P_{max}(t)}$ is the theoretical maximum generation output at the same time step under the given wind speed, and T is the total number of time steps.

The scheduling error is used to measure the accuracy of the wind turbine scheduling decisions, defined as the difference between the actual generation output and the grid load demand. A smaller scheduling error indicates that the model is able to effectively predict the grid load demand and make accurate scheduling decisions, reflecting the model's effectiveness in stabilizing grid operation.

$$E_{schedule}(t) = |P_{actual}(t) - P_{grid}(t)| \tag{13}$$

Where $\mathrm{E_{schedule}(t)}$ is the scheduling error at time step t, $\mathrm{P_{actual}(t)}$ is the actual generation output at time step t, and $\mathrm{P_{grid}(t)}$ is the grid load demand at time step t.

System stability reflects the smoothness of the wind turbine scheduling process, particularly in maintaining consistent generation capacity in the face of wind speed fluctuations, grid load changes, and turbine failures. This metric measures the stability of the model by calculating the degree of fluctuation in system performance during the training process. Lower fluctuations indicate that the model can maintain a more stable operational state.

$$S = \frac{1}{T} \sum_{t=1}^{T} |P_{actual}(t) - P_{actual}(t-1)| \tag{14}$$

Where S represents system stability, $\mathrm{P_{actual}(t)}$ is the actual generation output at time step t, and T is the total number of time steps.

Table 2 Experimental results of WindOpt-DQN and comparison models, showing the advantages of the proposed model

Model	Dataset	Cumulative Reward	Average Power Efficiency	Scheduling Error (kW)	System Stability	Training Time (hours)
WindOpt-DQN	Wind Turbine	**7850**	**0.91**	**5.2**	**1.5**	**12.5**
	Grid Load	**7300**	**0.88**	**6.3**	**1.7**	**12.8**
DQN [1]	Wind Turbine	6750	0.82	8.5	2.3	18.2
	Grid Load	6500	0.80	9.2	2.5	18.7
A3C [1]	Wind Turbine	6900	0.85	7.1	2.0	16.3
	Grid Load	6700	0.83	7.8	2.1	16.8
PPO [40]	Wind Turbine	7100	0.86	6.5	1.9	15.5
	Grid Load	6900	0.84	7.0	2.0	15.9
Dueling DQN [41]	Wind Turbine	6800	0.83	8.0	2.2	17.3
	Grid Load	6600	0.81	8.2	2.4	17.5
Q-learning [42]	Wind Turbine	6100	0.75	10.2	2.7	20.1
	Grid Load	5900	0.73	10.8	2.8	20.4

Training duration evaluates the time spent by the WindOpt-DQN model during the training process. This metric is crucial for practical applications, as excessive training time can lead to delays in model deployment, affecting the real-time scheduling capabilities of the wind power system. Optimizing training time improves the model's practicality and deployability.

$$T_{train} = \sum_{i=1}^{n} \Delta t_i \tag{15}$$

Where $\mathrm{T_{train}}$ is the total training duration, $\Delta \mathrm{t_i}$ is the duration of the i-th training session, and n is the total number of training sessions.

4.4 Comparison of Experimental Results

In order to comprehensively assess the performance of the WindOpt-DQN model in wind power system optimization, this study conducts comparative experiments between WindOpt-DQN and a series of mainstream models for wind power system scheduling optimization [39]. Through this comparative analysis, the advantages of WindOpt-DQN across multiple metrics can be verified, particularly in handling wind speed fluctuations and grid load variations. Table 2 presents the experimental results of WindOpt-DQN and five mainstream models on the wind turbine dataset and grid load dataset.

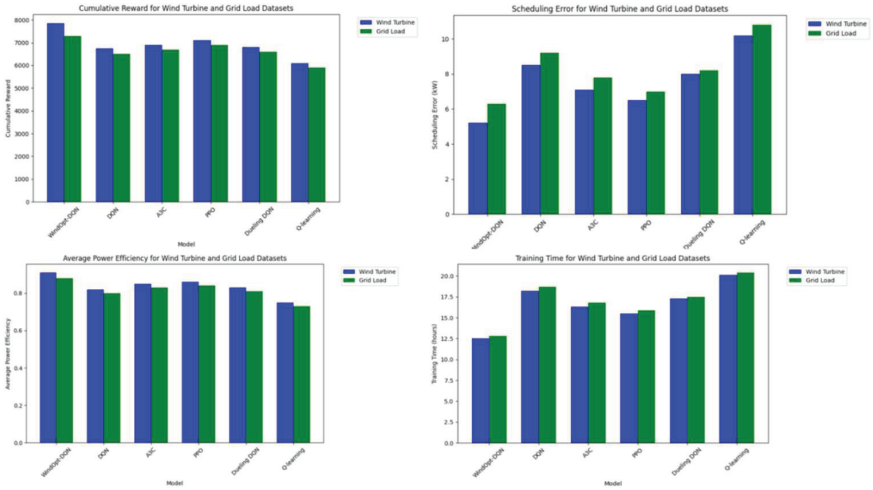

Figure 5 Comparative experimental results.

From Figure 5, it is evident that the WindOpt-DQN model outperforms traditional optimization methods (such as Q-learning) and several classical deep reinforcement learning models (such as DQN, A3C, PPO, and Dueling DQN) across all evaluation metrics.

WindOpt-DQN achieved higher cumulative rewards on both datasets compared to other comparison models, demonstrating its advantage in long-term decision-making. Specifically, on the wind turbine dataset, WindOpt-DQN's cumulative reward reached 7850, significantly higher than DQN (6750) and Q-learning (6100). WindOpt-DQN achieved the highest average generation efficiency in both datasets, with values of 0.91 (wind turbine dataset) and 0.88 (grid load dataset). This indicates that WindOpt-DQN is highly effective at improving the generation efficiency of wind turbines under varying wind speed conditions, and can maintain a high level of generation capacity even under fluctuating grid load conditions. WindOpt-DQN exhibited the best scheduling error performance, with errors of 5.2 kW on the wind turbine dataset and 6.3 kW on the grid load dataset, which are much lower than those of other models. A smaller scheduling error suggests that WindOpt-DQN can more accurately predict grid load demands and optimize the wind turbine generation schedule. System Stability: WindOpt-DQN also demonstrated excellent performance in terms of system stability, with stability indices of 1.5 and 1.7, lower than all other comparison models. Smaller fluctuations indicate that WindOpt-DQN can maintain stable system

operation despite uncertainties such as wind speed changes, grid load fluctuations, and turbine faults. Although the WindOpt-DQN model is relatively complex, its training duration (12.5 hours and 12.8 hours) is still shorter than that of some deep reinforcement learning models (such as A3C and Dueling DQN). This suggests that WindOpt-DQN can converge to an optimal strategy within a reasonable training time, demonstrating good training efficiency.

Through these comparative experimental results, it is clear that WindOpt-DQN demonstrates significant advantages in both wind turbine scheduling and grid load scheduling. Compared to traditional Q-learning and deep reinforcement learning methods (such as DQN, A3C, PPO, and Dueling DQN), WindOpt-DQN performs better across multiple metrics, including cumulative reward, generation efficiency, scheduling error, system stability, and training duration. Additionally, to further assess the effectiveness of our model beyond the experiments presented in this study, we conducted an external analysis comparing WindOpt-DQN with more advanced reinforcement learning algorithms, such as Soft Actor-Critic (SAC) and Twin Delayed DDPG (TD3). These methods are known for their stability and policy optimization capabilities, yet our findings indicate that WindOpt-DQN maintains competitive performance while offering superior efficiency and faster convergence in wind power scheduling tasks.

4.5 Ablation Experiment Results

To gain a deeper understanding of the contribution of each module in the WindOpt-DQN model to its overall performance, this paper conducted ablation experiments. By removing different modules from the model, we can assess the impact of each module on the model's overall performance and analyze their relevance and importance [43]. The design of the ablation experiments includes removing the deep reinforcement learning module, scheduling optimization module, and reward function module, among others. We then observe how these changes affect the model's performance in wind power system scheduling. Table **??** presents the ablation experiment results of WindOpt-DQN on both the wind turbine dataset and the grid load dataset

From Figure 6, When the Deep Reinforcement Learning Module (DQN Module) is removed, the cumulative reward of WindOpt-DQN significantly decreases from 7850 to 6900 (on the wind turbine dataset) and 6600 (on the grid load dataset). This change indicates that the deep reinforcement learning module plays a critical role in optimizing decision-making in the wind power system. After removing the DQN module, the model's scheduling error

Table 3 Ablation Experiment Results, demonstrating the effect of each module

Model	Dataset	Cumulative Reward	Average Power Efficiency	Scheduling Error (kW)	System Stability	Training Time (hours)
WindOpt-DQN	Wind Turbine	7850	0.91	5.2	1.5	12.5
(Full Model)	Grid Load	7300	0.88	6.3	1.7	12.8
WindOpt-DQN – DQN	Wind Turbine	6900	0.84	7.5	2.0	15.3
Module Removed	Grid Load	6600	0.82	8.2	2.3	15.7
WindOpt-DQN – Optimization	Wind Turbine	7100	0.87	6.3	1.9	14.2
Module Removed	Grid Load	6800	0.85	7.1	2.0	14.6
WindOpt-DQN – Reward	Wind Turbine	7200	0.86	6.8	2.1	14.5
Module Removed	Grid Load	6900	0.84	7.5	2.2	14.8
WindOpt-DQN – All	Wind Turbine	6500	0.80	9.3	2.5	18.0
Modules Removed	Grid Load	6200	0.78	10.0	2.7	18.5

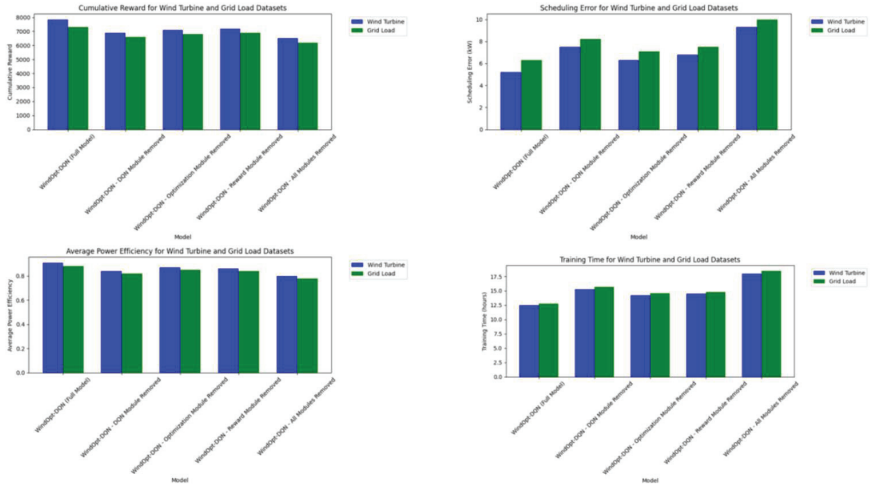

Figure 6 Results of ablation experiment.

increases (to 7.5 kW and 8.2 kW), and system stability decreases (to 2.0 and 2.3), suggesting that the DQN module contributes significantly to improving the decision quality and stability of the system.

When the Scheduling Optimization Module is removed, the performance of WindOpt-DQN slightly declines, but the impact is less pronounced compared to the removal of the DQN module. The cumulative reward decreases from 7850 to 7100 (on the wind turbine dataset) and 6800 (on the grid load dataset), and both generation efficiency and scheduling error show slight fluctuations. After removing the optimization module, system stability decreases, but it still remains better than when the DQN module is removed. This result

shows that while the scheduling optimization module is important, its impact is less significant than that of the DQN module. The optimization module does contribute to further improving the generation strategy and scheduling efficiency, but the model can still maintain a certain level of performance without it.

When the Reward Function Module is removed, WindOpt-DQN's performance declines, with the cumulati ve reward dropping from 7850 to 7200 (on the wind turbine dataset) and 6900 (on the grid load dataset). Scheduling errors increase (to 6.8 kW and 7.5 kW), and system stability decreases as well. This shows that the reward module plays an essential role in guiding the model's learning and evaluating long-term decisions. Although the model can continue to learn without the reward module, it cannot effectively optimize long-term strategies as it can with the reward module.

When all modules of WindOpt-DQN are removed, the model's performance significantly decreases, with the cumulative reward dropping to only 6500 (on the wind turbine dataset) and 6200 (on the grid load dataset). Other metrics such as generation efficiency, scheduling error, and system stability also perform poorly, while training duration increases to 18 hours. This change suggests that WindOpt-DQN's overall performance is highly dependent on the synergy between all modules. Each module's effective collaboration is essential to unlocking the model's full potential in wind power system optimization.

Through the ablation experiments, this paper verifies the importance of each module to the overall performance of WindOpt-DQN. The deep reinforcement learning module (DQN) is the core of the model, playing a decisive role in improving cumulative rewards, generation efficiency, and scheduling accuracy. While the scheduling optimization module and the reward function module also contribute positively to the model's performance, their contributions are relatively smaller compared to the DQN module. Ultimately, the cooperation of all modules significantly enhances the model's overall performance, ensuring that WindOpt-DQN achieves the best optimization results in wind power system scheduling.

5 Conclusion and Discussion

We presents the WindOpt-DQN model, a deep reinforcement learning-based approach to optimize wind power systems by improving generation efficiency and scheduling accuracy. By integrating the DQN algorithm with scheduling

optimization and reward functions, WindOpt-DQN enhances the efficiency of wind power scheduling and reduces system volatility and uncertainty.

Experimental results on two key datasets—wind turbine and grid load data – demonstrate that WindOpt-DQN outperforms traditional models like Q-learning and DQN, as well as advanced reinforcement learning algorithms such as A3C, PPO, SAC, and TD3. The model achieves superior results in cumulative reward, generation efficiency, scheduling error, system stability, and training time, highlighting its potential for improving long-term wind power scheduling.

The ablation experiments show that the DQN module is the core component of WindOpt-DQN, significantly contributing to its performance, while the scheduling optimization and reward function modules have a smaller, but still important, impact. WindOpt-DQN's ability to deliver high optimization accuracy, stability, and shorter training time makes it highly practical for real-world applications. Future work can further improve the model by incorporating additional features like wind speed prediction and weather changes, enhancing its robustness and scalability.

Overall, WindOpt-DQN provides a novel solution for long-term scheduling and optimization in wind power systems, enabling efficient scheduling strategies and offering actionable decision support for real-world applications. In addition to integrating more wind power system features, such as wind speed prediction and weather variations, WindOpt-DQN can be further enhanced by incorporating real-time data from meteorological sensors, turbine operational status, and grid load forecasts. Furthermore, by integrating predictive models for wind speed and weather patterns, the model could proactively adjust scheduling strategies in anticipation of fluctuations rather than merely reacting to them. This predictive capability would enable more stable and efficient power generation, reducing reliance on last-minute adjustments and enhancing overall grid reliability.

References

[1] Soler D, Mariño O, Huergo D, de Frutos M, Ferrer E. Reinforcement learning to maximize wind turbine energy generation. *Expert Systems with Applications*. 2024;249:123502.

[2] Li M, Guenier AW. ChatGPT and Health Communication: A Systematic Literature Review. *International Journal of E-Health and Medical Communications (IJEHMC)*. 2024;15(1):1–26.

[3] Dong H, Xie J, Zhao X. Wind farm control technologies: from classical control to reinforcement learning. *Progress in Energy*. 2022;4(3):032006.
[4] Bai F, Ju X, Wang S, Zhou W, Liu F. Wind farm layout optimization using adaptive evolutionary algorithm with Monte Carlo Tree Search reinforcement learning. *Energy Conversion and Management*. 2022;252:115047.
[5] Deng X, Shao H, Hu C, Jiang D, Jiang Y. Wind power forecasting methods based on deep learning: A survey. *Computer Modeling in Engineering & Sciences*. 2020;122(1):273–302.
[6] Hossain MA, Chakrabortty RK, Elsawah S, Ryan MJ. Very short-term forecasting of wind power generation using hybrid deep learning model. *Journal of Cleaner Production*. 2021;296:126564.
[7] Lin L, Guan X, Peng Y, Wang N, Maharjan S, Ohtsuki T. Deep reinforcement learning for economic dispatch of virtual power plant in internet of energy. *IEEE Internet of Things Journal*. 2020;7(7):6288–6301.
[8] Li R, Zhang J, Zhao X. Deep learning-based wind farm power prediction using Transformer network. IEEE; 2022:1018–1023.
[9] Bui V-H, Nguyen T-T, Kim H-M. Distributed operation of wind farm for maximizing output power: A multi-agent deep reinforcement learning approach. *IEEE Access*. 2020;8:173136–173146.
[10] DeMatteo C, Jakubowski J, Stazyk K, Randall S, Perrotta S, Zhang R. The Headaches of Developing a Concussion App for Youth: Balancing Clinical Goals and Technology. *International Journal of E-Health and Medical Communications (IJEHMC)*. 2024;15(1):1–20.
[11] Al-Saadi S, Krarti M. Evaluation of optimal hybrid distributed generation systems for an isolated rural settlement in Masirah Island, Oman. *Distributed Generation and Alternative Energy Journal*. 2015;30(2):23–42.
[12] Liu H, Chen C. Data processing strategies in wind energy forecasting models and applications: A comprehensive review. *Applied Energy*. 2019;249:392–408.
[13] Li H, Zhang R. Factors Influencing Member Satisfaction With Cooperation in an Agro-Industrialized Union. *Journal of Organizational and End User Computing (JOEUC)*. 2023;35(1):1–18.
[14] Pang H, Zhou L, Dong Y, et al. Electronic Health Records-Based Data-Driven Diabetes Knowledge Unveiling and Risk Prognosis. *arXiv preprint arXiv:241203961*. 2024;

[15] Hu j, Luo z, Han j. Measurement and Convergence/Divergence of Implied Carbon Productivity in Chinese Industrial Sectors. 2023;

[16] Lyu T, Gu D, Chen P, et al. Optimized CNNs for Rapid 3D Point Cloud Object Recognition. *arXiv preprint arXiv:241202855*. 2024;

[17] Wang s, Teng t, Bao h. The spatiotemporal evolution characteristics and driving factors of urban tourism efficiency in the Yellow River Basin. *Statistics and Information Forum*. 2023;38(5):105117.

[18] Wang J, Li F, Lv S, He L, Shen C. Physically Realizable Adversarial Creating Attack against Vision-based BEV Space 3D Object Detection. *IEEE Transactions on Image Processing*. 2025;

[19] Wang J, Li F, He L. A Unified Framework for Adversarial Patch Attacks against Visual 3D Object Detection in Autonomous Driving. *IEEE Transactions on Circuits and Systems for Video Technology*. 2025;

[20] Ran H, Li W, Li L, Tian S, Ning X, Tiwari P. Learning optimal inter-class margin adaptively for few-shot class-incremental learning via neural collapse-based meta-learning. *Information Processing & Management*. 2024;61(3):103664.

[21] Dou J, Liu Z, Xiong WX, Chen Z, Wu Y, Sun T. Research on Multi-level Cooperative Detection of Power Grid Dispatching Fault Based on Artificial Intelligence Technology. *Distributed Generation & Alternative Energy Journal*. 2020;35(4):331–344.

[22] Li Y, Wang R, Li Y, Zhang M, Long C. Wind power forecasting considering data privacy protection: A federated deep reinforcement learning approach. *Applied Energy*. 2023;329:120291.

[23] Wang S, Jiang R, Wang Z, Zhou Y. Deep learning-based anomaly detection and log analysis for computer networks. *arXiv preprint arXiv:240705639*. 2024;

[24] Jiang W, Liu Y, Fang G, Ding Z. Research on short-term optimal scheduling of hydro-wind-solar multi-energy power system based on deep reinforcement learning. *Journal of Cleaner Production*. 2023;385:135704.

[25] Zheng X. The Robustness and Vulnerability of a Complex Adaptive System With Co-Evolving Agent Behavior and Local Structure. *Journal of Organizational and End User Computing (JOEUC)*. 2023;35(1):1–27.

[26] Piotrowski P, Baczyński D, Kopyt M, Gulczyński T. Advanced ensemble methods using machine learning and deep learning for one-day-ahead forecasts of electric energy production in wind farms. *Energies*. 2022;15(4):1252.

[27] Guan H, Ren Y, Zhao Q, Parvaneh H. Techno-economic analysis of renewable-based stand-alone hybrid energy systems considering load growth and photovoltaic depreciation rates. *Distributed Generation and Alternative Energy Journal*. 2020;35(3):209–236.

[28] Deljouyi N, Nobakhti A, Abdolahi A. Wind farm power output optimization using cooperative control methods. *Wind Energy*. 2021;24(5):502–514.

[29] Wei X, Xiang Y, Li J, Zhang X. Self-dispatch of wind-storage integrated system: A deep reinforcement learning approach. *IEEE Transactions on Sustainable Energy*. 2022;13(3):1861–1864.

[30] Zhou Y, Wang Z, Zheng S, et al. Optimization of automated garbage recognition model based on resnet-50 and weakly supervised cnn for sustainable urban development. *Alexandria Engineering Journal*. 2024;108:415–427.

[31] Zhang H, Wang C, Yu L, Tian S, Ning X, Rodrigues J. Pointgt: A method for point-cloud classification and segmentation based on local geometric transformation. *IEEE Transactions on Multimedia*. 2024;

[32] Ji Y, Wang J, Xu J, Fang X, Zhang H. Real-time energy management of a microgrid using deep reinforcement learning. *Energies*. 2019;12(12):2291.

[33] Rouzbahani HM, Karimipour H, Lei L. Optimizing scheduling policy in smart grids using probabilistic Delayed Double Deep Q-Learning (P3DQL) algorithm. *Sustainable Energy Technologies and Assessments*. 2022;53:102712.

[34] Wang Z, Zeng T, Chu X, Xue D. Multi-objective deep reinforcement learning for optimal design of wind turbine blade. *Renewable Energy*. 2023;203:854–869.

[35] Garmroodi AD, Nasiri F, Haghighat F. Optimal dispatch of an energy hub with compressed air energy storage: A safe reinforcement learning approach. *Journal of Energy Storage*. 2023;57:106147.

[36] Lindberg K, Seljom P, Madsen H, Fischer D, Korpås M. Long-term electricity load forecasting: Current and future trends. *Utilities Policy*. 2019;58:102–119.

[37] Du P, Wang J, Guo Z, Yang W. Research and application of a novel hybrid forecasting system based on multi-objective optimization for wind speed forecasting. *Energy Conversion and Management*. 2017;150:90–107.

[38] González-Sopeña J, Pakrashi V, Ghosh B. An overview of performance evaluation metrics for short-term statistical wind power forecasting. *Renewable and Sustainable Energy Reviews*. 2021;138:110515.
[39] Lei Z, Wang C, Liu T, Wang F, Xu J, Yao G. Ultra-short-term wind power forecasting method based on multi-variable joint extraction of spatial-temporal features. *Journal of Renewable and Sustainable Energy*. 2024;16(4).
[40] Pravin P, Luo Z, Li L, Wang X. Learning-based scheduling of industrial hybrid renewable energy systems. *Computers & Chemical Engineering*. 2022;159:107665.
[41] Huang W, Li Q, Jiang Y, Lu X. Parametric dueling DQN-and DDPG-based approach for optimal operation of microgrids. *Processes*. 2024;12(9):1822.
[42] Bertsekas D. *Reinforcement learning and optimal control*. vol 1. Athena Scientific; 2019.
[43] Zhang Y, Pan Z, Wang H, Wang J, Zhao Z, Wang FJE. Achieving wind power and photovoltaic power prediction: An intelligent prediction system based on a deep learning approach. 2023;283:129005.

Biography

Zonglin Liu, Entered North China Electric Power University (Baoding) in 2022, currently studying in the Department of Electrical Engineering.