# Application Research of Deep Reinforcement Learning in Low-carbon Scheduling Mechanism for Power Grid Construction Tasks

**Kyle Kai Qin, Yi Zheng, Chaoyan Zeng, Shengzhu Wang**
Guangxi Power Grid Co., Ltd, Nanning 530000, Guangxi, China
Corresponding author: Kyle Kai Qin (e-mail: kaiqing13520@126.com)

**ABSTRACT** This paper proposes a low-carbon scheduling and carbon emission prediction framework for power grid construction machinery by integrating an improved genetic algorithm (IGA), deep reinforcement learning (DRL), and a gated recurrent unit (GRU) neural network. The proposed approach is applicable to engineering scenarios involving power transmission facilities, electromagnetic infrastructure, and communication-support systems where energy efficiency and carbon reduction are critical. A multi-objective scheduling model is established based on task dependency relationships, equipment operating characteristics, and carbon emission factors, while the GRU network is employed for real-time carbon emission forecasting. Experimental results demonstrate that the proposed method outperforms traditional GA and PSO approaches, reducing average scheduling time by 35.37% and total carbon emissions by 24.69%. The GRU model achieves an RMSE of 2.89 kgCO/h, showing superior prediction accuracy compared with RF and SVR models. The results verify the effectiveness, stability, and practical value of the proposed framework for intelligent low-carbon scheduling in complex engineering environments.

**INDEX TERMS** low-carbon scheduling, deep reinforcement learning, GRU, electromagnetic infrastructure

## I. INTRODUCTION

UNDER the background of the continuous promotion of the "dual carbon" strategy and the accelerated construction of a green energy system, the production process efficiency and carbon emission level of textile manufacturing, as an important part of global supply chains and a significant energy consumer, are becoming the key indexes for evaluating the effectiveness of the implementation of green projects (such as resourceefficient production and clean dyeing technologies). The traditional textile production scheduling method generally relies on empirical decision-making and static planning for resource-intensive stages like dyeing and finishing. This approach is difficult to meet the current demand for energy saving and emission reduction, precise manufacturing timelines, and optimal allocation of resources (such as water, chemical agents, and machine time). Especially in the reality of large-scale production of new sustainable textile products and the complex and changing environment of the manufacturing supply chain (including fluctuating raw material availability and regulatory requirements), production scheduling decision-making needs to be data-driven, require real-time response to process variations (like dyeing batch results), and operate under carbon constraints for every unit of output.

In recent years, the cross-application of AI, IoT and low-carbon modeling technologies in energy system and engineering dispatch has been deepening. Kurniawan et al [1] proposed a data-driven model based on predicting carbon emission hotspots, which effectively improved the emission visualization level. Alshammari [2] constructed a multilevel safety framework to guarantee the operational stability of renewable energy systems; Elgarahy et al [3] optimized the bioenergy process through AI, verifying the feasibility of AI in energy efficiency improvement. Jiang et al [4] constructed a low-carbon scheduling model in multimodal transportation scenarios, and proposed a hybrid Gray Wolf-Falcon algorithm to improve the global convergence performance, which provided inspiration for scheduling optimization in construction scenarios. Hoummadi et al [5] and Smahi et al [6], on the other hand, introduced AI techniques in micro-grid scheduling and grid inertia assessment, respectively, to expand the boundaries of energy system optimization. Jamil et al [7] further explored the coupling relationship of AI between electric vehicle charging strategy and grid response, to reveal the mechanism of the impact of intelligent regulation on the power load structure. Current research focuses on intelligent scheduling algorithm ontology or energy system modeling, and there is a lack of systematic research on scheduling problems in the context of multi-factor coupling of construction equipment, task constraints, real-time carbon emissions and working condition fluctuations, etc. In this paper, a low-carbon scheduling mechanism research is carried out around the grid construction tasks. In view of this, this paper focuses on the grid construction task to carry out low-carbon scheduling mechanism research, build the integration of carbon emission factor modeling, multi-objective optimization and depth prediction of the scheduling system, in order to improve the construction efficiency while significantly reducing the intensity of carbon emissions, to provide theoretical support for the construction of green projects and intelligent construction and technical path [8].

## II. GRID CONSTRUCTION MACHINERY SCHEDULING MODEL CONSTRUCTION

### A. CHARACTERIZATION OF POWER GRID CONSTRUCTION TASKS

Power grid construction tasks usually involve multi-stage processes such as foundation construction (e.g. tower foundation pouring), equipment lifting (e.g. transformer installation), line erection and commissioning, etc., which are diversified, cyclical and resource-intensive. Take 220 kV transmission line construction as an example, the average length of a single construction section is 5-10 km, involving more than 10 sets of machinery and equipment, including crawler cranes, concrete pump trucks, tension release trucks, etc., and its scheduling time window is generally concentrated in the morning at 6:00 a.m. - 18:00 p.m. [9]. The construction tasks show significant temporal dependencies, which can be modeled by a directed acyclic graph (DAG), defining the task set as $T = \{t_1, t_2, \ldots, t_n\}$ and the inter-task dependencies as $E = \{(t_i, t_j) \mid t_i \to t_j\}$. The intensity of demand for mechanical resources varies by task phase, with 65% of the demand for fuel-type heavy machinery in the foundation phase and a major reliance on motorized equipment in the commissioning phase. Table 1 shows the distribution of resource intensity in typical construction phases. The above characteristics pose a challenge to low-carbon scheduling, and it is necessary to match the characteristics of the task process with the structure of machinery use to realize the optimal allocation of carbon emission efficiency.

**TABLE 1.** Resource Intensity Distribution Across Construction Phases

| Construction Phase | Heavy Machinery | Medium Equipment | Electric Tools |
|---|---|---|---|
| Foundation Work (%) | 65 | 25 | 10 |
| Equipment Installation (%) | 40 | 40 | 20 |
| Commissioning Stage (%) | 15 | 25 | 60 |

### B. MODELING MECHANICAL RESOURCES AND SCHEDULING CONSTRAINTS

The machinery scheduling problem in grid construction can be modeled as a multi-resource constrained scheduling problem (RCPSP), which contains multiple constraints such as equipment type, job timing, time

window and carbon emission [10]. Let the set of construction tasks be $T = \{t_1, t_2, \ldots, t_n\}$, the set of mechanical equipment be $M = \{m_1, m_2, \ldots, m_k\}$, and each task need to satisfy a specific equipment demand $R_{ij}$, which represents the demand for equipment $m_j$ when task $t_i$ is executed. The start time of the task is $S_i$ and the finish time is $C_i = S_i + d_i$, where $d_i$ is the task duration. Scheduling constraints mainly include:

(1) Resource limitation constraints:

$$\sum\nolimits_{i \in T} x_{ij}(\tau) \le R_j^{\max}(\tau) \tag{1}$$

Where $R_j^{\max}(\tau)$ is the total number of devices of type $j$ available at any one time and $x_{ij}(\tau)$ indicates whether the task $i$ is allocated to one unit of device $j$ at the time $\tau$. Given the nature of construction machinery, this constraint is treated as a non-redistributable resource constraint, where each individual device unit can only be used by one task at any given time.

(2) Sequence of operations constraints:

$$C_i \le S_j, \quad \text{if } t_j \text{ depends on } t_i \tag{2}$$

(3) Time window constraints:

$$E_i \le S_i \le L_i \tag{3}$$

Here, $E_i$ and $L_i$ represent the earliest start time and the latest start time of task $i$, respectively, measured in time units. These parameters are typically set based on project contract standards, construction process experience (e.g., standard working hours), and site-specific time regulations.

To achieve the decarbonization goal, a carbon intensity factor needs to be incorporated to establish a carbon emission constraint related to the length of equipment use and essential idle time:

$$\sum\nolimits_{j \in M} \left( \sum\nolimits_{i \in T} \gamma_j^{active} \cdot \tau_{ij} \cdot x_{ij} + \gamma_j^{idle} \cdot \left(T_j^{total} - \sum\nolimits_{i \in T} \tau_{ij} \cdot x_{ij}\right)\right) \le Carbon_{max} \tag{4}$$

where $\gamma_j^{\text{active}}$ is the carbon emission factor per unit time of equipment $m_j$ under active load, $\gamma_j^{\text{idle}}$ is the factor under idle load, $\tau_{ij}$ is the task duration, $x_{ij}$ is the assignment variable, and $T_j^{\text{total}}$ is the total available time of device $j$ in the schedule.

Units for Key Carbon Variables: $\gamma_j^{\text{active}}$, $\gamma_j^{\text{idle}}$: $kgCO_2$/hour.

$\text{Carbon}_{\max}$: $kgCO_2$. The above model provides a mathematical basis for the optimization of subsequent scheduling algorithms and guarantees a reasonable and efficient allocation of mechanical resources [11].

### C. LOW-CARBON OBJECTIVE FUNCTION DESIGN

The core of low-carbon objective function design is to minimize carbon emissions as the main direction of scheduling optimization, taking into account the construction efficiency and equipment utilization. In the construction process, it is first necessary to establish a carbon emission factor database based on the energy consumption characteristics of different types of construction machinery, and take the carbon emission per unit of operating time of each piece of equipment as the basis for evaluation. Subsequently, the equipment configuration, operating hours and load intensity of each construction task are used as input parameters to calculate the total carbon emissions of the scheduling program within a given time interval [12]. The objective function not only evaluates the total amount of carbon emissions, but also introduces indirect carbon source factors such as equipment idle running time and scheduling waiting time to make the evaluation more comprehensive. In the scheduling process, the intelligent algorithm dynamically adjusts the task sequence and resource matching, prioritizes the equipment combination scheme with low carbon intensity and high energy efficiency, and avoids the high emission behaviors such as repeated starting and stopping and equipment idling [13]. This objective function serves as the constraint basis of the optimization algorithm, which directly drives the overall scheduling process towards the "carbon optimal" direction. The details are shown in Figure 1.

## III. DESIGN OF SCHEDULING OPTIMIZATION METHODS FOR ARTIFICIAL INTELLIGENCE

### A. ALGORITHM FRAMEWORK AND PROCESS DESIGN

In order to realize the low-carbon and intelligent scheduling of grid construction machinery, an intelligent optimization algorithm framework based on multi-source data driving is constructed, and the overall process includes four modules, namely, data input layer, scheduling modeling layer, optimization solving layer and output feedback layer (see Figure 2).

The data input layer integrates basic data such as construction task parameters, equipment carbon emission factors, and on-site working condition information; the scheduling modeling layer transforms the construction task into a multi-constraint scheduling graph model, and integrates the time window, equipment capacity, and carbon emission limitations to construct the scheduling structure; the optimization solution layer adopts the method based on the combination of improved genetic algorithms and deep reinforcement learning, and minimizes the total amount of carbon emissions under the premise of ensuring construction efficiency; the output feedback layer dynamically updates scheduling results and performs energy consumption analysis, thereby providing a decision-making basis for carbon emission control [14]. The whole framework has good module scalability and parameter adaptability, which is suitable for intelligent scheduling scenarios of construction projects of different scales, and has significant green construction and energy saving and emission reduction application value.

### B. INTELLIGENT SCHEDULING ALGORITHM DESIGN

A hybrid intelligent scheduling algorithm combining Improved Genetic Algorithm (IGA) and Deep Reinforcement Learning (DRL) [15] for optimizing the scheduling of grid construction machinery under multiobjective and low-carbon constraints. The initial stage generates a high-quality scheduling scheme population by IGA, and the individual fitness function is defined as:

$$\min Fitness = \omega_1 \cdot Duration^{\text{norm}} + \omega_2 \cdot CO_2^{\text{norm}} \quad (5)$$

Where $Duration^{\text{norm}}$ and $CO_2^{\text{norm}}$ are the normalized values of the total scheduling duration (Total Duration) and total carbon emissions (Total $CO_2$), respectively, and $\omega_1$ and $\omega_2$ are the target weighting factors ($\omega_1 + \omega_2 = 1$). Normalization is essential to eliminate the dimensional difference between time (e.g., hours) and carbon emissions (e.g., kgCO2). A common Min-Max normalization method is used to scale each objective value $f_k$ to the range [0,1]:

$$f_k^{\text{norm}} = \frac{f_k - f_k^{\min}}{f_k^{\max} - f_k^{\min}} \quad (6)$$

The specific values for these weights are typically determined through a sensitivity analysis or set empirically based on the project's explicit priority between scheduling efficiency and low-carbon goals. This fitness function is used to guide the search to converge towards the direction of work duration and carbon emission equilibrium.

The method adopts a hybrid optimization strategy combining IGA and DRL. IGA is utilized to generate highquality initial scheduling solutions, ensuring fast convergence to an optimal region.

Hyperparameters for IGA and DRL:

For the Improved Genetic Algorithm (IGA) component, the following hyperparameter settings were used:

Population Size: 100

Maximum Generations: 150

Selection Method: The population for the next generation is constructed in two steps. First, Elitist Selection is applied, where the top 5% of individuals are directly copied to the new generation, ensuring the best-found solutions are preserved. Second, the remaining 95% of the population is generated from offspring created via Crossover and Mutation, using parents selected through Tournament Selection (tournament size = 5) applied to the current population. Crossover Rate: Pc=0.85 (Uniform Crossover)

Mutation Rate: Pm=0.10 (Swap Mutation)

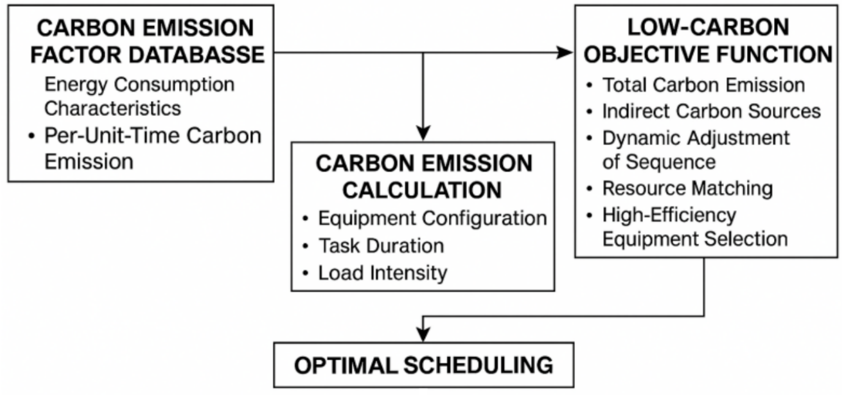


**FIGURE 1.** Low-carbon objective function design diagram

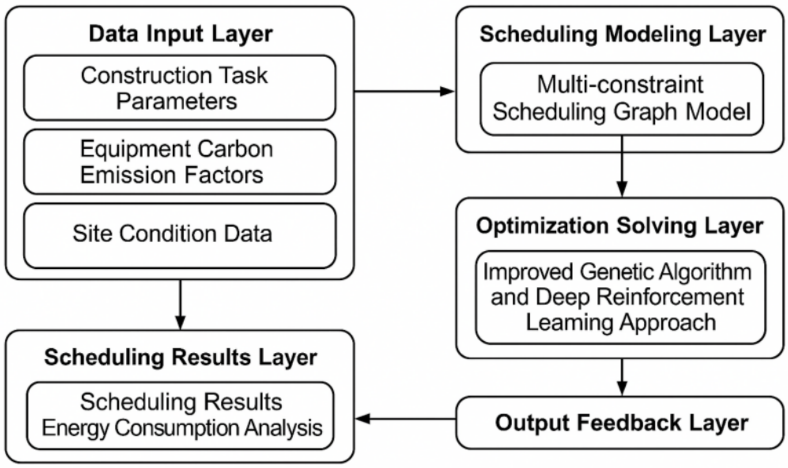


**FIGURE 2.** Algorithm Framework Flowchart

Encoding Scheme: Priority-based List Encoding, where the chromosome represents the task execution order and resource assignment priority.

The Deep Reinforcement Learning (DRL) component utilizes the Proximal Policy Optimization (PPO) algorithm, and the specific hyperparameters for its implementation are:

Learning Rate (Actor & Critic): $1 \times 10^{-4}$ (Adam Optimizer)

Discount Factor ($\gamma$): 0.99

Entropy Coefficient: 0.01 (For enhanced exploration)

PPO Clipping Parameter ($\epsilon$): 0.2

Batch Size (Mini-batch size for training): 64

Trajectory Length (N-step return): 128

PPO Update Steps per Epoch: 10

Network Architecture: A two-layer fully connected neural network for both the Actor and Critic, each with 128 neurons and a ReLU activation function. The final layer of the Actor uses a Softmax function to output a probability distribution over the discrete action space.

The coupling mechanism is realized by mapping the optimal solution derived from IGA to the initialization of the DRL agent's policy.

Specifically, the IGA's best-found scheduling sequence (task order and resource assignment) is used to pretrain the initial policy network. This is achieved through a Behavior Cloning pre-training approach. First, the optimal scheduling sequence generated by IGA serves as an expert demonstration. Second, this demonstration is executed by replaying the sequence in the environment simulator to accurately collect a detailed sequence of state-action pairs (s, a). The state s for the PPO agent encompasses all necessary environment information, including the current time, resource availability, the set of ready tasks, and future constraints. Third, these expert (s, a) pairs are used to pre-train the DRL policy network via supervised learning (Behavior Cloning loss). Finally, the pre-trained policy is fine-tuned using the PPO algorithm to further enhance performance.

A Markov Decision Process (MDP) is formally defined by a tuple that includes the State Space, Action Space, and Reward Function:

State Space (S - States): The set of all possible situations or conditions the agent can be in. A state ($s \in S$) is defined as a comprehensive summary of the current scheduling environment that allows the agent to make an optimal resource allocation decision, ensuring the Markov Property is satisfied.

The complete state vector $s_t$ is defined by the concatenation of (1) the current project time $t$ and task-specific deadlines, (2) the execution status of all tasks (ready, ongoing, completed) and their remaining durations, (3) the current status and remaining capacity of all mechanical resources, and (4) the currently observed external factors.

Action Space (A - Actions): The action $a \in A$ represents the agent's decision at the current time step. To manage the large search space characteristic of the Resource-Constrained Project Scheduling Problem (RCPSP) and prevent the combinatorial explosion of the action dimension that is unsuitable for PPO, the action space is simplified and represented as a fixed-size discrete selection mechanism. The action is defined by a two-step process to ensure a finite and controllable action set:

Task Selection: The agent first selects a task ($t_i$) from the current set of ready tasks ($T_{\text{ready}}$).

Resource Selection: The agent then selects one of the available and compatible equipment units ($m_j$) to execute the chosen task.

This mapping converts the composite space (Task $\times$ Equipment $\times$ Time Slot) into a discrete action vector a of size |T| $\times$ |M|, where |T| is the total number of tasks and |M| is the total number of equipment units. At any given time t, the agent outputs a probability distribution over this fixed action vector.

Crucially, the time component (the earliest feasible start time $S_i$) is NOT part of the agent's output vector but is determined deterministically by the scheduling environment kernel based on the selected $t_i, m_j$ assignment, ensuring that the PPO's policy network operates on a stable, fixed-dimensional discrete action space.

Reward Function (R - Reward): The immediate reward R(s,a) is designed as a composite function to maximize the cumulative expected return by simultaneously minimizing scheduling duration and carbon emissions. It is defined as a negative weighted linear combination of normalized penalty terms:

$$R(s,a)=\begin{cases}\omega_{dur}\cdot \text{TotalDuration}^{norm} & \text{if final task completes(Sparse)}\\ -(\omega_{co2}\cdot P^{norm}_{carbon}+\omega_{idle}\cdot P^{norm}_{idle}) & \text{otherwise(Shaped)}\end{cases} \tag{7}$$

Where:

$TotalDuration^{\text{norm}}$: Normalized penalty for the final total project duration. This penalty is sparse, meaning it is only applied upon the completion of the entire project/episode, thereby avoiding the credit assignment problem in step-$t$ rewards.

$P^{\text{norm}}_{\text{carbon}}$: Normalized penalty for the carbon emissions incurred by the action (based on activity and idle emission factors).

$P^{\text{norm}}_{\text{idle}}$: Normalized penalty for instantaneous resource inefficiency, including equipment switching overhead and any accrued idle time in the current time step $t$. This term is calculated solely based on the current state $s_t$ and action $a_t$, quantifying the non-productive resource expenditure.

$\omega_{\text{dur}}, \omega_{\text{co2}}, \omega_{\text{idle}}$: Calibrated weighting factors ($\sum \omega = 1$) that balance the optimization priorities. The penalties are normalized using a Min-Max method to eliminate dimensional differences (e.g., time vs. kgCO2) prior to weighting. The shaped components ($P^{\text{norm}}_{\text{carbon}}$ and $P^{\text{norm}}_{\text{idle}}$) provide dense, instantaneous feedback to guide the agent's behavior towards low-carbon and efficient resource use during the episode. It is typically denoted as $R(s, a)$ or $R(s, a, s')$. The reward guides the agent's behavior, with the goal being to find a policy that maximizes the expected cumulative future reward.

To achieve high-efficiency scheduling in this complex, discrete, and multi-objective environment, the Proximal Policy Optimization (PPO) algorithm is selected for the DRL component. PPO is an on-policy method renowned for its stability features stable policy updates, and can effectively handle high-dimensional discrete decision-making problems. The PPO's clipped objective function prevents overly aggressive policy updates, ensuring a stable convergence from the high-quality initial solution provided by the IGA coupling. The Actor network of the PPO directly learns the optimal scheduling policy $\pi\theta$(a|s), while the Critic network estimates the value function V$\phi$(s) based on the defined State Space and Reward Function.

The most prominent algorithms in Deep Reinforcement Learning (DRL) are Deep Q-Networks (DQN), Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO). They are distinguished by whether they learn a value function (Q-value) or a policy directly.

### 1) Deep Q-Networks (DQN)

DQN is a Value-Based, Off-Policy method that uses a neural network Q(s,a;$\theta$) to approximate the optimal action-value function Q*(s,a).

The network is trained to minimize the Mean Squared Error (MSE) between the predicted Q-value and the target Q-value, where the target uses the fixed Target Network $\theta$-for stability:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}\left[(Q(s,a;\theta) - y_t)^2\right] \tag{8}$$

where the target yt (or the Bellman target) is defined as:

$$y_t = r + \gamma \max_{a'} Q(s', a'; \theta^-) \tag{9}$$

Here: $r$ is the immediate reward. $\gamma$ is the discount factor. $\max_{a'} Q(s', a'; \theta^-)$ is the maximum expected discounted future return from the next state $s'$, calculated using the stable target network.

D is the Experience Replay buffer.

2) Advantage Actor-Critic (A2C)

A2C is an Actor-Critic, On-Policy method that uses two networks: an Actor ($\pi$(als;$\theta$)) to select the action and a Critic (V(s;$\emptyset$)) to estimate the state value.

A2C's objective function (for the Actor, L$\pi$) and value function loss (for the Critic, LV) are based on the Advantage Function (A):

$$A(s,a) = Q(s,a) - V(s) \approx r + \gamma V(s') - V(s) \quad (10)$$

The objective for the Actor is to maximize the expected advantage:

$$L_{\pi}(\theta) = \mathbb{E}\left[\log \pi(a|s;\theta) \cdot A(s,a)\right] \quad (11)$$

The Critic's loss minimizes the MSE of the value prediction:

$$L_V(\phi) = \mathbb{E}\left[(V(s;\phi) - (r + \gamma V(s')))^2\right] \quad (12)$$

3) Proximal Policy Optimization (PPO)

PPO is an Actor-Critic, On-Policy method designed for stable policy updates. It uses a Clipped Surrogate Objective Function to limit the change between the new policy ($\pi_\theta$) and the old policy ($\pi_{\theta_{\text{old}}}$).

The objective function to be maximized is:

$$L_{\text{CLIP}}(\theta) = \mathbb{E}_t\Big[\min\big(r_t(\theta)\hat{A}_t, \; \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big] \quad (13)$$

Here:

$r_t(\theta)$ is the probability ratio: $\quad r_t(\theta) = \dfrac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$.

$\hat{A}_t$ is the Advantage estimate. $\epsilon$ is a small clipping hyperparameter (e.g., 0.2).

The min operation ensures that the update is clipped if the ratio rt($\theta$) moves too far from 1, preventing destructive, large policy updates. PPO minimizes this policy loss combined with a value function loss and often an entropy bonus for exploration.

### C. MULTI-OBJECTIVE OPTIMIZATION FUNCTION CONSTRUCTION

In order to balance the construction efficiency, energy utilization and carbon emission control, an integrated multi-objective optimization function is constructed, covering three objectives of shortest scheduling duration, minimum total carbon emission and maximum equipment utilization [16]. The multi-objective function is defined as follows:

$$\min \begin{cases} f_1 = C_{\max}, \\ f_2 = \sum_{i=1}^{n} \sum_{j=1}^{k} \gamma_j \cdot \tau_{ij} \cdot x_{ij}, \\ f_3 = 1 - \dfrac{\sum_{j=1}^{k} T_j^{\text{idle}}}{\sum_{j=1}^{k} T_j^{\text{total}}}. \end{cases} \quad (14)$$

Where $f_1$ represents the total duration, $f_2$ is the total carbon emissions, $\gamma_j$ is the carbon emission factor per unit of the equipment $j$, $\tau_{ij}$ is the equipment operating hours, $x_{ij}$ is the task and equipment assignment variable, $T_j^{\text{total}}$ representing the total available time of device $j$, and $f_3$ is the equipment utilization target, reflecting the percentage of idle time.

A weighted summation method was used to convert the multi-objective into a single objective function:

$$\min F(t) = \alpha f_1 + \beta f_2 + \delta f_3 \quad (15)$$

where $\alpha$, $\beta$, and $\delta$ are adaptive weighting factors satisfying $\alpha + \beta + \delta = 1$. The values of these weights ($\alpha$,$\beta$,$\delta$) are fixed hyperparameters determined via a comprehensive sensitivity analysis. This approach ensures a stable and consistent optimization target, preventing the agent from manipulating the fitness landscape during training. Based on the project's explicit priority towards minimizing duration and carbon footprint, the optimal weights were empirically set to $\alpha = 0.45$, $\beta = 0.35$, and $\delta = 0.20$ [17].

## IV. CARBON EMISSION REAL-TIME FORECASTING MODEL DESIGN

### A. CONSTRUCTION EQUIPMENT CARBON EMISSION FACTOR BANK CONSTRUCTION

A database of carbon emission factors for typical grid construction machinery, covering the main types of fuel oil, self-propelled and electric equipment. The emission factors are measured in terms of carbon emissions per unit of operating time (kgCO2/h), which are corrected by combining the national emission accounting guidelines, equipment manufacturers' data and on-site measurements to form a standardized value system. The database is constructed according to the three principles of clear classification, traceability of data, and dynamic updating [18]. According to the operating scenarios and load levels, equipment working conditions are refined, such as empty, partially loaded and fully loaded states, and different emission factors are assigned to enhance the accuracy of carbon emission estimation. Figure 3 shows the distribution of carbon emission factors per unit of different types of construction equipment under typical working conditions, which helps to accurately introduce the impact of equipment energy consumption in the scheduling model and supports the subsequent low-carbon optimization and real-time prediction functions.

### B. REAL-TIME DATA ACQUISITION AND PRE-PROCESSING METHODS

The IoT-based multi-source data acquisition system combines GPS, fuel consumption sensors, ammeters and work timing controllers to continuously collect key variables such as the operating status, location, load and energy consumption of equipment at the construction site. The collected data is uploaded to the edge computing node at a frequency of

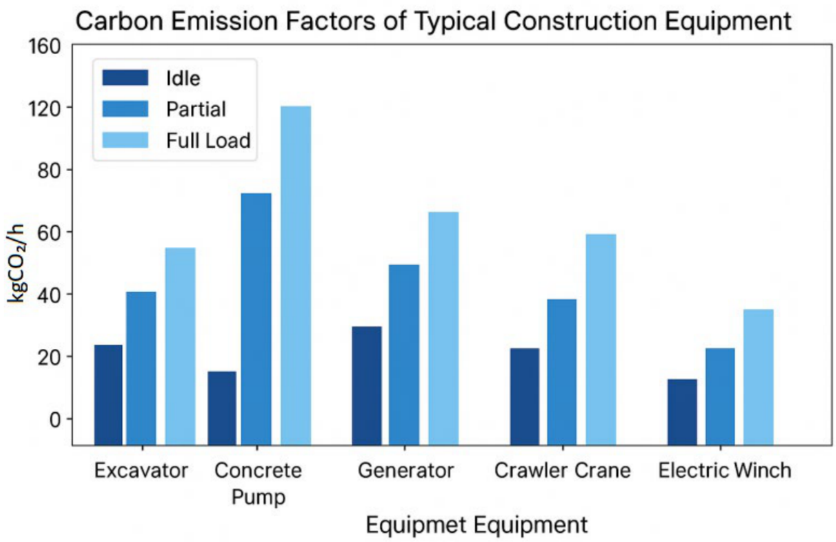


**FIGURE 3.** Carbon Emission Factors of Typical Construction Equipment

seconds, and after local pre-processing, it is transmitted to the cloud-based scheduling platform [19].

The preprocessing process mainly consists of outlier removal, data smoothing, normalization and time synchronization. Z-score normalization is performed on the continuous variable $x_t$:

$$x_t = \frac{x_t - \mu}{\sigma} \tag{16}$$

Where $\mu$ and $\sigma$ are the mean and standard deviation of the variable in the sliding time window, respectively. In order to maintain data consistency, unavailable data were estimated using two-way linear interpolation to ensure that the subsequent model inputs were complete and stable. Through the device ID and task number binding mechanism, real-time correlation between data and task model is realized, providing high-precision and low-latency data support for carbon emission prediction, and significantly enhancing the model response efficiency and deployment practicability.

### *C. CONSTRUCTION AND TRAINING OF DIFFERENT PREDICTION MODELS*

This paper constructs and compares a variety of machine learning and deep learning models, including Random Forest (RF), Support Vector Regression (SVR), Long Short-Term Memory (LSTM) and Gated Recurrent

Unit (GRU) networks. The model input features include: equipment type, workload, running time, energy consumption data, operating environment parameters (e.g., temperature, wind speed) and historical carbon emission records. The GRU network utilizes a multivariate input vector $X_t \in R^D$ with a total input feature dimension of $D = 9$. These 9 features are specifically: the equipment's load rate (refining the workload parameter), running time, energy consumption, temperature, wind speed, the historical carbon emission record ($C_{t-1}$), and the one-hot encoded equipment type (3 features: Heavy, Medium, Electric). The sliding window mechanism is used to construct time series samples with a time-series window length (Lag $L$) of 10 time steps. This means the input to the GRU for prediction at time $t$ is the historical sequence $[X_{t-L+1}, X_{t-L+2}, \ldots, X_t]$, with a resulting dimension of $10 \times 9$. During the model training process, the dataset is divided into training, validation and test sets according to the ratio of 7:2:1, and a validation set is used for early stopping to prevent overfitting. The LSTM and GRU models use a two-layer structure, with the number of neurons in each layer set to 64, the activation function is ReLU, the optimizer is Adam, and the loss function adopts the mean-square error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y_i} - y_i\right)^2 \tag{17}$$

Where $\hat{y}$ is the predicted value of the model and $y_i$ is the real carbon emission value. In order to enhance the generalization ability of the model, the Dropout mechanism is introduced, and the random dropout ratio is set to 0.2. The GRU model shows better prediction accuracy and stability under complex working conditions, and has stronger time-series feature extraction capability, which is suitable for deployment in dynamic carbon emission monitoring scenarios. The prediction module provides forward-looking decision support for the dispatching system and improves the level of carbon control management intelligence.

## V. EXPERIMENTAL DESIGN AND SIMULATION EVALUATION

### *A. EXPERIMENTAL ENVIRONMENT CONFIGURATION*

A hybrid experimental platform was constructed in MATLAB R2023b with Python 3.10 environment, integrating the scheduling optimization module and carbon emission prediction module. The hardware configuration is Intel Core i7-12700F processor, 32GB RAM, RTX 3070 GPU, and runs on Windows 11. The experimental data comes from a 220 kV transmission line construction project, including the operation records and carbon emission monitoring data of 45 pieces of equipment, spanning a period of 30 days and covering different meteorological and load conditions. The equipment types cover crawler cranes, concrete pump trucks, generators, electric winches, etc., and the data sampling frequency is 5 seconds/time. The scheduling algorithm is implemented in MATLAB, and the prediction model is trained using the TensorFlow 2.12 framework, which adopts a unified interface to realize data flow synergy and ensure the comparability of simulation results. All experiments were repeated 5 times under the same initial parameter settings to enhance the stability and credibility of the results.

### *B. PERFORMANCE PARAMETER TESTING*

The evaluation is carried out in two dimensions: scheduling computational efficiency and prediction model accuracy. In the experimental platform, the performance of classical algorithms (GA, PSO), improved algorithms (IGA) and the hybrid IGA+DRL algorithm proposed in this paper are tested respectively. The accuracy of the four mainstream carbon emission prediction models is compared under the same dataset conditions to comprehensively evaluate the adaptability, stability and practical deployment value of the algorithms. For the IGA+DRL method, the "Avg Scheduling Time (s)" refers to the Online Inference Time (i.e., the time required for the trained agent to generate a scheduling solution for a new task instance) and does not include the offline training time required for model convergence.

As seen in Table 2, the IGA+DRL algorithm proposed in this paper outperforms the traditional methods in terms of scheduling efficiency and optimization effect. Its average scheduling time is 33.8 seconds, which is significantly lower than the 52.3 seconds of GA and 47.6 seconds of PSO; the optimal fitness value reaches 0.665, which is improved by 12.6% compared with GA and 10.7% compared with PSO. In terms of convergence speed, IGA+DRL requires only 75 iterations to reach the optimal solution, which is 37.5% shorter compared to GA (120 iterations). This indicates that the algorithm possesses higher efficiency and accuracy in dealing with complex scheduling problems under multiple constraints, which saves computational resources and response time for real construction projects and enhances the engineering feasibility of the algorithm. The details are shown in Figure 4.

**TABLE 2.** Comparison of scheduling performance tests

| Algorithm | Avg Scheduling Time (s) | Best Fitness Value | Iterations to Convergence |
|---|---|---|---|
| GA | 52.3 | 0.761 | 120 |
| PSO | 47.6 | 0.745 | 110 |
| IGA | 39.4 | 0.692 | 90 |
| IGA+DRL | 33.8 | 0.665 | 75 |

The accuracy of carbon emission prediction models has a decisive impact on the optimization decision of the scheduling system. In order to comprehensively evaluate the performance of various types of models, this study introduces three metrics, RMSE, MAE and $R^2$, to assess the performance of Random Forest (RF), Support Vector Regression (SVR), LSTM and GRU in the prediction scenarios.

The data in Table 3 shows that the GRU model outperforms the other models in all metrics, with an RMSE of 2.89 $kgCO_2/h$, an MAE of 1.85 $kgCO_2/h$, and an $R^2$ value of 0.93, which provides optimal fitting accuracy with the lowest level of error. lSTM performs the next best, with an RMSE of 3.14, and an $R^2$ of 0.91. traditional machine learning models such as RF and SVR have an Traditional machine learning models such as RF and SVR have RMSEs of 4.21 and 4.56, respectively, and $R^2$ scores of 0.86 and 0.82, which are significantly lower than deep neural network models. The results show that GRU has stronger time-series feature capture and nonlinear modeling capabilities, which is suitable for the carbon emission prediction task under dynamic construction environment, and can effectively support the real-time decision-making and low-carbon optimization of the scheduling system. The details are shown in Figure 5.

**TABLE 3.** Comparison of model prediction performance parameters

| Model | RMSE ($kgCO_2/h$) | MAE ($kgCO_2/h$) | $R^2$ Score |
|---|---|---|---|
| RF | 4.21 | 2.97 | 0.86 |
| SVR | 4.56 | 3.12 | 0.82 |
| LSTM | 3.14 | 2.10 | 0.91 |
| GRU | 2.89 | 1.85 | 0.93 |

### *C. COMPARISON OF ENERGY SAVING EFFECT*

Four optimization strategies, GA, PSO, IGA and IGA+DRL, are selected for comparison experiments using the baseline scheduling scheme as a control. The energy saving effect of scheduling optimization is quantified by counting the total carbon emission and fuel consumption under each scheme. The energy saving rate index is introduced to evaluate the potential contribution of each algorithm in green construction from the dual dimensions of carbon emission control and energy utilization.

As shown in Table 4, the average carbon emission of the baseline scenario is 1284 $kgCO_2$ and the fuel consumption is 395 L. After the GA optimization, the carbon emission is reduced to 1156 kg, with an energy saving rate of 9.97%, and the PSO is further reduced to 1110 kg, with an energy saving rate of 13.55%. After optimization with IGA and IGA+DRL, the carbon emissions are 1023 kg and 967 kg, respectively, and the energy saving rate is significantly increased to 20.33% and 24.69%. In terms of fuel, the IGA+DRL solution consumes only 326 L of fuel, which is 69 L less than the baseline solution. This result shows that the scheduling method based on intelligent optimization

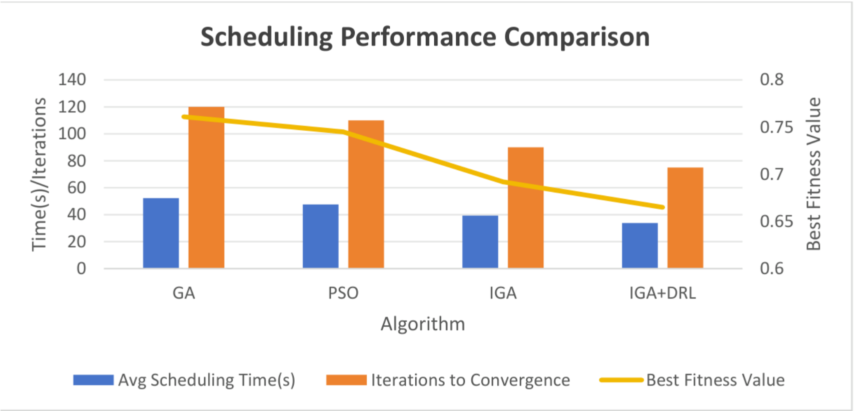


**FIGURE 4.** Scheduling Performance Comparison

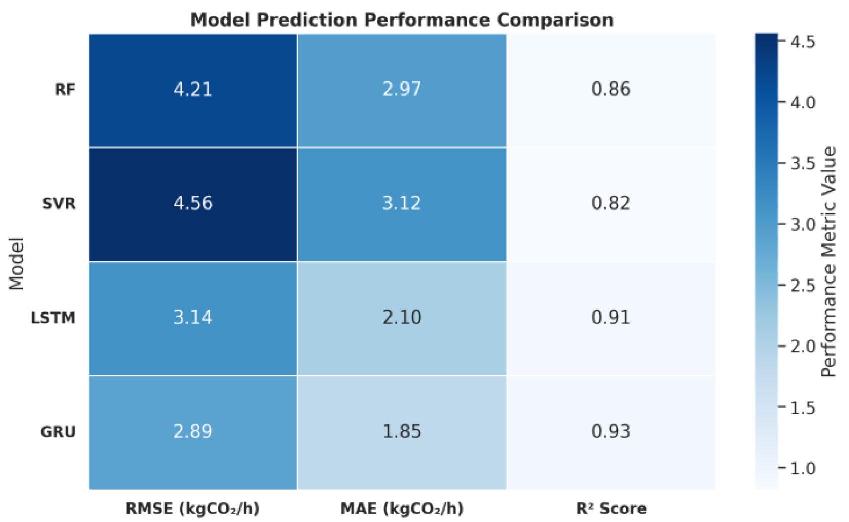


**FIGURE 5.** Model Prediction Performance Comparison

can significantly reduce the energy consumption and carbon emissions while guaranteeing the construction efficiency, which is an effective path to realize the green transformation of construction. The details are shown in Figure 6.

**TABLE 4.** Comparison of energy saving effect before and after scheduling

| Scenario | Avg Carbon Emission ($kgCO_2$) | Fuel Consumption (L) | Energy Saving Rate (%) |
|---|---|---|---|
| Baseline | 1284 | 395 | 0.00 |
| GA | 1156 | 362 | 9.98 |
| PSO | 1110 | 355 | 13.55 |
| IGA | 1023 | 340 | 20.34 |
| IGA+DRL | 967 | 326 | 24.68 |

### D. STABILITY ASSESSMENT

In the actual construction process, the working environment often changes with the weather, load fluctuation and other factors, and the robustness of the model becomes a key indicator of the stable operation of the scheduling system. In order to evaluate the stability of the constructed algorithm under a variety of typical working conditions, experimental tests are conducted in three dimensions: fluctuation of carbon emission prediction, dispersion of scheduling time, and fluctuation of modeling accuracy, which cover five kinds of working conditions such as sunny day, rainy day, high load, low load, and windy day. These scenarios are implemented as simulated disturbances based on the statistical properties of historical data to rigorously test the algorithm's robustness. Specifically, the disturbance modes are quantified as follows:

Sunny Day (Benchmark Condition): This scenario represents the standard, ideal operating condition with no external simulated disturbances or load fluctuations introduced. All parameters (duration, load factor, environment) are set to their statistical mean or baseline values.

Rainy Day simulates a 20% increase in task duration variance and 10% decrease in light equipment working efficiency; High Load simulates a 25% increase in total workload and 15% increase in equipment load factor; Low Load simulates a 15% decrease in total workload and a 10% decrease in equipment load factor (due to increased idle time and smaller task batches); and Windy Day simulates a 15% increase in the wind speed parameter and 10% increase in energy consumption per unit time (due to increased effort).

The results in Table 5 show that the system performs most

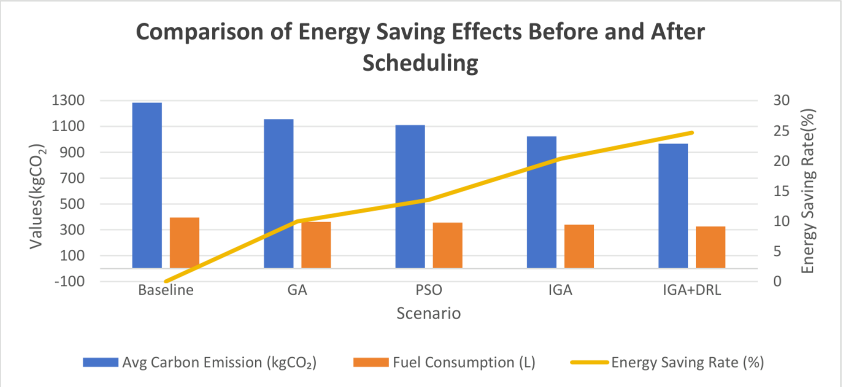


FIGURE 6. Comparison of Energy Saving Effects Before and After Scheduling

stably under sunny and low load conditions, with the standard deviation of carbon emission prediction of 2.15 $kgCO_2$ and 1.89 $kgCO_2$, the standard deviation of scheduling time of 3.4 seconds and 2.8 seconds, and the fluctuation of the model accuracy of 3.2% and 2.4%, respectively. In complex environments such as high load and rainy days, the model prediction volatility rises, with the standard deviation of 3.11 and 2.48, respectively, the scheduling time fluctuates up to 5.6 and 4.2 seconds, and the model accuracy fluctuates up to 5.8% and 4.7%, respectively. The details are shown in Figure 7.

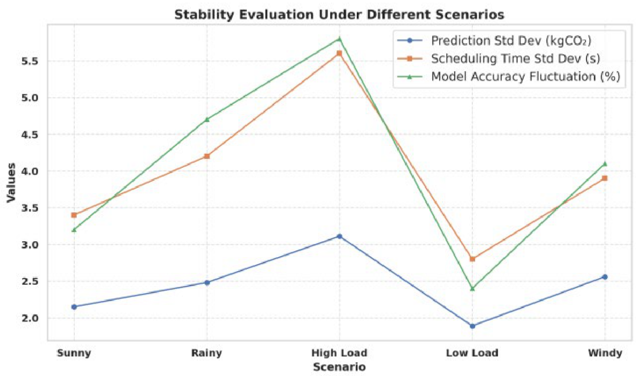


FIGURE 7. Stability Evaluation Under Different Scenarios

TABLE 5. Multi-case stability evaluation results

| Scenario | Prediction Std Dev ($kgCO_2$) | Scheduling Time Std Dev (s) | Model Accuracy Fluctuation (%) |
|---|---|---|---|
| Sunny | 2.15 | 3.4 | 3.2 |
| Rainy | 2.48 | 4.2 | 4.7 |
| High Load | 3.11 | 5.6 | 5.8 |
| Low Load | 1.89 | 2.8 | 2.4 |
| Windy | 2.56 | 3.9 | 4.1 |

Overall, the constructed scheduling system has strong adaptability to working conditions and can still maintain high operational stability in most scenarios, providing a solid guarantee for intelligent low-carbon scheduling in complex construction scenarios.

## VI. CONCLUSION

This study focuses on the low-carbon goal of grid construction machinery scheduling, and constructs a systematic methodology system that integrates equipment carbon emission characteristics modeling, multiobjective optimal scheduling and carbon emission prediction. The coupled design of improved genetic algorithm and deep reinforcement learning effectively improves the scheduling efficiency and carbon control performance; the results of multi-model prediction show that the GRU has a better carbon emission estimation capability under dynamic working conditions, and the experimental results verify the engineering feasibility and popularization value of the proposed method in terms of energy consumption control and system stability.

Future research can further expand the real-time adaptive capability of the optimization algorithm, combine heterogeneous data from multiple sources (such as machine sensor logs, raw material characteristics, and external energy market prices) to realize dynamic identification of carbon emissions across various production conditions (e.g., different fabrics or dye recipes) and linkage optimization of manufacturing strategies. Additionally, research could explore the scheduling game mechanism under parallel scenarios of multiple production lines or facilities, so as to provide theoretical support and technological support for the green management of large-scale and complex textile supply chains and carbon footprint assessment of finished products.

## AUTHOR CONTRIBUTIONS

Conceptualization – Kyle Kai Qin, Yi Zheng, Chaoyan Zeng and Shengzhu Wang; methodology – Kyle Kai Qin, Yi Zheng, Chaoyan Zeng and Shengzhu Wang; investigation – Kyle Kai Qin, Yi Zheng, Chaoyan Zeng and Shengzhu Wang; writing-original draft preparation – Kyle Kai Qin, Yi Zheng, Chaoyan Zeng and Shengzhu Wang. All authors have read and agreed to the published version of the manuscript.

## CONFLICTS OF INTEREST

The authors declare no conflict of interest.

## FUNDING

This research received no external funding.

## ACKNOWLEDGEMENTS

Not applicable.

## REFERENCES

[1] R. Kurniawan, F. Muttaqien H, I. Syafarina, H. Belgaman A, S. Dewi, and A. Latifah L, "Data-driven model for predicting peatland fire hotspots and carbon emissions in South Kalimantan," Wetlands Ecology and Management, Art. no. 33 (3): 44, 2025, doi: 10.1007/s11273-025-10058-z.

[2] A. Alshammari, "Securing smart microgrids with a novel multi-layer cybersecurity framework for Industry 4.0 renewable energy systems," Discover Computing, Art. no. 28 (1): 80, 2025, doi: 10.1007/s10791-025-09600-7.

[3] A. Elgarahy M, M. Eloffy, A. Alengebawy, D. Aboelela, A. Hammad, and K. Elwakeel Z, "Biowaste valorization: Integrating circular economy principles with artificial intelligence-driven optimization for sustainable energy solutions," Journal of Environmental Chemical Engineering, Art. no. 13 (3): 116673, 2025, doi: 10.1016/j.jece.2025.116673.

[4] M. Jiang, S. Lv, Y. Zhang, F. Wu, Z. Pei, and G. Wu, "A Low-Carbon Scheduling Method for Container Intermodal Transport Using an Improved Grey Wolf–Harris Hawks Hybrid Algorithm," Applied Sciences, Art. no. 15 (9): 4698, 2025, doi: 10.3390/app15094698.

[5] M. Hoummadi A, B. Bossoufi, M. Karim, A. Althobaiti, T. Alghamdi H A, and M. Alenezi, "Advanced AI approaches for the modeling and optimization of microgrid energy systems," Scientific Reports, vol. 15, no. 1, Art. no. 12599, 2025, doi: 10.1038/s41598-025-96145-w.

[6] A. Smahi and S. Makhloufi, "The Power Grid Inertia With High Renewable Energy Sources Integration: A Comprehensive Review," Journal of Engineering, vol. 2025, no. 1, Art. no. 7975311, 2025, doi: 10.1155/je/7975311.

[7] U. Jamil, R. Alva J, S. Ahmed, and Y. Jin, "Artificial Intelligence-Driven Optimal Charging Strategy for Electric Vehicles and Impacts on Electric Power Grid," Electronics, vol. 14, no. 7, Art. no. 1471, 2025, doi: 10.3390/electronics14071471.

[8] L. Li, P. Li, Y. Liu, and L. Li, "Research on low-carbon economy optimal scheduling of integrated energy system based on iterative adaptive dynamic programming," Journal of Physics: Conference Series, vol. 3000, no. 1, Art. no. 012006, 2025, doi: 10.1088/1742-6596/3000/1/012006.

[9] Z. Feng, J. Zhang, J. Lu, Z. Zhang, W. Bai, L. Ma, et al., "Low-Carbon Economic Dispatch Strategy for Integrated Energy Systems under Uncertainty Counting CCS-P2G and Concentrating Solar Power Stations," Energy Engineering, vol. 122, no. 4, pp. 1531-1560, 2025, doi: 10.32604/ee.2025.060795.

[10] M. Almihat M G and J. Munda L, "The Role of Smart Grid Technologies in Urban and Sustainable Energy Planning," Energies, vol. 18, no. 7, Art. no. 1618, 2025, doi: 10.3390/en18071618.

[11] W. Ayadi, J. Tavoosi, A. Sarvenoee K, and A. Mohammadzadeh, "Soft-switching predictive Type-3 fuzzy control for microgrid energy management," Energy Informatics, vol. 8, no. 1, Art. no. 38, 2025, doi: 10.1186/s42162-025-00508-6.

[12] B. Lami, M. Alsolami, A. Alferidi, and S. Slama B, "A Smart Microgrid Platform Integrating AI and Deep Reinforcement Learning for Sustainable Energy Management," Energies, vol. 18, no. 5, Art. no. 1157, 2025, doi: 10.3390/en18051157.

[13] N. Sinha, V. Jain, Himanshu, R. Sehrawat, and S. Dhingra, "Synergizing the Future: Electric Vehicles, Artificial Intelligence, and Smart Grids," Smart Grids and Sustainable Energy, vol. 10, no. 1, Art. no. 17, 2025, doi: 10.1007/s40866-025-00247-3.

[14] Q. Huang, Z. Zhuang, M. Duan, S. Yang, J. Sheng, Y. Huang, et al., "A Stackelberg game-based model for lowcarbon scheduling of commercial building loads considering lifecycle unit carbon-emission factors," Energy Conversion and Economics, vol. 6, no. 1, pp. 26-40, 2025, doi: 10.1049/enc2.70000.

[15] T. Senthilkumar, S. Sivaraju S, T. Anuradha, and C. Vimalarani, "An Intelligent Electric Vehicle Charging System in a Smart Grid Using Artificial Intelligence," Optimal Control Applications and Methods, vol. 46, no. 3, pp. 1180-1192, 2025, doi: 10.1002/oca.3252.

[16] J. Bian, Y. Wang, Z. Dang, T. Xiang, Z. Gan, and T. Yang, "Low-Carbon Dispatch Method for Active Distribution Network Based on Carbon Emission Flow Theory," Energies, vol. 17, no. 22, Art. no. 5610, 2024, doi: 10.3390/en17225610.

[17] C. Wu, Z. Wei, Y. Cao, Y. Xu, T. Wei, H. Han, et al., "Low-carbon scheduling model of multi-virtual power plants based on cooperative game considering failure risks," IET Renewable Power Generation, vol. 18, no. 16, pp. 3923-3935, 2024, doi: 10.1049/rpg2.13078.

[18] L. Chen, W. Tang, L. Zhang, Z. Wang, and J. Liang, "Two-stage self-adaption security and low-carbon dispatch strategy of energy storage systems in distribution networks with high proportion of photovoltaics," IET Smart Grid, vol. 7, no. 3, pp. 251-263, 2023, doi: 10.1049/stg2.12118.

[19] Y. Wu, Y. Li, X. Xu, J. Huang, and C. Li, "Research on Low Carbon Dispatching of Hybrid Power Generation System with Wind Power and Pumped Storage Station," IOP Conference Series: Earth and Environmental Science, vol. 153, no. 3, Art. no. 032047, 2018, doi: 10.1088/1755-1315/153/3/032047.