Two-stage Real-time Carbon Emission Monitoring for Low-carbon Power System Operation: a Graph Neural Network-based Approach

Zixuan Zhu, Ruoheng Wang, Siqi Bu, *Senior Member*, *IEEE*, and Roberto Guglielmi

***Abstract*—As carbon emissions reduction is becoming increasingly important for sustainable development and carbon neutrality targets, the concept of carbon emission market has been recently proposed in order to essentially manage carbon emission on the demand side by allowing electricity consumers to purchase or sell carbon emission quotas. Hence, real-time demand-side carbon emission monitoring (DCEM), indicating the amount of carbon emission that each electricity consumer should be responsible for, becomes a necessity for the operation of the carbon emission market. However, the real-time DCEM cannot be achieved by carbon emission flow (CEF) analysis due to the unavailability of real-time power demand data. In this connection, this paper proposes a two-stage real-time DCEM method based on the graph neural network (GNN). In the first stage, power system operating scenario data, including the power generation capacity and power demand data, are collected to calculate carbon emission patterns through CEF analysis. In the second stage, a data-driven GNN-based model is designed to learn from historical daily carbon emission patterns and then realize accurate real-time DCEM with real-time available generation-side measurements only. Case studies on the 118-bus power system operated with day-ahead planning considering carbon emission are performed to demonstrate the accuracy and effectiveness of the proposed method.**

***Index Terms*—Carbon emission flow (CEF), power transmission networks, low-carbon power system, graph neural network.**

I. Introduction

Driven by the carbon emission peak and neutrality target, reducing CO_2 emissions is one of the most crucial missions in the path to the decarbonization goal [1], [2]. As a major contributor to carbon emissions, the power system has a profound impact on the world's carbon footprint, highlighting the need for the establishment of a low-carbon power system [3], [4]. Since the integration of renewable power generation is rapidly advancing [5], [6], the adoption of carbon emission quota (CEQ) policies will incentivize electricity consumers to use clean energy [7], [8]. The carbon emission market is indispensable to promoting CEQ policies, but its efficient operation is based on the understanding of real-time demand-side carbon emission (DCE) patterns (carbon emission originated from demand-side electricity consumers) to plan the CEQ price and implement relevant strategies. Given that, precise real-time demand-side carbon emission monitoring (DCEM) is essential to enable CEQ policies in the carbon emission market.

Several research efforts have been conducted to model, calculate, and analyze carbon emission flow (CEF) in the power system. Reference [9] provides a green energy scheduling method that considers carbon emission reduction. References [10] and [11] utilize life cycle analysis (LCA) to analyze greenhouse gas emissions from wind turbines and power plants over a timescale from manufacturing and operation to retirement, which means that LCA can analyze carbon emission in the generation side along the usage of generation units. However, these methods are not feasible for calculating the DCEs because LCA cannot determine carbon emission induced in the demand side during a specific time period. To calculate the DCE, the concept of CEF is developed, referring to the virtual carbon distribution pattern following energy flow [12].

On the other hand, references [13] and [14] propose a CEF analysis model for regional and user-level carbon emission assessment using a network-based model. As illustrated in Fig. 1, CEF depends on both the charac-

Received: May 20, 2024
Accepted: September 5, 2024
Published Online: May 1, 2025

Zhu Zixuan and Ruoheng Wang are with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China (e-mail: zi-xuan.zhu@connect.polyu.hk; iswangruoheng@foxmail.com).

Siqi Bu (corresponding author) is with the Department of Electrical and Electronic Engineering, Research Centre for Grid Modernisation, International Centre of Urban Energy Nexus, Centre for Advances in Reliability and Safety, Research Institute for Smart Energy, Policy Research Centre for Innovation and Technology, and Shenzhen Research Institute, The Hong Kong Polytechnic University, Hong Kong 999077, China (e-mail: siqi.bu@polyu.edu.hk).

Roberto Guglielmi is with the Department of Applied Mathematics, University of Waterloo, Waterloo N2L 3G1, Canada (e-mail: roberto.guglielmi@uwaterloo.ca).

DOI: 10.23919/PCMP.2023.000172

teristics of power plants and the power flow in the transmission lines. Nevertheless, existing research like [13] and [14] requires full knowledge of power distribution, making it challenging for real-time DCEM due to the possible unavailability of real-time power demand data [15]. For this reason, real-time DCEM should better rely on easy-to-manage and highly-available generation-side data. Hence, this paper proposes a data-driven method to enable DCEM with highly available supply-side data only instead of requiring a full knowledge of power distribution data.

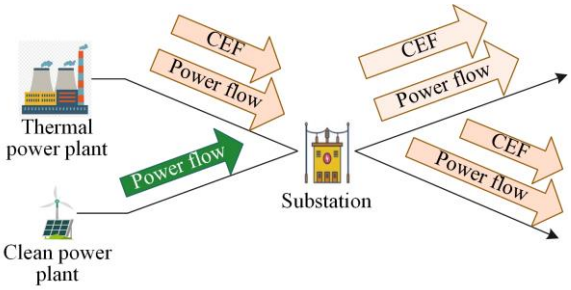


Fig. 1. Illustration of power flow and CEF.

However, the carbon emission patterns in a power system are very complex, especially for the power system integrating with renewable power generations and operating in multi-aspects-considered condition, such as both carbon emission and operation cost, due to the variation in power generation. This brings more challenges for real-time DCEM. In this case, traditional machine learning methods, such as support vector machine and decision tree, may be inadequate to address the intricacies of such a complex system according to the performance of traditional learning methods in other relevant work in power system [16], [17].

This paper proposes an innovative two-stage real-time DCEM method based on graph learning to address the aforementioned research gap. In the first stage, the power flow data of a specific power system is collected, which is needed to calculate DCE patterns for the next stage. A network model is trained using generation-side data and DCE patterns, which can monitor DCE patterns with only generation-side data in the second stage. To solve such a complex system, graph neural network (GNN), an emerging deep learning algorithm that can extract features and topological information from graph-structured data, is utilized.

GNN has been investigated in power system applications due to its exceptional ability to extract spatial features based on network topology. In [16], GNN is used for power system fault classification for power distribution systems, and GNN demonstrates its potential to provide a better generalization ability. Reference [17] utilizes GNN to estimate the power system's transient angle stability. The graph convolution operation can improve the prediction precision and anti-noise ability. GNN is also used to perform power flow calculation, given its capability to deal with graph-structured data effectively [18]. Compared with traditional machine learning methods, the GNN has advantages in dealing with data-oriented in specific topologies [19], [20].

Various experiments related to GNN structure designs are conducted to identify the better-performing GNN structures and suitable graph convolution techniques. A day-ahead planning considering carbon emission (DPCCE) for future power system operation is developed, which is implemented with the IEEE 118-bus system to test the effectiveness and utility of the proposed method. The performance of proposed GNNs is also compared with multi-layer perceptron (MLP) and convolution neural network (CNN) to demonstrate GNN's advantage.

The major contributions of the paper are summarized as follows.

1) This paper proposes a two-stage real-time DCEM scheme based on GNN by using real-time manageable data. In particular, the data collected from future low-carbon scenarios based on a carbon-emission-considered optimization strategy are employed as an example to enable the real-time DCEM.

2) This paper designs GNN structures for real-time DCEM, which can boost real-time DCEM accuracy by extracting and learning spatial patterns of carbon emission. The performances of the proposed GNN are compared with other typical methods, including MLP and CNN, to demonstrate its advantages.

3) This paper reveals seasonal nodal carbon emission intensity (NCEI) patterns, and a negative correlation is discovered between average NCEI and renewable power penetration (RPP), which can effectively evaluate NCEI by looking at the percentage of clean energy injections.

4) This paper develops a DPCCE for the power system operating in a future low-carbon scenario. The proposed method is also tested with DPCCE to demonstrate its effectiveness in future complex power system operating scenarios.

The paper is organized as follows. Section II introduces the CEF calculation model and CEF optimization. Section III discusses the real-time DCEM details and GNN designs. Section IV applies the previously described methodology to two case studies using the 118-bus power system to demonstrate the procedure and performance of the proposed method, and observed NCEI patterns are also discussed. Section V concludes this paper.

II. CEF Modeling and Data Collection

Utilizing power flow data to identify the DCE patterns is the key for the proposed real-time DCEM method. The CEF modeling is discussed in Section II.A. First, the CEF parameters are defined to describe the

DCE. Then, a CEF model is introduced to calculate DCE from power flow data.

As introduced before, the carbon emission market will apply a CEQ price, that is, imposing an additional cost on carbon emission, to encourage the reduction in carbon emissions. Thus, the procedure to generate a low-carbon power system scenario is discussed in Section II.B. Then, the power flow data can be collected to generate DCE data.

A. CEF Parameters and Calculation Model

Four carbon-related parameters are defined for CEF analysis in Table I. Among them, generator carbon intensity (GCI) is regarded as a known parameter describing generator characteristics, and the other parameters can be obtained with the data obtained by power flow analysis. When analyzing CEF, only active power flow, which induces the majority of carbon emission, is considered [14].

TABLE I
DEFINITION OF CEF PARAMETERS

Parameters	Explanation
Generator carbon intensity	Generator characteristic
Generator CEF	Carbon emission from generators
Nodal carbon emission intensity	Carbon emission induced by each unit of power consumption at power demand side
Nodal carbon emission	Carbon emission induced by power consumption at power demand side

The carbon emission from the generator, generator carbon emission flow (GCEF), is defined in (1), where R_{G}, P_{G}, and E_{G} denote the GCEF, power injection, and GCI of a generator, respectively. Given that CEF is concurrent with energy flow [13], [14], the proportional sharing principle (PSP) assumption is taken in CEF analysis. The main idea of PSP is that, in a network system, the energy flow in the outflow branch is proportionally distributed in outflow branches [21]. Therefore, the NCEI is dependent on the inflow power and the corresponding carbon intensity only, which is the carbon emission associated with a single unit of power consumption. In particular, the NCEI at the *i*th node is the ratio of the injected carbon emission to the injected power flow, as shown in (2), where P_i and e_i are the inflow power and the corresponding NCEIs of the previous node; X^+ indicates the previous bus from which the power flow is flowing into bus i; $P_{\mathrm{G},j}$ and $e_{\mathrm{G},j}$ are the generator output and the corresponding GCI. If there is no generator at node j, $P_{\mathrm{G},j}$ is zero. Since the GCI and power flow distribution are known, the NCEI of each node can be calculated one by one. After obtaining the NCEI, the NCE of bus *i*, referring to the amount of the DCE, can be obtained by multiplying the corresponding NCEI, e_i, and power demand, P_i, as given in (3).

$$R_{\mathrm{G}} = P_{\mathrm{G}} E_{\mathrm{G}} \tag{1}$$

$$e_i = \frac{\sum_{i\in X^+} P_i e_i + P_{\mathrm{G},j} e_{\mathrm{G},j}}{\sum_{i\in X^+} P_i + P_{\mathrm{G},j}} \tag{2}$$

$$R_i = P_i e_i \tag{3}$$

The model proposed by reference [14] is used to perform CEF calculation. Since GCEF is the product of generation output and GCI, the GCEF matrix containing the GCEF and the corresponding location, $\boldsymbol{R}_{\mathrm{G}}$, can be obtained by:

$$\boldsymbol{R}_{\mathrm{G}} = \boldsymbol{P}_{\mathrm{G}} \boldsymbol{E}_{\mathrm{G}} \tag{4}$$

where $\boldsymbol{P}_{\mathrm{G}}$ is an $n_{\mathrm{b}} \times n_{\mathrm{g}}$ generator output matrix (n_{b} is the number of buses in the system, and n_{g} is the number of generators); $\boldsymbol{E}_{\mathrm{G}}$ is a $n_{\mathrm{g}} \times 1$ GCI matrix containing the GCI of each generator in the system. For instance, if the *x*th generator located at node *y* has a power injection of *a*, the (y, x)th element of $\boldsymbol{P}_{\mathrm{G}}$ is *a*, and $P_{\mathrm{G}yx} = a$. On the other hand, if the *y*th node has no generator, $P_{\mathrm{G}yx}$ is zero. Note that $\boldsymbol{R}_{\mathrm{G}}$ is a column vector, the element of which represents the GCEF at each note. For example, the first element of $\boldsymbol{R}_{\mathrm{G}}$ is the GCEF of node 1.

The general equation to calculate NCEI shown in (2) can be rewritten into:

$$\sum_{i\in X^+} P_i e_i + P_{\mathrm{G},j} e_{\mathrm{G},j} = e_i \left(\sum_{i\in X^+} P_i + P_{\mathrm{G},j} \right) \tag{5}$$

The right-hand side of (5) can be expressed in the form of a matrix as:

$$e_i \left(\sum_{i\in X^+} P_i + P_{\mathrm{G}j} \right) = e_i \boldsymbol{\eta}_X^{(i)} \boldsymbol{F}_{\mathrm{N}} \left(\boldsymbol{\eta}_X^{(i)} \right)^{\mathrm{T}} \tag{6}$$

where $\boldsymbol{\eta}_X^{(i)}$ is an *i*th unit row matrix that has n_g dimensions; and $\boldsymbol{F}_{\mathrm{N}}$ denotes the power flow flux at each node, which can be obtained via:

$$\boldsymbol{F}_{\mathrm{N}} = \mathrm{diag}\left\{ \boldsymbol{\delta}_{X+Y} \begin{bmatrix} \boldsymbol{P}_{\mathrm{B}} \\ \boldsymbol{P}_{\mathrm{G}}^{\mathrm{T}} \end{bmatrix} \right\} \tag{7}$$

where $\boldsymbol{P}_{\mathrm{B}}$ is an $n_b \times n_b$ matrix consolidating the power flow distribution, if the power flow goes from node *i* to node *j*, $\boldsymbol{P}_{\mathrm{B}ij} = P$, denoting the (i, j)th element of $\boldsymbol{P}_{\mathrm{B}}$, ($P$ is the quantity of the power flow, and the value of P is non-negative), in particular, if no power flow passes through branch *ij*, $\boldsymbol{P}_{\mathrm{B}ij}$ is zero; $\boldsymbol{\delta}_{X+Y}$ is a unit row matrix with a shape of $1 \times (n_b + n_g)$; and it is noticeable that

$\boldsymbol{F}_{\mathrm{N}}$ is a diagonal matrix and diag{·} denotes the diagonal matrix operator.

The left hand side of (5) can be expressed as the following by expanding e_i to $\boldsymbol{I}_{\mathrm{N}}$ that contains NCEIs of different nodes, as:

$$\sum_{i\in X^{+}} P_i\rho_i + P_{\mathrm{G}j}e_{\mathrm{G}j} = \boldsymbol{\eta}_X^{(i)}\left(\boldsymbol{P}_{\mathrm{B}}^{\mathrm{T}}\boldsymbol{I}_{\mathrm{N}} + \boldsymbol{R}_{\mathrm{G}}\right) \tag{8}$$

where $\boldsymbol{I}_{\mathrm{N}}$ is a column matrix consolidating the NCEI of each node, which is the goal of CEF analysis. With (6) and (8), equation (5) can be expressed as:

$$\boldsymbol{F}_{\mathrm{N}}\boldsymbol{I}_{\mathrm{N}} = \boldsymbol{P}_{\mathrm{B}}^{\mathrm{T}}\boldsymbol{I}_{\mathrm{N}} + \boldsymbol{R}_{\mathrm{G}} \tag{9}$$

And $\boldsymbol{I}_{\mathrm{N}}$ can be calculated as:

$$\boldsymbol{I}_{\mathrm{N}} = \left(\boldsymbol{F}_{\mathrm{N}} - \boldsymbol{P}_{\mathrm{B}}^{\mathrm{T}}\right)^{-1}\boldsymbol{R}_{\mathrm{G}} \tag{10}$$

By substituting $\boldsymbol{F}_{\mathrm{N}}$, $\boldsymbol{I}_{\mathrm{N}}$ can be calculated by:

$$\boldsymbol{I}_{\mathrm{N}} = \left(\mathrm{diag}\left\{\boldsymbol{\delta}_{X+Y}\begin{bmatrix}\boldsymbol{P}_{\mathrm{B}}\\ \boldsymbol{P}_{\mathrm{G}}^{\mathrm{T}}\end{bmatrix}\right\} - \boldsymbol{P}_{\mathrm{B}}^{\mathrm{T}}\right)^{-1}\boldsymbol{R}_{\mathrm{G}} \tag{11}$$

Note that all variables in (11) are known if the power flow distribution is known, and $\boldsymbol{I}_{\mathrm{N}}$ can be therefore calculated with power flow data.

B. Day-ahead Planning Considering Carbon Emission for Low-carbon Operation

The power system will operate under low-carbon conditions in the scenario of the CEQ market, which means that the power system operator needs to consider both operation cost and CEQ price instead of operation cost only. In this case, a DPCCE for power systems operating in low-carbon scenarios is developed in this paper. Two matrices consolidating the generation costs and CEQ prices for generation units are involved to construct the objective function for CEF optimization. In particular, $\boldsymbol{G}_{\mathrm{cost}}$ and $\boldsymbol{E}_{\mathrm{G}}$ in (12) are $n_g \times 1$ matrices containing the generation cost and GCI of each generation, respectively; two weights are assigned to carbon emission and generation cost by w_{carbon} and w_{cost}. The overall economic parameter of each generator is obtained through (12) with element-wise multiplication, where $\boldsymbol{W}_{\mathrm{overall}}$ is a $n_g \times 1$ matrix needed for CEF optimization.

$$\boldsymbol{W}_{\mathrm{overall}} = \boldsymbol{G}_{\mathrm{cost}} \odot w_{\mathrm{cost}} + \boldsymbol{E}_{\mathrm{G}} \odot w_{\mathrm{carbon}} \tag{12}$$

The objective function for CEF optimization is given in (13), where $W_{\mathrm{overall},i}$ refers to the ith element in $\boldsymbol{W}_{\mathrm{overall}}$; and P_{G}^{i} and Q_{G}^{i} are the active power and reactive power output of the ith generator.

$$f(P_{\mathrm{G}}, Q_{\mathrm{G}}) = \sum_{i=1}^{Y} W_{\mathrm{overall},i} \times (P_{\mathrm{G}}^{i} + Q_{\mathrm{G}}^{i}) \tag{13}$$

The optimization model can be constructed as described in [22]. The model used in this paper is provided in Appendix A.

With this algorithm, a low-carbon power system scenario can be generated. As illustrated in Fig. 2, power generation is planned with forecasted power demand and renewable power generation data. The power flow data can then be collected, and the nodal carbon emission (NCE) and NCEI can be calculated as discussed in Section II.A.

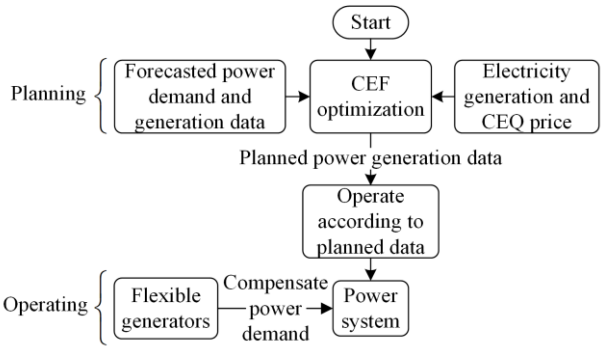


Fig. 2. DPCCE for low-carbon power system operation.

After CEF modeling, each bus has four features, including power injection, GCEF, NCE, and NCEI, as shown in Fig. 3. NCE and NCEI are chosen to describe the DCE, while power injection and GCEF are collected as the generation-side data for real-time DCEM due to the real-time availability of the generation-side data. In particular, if a node does not have power injection, the corresponding power injection, as well as GCEF, are set to zero. The collected data will be used to train the GNN for real-time DCEM, which will be discussed in Section III.

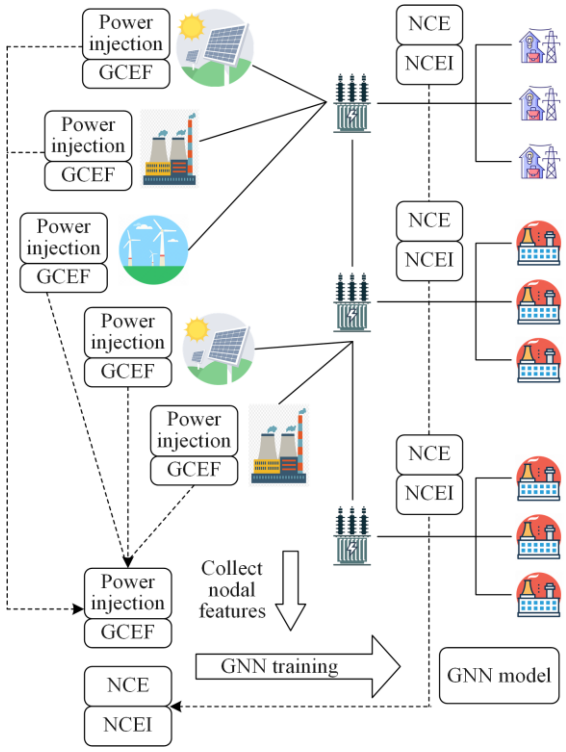


Fig. 3. Illustration of operation data collection.

III. GNN-BASED APPROACH FOR REAL-TIME DCEM

A. Construction of Adjacent Matrix

The adjacent matrix is a sparse matrix used to represent the connections or relationships between nodes in a graph. In the scenario of the power system, the adjacent matrix is constructed based on the power system topology. The substations can be regarded as nodes, and branches can be regarded as edges. For example, the IEEE 14-bus system can be expressed in a graph, as shown in Fig. 4. The nodal features are assigned based on the collected generation-side data and DCE data as explained in Fig. 3.

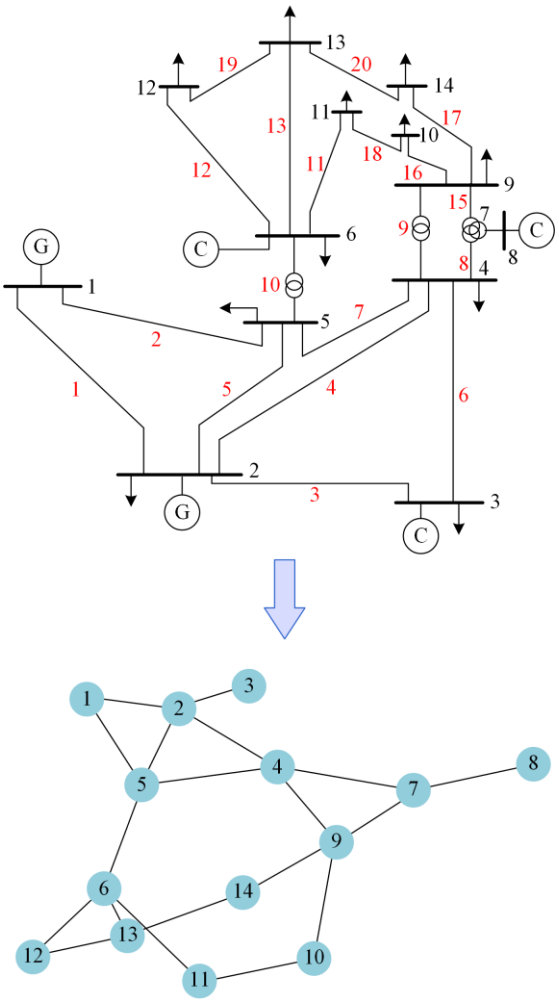


Fig. 4. Power system expressed in graph.

For a power system with a number of n_b, a $n_b \times n_b$ square matrix denoted by $\boldsymbol{A}$ filled with all zeros should be initialized. If bus i and bus j of a power system are connected, the (i, j)th element of $\boldsymbol{A}$ is 1. Since we use undirected graph, the (j, i)th element of $\boldsymbol{A}$ is also 1. And for a simple 3-bus power system in Fig. 5, the adjacency matrix can be created according to the network topology.

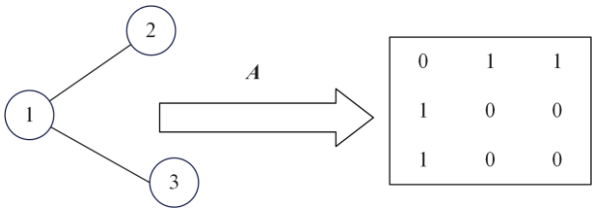


Fig. 5. Adjacent matrix construction.

B. Graph Convolution

In this paper, we will investigate three graph convolution techniques: graph convolution network (GCN) [23], graph attention network (GAT) [24], and generalized graph network (GEN) [25]. GCN is a typical GNN that encodes both graph edges and features. GAT is a typical GNN that also includes importance assignments to enhance the effectiveness of the neural network. According to [26], GAT can consistently outperform GCN. GEN is one of the state-of-the-art GNNs that employs generalized aggregation functions. According to [25], GEN achieves the best performance over other GNNs.

1) Graph Convolution Network

The GCN proposed in [23] is motivated by spectral graph convolution. In the context of real-time DCEM, the input graph-structured data can be expressed by $f = (\boldsymbol{H}, \boldsymbol{A})$, where $\boldsymbol{H}$ and $\boldsymbol{A}$ denote the node feature matrix and adjacent matrix representing power system topology, respectively. For a power system with N buses, $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, which is a binary matrix, and $\boldsymbol{H} \in \mathbb{R}^{N \times 2}$ for 2 features for each node.

For a signal $\boldsymbol{h} \in \mathbb{R}^N$, which is a scalar for every node, and a filter $G_\theta \in \mathrm{diag}(\boldsymbol{\theta})$ that is parameterized in the Fourier domain by $\boldsymbol{\theta} \in \mathbb{R}^N$, the spectral convolution can be expressed as:

$$\boldsymbol{h} * G_\theta = \boldsymbol{U} G_\theta \boldsymbol{U}^{\mathrm{T}} \boldsymbol{h} \tag{14}$$

where $\boldsymbol{U}$ is the eigenvector matrix of the normalized graph Laplacian matrix with eigenvalue λ, which can be expressed by $\boldsymbol{L} = \boldsymbol{I}_{\mathrm{N}} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}} = \boldsymbol{U} \lambda \boldsymbol{U}^{\mathrm{T}}$.

As proved in [24], the spectral convolution can be approximated by (15) with a single parameter $\boldsymbol{\theta} = \boldsymbol{\theta}_0' = -\boldsymbol{\theta}_1'$. Note that $\boldsymbol{\theta}'$ is a vector of the Chebyshev coefficient, as discussed in [27].

$$\boldsymbol{h} * G_\theta \approx \theta \left(\boldsymbol{I}_{\mathrm{N}} + \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}} \right) \boldsymbol{h} \tag{15}$$

The convolution in (15) can be generalized into (16), where $\boldsymbol{D}_{ii} = \sum_{j=1}^{n} \hat{\boldsymbol{A}}_{ij}$, $\hat{A} = \boldsymbol{A} + \boldsymbol{I}_{\mathrm{N}}$ is the adjacent matrix with $\boldsymbol{I}_{\mathrm{N}}$ denoting the identity matrix, and $\boldsymbol{\Phi} \in \mathbb{R}^{2 \times F}$ is

the filter parameter matrix. $\boldsymbol{H} * G_\theta$ is the convolved matrix, and $\boldsymbol{H} * G_\theta \in \mathbb{R}^{N \times F}$.

$$\boldsymbol{H} * G_\theta = \hat{\boldsymbol{D}}^{-\frac{1}{2}} \hat{\boldsymbol{A}} \hat{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{H} \boldsymbol{\Phi} \tag{16}$$

Based on (16), the GCN propagation rule is given in (17), where $\sigma(\cdot)$ is the activation operator; $\boldsymbol{W}^l$ is a layer-specific trainable weight matrix; and $\boldsymbol{H}^l \in \mathbb{R}^{N \times D}$ is the activation matrix at lth layer; the l^0 layer equals to the input feature, $\boldsymbol{H}^0 = \boldsymbol{H}$.

$$\boldsymbol{H}^{l+1} = \sigma\left(\hat{\boldsymbol{D}}^{-\frac{1}{2}} \hat{\boldsymbol{A}} \hat{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{H}^l \boldsymbol{W}^l \right) \tag{17}$$

2) Graph Attention Network

The GAT proposed in [24] integrates attention mechanisms into graph domains. The input here is also a set of node features: $\boldsymbol{H}^0 = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \cdots, \boldsymbol{f}_N\}$ for $\boldsymbol{f}_i \in \mathbb{R}^2$. The features of the next layer are $\boldsymbol{H}^1 = \{\boldsymbol{f}_1^1, \boldsymbol{f}_2^1, \cdots, \boldsymbol{f}_N^1\}$ for $\boldsymbol{f}_i \in \mathbb{R}^F$. As provided in [21], the next-layer features can be calculated by K independent attention mechanism:

$$\boldsymbol{f}_i^{l+1} = \mathop{\Big\|}_{k=1}^{K} \sigma\left(\sum_{j \in N_i} \alpha_{ij}^k \boldsymbol{W}^k \boldsymbol{f}_j^l \right) \tag{18}$$

where $\boldsymbol{f}_i^l$ is a set of node features of node i at layer l; N_i represents the adjacent nodes of node i; $\boldsymbol{W}^k$ denotes the input linear transformation's weight matrix corresponding to α_{ij}^k; α_{ij}^k is the normalized attention coefficient computed in the kth attention mechanism; and $\|$ denotes concatenation. The attention coefficients can be calculated by:

$$a_{ij} = \frac{\exp\left[\text{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}} \left[\boldsymbol{W}\boldsymbol{f}_i^l \,\|\, \boldsymbol{W}\boldsymbol{f}_j^t \right] \right) \right]}{\sum_{k \in N_i} \exp\left[\text{LeakyReLU}\left(\boldsymbol{a}^{\mathrm{T}} \left[\boldsymbol{W}\boldsymbol{f}_i^t \,\|\, \boldsymbol{W}\boldsymbol{f}_k^t \right] \right) \right]} \tag{19}$$

The negative input slope of LeakyReLU is 0.2, which is rectified linear activation function with a slope when the input is negative; and $\boldsymbol{a}^{\mathrm{T}}$ is a weight vector such that $\boldsymbol{a}^{\mathrm{T}} \in \mathbb{R}^{2F'}$, parameterizing the attention mechanism.

GAT enables importance assignments for better prediction accuracy. However, the attention mechanism may require a large amount of data, and limited power system operation data may not fully explore the potential of GAT.

3) Generalize Graph Network

GEN is proposed by [25], which works with fully differentiable message construction function, message aggregation function, and vertex update function.

The message construction function for GEN for individual message construction for each neighbor, $\boldsymbol{h}_{ij}^{(l)}$, is defined in (20), which is applied to features at node i, $\boldsymbol{h}_i$, and its neighbor features $\boldsymbol{h}_j$ such as $j \in N_i$. It is noticeable that some message aggregators are only applicable to positive features but will not affect the implementation of DCEM because all features involved are positive.

$$\boldsymbol{h}_{ij}^{(l)} = \text{ReLU}(\boldsymbol{h}_j^{(l)}) + \epsilon, j \in N_i \tag{20}$$

where ReLU(·) is rectified linear unit activation function; and ϵ is a positive constant, i.e., 10^{-7}.

The message aggregation function can output an aggregated or reduced nodal features, as described in (21), where AGG(·) denotes aggregation operation. Two aggregation functions are usually involved in GEN, SoftMax aggregation and PowerMean aggregation [28]. For a message set $\{\boldsymbol{h}_{ij} | j \in N_i\}$ and $\boldsymbol{h}_{ij} \in \mathbb{R}^D$, the SoftMax aggregation function is defined in (22)., whereas for a message set $\{\boldsymbol{h}_{ij} | j \in N_i\}$ and $\boldsymbol{h}_{ij} \in \mathbb{R}_+^D$, the PowerMean aggregation function is defined in (23) controlled by a parameter p. Our experiments show that SoftMax aggregation and PowerMean are literally equally effective in feature aggregation real-time DCEM.

$$\boldsymbol{h}_i = \text{AGG}\left(\left\{ \boldsymbol{h}_{ij} \mid j \in N_i \right\} \right) \tag{21}$$

$$\text{SoftMax\_AGG}(\cdot) = \sum\nolimits_{j \in N_i} \frac{\exp(\beta \boldsymbol{h}_{ij})}{\sum_{k \in N_i} \exp(\beta \boldsymbol{h}_{ik})} \times \boldsymbol{h}_{ij} \tag{22}$$

$$\text{PowerMean\_AGG}(\cdot) = \left(\frac{1}{|N_i|} \sum\nolimits_{j \in N_i} \boldsymbol{h}_{ij}^p \right)^{\frac{1}{p}} \tag{23}$$

PowerMean aggregation can only be applied when all features are positive, and SoftMax aggregation is applicable for both positive and negative features. The two aggregation operations have similar operations. We use PowerMean aggregation in this paper because PowerMean is less computational complex.

The vertex update function can be then expressed in (24), where MLP is used to output the updated features.

$$\boldsymbol{h}_i^{(l+1)} = \text{MLP}\left(\boldsymbol{h}_i^{(l)} + \text{AGG}\left(\boldsymbol{h}_{ij}^{(l)} \right) \right), j \in N_i \tag{24}$$

Compared with other GCN, GEN can strengthen the feature aggregation functions of GNN and enhance the capability of node features learning.

C. GNN Architecture for Real-time DCEM

The graph-structured data are constructed to represent the power system topology. The edges and nodes of the graph-structured data can represent transmission lines and substations, as explained in Section III.A.

Based on the graph convolution operations discussed above, we design two GNN structures for real-time DCEM: single-layer GNN and multi-layer GNN.

The convolution layer is directly connected to a series of FCLs in the single-layer GNN design, as illustrated in Fig. 6. In particular, P_{G}^{n} and R_{G}^{n} represent the power injection and GCEF at node n as the generation-side data; e_n and R_n denote the demand-side data, NCEI and NCE at node n. This setup enables the neural networks to possess a fundamental graph feature extraction capability.

For the multi-layer GNN network, the initial graph convolution layers are connected to an intra-FCL (fully-connected layer) for feature downsize, which is connected to the subsequent graph convolution layers, as shown in Fig. 7. The intra-FCL serves the purpose of reducing feature dimensions to its input feature size. Specifically, if the graph convolution layers extract more nodal features, the intra-FCL will reduce these nodal features to match the original size. Our experiments show that this feature downsize can reduce the training time and enhance the network's performance, which will also be discussed in the case study section.

While GNN networks can contain multiple graph convolution layers, we empirically discourage the use of deep GNN networks. Our experiments indicate that deep GNN networks may lead to a degradation in network performance and slow down the training process. Based on empirical evidence, we recommend GNNs with two graph convolution layers, as they can achieve desirable accuracy and training speed for most power systems.

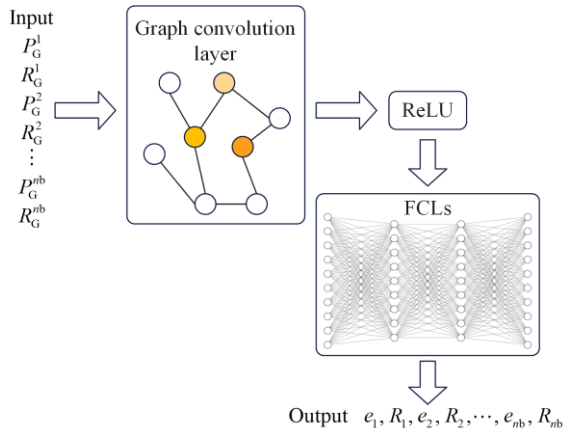


Fig. 6. Single-layer GNN network structure for real-time DCEM.

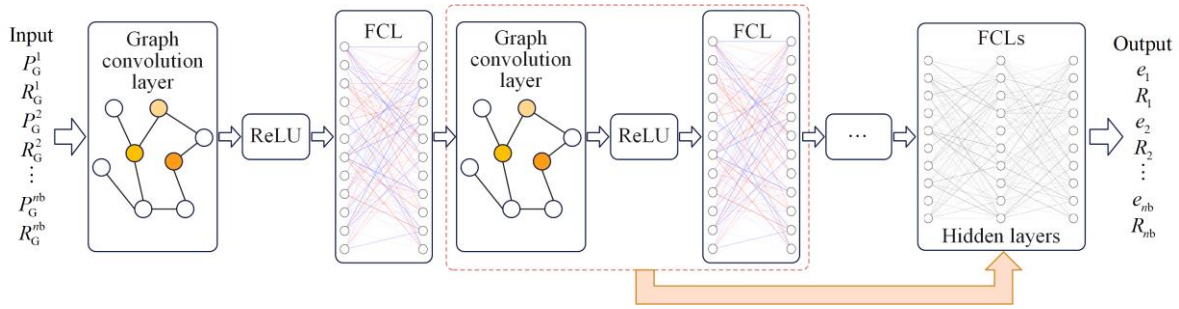


Fig. 7. Multiple-layer GNN network structure for real-time DCEM.

D. Assessment Metrics and Implementation Procedure

GNNs use supervised learning algorithms to train the model based on observed and predicted values. During the training process, mean square error (MSE) is used to estimate the network's performance, and network weights in the neural network are adapted through backward propagation of error. Once the training process is completed, other metrics can assess the networks from different perspectives, as given in (25). Note that $\hat{y}_i$ indicates the actual value. However, it should be noted that mean absolute percentage error (MAPE) and symmetric mean absolute percentage error (SMAPE) are not suitable for estimating the performance of the trained network for real-time DCEM because the values of NCEs or NCEIs of some nodes may be zero, which may lead to a problem of "dividing by zero" when using these metrics.

$$
\begin{aligned}
E_{\mathrm{MSE}} &= \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \\
E_{\mathrm{MAE}} &= \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \\
E_{\mathrm{R}^2} &= 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \\
E_{\mathrm{MAPE}} &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \\
E_{\mathrm{SMAPE}} &= \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\frac{\left|\hat{y}_i - y_i\right|}{\left|\hat{y}_i\right| + \left|y_i\right|}
\end{aligned}
\tag{25}
$$

The overall procedure to implement the proposed real-time DCEM is shown in Fig. 8. A low-carbon power system scenario simulating the power system operating scenario under the context of the carbon

emission market is first generated, as discussed in Section II.B. Then, the power flow data of the low-carbon scenario is collected. The generation-side data (GCEF and power injection) and the demand-side data (NCE and NCEI) are calculated through the aforementioned CEF model, which is essential to enable the data-driven method.

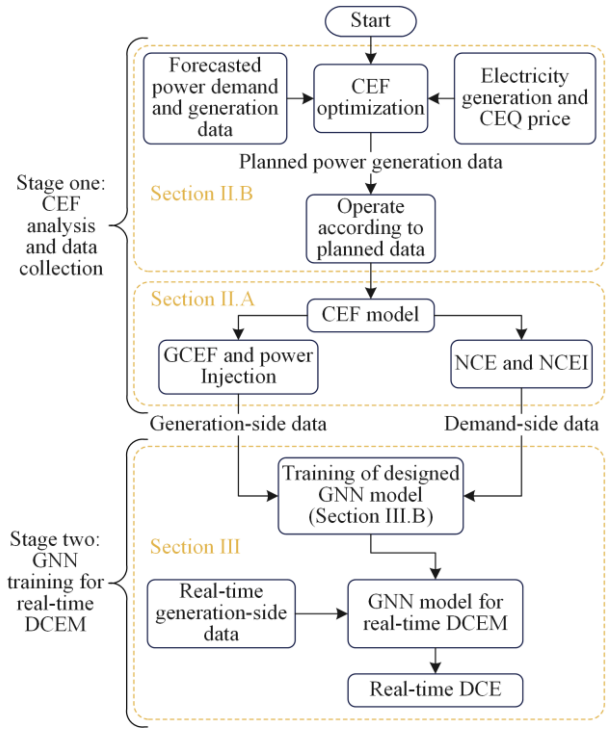


Fig. 8. Overall procedure for CEF optimization and neural network training for real-time DCEM.

IV. Case Study

A case study based on the IEEE 118-bus power system is presented using the proposed method. The test system includes 54 generators, 118 substations, and 186 branches, the details of which are available in MATPOWER [29].

The generation cost and CEQ price are determined according to IEA's report in 2020 [30]. The actual and forecasted data of power demand and renewable power generation, as shown in Fig. 8, are derived from real power system operation data from open power system data and Elia open data [31], [32]. Specifically, the training data is selected from the hourly sampled data between 2015 and 2018, while the test data is collected from 2018 to 2019. Given that only active power data is available, we assume that the reactive power demand shares the same trends with active power.

We first implement the DPCCE to generate the low-carbon power system scenario and data collection using Matlab and MATPOWER. Subsequently, under the low-carbon power system operation scenario, we train the GNNs proposed in Section III for real-time DCEM. The GNNs are built and trained with PyTorch and PyG on a laptop with Intel(R) Core(TM) i5-10300 H CPU @ 2.50 GHz. Other typical methods for comparison are implemented with PyTorch (MLP and CNN) and scikit-learn. The performance of the GNNs is presented to illustrate the effectiveness of the proposed method. The advantages of the proposed GNN architecture and observed NCEI patterns are also discussed in this section.

A. Low-carbon Scenarios Generated by DPCCE

The power system in this case study is composed of different types of generation units, including 13 coal-fired generators, 22 gas turbines, 5 wind farms, and 14 solar power units. The setting of these generators is shown in Table II, and all the generation units are assigned to buses randomly, as given in Table III.

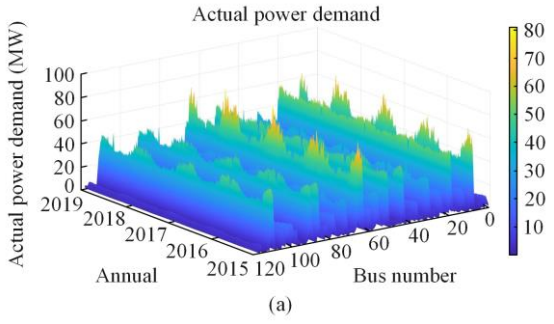
TABLE II
Generation Unit Setting

Generator type	GCI	Carbon price ($/tCO$_2$)	Generation cost ($/MW)
Coal-fired	0.875	30	78
Gas turbine	0.6	30	68
Wind	0	30	66
Photovoltaic	0	30	73

TABLE III
IEEE 118-bus System Generator Assignment

Generator type	Generator number
Coal-fired power	10, 12, 13, 14, 22, 30, 34, 42, 43, 44, 45, 52, 53
Gas power	1, 3, 8, 11, 15, 16, 18, 20, 24, 29, 32, 35, 36, 37, 38 39, 40, 41, 46, 47, 48, 54
Wind power	7, 17, 21, 28, 51
Solar power	2, 4, 5, 6, 9, 19, 23, 25, 26, 27, 31, 33, 49, 50

Using the DPCCE discussed in Section II.B, the system can be operated in a low-carbon scenario. To emulate the profit-driven characteristics of all stakeholders, the carbon emissions and generation cost weight are both set to 0.5 [33]. The day-ahead power generation is planned with the forecasted power demand and generation data in Figs. 9(b) and (d). The actual power demand and renewable power generation are shown in Figs. 9(a) and (c). In real-time operation, the generation units operate as planned, and the flexible generator will compensate for the day-ahead prediction error. The power injection, GCEF, NCE, and NCEI data in such a low-carbon scenario are provided in Figs. 10 (a)–(d), respectively.



(a)

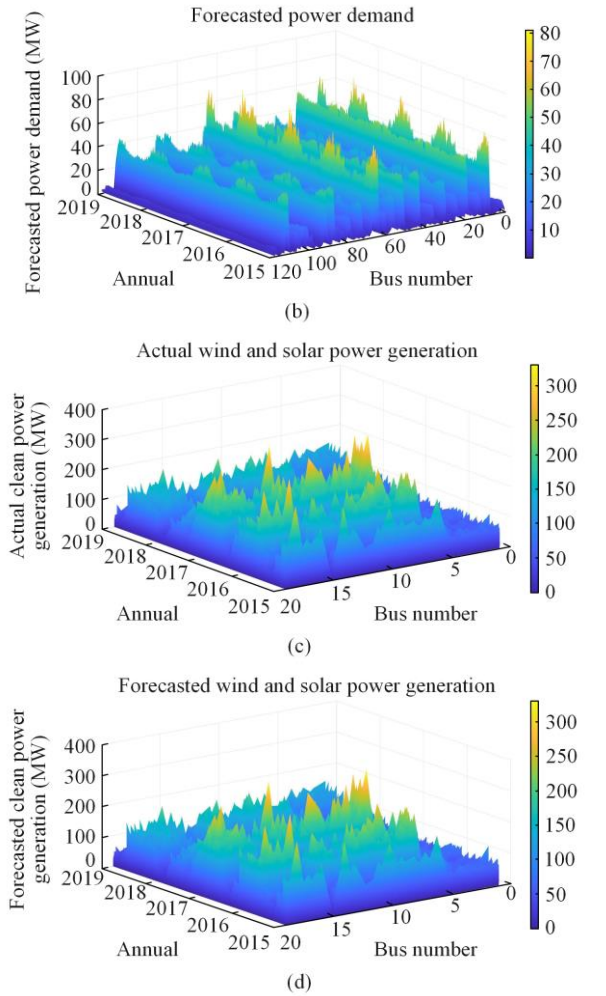
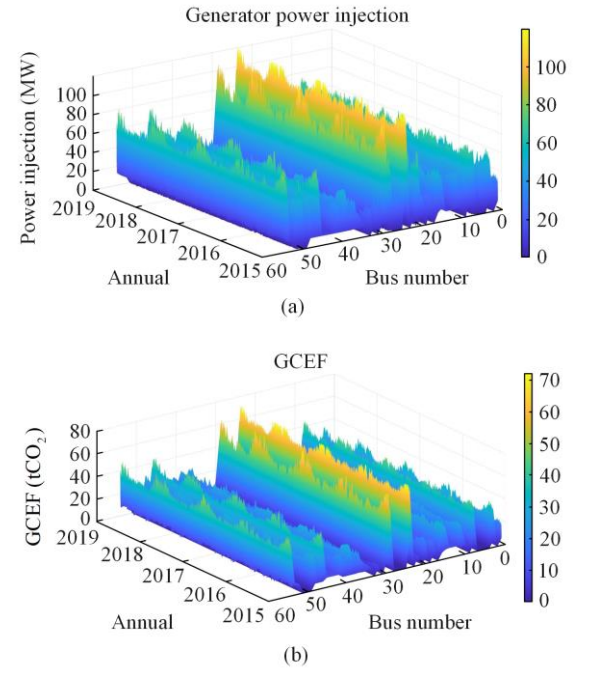


Fig. 9. Power system operation data. (a) Actual power demand. (b) Day-ahead forecasted power demand. (c) Actual renewable power generation. (d) Day-ahead forecasted renewable power generation.



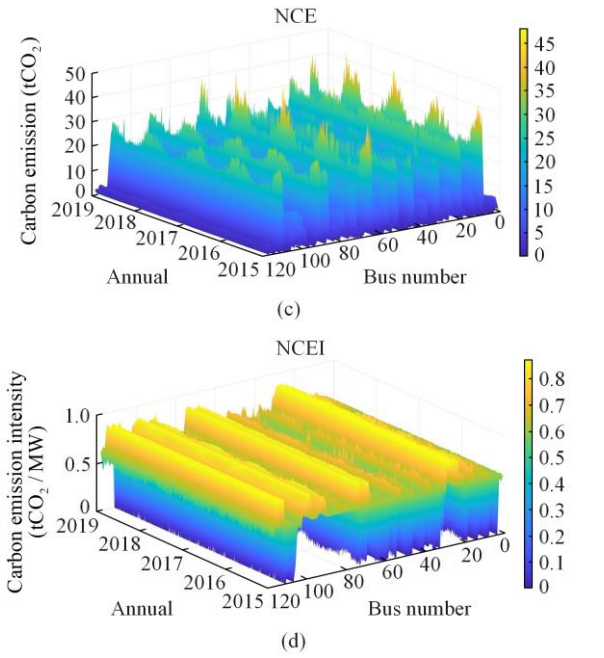


Fig. 10. Generation-side and demand-side data. (a) Power injection. (b) GCEF. (c) NCE. (d) NCEI.

However, the inevitable prediction errors will lead to a higher carbon emission, as given in Table IV. An additional regular power system that only takes the electricity generation cost into account is also presented as a basis for comparison. It is observed that the CEF-considered planning has a lower carbon emission compared with regular generation cost planning.

TABLE IV
ESTIMATED CARBON EMISSION AND ACTUAL POWER EMISSION

Stage	CEF-considered planning (tCO_2)	Regular cost planning (tCO_2)
Planning	1.5976×10^7	1.7825×10^7
Actual operation	1.7134×10^7	1.8222×10^7
With historical data	1.6165×10^7	1.6810×10^7

The nature of the low-carbon power system scenario is to prioritize the utilization of energy sources with lower GCI. For instance, regular cost planning tends to shut down solar power and utilize gas power instead. On the other hand, in the low-carbon scenario, renewable sources are preferred due to the involvement of the CEQ price. The advantage of that in carbon emission reduction can also be evaluated by RPP, which is defined in (26), where P_{TR} is the total renewable power generation and P_{TD} is the total power demand, to describe the proportion of clean energy in a power system. Figure 11 shows a case of monthly variations of RPPs of regular and low-carbon power system scenarios. It is observed that the low-carbon scenario has higher RPPs compared with the regular scenario, and a higher RPP indicates a higher percentage of clean energy and a lower carbon emission.

$$E_{\mathrm{RPP}} = \frac{P_{\mathrm{TR}}}{P_{\mathrm{TD}}} \tag{26}$$

The operation data can be collected to enable the proposed data-driven method. The power injection and GCEF are collected as the generation-side data, and the NCE and NCEI are collected as the demand-side data. The performance of GNN will be discussed in the next section.

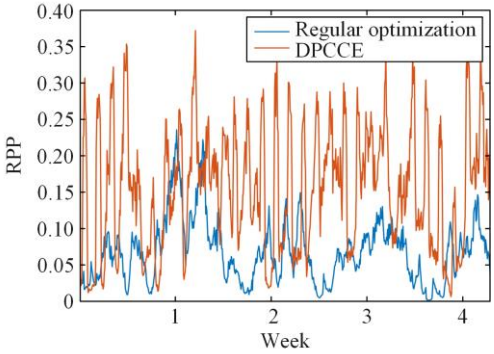


Fig. 11. RPP comparison between DPCCE and regular optimization.

B. CNN and MLP for Comparison

MLP, consisting of a series of FCLs, is one of the typical models of the artificial neural network consisting of the input layer, output layer, and hidden layers, which is also built as a comparison basis to highlight the advantages of the proposed GNNs.

Furthermore, typical CNNs are built and tested to further demonstrate the advantage of GNN according to [34]–[36]. Reference [34] uses CNN-LSTM for short-term forecasting of photovoltaic power production, reference [35] uses a hybrid CNN-based network to predict global solar radiation, and reference [36] proposes a CNN-based method for distribution system fault classification. Since the DCEs and generation-side data only have the spatial connection, we reasonably replace LSTMs with FCLs. The CNNs used in this paper consist of convolution layers, pooling layers, and FCLs, as shown in Fig. 12. We use both 1D CNN and 2D CNN to explore the potential of CNNs, as discussed in [37].

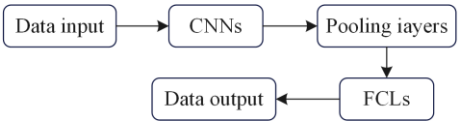


Fig. 12. Architecture of typical CNNs.

CNNs normally incorporate pooling layers to further extract features and accelerate network training. However, pooling layers may also lose some information and result in poorer CNN performance. Therefore, we test each CNN architecture with and without pooling layers to decide whether a CNN should include a pooling layer. We find that the average pooling layer is more suitable compared with max pooling for real-time DCEM, and the pooling layer may strengthen the CNN's generalization ability for some structures. We find that CNN(1D)x2-Avg-FCLs and CNN(2D)x2-FCLs achieve the best performance among other CNNs, and their performance in terms of MSE, MAE, and R^2 are also given in Table V. Although we have selected the best CNNs, their performances are worse than GNNs though they are better than MLP under the same training setting that will be given later.

TABLE V
PERFORMANCE COMPARISON OF DIFFERENT METHODS FOR DCEM

Type	Method	Graph convolution dimension	MSE	MAE	R^2 Score
Traditional machine learnings	Gradient boosting		0.906	0.336	0.391
	Support vector machine		0.198	0.178	0.516
	Decision tree		0.186	0.157	0.569
	Random forest		0.139	0.113	0.732
Typical deep learnings	MLP		0.0644	0.0966	0.820
	CNN(1D)x2-Avg-FCLs	32, 64, 128	0.0545	0.0931	0.817
	CNN(2D)x2-FCLs	32, 64, 128	0.0530	0.0912	0.810
GNN	GCN	9	0.0580	0.0949	0.819
	GAT	9	0.0543	0.0918	0.821
	GEN	9	0.0524	0.0920	0.820
	GEN-GAT	3, 6	0.0513	0.0901	0.817
	GEN-GEN	3, 6	0.0477	0.0883	0.820
	GEN-GCN	**3, 6**	0.0445	0.0850	0.823
	GEN-GCN-WFD	3, 6	0.0474	0.0891	0.820
	GEN-GCN-GCN	3, 3, 3	0.0507	0.0891	0.821

C. Real-time DCEM Performance Comparison

With the generation-side and demand-side data collected before, we can train the GNN designed for real-time DCEM. In this case study, 90% of the dataset is the training set, and the remaining data is used as the validation set. Several GNNs following the

structures discussed in Section III.A are tested in this section to compare their performances and identify the best GNN structure.

1) GNN Hyper-parameters and Training Details

The GNN structures discussed in Section III.A are used here with the data collected in the previous section. To maintain uniform complexity across all neural networks, the sum of the graph convolution dimension and the number of layers remains consistent. In this case study, the MLP comprises 5 FCLs, and the other GNNs also consist of 5 layers, which include both convolution layers and fully connected layers. For example, if a GNN is composed of 2 graph convolution layers, this GNN will have 3 FCLs. In this paper, every FCL has 100 hidden units. Also, the one-layer GNNs have a convolution dimension of 9, whereas the two-layer GNNs have convolution dimensions of 3 and 6, respectively. This approach ensures that the neural networks have the same complexities for a fair comparison of their performances.

The training setting for both the GNN and other typical methods are the same: 1) a maximum training epoch of 1000; 2) a learning rate of 0.0001; and 3) a stopping criterion that prevents overfitting by halting training if there is no improvement in the validation loss after 10 epochs. Despite these basic training strategies, learning rate decay is also tested in our neural network training. However, we observe that the performance of dynamic learning rate adjustment is not stable and will not consistently yield better results.

To compare the accuracy of different methods, we also include traditional machine learning methods such as decision tree and random forest. The results of all methods will be discussed in the next section for a comprehensive evaluation of their performances alongside the proposed GNN approach.

2) The Role of Feature Downsize

The proposed GNN architecture has an intra-FCL for feature downsize. Feature downsize refers to the operation to reduce the nodal feature size that may be increased by graph convolution operation, which can be achieved by pooling, convolution, or other techniques. We use a FCL to achieve feature downsize because FCL can be trained faster under the same training setting without information loss like pooling. For example, the input nodal feature number is 2 in this paper. Then, the nodal features will be increased by the graph convolution operation. Then, the intra-FCL for feature downsize will reduce the nodal feature number to 2. The changes in nodal feature size of the proposed GNN architecture are given in Fig. 13(a).

Our experiments show that the feature downsize can improve the network performance and speed up the network training time. A typical GNN without feature downsize is built by putting the FCL for feature downsize after GCN, as shown in Fig. 13(b), which is denoted by GEN-GCN-WFD. The learning curves and test performance of GEN-GCN and GEN-GCN-WFD are shown in Fig. 14 and Table V. It is observed that the GNN with feature downsize will converge faster: the training time for GEN-GCN and GEN-GCN-WFD are 2949 s and 3325 s, respectively. GEN-GCN also outperforms GEN-GCN-WFD in terms of MSE, MAE, and R^2.

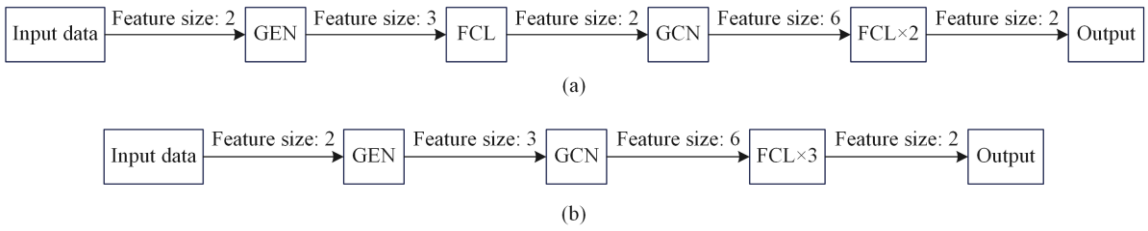


Fig. 13. Feature size details. (a) GEN-GCN. (b) GEN-GCN-WFD.

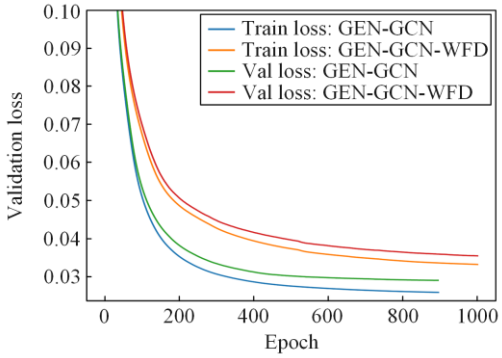


Fig. 14. Learning curves of GEN-GCN and GEN-GCN-WFD.

3) Comparison Between GNN and Other Typical Methods

To assess the effectiveness of graph convolution layers, we construct various GNNs following the structure in Figs. 6 and 7. These GNNs consist of graph convolution layers of GCN, GAT, and GEN, along with MLP layers. We then compare the single-layer GNNs and multi-layer GNN with MLP, CNNs, and other traditional machine learning methods in terms of learning curve, MSE, MAE, and R^2. The performance of different methods and the corresponding learning curves are provided in Table V and Fig. 15.

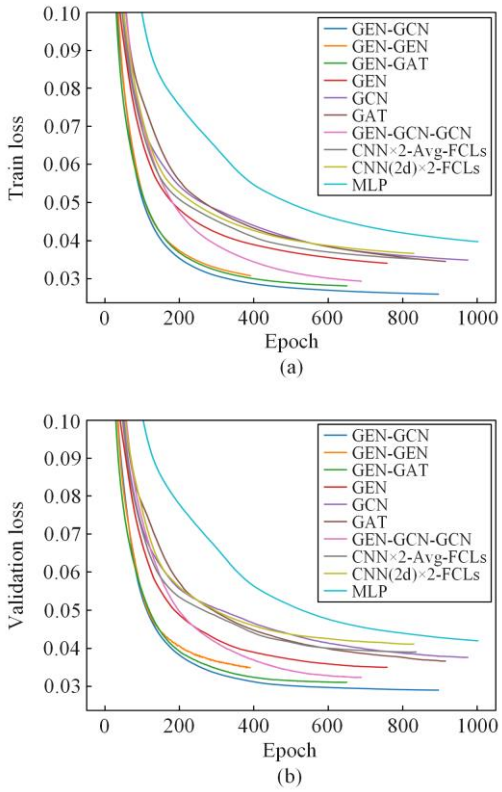


Fig. 15. Comparison of learning curves. (a) Training loss. (b) Validation loss.

As shown in Table V, MLP exhibits clear advantages over traditional machine learning methods, including decision tree (DT) and random forest (RF). CNNs are better than MLP. GNNs, on the other hand, demonstrate a notable improvement compared with MLP and single-layer GNN. From Fig. 15, it is observed that single-layer GNNs have smaller losses in the training set and validation set than MLP. Single-layer GNNs also have lower validation losses than CNNs. It is also noticeable that some CNNs have better performances than single-layer GCN and GAT. On the other hand, the two-layer GNNs demonstrate clearer advantages over other methods, including CNNs. According to Fig. 15, two-layer GNNs have lower training curves and validation curves, which is also reflected in Table V.

4) Graph Convolution Comparison

To further exploit the potential of GEN, we also investigate other graph convolution operations, including GAT, GCN, and GEN for single-layer and multi-layer GNNs. In particular, the performance of GEN is the best among single-layer GNN, indicating the potential of GEN. It is also noteworthy that single-layer GEN does not show overwhelming superiority over other GNNs on the test set: the MAE of GEN is higher than that of GAT, and that of GEN is lower than that of GAT. After analyzing the graph convolution's performance with single-layer GNNs, we also integrate other graph convolution layers into the original single-layer GEN by constructing two-layer GNNs depicted in Fig. 7. It is observed that two-layer GNNs outperform single-layer GNNs in the view of learning curve. Moreover, GEN-GCN achieves the smallest loss on the training set and validation set, and GEN-GAT also has a good performance second only to GEN-GCN. Based on the results presented in Table V, the performance of GEN-GCN is still the best, but GEN-GAT's performance is worse than GEN-GEN though GEN-GAT has a lower train loss and validation loss. GEN-GAT also takes more time in the training. The training times for GEN-GAT, GEN-GCN, and GEN-GEN are 5170 seconds, 2949 seconds, and 2645 seconds. In this case, GAT may not be suitable for real-time DCEM due to its unimpressive performance and long training time.

Afterward, we construct a deeper GNN network comprising three convolution layers with GEN and two GCN following the same GNN structure in Fig. 7. In particular, the structure of the three-layer GNN is GEN-GCN-GCN, and MLP is applied between GEN and GCN, as well as GCN and GCN. However, the performance of the deeper GEN-GCN-GCN network is degraded compared to the two-layer networks, which is reflected in the learning curves and test performance in Fig. 15 and Table V. GEN-GCN-GCN takes 3670 seconds to be trained, longer than GEN-GCN. Given the training time and performance of GEN-GCN-GCN, GEN-GCN is more suitable for real-time DCEM compared with the three-layer GNN.

We also experiment with residual connections, and slight degradation and faster convergence are found. Since residual connections are primarily used to facilitate the training of deeper networks, their impact may be limited for two-layer GNNs.

To further demonstrate the advantages of GNN over other methods, we calculate the MSE and overall relative errors (ORRs) of NCE and NCEI of every bus using GEN-GCN as the GNN model and other machine learning methods. The ORR denoted by ε_{sum} given in (27) is the relative error of the sum of NCE as given below, which can evaluate the prediction bias, where ω_i is the difference between the predicted value and actual values; and η_i is the actual value. If ORR is large, the model may produce overall higher or lower results, which are unfair for all stakeholders.

$$\varepsilon_{\text{sum}} = \frac{\sum_{i=0}^{n_g} \omega_i}{\sum_{i=0}^{n_g} \eta_i} \times 100\% \tag{27}$$

As depicted in Fig. 16, GNN demonstrates superior performance in terms of MSE and ORR for the majority of nodes. A specific example in Fig. 17 provides a daily-based real-time DCEM example for bus 20 using GNN, MLP, and other machine learning methods, further illustrating the advantage of GNN intuitively. The prediction errors of GNN are smaller compared with other methods for both NCE and NCEI. By comparing

GNN with other machine learning methods, it is observed that GNNs are more effective in accurately estimating real-time DCEM in power systems, realizing more accurate real-time DCEM and less potential bias.

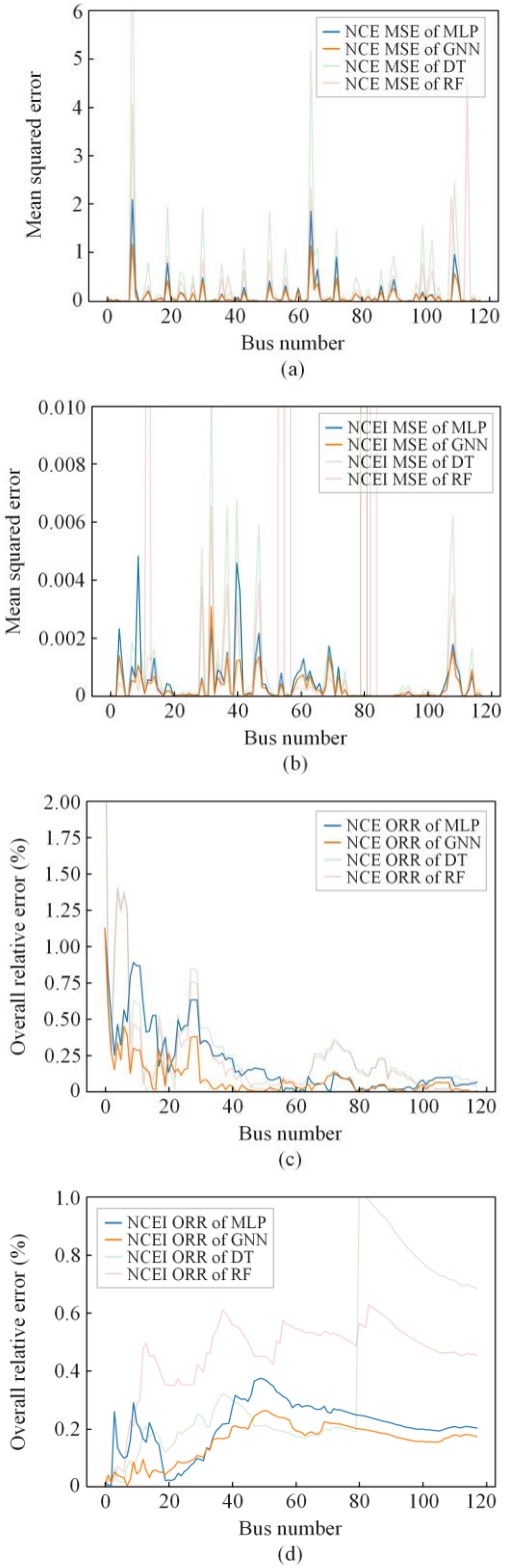


Fig. 16. GNN, MLP, and other machine learning methods' performances in terms of MSE and ORR. (a) MSE of NCE. (b) MSE of NCEI. (c) ORR of NCE. (d) ORR of NCEI.

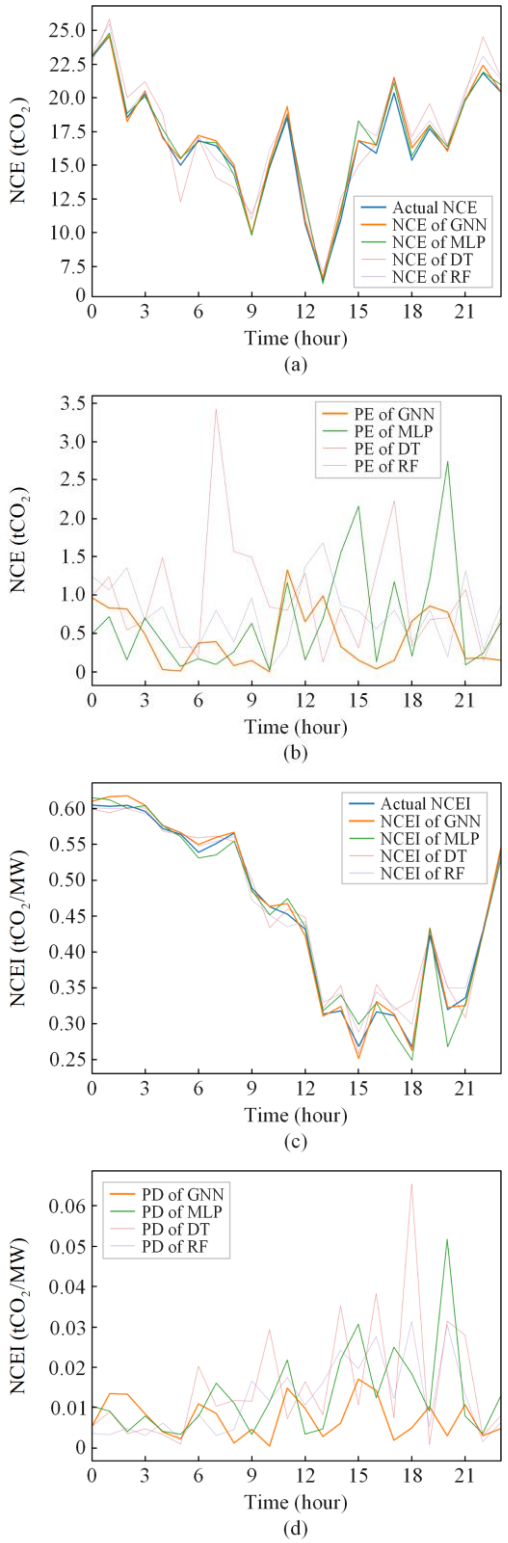


Fig. 17. Real-time DCEM results comparison on the daily basis for GNN, MLP, DT, and RF. (a), (b) NCE. (c), (d) NCEI.

D. Discussions on the Carbon Emission Patterns

Seasonal patterns in NCEIs, as well as a negative correlation between NCEI and RPP, are observed through our experiments. Specifically, the NCEI tends to become larger in winter and lower in summer, and

NCEIs are lower when the RPP is larger. In this way, the RPP in winter is relatively small and reaches its peak value in summer.

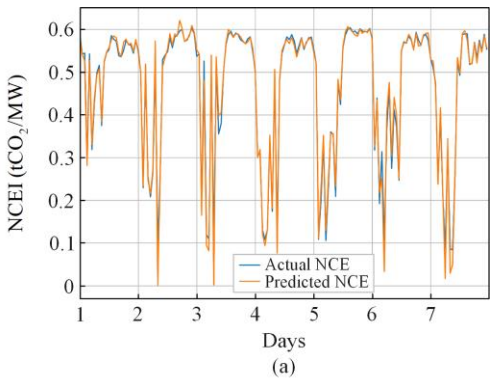
Bus 20 and bus 60, that are far apart in the IEEE 118-bus system, are selected to demonstrate the seasonal pattern. From Table VI, the NCEIs of buses 20 and 60 in December are larger than those in June. The variations in NCEIs originate from the changes in power demand and renewable power generation, which can be evaluated by RPP. Therefore, there is a negative correlation between RPP and NCEI. As shown in Table VI, in June, the RPP reaches its highest level, leading to the lowest NCEI. In December, the value of RPP is the lowest, and the corresponding NCEI is the highest.

TABLE VI
RPP AND NCEI OF BUS 20 AND BUS 60

Time period	Average RPP	NCEI of bus 20 (tCO_2/MW)	NCEI of bus 60 (tCO_2/MW)
March	0.161	0.432	0.364
June	0.169	0.427	0.341
September	0.157	0.460	0.475
December	0.141	0.490	0.537
Overall	0.157	0.449	0.432

The variations in the average NCEIs for bus 20 and bus 60 also match the carbon emission pattern of the whole system. In fact, the average NCEIs of the system in Q1, Q2, Q3, Q4, and the whole year are 0.541, 0.537, 0.544, 0.557, and 0.545 tCO_2/MW, respectively.

Similar to power demand patterns, the NCE and NCEI are sometimes periodic and random in some cases. Typical weekly-based examples of periodic DCE are given in Figs. 18(a) and (b), the peaks and valleys of which can be easily observed. In contrast, periodic patterns are hard to identify for those in Figs. 18(c) and (d). The predictions for non-periodic data are usually more challenging than periodic data if temporal-related components are involved. The proposed real-time DCEM method, on the contrary, can deal with both periodic and non-periodic data effectively because only spatial connections are considered. In particular, the prediction MAEs of Figs. 18(a)–(d) are 0.0115, 0.560, 0.008 62, and 0.465, respectively.



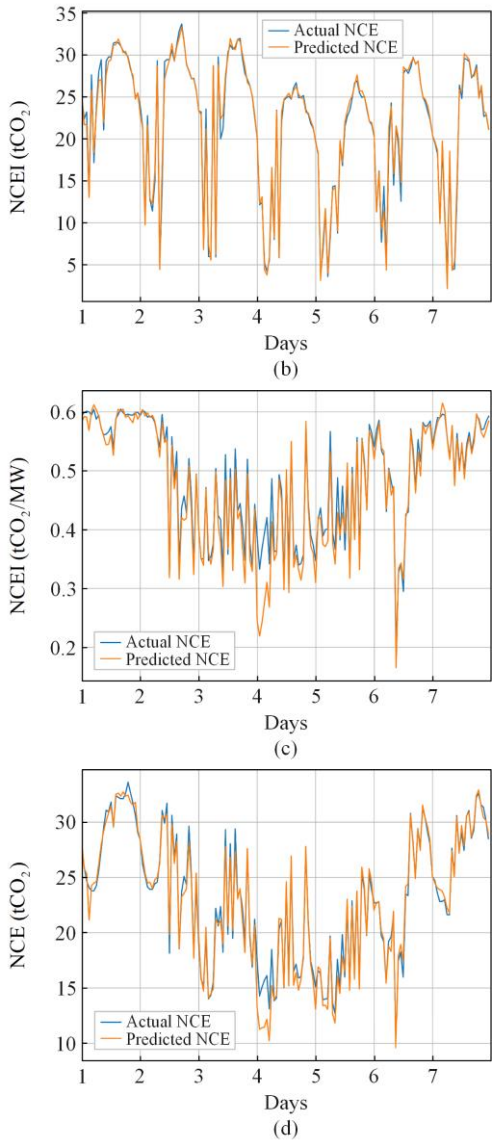


Fig. 18. Typical examples of periodic DCE. (a), (b) Periodic DCE. (c), (d) Non-periodic DCE.

V. CONCLUSION

This paper introduces a two-stage real-time DCEM method for low-carbon power systems. In the first stage, the power flow data of a power system is collected. Subsequently, the NCE and NCEI are calculated using the CEF calculation model, which are then used as the demand-side data, along with power injections and GCEF, to train a GNN for real-time DCEM in the second stage. The GNN can determine DCE with real-time available generation-side data only, thus realizing real-time DCEM.

A case study using the IEEE 118-bus power system is presented to demonstrate the accuracy and effectiveness of the proposed real-time DCEM method. In particular, a low-carbon power system scenario is generated with the day-ahead power generation planning method considering the CEQ price, which prioritizes clean energy,

and the power flow data of that is collected to generate the generation-side data and demand-side data to enable the data-driven method. A variety of GNN structures for real-time DCEM are investigated to identify the best GNN. GNNs and other machine learning methods are also compared, demonstrating the advantages of GNN for real-time DCEM in terms of different metrics. In this case study, seasonal regularities and the correlation between RPP and NCEI are observed. Moreover, the proposed method can effectively deal with carbon emission patterns regardless of whether there are regularities in the pattern.

The proposed real-time DCEM method opens the possibility for implementing the CEQ policy in the future, encouraging the widespread adoption of renewable power and driving the transition towards next-generation low-carbon power systems, which holds promising prospects for achieving carbon emission peak and neutrality in the power sector.

The proposed GNN-based method for real-time DCEM also has certain limitations. GNNs rely on graph representations that explicitly capture the node connection. Therefore, the proposed method requires more detailed information about the power system topology. If the power system topology is changed due to maintenance, the GNN may need to be retrained. In addition, GNNs incorporate an iterative message-passing mechanism that is computationally complex, which may also decelerate the training process. Fortunately, the advancements in power system monitoring technology and computation hardware can mitigate the potential problems, and the offline training of the proposed will not affect its online application, thus making the proposed method less susceptible to long network training time and retraining.

The proposed method can be further refined with edge features, and new graph convolution layers can be designed to improve GNN performance and reduce training time.

APPENDIX A

The standard model to perform power system optimization is indicated in (A1), where $f(\boldsymbol{x})$ is the objective function; $g(\boldsymbol{x})$ and $h(\boldsymbol{x})$ are the power flow constraints; $\underline{x}$ and $\overline{x}$ are the constrained lower and upper values; $\boldsymbol{x}$ is an optimization vector consisting of $n_{\mathrm{g}} \times 1$ generator real and reactive power injections (denoted by $\boldsymbol{P}_{\mathrm{G}}$ and $\boldsymbol{Q}_{\mathrm{G}}$, respectively) and $n_{\mathrm{b}} \times 1$ vectors of voltage magnitude and angles (denoted by $\boldsymbol{V}$ and $\boldsymbol{\theta}$), as given in (A2).

$$\begin{cases} \mathrm{Min} f(\boldsymbol{x}) \\ g(\boldsymbol{x}) = 0 \\ h(\boldsymbol{x}) \leqslant 0 \\ \underline{\boldsymbol{x}} < \boldsymbol{x} < \overline{\boldsymbol{x}} \end{cases} \tag{A1}$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{P}_{\mathrm{G}} \\ \boldsymbol{Q}_{\mathrm{G}} \\ \boldsymbol{V} \\ \boldsymbol{\theta} \end{bmatrix} \tag{A2}$$

Equation (A3) is the objective function, where W_i refers to the ith element in $\boldsymbol{W}_{\mathrm{overall}}$; while P_{G}^i and Q_{G}^i are the active power and reactive power output of the ith generator, respectively.

$$f(P_{\mathrm{G}}, Q_{\mathrm{G}}) = \sum_{i=1}^{Y} W_i \times (P_{\mathrm{G}}^i + Q_{\mathrm{G}}^i) \tag{A3}$$

Then, we can model an $n_{\mathrm{b}} \times n_{\mathrm{b}}$ network admittance matrix, $\boldsymbol{Y}_{\mathrm{net}}$, with the $n_{\mathrm{bn}} \times n_{\mathrm{b}}$ "from end" and "to end" admittance matrix $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$. Note that n_{bn} denotes the number of branches. $\boldsymbol{Y}_{\mathrm{net}}$ can be computed as shown in (A4) by $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, and the corresponding connection $n_{\mathrm{bn}} \times n_{\mathrm{b}}$ matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, which are defined according to network connections: for each branch i connection bus a to bus b, (i,a)th and (i,b)th elements of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are 1, and other elements are zero.

$$\boldsymbol{Y}_{\mathrm{net}} = \boldsymbol{C}_1^{\mathrm{T}} \boldsymbol{Y}_1 + \boldsymbol{C}_2^{\mathrm{T}} \boldsymbol{Y}_2 \tag{A4}$$

The complex power injections can be therefore computed in terms of the $n_{\mathrm{b}} \times 1$ complex bus voltage, $\boldsymbol{V}$, as:

$$\boldsymbol{S}_{\mathrm{bus}}(\boldsymbol{V}) = \mathrm{diag}\{\boldsymbol{V}\} \boldsymbol{Y}_{\mathrm{net}}^* \boldsymbol{V}^* \tag{A5}$$

Equation (A5) can be expressed by active and reactive aspects with voltage angles, and the bus injection can form equations with injections of generators and loads. Since each node has two constraints for active power and reactive power, for an n-bus system, there are $2n$ equality constraints as:

$$\begin{cases} G_{\mathrm{P}}(\boldsymbol{V}, \boldsymbol{\theta}, \boldsymbol{P}_{\mathrm{G}}) = P_{\mathrm{net}}(\boldsymbol{V}, \boldsymbol{\theta}) + \boldsymbol{P}_{\mathrm{d}} - \boldsymbol{P}_{\mathrm{G,inj}} \\ Q_{\mathrm{P}}(\boldsymbol{V}, \boldsymbol{\theta}, \boldsymbol{P}_{\mathrm{G}}) = Q_{\mathrm{net}}(\boldsymbol{V}, \boldsymbol{\theta}) + \boldsymbol{Q}_{\mathrm{d}} - \boldsymbol{Q}_{\mathrm{G,inj}} \end{cases} \tag{A6}$$

where $P_{\mathrm{net}}(\boldsymbol{V}, \boldsymbol{\theta})$ and $Q_{\mathrm{net}}(\boldsymbol{V}, \boldsymbol{\theta})$ denote the active and reactive power injection from the network; $\boldsymbol{P}_{\mathrm{d}}$, $\boldsymbol{Q}_{\mathrm{d}}$, $\boldsymbol{P}_{\mathrm{G,inj}}$, and $\boldsymbol{Q}_{\mathrm{G,inj}}$ are the known active and reactive injections from loads and generators, respectively; the expressions of $\boldsymbol{P}_{\mathrm{G,inj}}$ and $\boldsymbol{Q}_{\mathrm{G,inj}}$ are given in (A7):

$$\begin{cases} \boldsymbol{P}_{\mathrm{G,inj}} = \boldsymbol{C}_{\mathrm{G}} \boldsymbol{P}_{\mathrm{G}} \\ \boldsymbol{Q}_{\mathrm{G,inj}} = \boldsymbol{C}_{\mathrm{G}} \boldsymbol{Q}_{\mathrm{G}} \end{cases} \tag{A7}$$

where $\boldsymbol{C}_{\mathrm{G}}$ is an $n_{\mathrm{b}} \times n_{\mathrm{g}}$ connection matrix such as $\boldsymbol{C}_{\mathrm{G},xy}$, the (x,y)th element of $\boldsymbol{C}_{\mathrm{G}}$ is 1 if there is a generator y located at node x.

The inequality constraints $h(x)$ can be written as:

$$\begin{cases} \left| F_1(\boldsymbol{\theta}, \boldsymbol{V}) \right| - F_{\max} \leqslant 0 \\ \left| F_2(\boldsymbol{\theta}, \boldsymbol{V}) \right| - F_{\max} \leqslant 0 \end{cases} \tag{A8}$$

where $|F_1(\boldsymbol{\theta},\boldsymbol{V})|$ and $|F_2(\boldsymbol{\theta},\boldsymbol{V})|$ denote the magnitude of power flowing for the "from" end and "to" end in a transmission line; and $F_{\max}$ is the maximum power allowed in a branch. Note that $|F_1(\boldsymbol{\theta},\boldsymbol{V})|$ and $|F_2(\boldsymbol{\theta},\boldsymbol{V})|$ can be written in the form of apparent power, active power, and reactive power. In this paper, the flow in (A8) is regarded as apparent flow. Constraints imposed on generators in a power system, including generator outputs, generator voltages, and voltage reference angles, can be written in terms of inequality constraints in (A9), (A10), (A11), and (A12), respectively.

$$P_{Gi}^{\min} \leqslant P_{Gi} \leqslant P_{Gi}^{\max},\ i=1,\cdots,n_{\mathrm{g}} \tag{A9}$$

$$Q_{Gi}^{\min} \leqslant Q_{Gi} \leqslant Q_{Gi}^{\max}, i=1,\cdots,n_{\mathrm{g}} \tag{A10}$$

$$V_{Bi}^{\min} \leqslant V_{Gi} \leqslant V_{Bi}^{\max}, i=1,\cdots,n_{\mathrm{b}} \tag{A11}$$

$$\theta_i^{\mathrm{ref}} \leqslant \theta_i \leqslant \theta_i^{\mathrm{ref}}, i \in I^{\mathrm{ref}} \tag{A12}$$

For the model introduced above, mathematical methods such as mix-integer linear programming can solve the optimization problem.

ACKNOWLEDGMENT

The authors would like to thank the supports of Undergraduate Research and Innovation Scheme (URIS) of The Hong Kong Polytechnic University, Matics Globalink Research Internship Program, and the Department of Applied Mathematics, University of Waterloo.

AUTHORS' CONTRIBUTIONS

Zhu Zixuan: conceptualization, investigation, methodology, simulation, and full-text writing. Ruoheng Wang: conceptualization, investigation, methodology, simulation, and review & editing. Siqi Bu: conceptualization, investigation, supervision, funding acquisition, project administration, and review & editing. Roberto Guglielmi: review, and supervision. All authors read and approved the final manuscript.

FUNDING

This work is supported in part by the National Natural Science Foundation of China (No. 52077188), and in part by the PolyU for the PReCIT Seed Project (No. 1-CE16).

AVAILABILITY OF DATA AND MATERIALS

Please contact the corresponding author for data material request.

DECLARATIONS

Competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

AUTHORS' INFORMATION

Zhu Zixuan received the B.E. degree from The Hong Kong Polytechnic University with the first class honor. He is currently a PhD student at the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. His research interests include reinforcement learning, deep learning, large language model, and their application in power system integrated with renewable energy and electrified transportation system.

Ruoheng Wang received the B.E. degree from Wuhan Institute of Technology, the M.E. degree from Huazhong University of Science and Technology, and the Ph.D. degree from The Hong Kong Polytechnic University. He is currently a post-doctoral fellow with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. His research interests include deep learning, reinforcement learning, and their applications in power systems integrated with renewable energy and electrified transportation systems.

Siqi Bu received the Ph.D. degree from the electric power and energy research cluster, The Queen's University of Belfast, Belfast, U.K., where he continued his postdoctoral research work before entering industry. Then he was with the National Grid UK as an experienced UK national transmission system planner and operator. He is an associate professor and associate head with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, and associate director of Research Centre for Grid Modernisation. He is also a chartered engineer with UK Royal Engineering Council, London, U.K. and a fellow of IET. His research interests include power system stability, operation and economics considering renewable energy integration, smart grid application and transport electrification. He is an assistant editor of *Protection and Control of Modern Power Systems*, *IEEE Transactions on Power Systems*, *IEEE Transactions on Consumer Electronics*, *IEEE Power Engineering Letters*, *IEEE Access*, *IEEE Open Access Journal of Power and Energy*, *CSEE Journal of Power and Energy Systems*, *Journal of Modern Power Systems and Clean Energy*, and *Advances in Applied Energy*.

Roberto Guglielmi is an assistant professor in the Department of Applied Mathematics at the University of Waterloo. His research interests aim at developing control and optimization strategies for complex dy-

namics, described by ordinary or partial differential systems of evolution, with applications to engineering and life sciences. Main areas of applications range from environment and sustainability, health and energy. He is currently a member of the Technical Committee 2.6—Distributed Parameter Systems—of the International Federation of Automatic Control (IFAC), and sit on the Advisory committee of the Cybersecurity and Privacy Institute (CPI) of the University of Waterloo. Besides, he is a member of the University of Waterloo's Water Institute (WI), the Waterloo Climate Institute (WCI), the Waterloo Institute for Sustainable Aeronautics (WISA), the Waterloo Data and Artificial Intelligence Institute (Waterloo.ai), and the Waterloo Institute for Sustainable Energy (WISE).

REFERENCES

[1] N. Stern, "Part I—climate change—our approach," in *The Economics of Climate Change: The Stern Review*, Cambridge, UK: Cambridge University Press, 2007.

[2] K. Wang, S. Mei, and W. Wei *et al*., "Fast transient security assessment based on graph neural networks," *Power System Protection and Control*, vol. 51, no. 1, pp. 43-51, Jan. 2023. (in Chinese)

[3] J. Sjödin and S. Grönkvist, "Emissions accounting for use and supply of electricity in the Nordic market," *Energy Policy*, vol. 32, pp.1555-1564, Sept. 2004.

[4] B. Zeng, J. Zhang, and X. Yang *et al*., "Integrated planning for transition to low-carbon distribution system with renewable energy generation and demand response," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1153-1165, Jun. 2014.

[5] F. Ahsan, N. H. Dana, and S. K. Sarker *et al.*, "Data-driven next-generation smart grid towards sustainable energy evolution: techniques and technology review," *Protection and Control of Modern Power Systems*, vol. 8, no. 3, pp. 1-42, Jul. 2023.

[6] X. Fu, X. Wu, and C. Zhang *et al*., "Planning of distributed renewable energy systems under uncertainty based on statistical machine learning," *Protection and Control of Modern Power Systems*, vol. 7, no. 1, pp. 1-27, Jan. 2022.

[7] Y. Li, M. K. Lim, and J. Hu *et al*., "Investigating the effect of carbon tax and carbon quota policy to achieve low carbon logistics operations," *Resources*, *Conservation and Recycling*, vol. 154, Mar. 2020.

[8] R. Liu and Z. Jing, "The impacts of introducing carbon prices in the electricity market under China's carbon neutrality goal," in *2021 International Conference on Power Grid System and Green Energy (PGSGE 2021)*, Jan. 2021, pp. 1-4.

[9] C. Wang, Z. Lu, and Y. Qiao, "A consideration of the wind power benefits in day-ahead scheduling of wind-coal intensive power systems," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 236-245, Feb. 2013.

[10] M. Lenzen and J. Munksgaard, "Energy and CO_2 life-cycle analyses of wind turbines—review and applications," *Renewable Energy*, vol. 26, no. 3, pp. 339-362, Jul. 2002.

[11] D. Weisser, "A guide to life cycle green-house gas (GHG) emission fromelectric supply technologies," *Energy*, vol. 32, no. 9, pp. 1543-1559, Sept. 2007.

[12] C. Kang, T. Zhou, and Q. Chen *et al.*, "Carbon emission flow in networks," *Scientific Reports*, vol. 2, no. 1, Jun. 2012.

[13] B. Li, Y. Song, and Z. Hu, "Carbon flow tracing method for assessment of demand side carbon emissions obligation," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 4, pp. 1100-1107, Oct. 2013.

[14] C. Kang, T. Zhou, and Q. Chen *et al*., "Carbon emission flow from generation to demand: a network-based model," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2386-2394, Sept. 2015.

[15] T. H. Ruggles, D. J. Farnham, and D. Tong *et al*., "Developing reliable hourly electricity demand data through screening and imputation," *Scientific Data*, vol. 7, no. 1, pp. 145-155, Jan. 2020.

[16] B. L. H. Nguyen, T. V. Vu, and T. T. Nguyen *et al*., "Spatial-temporal recurrent graph neural networks for fault diagnostics in power distribution systems," *IEEE Access*, vol. 11, pp. 46039-46050, May 2023.

[17] F. Shi, J. Wu, and Y. Wang *et al*., "Integrated advance assessment of power system transient voltage and transient angle stability based on two-stage ensemble spatio-temporal graph neural network," *Measurement*, vol. 221, Nov. 2023.

[18] T. B. Lopez-Garcia and J. A. Domínguez-Navarro, "Power flow analysis via typed graph neural networks," *Engineering Applications of Artificial Intelligence*, vol. 117, Jan. 2023.

[19] B. Chen, Q. Wu, and M. Li *et al*., "Detection of false data injection attacks on power systems using graph edge-conditioned convolutional networks," *Protection and Control of Modern Power Systems*, vol. 8, no. 1, pp. 1-12, Jan. 2023.

[20] W. Liao, B. Bak-Jensen, and J. R. Pillai *et al*., "A review of graph neural networks and their applications in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 2, pp. 345-360, Mar. 2022.

[21] Z. Jing and F. Wen, "Discussion on the proving of proportional sharing principle in electricity tracing method," in *2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific*, Dalian, China, Aug. 2005, pp. 1-5.

[22] Z. Ji, C. Kang, and Q. Chen *et al.*, "Low-carbon power system dispatch incorporating carbon capture power plants," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4615-4623, Nov. 2013.

[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv: 1609.02907, Feb. 2017. Available: https://doi.org/10.48550/arXiv.1609.02907

[24] P. Veličković, G. Cucurull, and A. Casanova *et al*., "Graph attention networks," arXiv:1710.10903, Feb. 2018. Available: https://doi.org/10.48550/arXiv.1710.10903

[25] G. Li, C. Xiong, and A. Thabet *et al*., "Deeper GCN: all you need to train deeper GCNs," arXiv:2006.07739, Jun.

2020. Available: https://doi.org/10.48550/arXiv.2006.07739

[26] V. P. Dwivedi, C. K. Joshi, and A. T. Luu *et al*., "Benchmarking graph neural networks," *Journal of Machine Learning Research*, vol. 24, no. 43, pp. 1-48, 2023.

[27] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29*, California, USA: Neural Information Processing Systems Foundation, 2016.

[28] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129-150, Apr. 2011.

[29] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12-19, Feb. 2011.

[30] IEA, (2020), "Projected costs of generating electricity 2020," IEA, Paris, France [Online]. Available: https://www.iea.org/reports/projected-costs-of-generating-electricity-2020

[31] F. Wiese, I. Schlecht, and W. Bunke *et al.*, "Open power system data–frictionless data for electricity system modelling," *Applied Energy*, vol. 236, pp. 401-409, Feb. 2019.

[32] Elia Open Data Portal (accessed July 31, 2023). Available: https://opendata.elia.be/explore/

[33] H. Ming, B. Xia, and K. Y. Lee *et al*., "Prediction and assessment of demand response potential with coupon incentives in highly renewable power systems," *Protection and Control of Modern Power Systems*, vol. 5, no. 1, pp. 1-14, Jan. 2020.

[34] A. Agga, A. Abbou, and M. Labbadi *et al*., "CNN-LSTM: an efficient hybrid deep learning architecture for predicting short-term photovoltaic power production," *Electric Power Systems Research*, vol. 208, Jul. 2022.

[35] S. Ghimire, R. C. Deo, and D. Casillas-Pérez *et al*., "Deep learning CNN-LSTM-MLP hybrid fusion model for feature optimizations and daily solar radiation prediction," *Measurement*, vol. 202, Oct. 2022.

[36] P. Rai, N. D. Londhe, and R. Raj, "Fault classification in power system distribution network integrated with distributed generators using CNN," *Electric Power Systems Research*, vol. 192, Mar. 2021.

[37] L. Zhu, K. Long, and Z. Dong *et al*., "Two-dimensional convolution-based power system reliability assessment," *Energy Reports*, vol. 9, pp. 472-478, May 2023.