International Journal of Innovative
Computing, Information and Control
Volume **22**, Number **1**, February **2026**

ICIC International ©2026 ISSN 1349-4198
pp. **31–46**

HYBRID TRANSFORMER-CNN ARCHITECTURE WITH ATTENTION POOLING FOR ACCURATE PREDICTION OF DYNAMIC ELECTRIC CARBON FACTORS IN POWER SYSTEMS

Jianlin Tang[1,2,*], Yong Xiao[1,2], Xiaoming Lin[1,2] and Bin Qian[1,2]

[1]CSG Electric Power Research Institute
No. 11, Kexiang Road, Huangpu District, Guangzhou 510663, P. R. China
{ xiaoyong; linxm4; qianbin }@csg.cn
*Corresponding author: tangjl2@csg.cn

[2]Guangdong Provincial Key Laboratory of Intelligent Measurement and Advanced Metering of Power Grid
No. 11, Kexiang Road, Huangpu District, Guangzhou 510663, P. R. China

Received April 2025; revised July 2025

Abstract. *This paper proposes a hybrid Transformer-CNN model enhanced with an attention pooling mechanism for accurately predicting dynamic electric carbon factors on the user side of power grids. The model integrates the temporal sequence modeling capabilities of the Transformer with the spatial feature extraction strengths of CNN, while the attention pooling component adaptively emphasizes relevant features to improve robustness under dynamic power grid conditions. The model is evaluated on both the IEEE 39-bus and 118-bus systems, under static and dynamic scenarios. In the 39-bus system, the proposed model achieves an error rate of 0.012, outperforming linear regression (0.055), decision trees (0.038), LSTM (0.020), and MLP (0.025). In the 118-bus system, the model maintains superior performance with an error rate of 0.015, compared to 0.048 (linear regression), 0.065 (decision trees), 0.023 (LSTM), and 0.029 (MLP). Under simulated dynamic grid conditions, the model achieves an error rate of 0.022, significantly outperforming other methods. These results confirm the model's strong accuracy, robustness, and generalization capability, supporting its potential for practical use in carbon emission management and policy-making in the power system domain.*
Keywords: Transformer, Convolutional neural network (CNN), Attention pooling, Dynamic electric carbon factor, Deep learning, Carbon emission prediction, Power systems

1. **Introduction.** The power system is the backbone of modern society; however, its excessive carbon emissions present significant threats to both the climate and the environment [1]. Among these issues, the carbon emissions on the user side of the power system have become increasingly prominent, primarily due to the dynamic and diverse nature of energy consumption patterns [2]. Traditional modeling techniques often fail to account for these spatiotemporal variations, resulting in inaccurate estimates of user-side carbon emissions. This challenge is exacerbated by the growing integration of new and distributed energy sources, underscoring the urgent need for more sophisticated computational models that can adapt to these dynamic changes [3].

In recent years, there has been a surge of interest in applying deep learning techniques to modeling spatiotemporal dynamic data within power systems. For instance, several studies have explored the use of long short-term memory (LSTM) networks and recurrent neural networks to capture the dynamic behavior of power systems, with varying

DOI: 10.24507/ijicic.22.01.31

degrees of success. [4] utilized regression models to derive marginal carbon emission factors and employed multi-layer perceptron (MLP) networks for short-term emission factor prediction. Another study [5] combined grey models with cubic exponential smoothing to predict CO_2 emissions, and a hierarchical machine learning approach, designed to predict grid carbon intensity over multiple days, was introduced in [6]. Further, a quadratic optimization algorithm was used to transform annual data into monthly intervals, and a time series autoregressive distributed lag model was developed for electric carbon analysis [7]. Additionally, a particle swarm optimization-extremely random tree regression model was proposed to forecast emission intensity in the Australian national electricity market [8], while multiple techniques, such as least absolute shrinkage and selection operator, principal component analysis, and support vector regression, were integrated for improved carbon emission forecasting [9]. Other research, such as differential evolution-grey wolf optimizer [10], the time-delay optimization optimizer [11], and online knowledge distillation [12], has focused on machine learning models like bidirectional long short-term memory and projection pursuit regression to improve the accuracy of regional carbon emission predictions. More recently, advanced models incorporating graph neural networks were proposed to predict carbon intensity in cross-border grids [13], providing real-time insights for grid management [14]. A dynamic pollution emission prediction method for combined heat and power systems was presented, utilizing a hybrid CNN-LSTM model integrated with an attention mechanism [15]. A novel approach for predicting carbon emissions was introduced, using a hybrid model based on LSTM and graph attention networks [16]. In addition, recent studies have explored innovative CNN-based models in other domains, such as using nighttime infrared images for human action analysis [17], using residual networks and dense-connected architectures for facial expression recognition [18], vehicle detection through infrared thermal imaging [19], and a hybrid CNN-transformer model for predicting ozone concentrations was proposed, achieving high accuracy and demonstrating the effectiveness of combining convolutional and transformer architectures in environmental monitoring [20]. These approaches underscore the versatility of CNN models and further highlight their potential applicability to dynamic carbon emission prediction in power systems.

While methods like [15] effectively integrate spatial and temporal features, its architecture is primarily designed for relatively stable co-generation systems and lacks scalability when applied to large-scale, dynamic power grids with complex user-side behaviors. Moreover, the attention mechanism in [15] operates within the LSTM substructure and does not fully exploit hierarchical spatial relationships inherent in power network topologies. In contrast, the present study advances the literature by introducing a Transformer-CNN hybrid model that separates temporal and spatial processing using dedicated modules and incorporates an attention pooling mechanism designed to selectively aggregate high-impact features across both dimensions. This enables the model to more accurately capture fine-grained, dynamic variations in user-side electric carbon factors under fluctuating conditions. Furthermore, unlike [15], the proposed method is validated on both the IEEE 39-bus and 118-bus systems under dynamic scenarios, demonstrating superior accuracy and generalization. Compared with [20] that applied Transformer-CNN models primarily for environmental data prediction tasks such as ozone concentration, this study tailors the Transformer-CNN architecture specifically for spatiotemporal modeling of user-side carbon emissions in power systems. It integrates power system topology and dynamic load features, enhancing temporal attention while preserving local spatial patterns, thus achieving more accurate and interpretable emission forecasts.

Despite these advancements, existing studies primarily focus on static models of overall carbon emissions, often neglecting the spatiotemporal dynamics of user-side emissions.

These models struggle to account for variations during peak and off-peak periods, seasonal fluctuations, and regional differences in energy consumption. While some studies have relied on simplified models or rule-based approaches, these methods are often inadequate for handling the complexities of real-world scenarios. Furthermore, although deep learning approaches have demonstrated success, they typically overlook the dynamic characteristics specific to user-side carbon emissions [21]. A significant gap in current research is the absence of a comprehensive deep learning model capable of accurately capturing the dynamic patterns of user-side carbon emissions across spatiotemporal dimensions.

To address this gap, the present study proposes a novel approach by integrating transformer and convolutional neural network (CNN) to more effectively capture the dynamic characteristics of user-side carbon emissions. The combination of the Transformer's ability to model long-range dependencies and CNN's strength in capturing local features, along with an attention-based pooling mechanism, enables the model to efficiently handle spatiotemporal data. The primary contributions of this work are as follows.

1) The development of a Transformer-CNN hybrid model that comprehensively captures the complex, dynamic behavior of user-side carbon emissions.
2) The integration of an attention-based pooling mechanism that enhances the model's focus on the most relevant features, improving prediction accuracy.
3) The successful application of this model to real-world power systems, resulting in more precise predictions of dynamic electric carbon factors within the user domain, offering new insights for carbon emission management.

The remainder of this paper is organized as follows. Section 2 introduces the Transformer-CNN model, outlining the problem definition, data representation, feature extraction, and the model's application in power systems. Section 3 discusses the attention pooling mechanism, elaborating on its principles and specific application to the modeling of user-side carbon emissions. Section 4 presents the experimental setup, performance evaluation metrics, and results, followed by a discussion on the model's strengths, limitations, and implications. Finally, Section 5 concludes the paper, summarizing the key findings.

2. **Dynamic Electric Carbon Factor Modeling and Data Encoding.**

2.1. **Problem definition.** In practical applications, the dynamic electric carbon factor on user side could be calculated using the subsequent formula:

$$\delta_{\mathrm{p}}^{i} = \frac{P_{\mathrm{g}}^{i} \cdot \delta_{\mathrm{g}}^{i} + \sum_{j \in \Omega_1} P_i^j \cdot \delta_{\mathrm{p}}^{j}}{D_i + \sum_{k \in \Omega_2} P_k^i} \tag{1}$$

where δ_{p}^{i}, δ_{p}^{j} ($\mathrm{kgCO_2/kWh}$) represent the carbon emission factors of nodes i and j, respectively; δ_{g}^{i} ($\mathrm{kgCO_2/kWh}$) is the carbon emission factor of the generator connected to node i; P_{g}^{i} (kW) denotes the active power of the generator connected to node i; P_i^j (kW), P_k^i (kW) represent the active power flow from node j to node i and from node i to node k, respectively. Ω_1 is the set of branch nodes flowing into node i, and Ω_2 is the set of branch nodes flowing out of node i; D_i (kW) represents the load at node i. The dynamic electric carbon factor $DCF_{i,t}$ indicates the ratio of the carbon emissions generated by the energy consumption at time step t of user i to the actual electricity consumption.

By processing Equation (1), we obtain

$$\left(D_i + \sum_{k \in \Omega_2} P_k^i\right) \cdot \delta_{\mathrm{p}}^{i} - \sum_{j \in \Omega_1} P_i^j \cdot \delta_{\mathrm{p}}^{j} = P_{\mathrm{g}}^{i} \cdot \delta_{\mathrm{g}}^{i} \tag{2}$$

According to the calculation principle of (2), the overall expression can be written as

$$\boldsymbol{A\delta} = \boldsymbol{E}_{\mathrm{g}} \tag{3}$$

$$A_{ij} = \begin{cases} D_i + \sum\limits_{k\in\Omega_2} P_k^i & (j = i) \\ -P_i^j & (j \in \Omega_1) \\ 0 & (\text{otherwise}) \end{cases} \tag{4}$$

$$\boldsymbol{E}_{\mathrm{g}} = \left[E_{\mathrm{g}}^1, \ldots, E_{\mathrm{g}}^i, \ldots, E_{\mathrm{g}}^n\right]^{\mathrm{T}} \tag{5}$$

$$E_{\mathrm{g}}^i = P_{\mathrm{g}}^i \times \delta_{\mathrm{g}}^i, \quad i = 1, 2, \ldots, n \tag{6}$$

where $\boldsymbol{\delta}$ (kgCO$_2$/kWh) is the carbon emission factor vector of each node of the power grid; $\boldsymbol{A}$ (kW) is the solution coefficient matrix, where $\boldsymbol{A} = [A_{ij}] \in \boldsymbol{R}^{n\times n}$; $\boldsymbol{E}_{\mathrm{g}}$ (kgCO$_2$/h) is the injected power vector for all generators and n denotes the number of system nodes.

Given that Equation (2) represents the node energy conservation equation, Equation (3) constitutes a set of n independent formulas. Formula (3) could be transformed into

$$\boldsymbol{\delta} = \boldsymbol{A}^{-1}\boldsymbol{E}_{\mathrm{g}} \tag{7}$$

By utilizing the above formula, it is possible to accurately estimate the carbon emissions generated by users' energy consumption at different times and locations, thereby providing critical reference data for carbon emission management in power systems. Additionally, the calculation of user-side dynamic electric carbon factors may involve considerations such as seasonal variations and regional differences, requiring more complex models for accurate modeling and prediction.

The problem can be formalized as the following function mapping:

$$DCF_{i,t} = f(X_{i,t}) \tag{8}$$

In the above mathematical definition, the following notations are introduced.

$i = 1, 2, \ldots, n$: User indices, where n is the total number of users.

$t = 1, 2, \ldots, T$: Time steps indices, with T being the overall count of time steps.

$X_{i,t}$: The node-related data for user i at time step t, typically depicted as a vector or matrix containing the power system node data for that user at that time.

$DCF_{i,t}$: The dynamic electric carbon factor for user i at time step t, usually a real number reflecting the ratio of carbon emissions from energy consumption to actual electricity usage at that time.

$f(\cdot)$: The function to be learned, used to predict the dynamic electric carbon factor. The specific form and parameters of this function will be determined during model design and training.

2.2. **Data representation and feature extraction.** For the prediction of dynamic electric carbon factors within the power grid's user domain, this study primarily uses relevant node information from the power system to forecast these factors. This node information includes power, voltage, and the topological structure between nodes. In this context, the data can be represented in the following aspects.

1) Node Data Representation:

For example, node power data reflects the power consumption at different time steps. This data can be used in its raw form or normalized to fall within a consistent numerical range. Assuming there are N nodes and T time steps, the power data at time step t for node i can be depicted as

$$P_{i,t} = \left[p_{i,t}^{(1)}, p_{i,t}^{(2)}, \ldots, p_{i,t}^{(m)}\right] \tag{9}$$

where, $p_{i,t}^{(m)}$ represents the power node i data in time step t. Others such as voltage $V_{i,t}$, are expressed by analogy with the above definition.

2) Node Topological Structure Information Representation:

Node topological structure information describes the connections between nodes, typically represented by metrics such as distances or connection types. This can be expressed using an adjacency matrix or adjacency list. Assuming we have N nodes, the adjacency matrix $\boldsymbol{A}$ is represented as

$$\boldsymbol{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{bmatrix} \tag{10}$$

where $a_{i,j}$ represents the connection connecting node i and node j.

Based on the above representation methods, the input of the model can be expressed as

$$X_{i,t} = [P_{i,t}, V_{i,t}, \ldots, \boldsymbol{A}] \tag{11}$$

In the data representation phase, it is crucial to carefully design the data processing workflow to guarantee both the validity and interpretability of the data. By employing the aforementioned data representation methods, we can effectively utilize the relevant node information within the power system and provide sufficient the model's predictions data. This approach ensures that the model has access to comprehensive and accurate information, which is essential for making reliable forecasts of dynamic electric carbon factors.

3. Transformer-CNN Model and Pooling of Attention Mechanism.

3.1. Transformer-CNN model design.

The Transformer-CNN model is a deep learning framework that combines Transformer and CNN components to handle sequential data by capturing both long-term dependencies and local features.

1) Transformer Module

The Transformer module, grounded in the self-attention mechanism, is designed to manage overall dependencies within sequential data. It mainly includes several layers of feed-forward neural networks and self-attention mechanisms.

Self-Attention Layer: The self-attention layer calculates every position's attention weights in the input sequence, allowing interactions and integration of information across different positions. Given an input sequence $x = \{x_1, x_2, \ldots, x_n\}$, where x_i represents the feature vector at position i, the result from the self-attention layer could be expressed as

$$\text{Attention}(X) = \text{softmax}\left(\frac{XW_q(XW_k)^T}{\sqrt{d_k}}\right)(XW_v) \tag{12}$$

where W_k, W_v and W_q represent the weight matrices for key, value, and query, respectively; d_k denotes the attention head dimension.

Feed-Forward Neural Network Layer: It receives the output from the attention layer and performs a nonlinear transformation via the fully connected layer and activation function in order to get the final feature representation.

2) CNN Module

The CNN module is a neural network structure specialized in processing local features through convolution and pooling operations for extracting spatial details from the data. It mainly consists of convolutional layers, pooling layers, and activation functions.

Convolutional Layer: It performs feature extraction from the input data by applying convolution operations with filters (also known as kernels) and subsequently applying

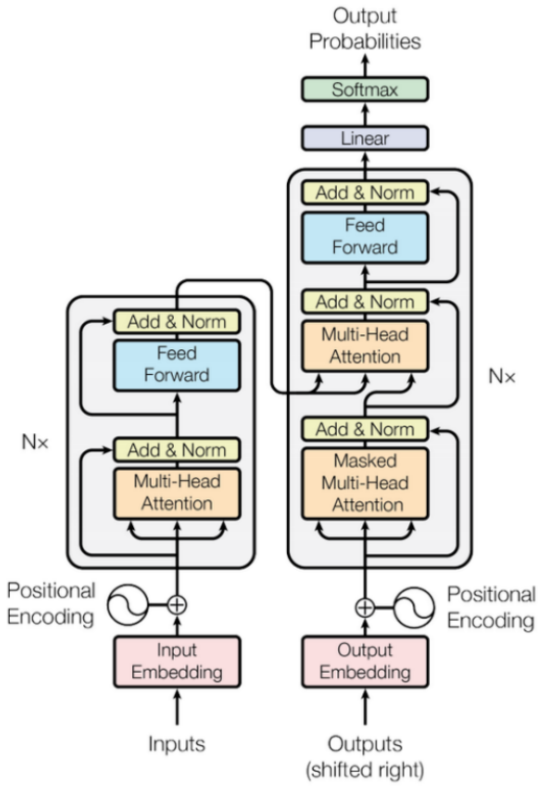


FIGURE 1. Basic structure principle of Transformer

activation functions. Given an input sequence $x = \{x_1, x_2, \ldots, x_n\}$ and a set of filters, the result produced by the convolutional layer can be formulated as

$$\mathrm{Conv}(X) = \mathrm{ReLU}(X * W + b) \tag{13}$$

Here, $*$ denotes the convolution operation, while b and W signify the bias term and convolutional kernel, respectively.

Pooling Layer: Pooling layer reduces the feature dimension and retains the main information by pooling the output of the convolutional layer. Typical pooling methods include max pooling and average pooling.

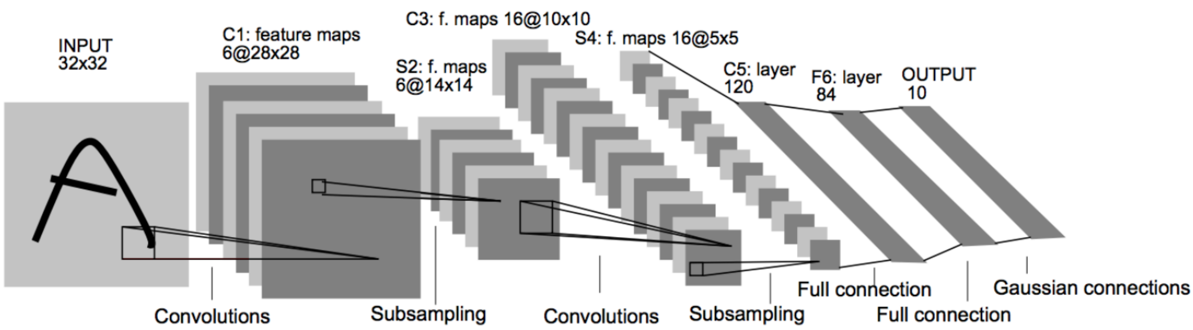


FIGURE 2. CNN network structure

3) Integration Module:

The Transformer-CNN model integrates the outputs from the Transformer and CNN modules using methods such as concatenation or weighted sum. This paper uses a weighted sum approach to combine features. The final integrated feature representation is subsequently fed into the model's output layer for classification or regression forecasting.

When using weighted sum for feature integration, each module's output is assigned a weight, and these outputs are combined according to these weights. The weights can be fixed, learned, or determined by other methods. In this paper, a fixed weight is assigned to each module's output, with the constraint that the sum of all weights equals 1. This approach maintains a balanced weight distribution, preventing any single module's output from disproportionately affecting the overall result.

Assuming the output of the Transformer module is T and the output of the CNN module is C, the integrated output I can be calculated using the weighted sum formula:

$$I = \alpha \cdot T + (1 - \alpha) \cdot C \tag{14}$$

where α is the weight for the Transformer module's output and $(1-\alpha)$ is the weight for the CNN module's output. By adjusting α, the relative importance of the outputs from the two modules can be controlled. If the goal is to have the model learn the weights automatically, the weights can be treated as learnable parameters, which are optimized through training. This allows the model to adjust the significance of each module's output based on data characteristics, enhancing adaptability to various tasks and data distributions.

Below is a detailed description of the Transformer-CNN model's working mechanism, including the data processing flow, forward propagation process, and loss function definition.

1) Input Data Processing Flow

At the input end of the model, we first convert the raw node data into tensor form and perform normalization or standardization as preprocessing. Assume the input sequence represents $x = \{x_1, x_2, \ldots, x_n\}$, where x_i represents the feature representation at position i. The data processing flow is as follows.

a) Data Representation: Convert the raw node data into tensor form to facilitate input into the model for processing.

b) Normalization: Apply normalization to the input data to constrain the value range within a suitable interval. This step helps accelerate model convergence and improve model stability.

2) Model Forward Propagation Process

The forward propagation process of the model can be divided into several steps: processing by the Transformer module, processing by the CNN module, feature integration, and model prediction.

a) Transformer Module Processing: The Transformer module consists of multiple attention layers and feedforward neural network layers. Each attention layer includes self-attention mechanisms and feedforward networks to learn global dependencies in the input sequence. The specific process is as follows.

Self-Attention Mechanism: The input sequence x is transformed linearly to derive the value matrix $\boldsymbol{V}$, query matrix $\boldsymbol{Q}$, and key matrix $\boldsymbol{K}$. Attention weights are calculated, and the value matrix is weighted and summed to produce the output of self-attention.

Feedforward Neural Network: The self-attention output passes through feedforward neural network layers, where nonlinear transformations and feature extraction occur, resulting in the output of the Transformer module.

b) CNN Module Processing: The CNN module is composed of convolutional layers and pooling layers to capture local features on the input data. The detailed procedure is outlined below.

Convolution Operation: The input sequence x undergoes convolution operations with multiple filters to extract features, resulting in convolutional feature maps.

Pooling Operation: The convolutional feature maps are processed by pooling operations, which perform downsampling, retain the main features, and reduce data dimensions.

Feature Integration Process: The outputs of the Transformer and CNN modules are combined through weighted summation to derive the ultimate feature representation.

Model Prediction: The ultimate feature representation is inputted into the output layer of the model, which uses a softmax function or other activation functions for classification or regression predictions, resulting in the model's final prediction output.

3) Loss Function Definition

During model training, an appropriate loss function is defined to evaluate the difference between the model's predictions and the true labels, allowing for parameter updates through backpropagation. Typical loss functions are Cross-Entropy Loss and Mean Squared Error (MSE). Suppose the model output is $\hat{y}$, the true label is y, and the MSE loss function is defined as detailed below:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{15}$$

where n denotes the number of samples, $\hat{y}_i$ denotes the prediction result of the model for the sample i, and y_i denotes the true label of the sample i.

3.2. **Basic concept of attention mechanism.** The attention mechanism originally stems from human visual and cognitive processes. When humans perceive and process information, the attention mechanism enables us to focus on specific information or regions while ignoring irrelevant data, allowing for more efficient processing of input data. This mechanism has been integrated into deep learning, achieving notable success in tasks like natural language processing and computer vision. The basic principle of the attention mechanism is to mimic the human attention process by assigning different attention weights to various parts of the input so that the model can focus more on important information when processing sequential data. In the attention mechanism, attention weights are usually calculated for each position within the input sequence, and these weights are then applied to the input features to obtain the weighted feature representation.

Given the input sequence $x = \{x_1, x_2, \ldots, x_n\}$ and the query vector q, the formula for computing attention weights is as follows:

$$\text{Attention}(x_i, q) = \frac{\exp(e(x_i, q))}{\sum_{j=1}^{n}\exp(e(x_j, q))} \tag{16}$$

where $e(x_i, q)$ is a function that calculates the degree of association between input x_i and the query vector q.

The introduction of the attention mechanism allows the model to automatically learn dependencies across different positions, enhancing the capture of long-range dependencies within the input sequence and improving the model's performance and generalization ability.

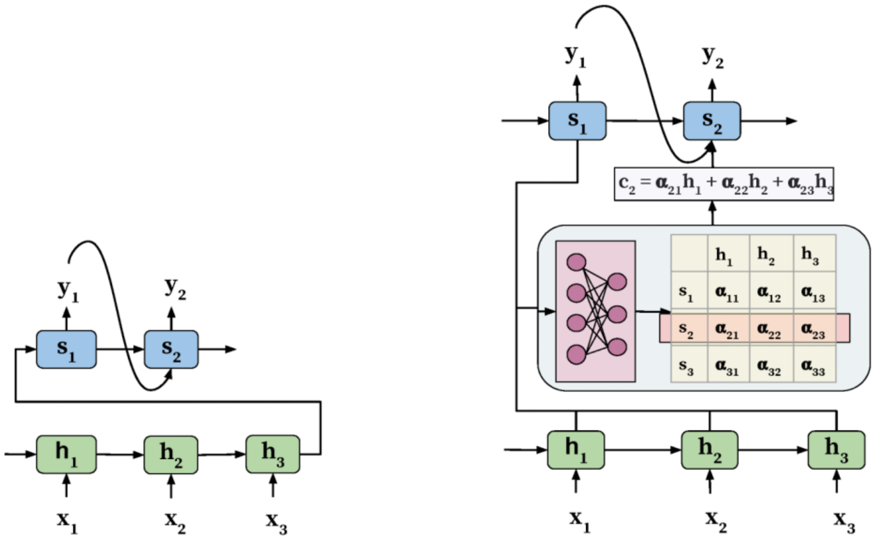
3.3. **Pooling of attention mechanism.** Attention pooling is a pooling method based on the attention mechanism, designed to dynamically adjust the weight of each feature according to its importance in the input, thereby generating the final pooled representation. Traditional pooling methods typically use fixed pooling operations and cannot adapt to changes in the significance of features at various positions. In contrast, attention pooling introduces the attention mechanism, enabling the model to dynamically adjust feature weights according to the specific conditions of the input, thereby more effectively capturing the important information in the input sequence.

The calculation of attention pooling is usually divided into two steps: calculating the attention weights and weighted pooling.

1) Calculating Attention Weights: First, attention weights are calculated for every position in the input sequence to measure the importance of that position in the final pooled representation. The attention weights are typically calculated using the attention mechanism, and this process can be implemented using the softmax function in the attention mechanism to convert the raw scores into a probability distribution, allowing each position to be weighted accordingly. This is shown in Formula (17).

2) Weighted Pooling: Next, based on the calculated attention weights, the input features are pooled with the respective weights to obtain the final pooled representation. Specifically, the feature at each input position is scaled by its respective attention weight, and then the weighted features are aggregated to form the final pooled representation. The mathematical expression for weighted pooling is as follows:

$$\text{Attention\_output} = \sum_{i=1}^{n} \text{Attention}(x_i, q) \cdot x_i \tag{17}$$



(a) (b)

FIGURE 3. Attention mechanism

By introducing attention pooling, the model can dynamically adjust feature weights according to their importance in the input, generating a more representative pooled output, thereby improving the model's performance and generalization ability.

3.4. **Model application.** Determining dynamic electric carbon factors on the user side of the power grid requires extensive data from the power system and is influenced by numerous factors. Traditional calculation methods often fail to adequately consider the complex relationships between these factors, resulting in suboptimal model performance. Therefore, this paper introduces an attention pooling mechanism to help the model concentrate more effectively on key input features and improve the accuracy of dynamic electric carbon factor predictions.

The model introduced is based on a Transformer-CNN architecture, which combines the Transformer model and CNN to process dynamic data from the power grid's user domain and predict changes in electric carbon factors. In this model, attention pooling is incorporated to enhance the model's focus on crucial input features and improve its capability to capture significant information.

Specifically, in the encoder part of the Transformer model, the self-attention mechanism is used to implement attention pooling. The self-attention mechanism enables the model to dynamically attend to information from other positions in the input sequence while processing each individual position. In each layer of self-attention computation, the significance of each input position to the entire sequence is calculated, and corresponding attention weights are generated. These attention weights are then applied to the input features to generate weighted representations. This allows the model to dynamically adjust the representation of each position based on the importance of the input data, thereby more effectively capturing the dynamic characteristics of the power grid's user side.

In attention pooling, the model first calculates the relevance scores between each input position and a query vector, and then uses the softmax function to convert these scores into attention weights. These attention weights indicate the significance of each input position within the entire sequence. Finally, the model applies these attention weights to the input features and generates the final pooled representation through weighted summation.

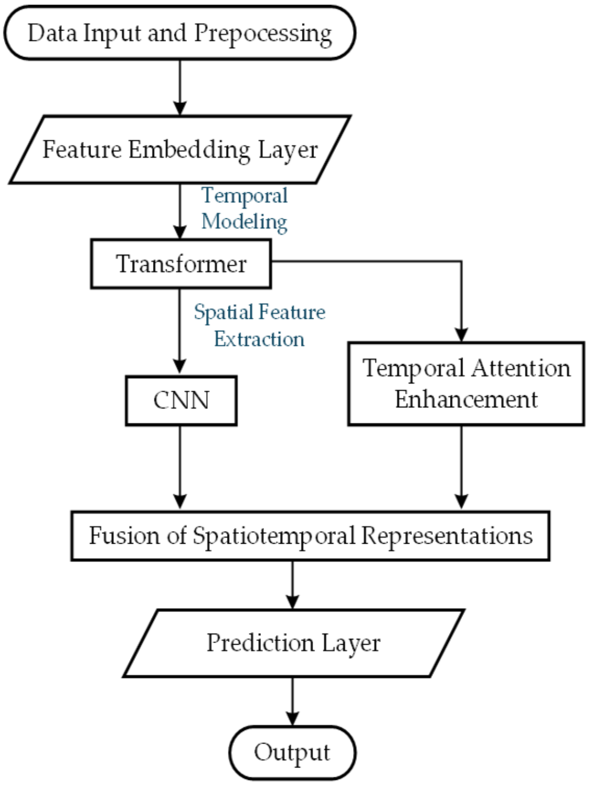


FIGURE 4. Model work process

By introducing the attention pooling mechanism, our model can better concentrate on important input features, thereby improving model performance and prediction accuracy, and contributing to enhanced calculation performance of dynamic electric carbon factors within the power grid's user domain. A general framework diagram illustrating the complete model work process is shown in Figure 4. As shown in Figure 4, the model work process involves the following sequential and parallel key steps.

- ***Step 1:*** The Transformer module is applied immediately after the feature embedding layer. It is responsible for capturing the temporal dependencies in the input sequence through the self-attention mechanism.
- ***Step 2:*** The output of the Transformer is then passed along two parallel paths. One path goes to the CNN, which is used to extract spatial features from the same

input representation. The other path goes to the Temporal Attention Enhancement module, which operates on the temporal features produced by the Transformer. This module refines and re-weights the temporal representations to emphasize important time steps, leveraging attention pooling.

- ***Step 3:*** The outputs of CNN (spatial features) and Temporal Attention Enhancement (refined temporal features) are then fused, forming a comprehensive spatiotemporal representation. This fused representation is subsequently fed into the prediction layer, which generates the final model output.

4. Experiment and Result Analysis.

4.1. **Experimental setup.** Two classical power grid datasets are used in this paper, namely IEEE 118-bus system and IEEE 39-bus system. These data sets are publicly available standard test data and are widely used in research and experiments in the field of power systems. The model parameter settings (flexibly adjusted) are presented in Table 1.

TABLE 1. Model parameter settings

Parameter name	Numerical value
Learning rate	0.001
Batch size	64
Epochs	1000
Optimizer	Adam
Weight decay	0.0001
Learning rate decay	0.01
Loss function	Mean square error (MSE)
Early stopping	Stop training when the loss on the verification set no longer falls
Model checkpoint	Save the model that achieves the highest performance on the validation set

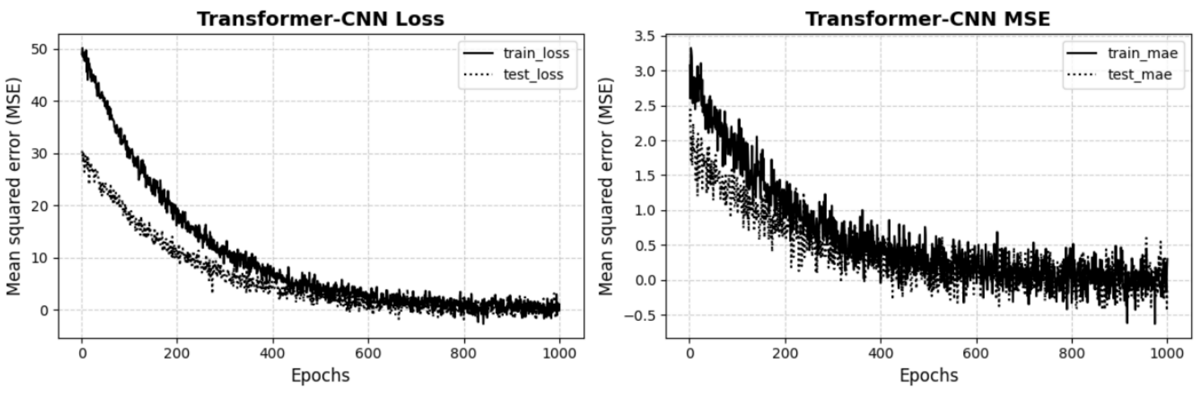


FIGURE 5. Model training and validation loss

4.2. **Dynamic electric carbon factor prediction experiment.** This study aims to explore the effectiveness of the proposed Transformer-CNN model in predicting dynamic electric carbon factors on the power grid user side. The first experiment was conducted on the IEEE 39-bus system, with the results shown in Figure 6. The study first compares the model with traditional methods, including linear regression and decision tree models.

The results demonstrate that the proposed model offers notable advantages in predictive performance in the 39-bus system. Specifically, compared to the linear regression model, the proposed model exhibits superior performance in terms of error rate, with only 0.012 compared to 0.055 for the linear regression model, indicating that the proposed model can more accurately capture complex relationships in the power grid system. In contrast, the decision tree model has an error rate of 0.038, which is notably higher than that of the proposed model, demonstrating that the Transformer-CNN model based on deep learning has better generalization capabilities when handling power grid data.

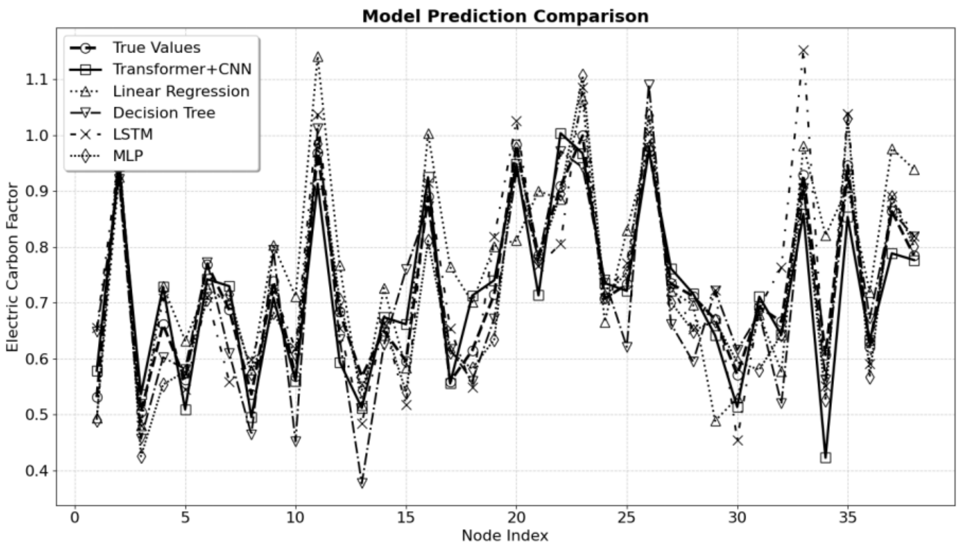


FIGURE 6. Comparison of IEEE 39-bus dynamic electric carbon factor prediction

Additionally, the study compares the model with other deep learning-based models, including MLP models and LSTM. The results indicate that the proposed model also surpasses these deep learning models in the 39-bus system. The LSTM-based model has an error rate of 0.020, while the MLP-based model has an error rate of 0.025, both higher than that of the proposed model. This suggests that the Transformer-CNN model has better feature extraction and representation capabilities when processing power grid data, enabling it to more effectively capture complex relationships in the data.

In the second experiment, the study further extends the dataset by evaluating the model on a larger 118-bus system, with results shown in Figure 7. The purpose of this experiment is to validate the prediction performance of the proposed Transformer-CNN model in a larger-scale power grid system. Similarly, the proposed model is relative to traditional methods and deep learning-based models.

First, relative to traditional methods, the proposed model yields satisfactory prediction results in the 118-bus system. The error rate of the proposed model is significantly reduced to only 0.015 compared to 0.048 for the linear regression model. Moreover, the proposed model also demonstrates better predictive performance compared to the decision tree model, which has an error rate of 0.065. This indicates that the Transformer-CNN model can effectively handle larger-scale power grid data and accurately capture dynamic changes in the system.

Furthermore, the study compares the proposed model with deep learning-based models. The LSTM-based model has an error rate of 0.023 in the 118-bus system, while the MLP-based model has an error rate of 0.029, both higher than that of the proposed model. This further confirms the superiority of the proposed model in processing larger-scale power

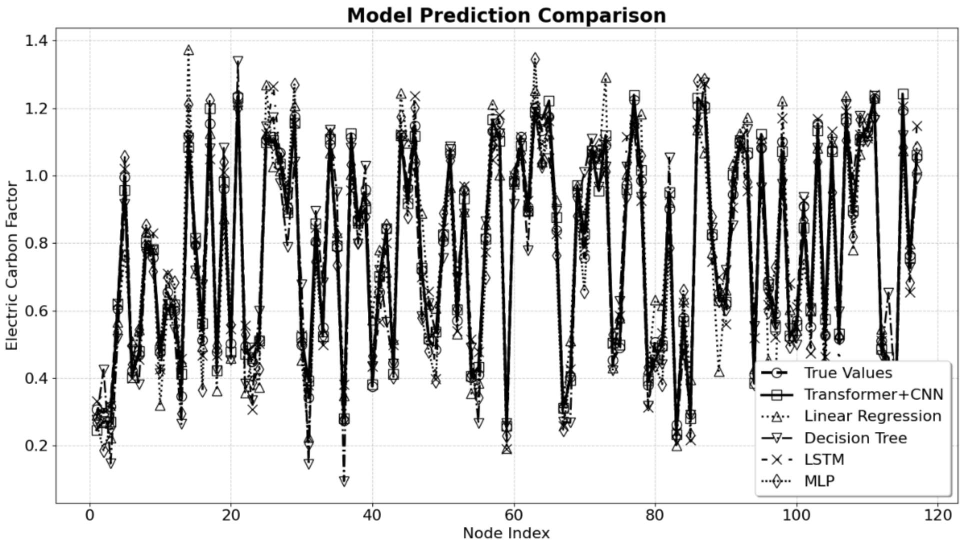


Figure 7. Comparison of IEEE 118-bus dynamic electric carbon factor prediction

Table 2. Average error rate of model test

	Transformer-CNN	Linear regression	Decision tree	LSTM	MLP
IEEE 39-bus average error rate	0.22	0.57	0.58	0.38	0.41
IEEE 118-bus average error rate	0.28	0.62	0.71	0.45	0.46

grid data and its ability to more accurately predict dynamic electric carbon factors in the 118-bus system.

In summary, the results of the two experiments (as presented in Table 2) demonstrate that the proposed Transformer-CNN model exhibits significant advantages and strong generalization capabilities in predicting dynamic electric carbon factors on the power grid user side. Compared with traditional methods, the proposed model can more accurately predict dynamic electric carbon factors in the power grid system, providing an effective tool and support for grid carbon emission management. In comparison to deep learning-based models, the proposed model more effectively captures complex relationships in the data and has better feature extraction and representation capabilities.

4.3. **Environment dynamic change experiment.** In this experiment, a representative node was selected from the 39-bus system, and the dynamic variations in the power grid system were simulated. The power data of the selected node and its connected nodes were randomly altered to simulate the dynamic variations that the power grid system might encounter during actual operation. By tracking and predicting the dynamic electric carbon factors of the selected nodes, the predictive performance of the proposed Transformer-CNN model under dynamic conditions was evaluated, as shown in Figure 8.

The proposed model demonstrated high prediction accuracy under dynamic conditions. For the dynamic electric carbon factor prediction of the selected nodes, the error rate of the proposed model was 0.022, significantly outperforming other methods. The error rate of the dynamic electric carbon factor prediction using traditional methods, specifically linear regression, was 0.057. For deep learning-based models, the error rates were 0.038 and 0.041 for the LSTM-based and MLP-based models, respectively. Relative to both conventional methods and deep learning-based models, the proposed model achieved the

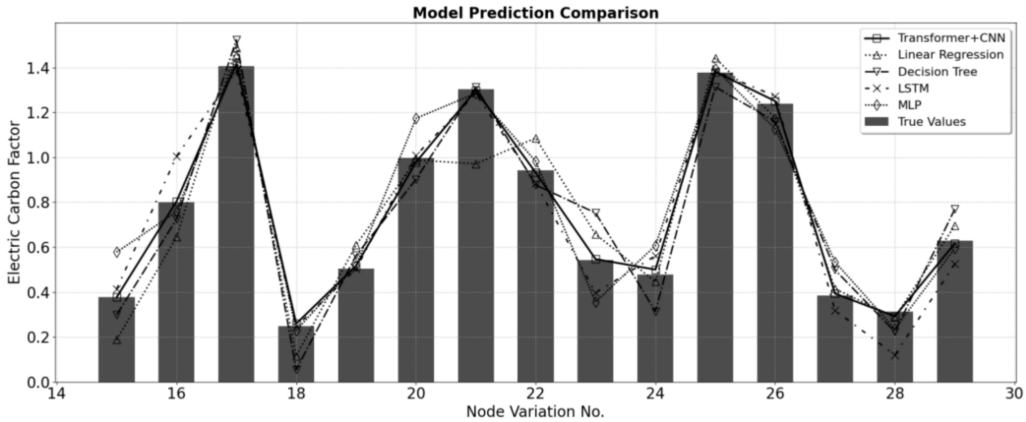


FIGURE 8. Comparison of IEEE 39-bus node change tracking prediction

best predictive performance under these conditions. The proposed model was more accurate in capturing the dynamic changes within the system and in predicting dynamic electric carbon factors compared to traditional methods. It also exhibited superior feature extraction and representation capabilities compared to deep learning-based models, effectively capturing complex relationships within the data.

The findings from the experiments validate the superiority of the proposed model in handling dynamic conditions, accurately capturing system changes, and predicting dynamic electric carbon factors. The model demonstrated higher forecast accuracy and generalization capability.

An analysis of the findings from the experiments revealed that the proposed model effectively responded to changes in the power grid system under dynamic conditions, exhibiting strong robustness and generalization capabilities. The simulation of dynamic changes showed that the model could capture the complex spatiotemporal relationships within the system, leading to accurate predictions of dynamic electric carbon factors. This provides strong support and validation for the potential application of the proposed model in real-world power grid systems.

5. **Conclusion.** This paper presents an innovative Transformer-CNN hybrid model for predicting dynamic electric carbon factors within the user domain of power grids. By integrating the Transformer's capability to capture long-range temporal dependencies with CNN's strength in spatial feature extraction, along with the incorporation of an attention pooling mechanism, the model demonstrates substantial improvements in prediction accuracy. Experimental results show that the proposed model significantly outperforms traditional methods and standalone models, exhibiting superior accuracy, stability, and generalization. These findings highlight the model's potential as a powerful and versatile tool for predicting dynamic electric carbon factors, representing a significant advancement in the field and offering valuable insights for carbon emission management in power grids.

Acknowledgments. This work was financially supported by the Major Science and Technology Project of China Southern Power Grid Co., Ltd. (ZBKJXM20232456).

REFERENCES

[1] Y. Hu and M. Yi, Energy consumption and carbon emissions forecasting for industrial processes: Status, challenges and perspectives, *Renewable and Sustainable Energy Reviews*, vol.182, 113405, 2023.

[2] J. Wang and L. Li, Sustainable energy development scenario forecasting and energy saving policy analysis of China, *Renewable and Sustainable Energy Reviews*, vol.58, pp.718-724, 2016.

[3] X. Ma, W. Yong and W. Chen, Low-carbon development of China's thermal power industry based on an international comparison: Review, analysis and forecast, *Renewable and Sustainable Energy Reviews*, vol.80, pp.942-970, 2017.

[4] J. Huber et al., Carbon efficient smart charging using forecasts of marginal emission factors, *Journal of Cleaner Production*, vol.284, 124766, 2021.

[5] Y. Li, T. Li and S. Lu, Forecast of urban traffic carbon emission and analysis of influencing factors, *Energy Efficiency*, vol.14, 84, 2021.

[6] D. Maji, S. Prashant and K. S. Ramesh, Multi-day forecasting of electric grid carbon intensity using machine learning, *ACM SIGENERGY Energy Informatics Review*, vol.3, no.2, pp.19-33, 2023.

[7] J. Ji et al., Real-time monitoring and short-term forecasting of regional carbon emissions based on coupled electric-carbon models, *The 12th International Conference on Renewable Power Generation (RPG 2023)*, Shanghai, China, pp.791-799, 2023.

[8] V. Aryai and G. Mark, Day ahead carbon emission forecasting of the regional National Electricity Market using machine learning methods, *Engineering Applications of Artificial Intelligence*, vol.123, 106314, 2023.

[9] M. Shi, Forecast of China's carbon emissions under the background of carbon neutrality, *Environmental Science and Pollution Research*, vol.29, no.28, pp.43019-43033, 2022.

[10] H. Yang, M. Wang and G. Li, A multi-factor forecasting model for carbon emissions based on decomposition and swarm intelligence optimization, *Measurement*, vol.222, 113554, 2023.

[11] L. Chang et al., Exploring carbon dioxide emissions forecasting in China: A policy-oriented perspective using projection pursuit regression and machine learning models, *Technological Forecasting and Social Change*, vol.197, 122872, 2023.

[12] R. Lin et al., Dual-stage ensemble approach using online knowledge distillation for forecasting carbon emissions in the electric power industry, *Data Science and Management*, vol.6, no.4, pp.227-238, 2023.

[13] X. Zhang and W. Dan, A GNN-based day ahead carbon intensity forecasting model for cross-border power grids, *Proc. of the 14th ACM International Conference on Future Energy Systems*, 2023.

[14] Y. Zhang et al., Graph convolution network for carbon emission factor forecast of power grid, *2023 5th International Conference on Power and Energy Technology (ICPET)*, 2023.

[15] A. Wan et al., Dynamic pollution emission prediction method of a combined heat and power system based on the hybrid CNN-LSTM model and attention mechanism, *Environmental Science and Pollution Research*, vol.29, no.46, pp.69918-69931, 2022.

[16] X. Wu et al., Carbon emissions forecasting based on temporal graph transformer-based attentional neural network, *Journal of Computational Methods in Science and Engineering*, vol.24, no.3, pp.1405-1421, 2024.

[17] Y. Liu, K. Matsui, Y. Kageyama, H. Shirai and C. Ishizawa, A CNN-based method for human action analysis using nighttime infrared images, *International Journal of Innovative Computing, Information and Control*, vol.19, no.6, pp.1861-1875, 2023.

[18] D. T. Long, Efficient CNN model based on combining residual network and dense-connected network architectures for facial expression recognition, *International Journal of Innovative Computing, Information and Control*, vol.19, no.5, pp.1661-1678, 2023.

[19] Y. Liu et al., Improving a faster R-CNN model for vehicle detection and human action recognition at night via infrared thermal imaging using transfer learning, *The 10th IIAE International Conference on Intelligent Systems and Image Processing*, GS8-1, 2023.

[20] Y. Chen et al., A hybrid CNN-Transformer model for ozone concentration prediction, *Air Quality, Atmosphere & Health*, vol.15, no.9, pp.1533-1546, 2022.

[21] L. Suganthi and A. S. Anand, Energy models for demand forecasting – A review, *Renewable and Sustainable Energy Reviews*, vol.16, no.2, pp.1223-1240, 2012.

Author Biography

Jianlin Tang received bachelor's degree in Electrical Engineering from South China University of Technology, China, in 2017. In 2020, he received master's degree in Electrical Engineering from the same university. He is currently an Engineer with Guangdong Provincial Key Laboratory of Intelligent Measurement and Advanced Metering of Power Grid, and CSG Electric Power Research Institute, China. His research interest includes measurement of electricity and its carbon.

Yong Xiao received master's degree in Computer Software and Theory from University of Electronic Science and Technology of China, China, in 2005. In 2017, he obtained his Ph.D. degree in Electrical Engineering from Wuhan University, China. He is currently a Professorate Senior Engineer with Guangdong Provincial Key Laboratory of Intelligent Measurement and Advanced Metering of Power Grid, and CSG Electric Power Research Institute, China. His research interest includes measurement of electricity and its carbon.

Xiaoming Lin received bachelor's degree in Electrical Engineering from Huazhong University of Science and Technology, China, in 2016. In 2019, he received master's degree in Electrical Engineering from South China University of Technology, China. He is currently an Engineer with Guangdong Provincial Key Laboratory of Intelligent Measurement and Advanced Metering of Power Grid, and CSG Electric Power Research Institute, China. His research interest includes charging operation technique of electric vehicle.

Bin Qian received bachelor's degree in Electrical Engineering from Huazhong University of Science and Technology, China, in 2011. In 2014, he received master's degree in Electrical Engineering from the same university. He is currently a Senior Engineer with Guangdong Provincial Key Laboratory of Intelligent Measurement and Advanced Metering of Power Grid, and CSG Electric Power Research Institute, China. His research interest includes electricity measurement.