machines

MDPI

Article

Predicting Real-Time Carbon Emissions for Power Grids Using Graph Convolutional Networks

Qian Zhao [1], Jianhua Chen [1], Qianwei Jia [2], Cong Sun [1], Xi Chen [3,*] and Hongtian Chen [3]

[1] State Grid Jibei Electric Power Company Ltd., Beijing 100054, China; zhao.qian.a@jibei.sgcc.com.cn (Q.Z.); dffg111@163.com (J.C.); suncong_ncepu@163.com (C.S.)
[2] State Grid Jibei Electric Power Company Limited Management Training Center, Beijing 102401, China; jqw_2907@163.com
[3] State Key Laboratory of Submarine Geoscience, Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China; hongtian.chen@sjtu.edu.cn
* Correspondence: xi-chen@sjtu.edu.cn

Abstract

Accurate prediction of carbon emissions is crucial for both providing effective carbon reduction guidance to the power grid sector and driving society-wide carbon emission reduction. Existing methods based on power flow calculation theory heavily rely on real-time grid parameters and cannot accurately predict grid carbon emissions. To address the challenge, this paper proposes a novel spatiotemporal prediction model by integrating graph convolutional networks (GCNs) and temporal convolutional networks (TCNs). The GCN layer enhances node representation by aggregating neighborhood information, while the TCN layer captures long-term temporal dependencies through expanded causal convolutions. Experiments conducted on the IEEE 14-bus system demonstrate that the model achieves high accuracy and real-time processing capabilities, confirming its robustness and practical value for dynamic carbon emission predictions.

Keywords: carbon emissions; graph convolutional networks (GCNs); temporal convolutional networks (TCNs); carbon emissions; power grid systems

check for updates

Academic Editor: Ahmed Abu-Siada

Received: 8 October 2025
Revised: 5 November 2025
Accepted: 13 November 2025
Published: 18 November 2025

Citation: Zhao, Q.; Chen, J.; Jia, Q.; Sun, C.; Chen, X.; Chen, H. Predicting Real-Time Carbon Emissions for Power Grids Using Graph Convolutional Networks. *Machines* **2025**, *13*, 1061. https://doi.org/10.3390/machines13111061

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The carbon emissions provide key guidance for developing new power systems, conducting precise carbon accounting, and evaluating the value of green electricity consumption [1–4]. Calculating regional dynamic carbon emissions is challenging because it involves modeling power flow tracking for electricity consumption across different time intervals within a region. Due to the multi-voltage-level feature in power grids, the power flow at each level undergoes dynamic changes. Consequently, constructing an accurate and efficient model is important yet complex [5–7].

Prediction methods for the carbon emissions generally fall into three major categories: scenario prediction models, linear regression models, and machine learning models [8]. The long-range energy alternatives planning system (LEAP) is a scenario prediction model that can be used to predict carbon emissions from power grids when combined with scenario analysis. Wang and Watson [9] developed four scenario pathways under a cumulative carbon budget to explore China's low-carbon transition in power, transport, industry, and household sectors. Furthermore, the LEAP model is applied to construct and compare four exploratory scenarios for Panama's electricity sector, quantifying the trade-offs between system costs, global warming potential, and resource diversity under different renewable

energy deployment pathways through 2026 [10]. The core advantage of linear regression models is their exceptional interpretability, which enables the clear quantification of the relationship between power generation and total carbon emissions. This provides a direct and transparent basis for energy dispatch decisions. Furthermore, the model is computationally efficient and easy to implement, enabling the rapid establishment of predictive benchmarks [11,12]. A hyperbolic tangent function-based linear regression model [13] is proposed to capture seasonal and diurnal trends in electricity consumption for short-term forecasting and carbon emission estimation. Ref. [14] improves the existing stirpat model by incorporating carbon emissions into the modelling analysis. It uses ridge regression to address biased estimates in historical data and establishes a carbon emission prediction model for Shanghai's electricity and energy sectors based on elasticity relationships.

Machine learning methods have been widely applied in the field of carbon emissions predictions and have demonstrated excellent performance in recent years. Ref. [15] proposes a CNN-LSTM-attention model combined with Bayesian optimization for precise prediction of carbon dioxide emissions from coal-fired power plants. Machine learning methods combining feature selection, linear regression, and the auto-regressive integrated moving average model (ARIMA) [16] can be employed to forecast short-term carbon emission intensity in power grids, providing precise predictive support for low-carbon dispatch. Furthermore, advanced deep learning architectures, including long short-term memory (LSTM) networks [17], support vector machine (SVM) [18], and TCN [19] offer viable approaches for predicting carbon emissions. These methods are capable of automatically capturing complex nonlinear dependencies and temporal patterns from large-scale historical operational datasets, thereby facilitating accurate and reliable forecasting of carbon emissions in electric power systems. Current research on carbon emissions prediction is increasingly focusing on the integration of multi-modal data, hybrid modeling approaches, and real-time adaptive learning techniques to enhance the accuracy of predictions across complex and dynamic power systems.

Inspired by [20], which shows that graph neural networks (GNNs) can be employed for spatio-temporal forecasting modeling, this paper proposes a novel carbon emissions prediction model based on GNNs for power grid systems. The proposed method leverages GNNs combined with advanced spatio-temporal feature extraction, which enables dynamic and accurate prediction of carbon emissions in power grid systems. The contributions of the study are summarized as follows:

- Complex spatiotemporal data in power grid systems are structured into graph data, leveraging the powerful relational reasoning capabilities of graph neural networks to capture dynamic interactions between regional nodes.
- The TCNs layer captures complex time-dependent relationships, enabling highly accurate predictions of power grid carbon emissions.

The remainder of this paper is organized as follows. The preliminaries and problem formulation is provided in Section 2, and the proposed carbon emissions prediction method is presented in Section 3. Then, the experimental results are given in Section 4. Section 5 concludes this study.

2. Preliminaries and Problem Formulation

2.1. A Carbon Calculation Model for Power Grid Nodes

The operation process of the power grid is shown in Figure 1. The power grid system first converts various energy sources including fuel, wind power, and hydropower into electricity through gas turbines, wind turbines, and hydroelectric stations. Subsequently, the generated electricity is transmitted via a unified grid network and intelligently dispatched and distributed by an energy management system to ensure real-time balance

between power supply and demand. Stable electrical current is precisely delivered to diverse end-use terminals such as industrial zones and charging stations, effectively meeting the energy requirements for urban operations and development. Ultimately, this modern grid—integrating generation, transmission, distribution, and consumption—forms a highly efficient, reliable, and secure closed-loop energy supply system.

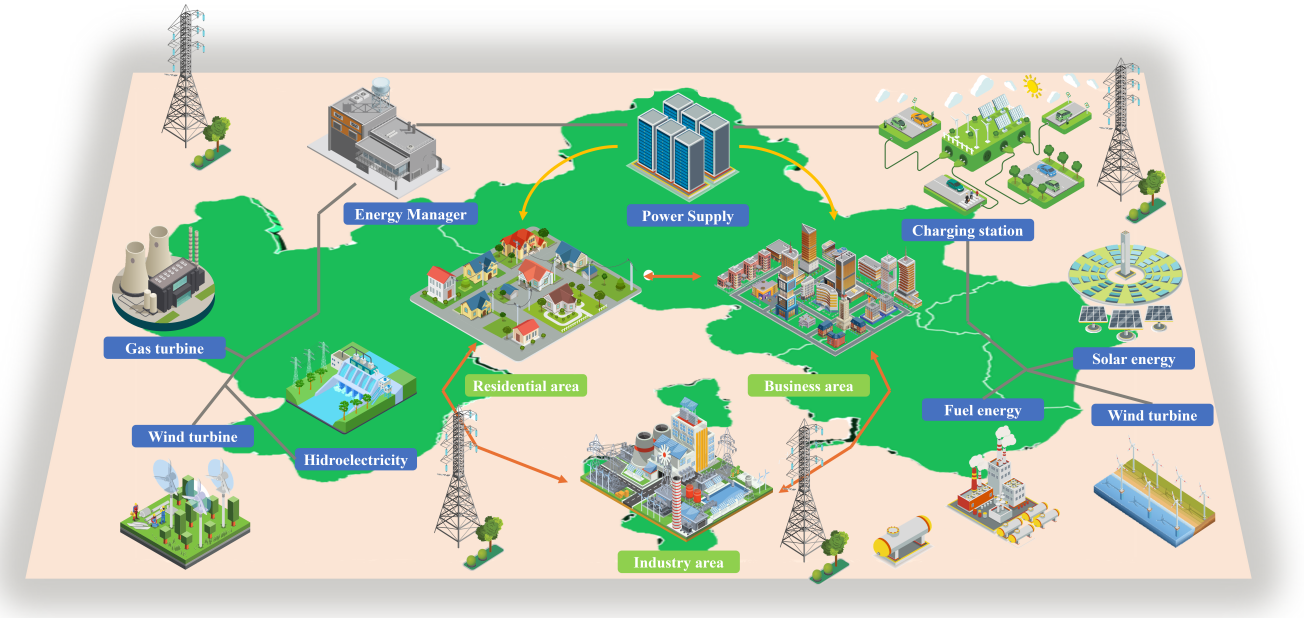

Figure 1. The operation process of the power grid.

Carbon emission flows are modeled as virtual network flows coupled with the active power flow in the power grid, representing the carbon emissions associated with power transmission through branch circuits. A power grid system can be abstracted into the model shown in Figure 2:

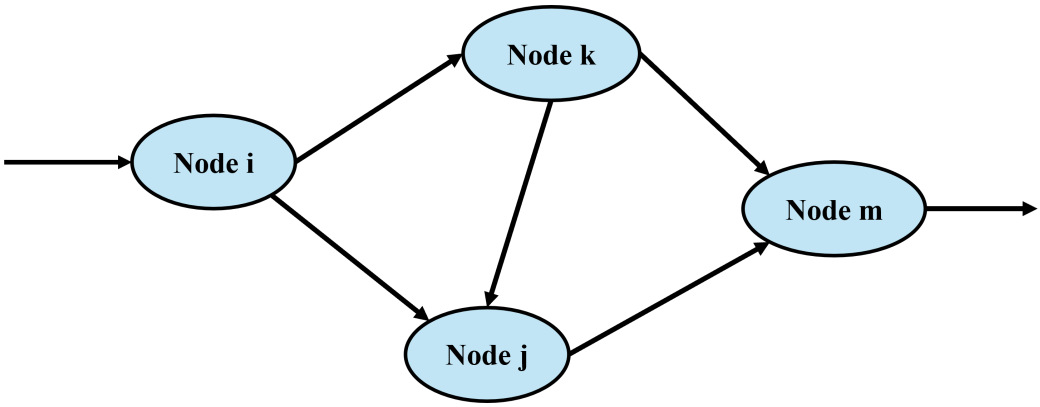

Figure 2. A topology structure of power grids.

Nodes i, j, k, and m represent entities such as substations, power plants, users, and so on. Typically, these nodes are used for consuming electricity, generating electricity, or voltage transformation. Each node satisfies the power balance and carbon emission balance equations as follows:

$$S_k = \sum_{i=1}^{n} S_{ik} = \sum_{j=1}^{m} S_{kj} + \Delta S$$
$$E_k = \sum_{i=1}^{n} E_{ik} = \sum_{j=1}^{m} E_{kj} + E_{\Delta S} \quad (1)$$

where S_k represents the total electricity consumption of node k, S_{ik} denotes the electricity supplied from node i, S_{kj} indicates the electricity flowing to node j, ΔS represents electricity consumed internally at node k. E_k represents the total carbon emissions of node k, E_{ik} denotes the carbon emissions originating from node i, E_{kj} denotes the carbon emissions flowing to node j, $E_{\Delta S}$ denotes the carbon emissions generated by internal consumption of node k. n and m represent the number of incoming and outgoing nodes respectively. Formula (1) represents the balance equations for electricity and carbon flows in power grids and serves as a key basis for predicting grid carbon emissions.

2.2. Problem Description

GCNs enable hierarchical feature extraction directly from graph-structured data through efficient convolutional operations, significantly enhancing performance in classification and prediction tasks while maintaining high computational efficiency [21,22]. The specific architecture is illustrated in Figure 3.

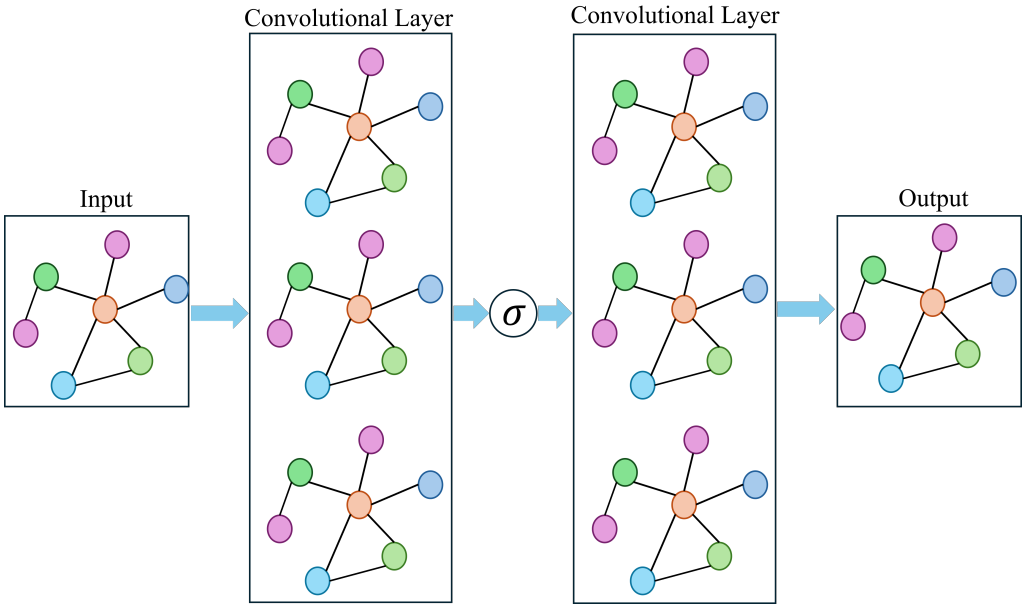

Figure 3. Schematic depiction of multi-layer Graph Convolutional Networks.

Consider a graph $G = (V, E)$ with $N = |V|$ vertices. To include self-loops, the modified adjacency matrix can be defined as follows:

$$\hat{A} = A + I_N, \hat{D}_{ii} = \sum_j \hat{A}_{ij}, \widetilde{A} = \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2} \quad (2)$$

where $\hat{D}$ is the degree matrix of $\hat{A}$, $\widetilde{A}$ is the symmetrically normalized adjacency matrix, adjacency matrix $A \in \mathcal{R}^{N\times N}$.

In this work, the adjacency matrix A is defined as a binary matrix, where $A_{ij} = 1$ if there is a direct transmission connection between node i and node j, and $A_{ij} = 0$ otherwise. This binary representation captures the topological connectivity of the power grid without

incorporating physical line parameters such as impedance or capacity, as the primary focus is on modeling spatial dependencies between regions. The graph structure is based on the IEEE 14-node system mentioned later, where nodes represent buses and edges represent transmission lines.

Given the node feature matrix $X \in \mathcal{R}^{N \times F}$, the propagation rule of a GCN layer is defined as:

$$H^{(l+1)} = \sigma\Big(\widetilde{A}H^{(l)}W^{(l)} + \mathbf{1}(b^{(l)})^{\top}\Big) \tag{3}$$

where $H^{(l)} \in \mathcal{R}^{N \times F^{(l)}}$ denote the node representations at the l-th layer, with $H^{(0)} = X$, $W^{(l)} \in \mathcal{R}^{F^{(l)} \times F^{(l+1)}}$ is a trainable weight matrix, $b^{(l)} \in \mathcal{R}^{F^{(l+1)}}$ is a bias vector, $\sigma(\cdot)$ is a nonlinear activation function, and $\mathbf{1}$ is an all-ones column vector used to broadcast the bias.

In an element-wise form, the update rule for node i is expressed as:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i) \cup \{i\}} \widetilde{A}_{ij}\, h_j^{(l)} W^{(l)} + b^{(l)}\right) \tag{4}$$

where $\mathcal{N}(i)$ denotes the set of neighbors of node i. This formulation allows each node to aggregate feature information from its neighborhood and transform it through learnable parameters.

For node-level tasks, the final node embeddings $H^{(L)}$ are directly used for downstream predictions. For graph-level tasks, a permutation-invariant readout function is applied to obtain a global representation:

$$z_G = \alpha\Big(H^{(L)}\Big) \tag{5}$$

where α can be implemented as sum-pooling, mean-pooling, or max-pooling over the node embeddings.

The computational complexity of a GCN layer is presented as follows:

$$\mathcal{O}(|E|F^{(l)} + NF^{(l)}F^{(l+1)}) \tag{6}$$

where $|E|$ is the number of edges, N the number of nodes, and $F^{(l)}$, $F^{(l+1)}$ the input and output feature dimensions, respectively. The first term arises from sparse matrix multiplication with $\widetilde{A}$, while the second corresponds to dense linear transformation.

Given a power grid system has M regions, each region contains both power generation and consumption facilities in a power grid system. The entire multi-region power grid system is modeled as a graph structure:

$$G_t = (V, E) \tag{7}$$

where Each node V represents a power grid region, edges E represent transmission connections between regions. Each node has a set of feature vectors x_t. The data of each node are aggregated into the feature matrix X:

$$X = \begin{bmatrix} x_1^T & x_2^T & \cdots & x_N^T \end{bmatrix}^T \tag{8}$$

The ultimate goal is to predict the carbon emissions Y_t of each node at time step t by constructing GCNs.

$$Y_t = G_t(X) \tag{9}$$

Based on the above discussion and analysis, in order to achieve effective prediction of grid carbon emissions, the following issues must be addressed.

1. Establish an accurate measurement model based on graph convolution networks for power grid systems.
2. Transform the output of the network topology matrices into a predictive form for carbon emissions.
3. Train the entire network to minimize prediction errors for the carbon emissions.

3. GCN-Based Carbon Emissions Prediction

Figure 4 shows the architecture of the proposed GCN-based approach for predicting the carbon emissions, comprising three components: a graph convolutional layer, a pooling layer and a TCN layer. This section first introduces the architecture of the graph convolutional layer, then covers the pooling layer and the TCN layer, and finally presents the training method and objective function.

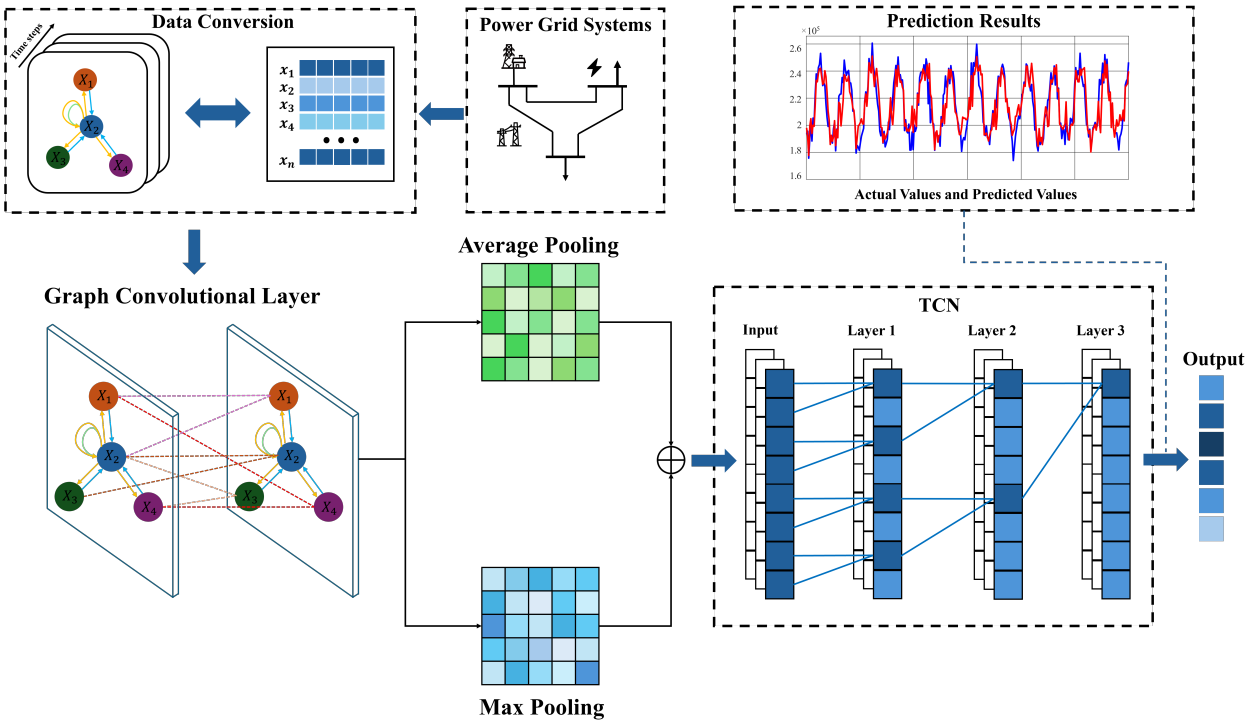

Figure 4. The architecture of the proposed GCN-based carbon emissions prediction approach.

3.1. Graph Convolutional Layer

The propagation rules for each convolutional layer have been provided in Equation (3). Node regression output can be computed as:

$$\hat{y}_i = w^T h_i^{(L)} + b \tag{10}$$

where $h_i^{(L)}$ is the feature representation of the node at the final layer. Power networks exhibit distinct graph-structured characteristics, where each region can be regarded as a node in the graph, and power exchanges between regions form the edges [23,24]. The GCN layer effectively captures the mutual influence between regions by aggregating information from nodes and their neighboring nodes. Figure 5 illustrates the structure of a GCN layer.

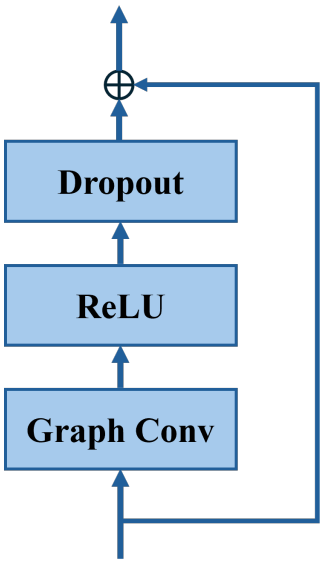

Figure 5. Structure of a GCN layer.

Each regional node possesses multiple attributes. The GCN layer integrates these attributes with graph structural information to generate context-rich node embeddings. These embeddings not only reflect the node's intrinsic states but also incorporate insights from its local graph structure, thereby providing a more comprehensive representation of the system operational status. The introduction of the GCN layer addresses the spatial limitations of purely temporal models. By incorporating graph structure information from each time step into node representations, the model can consider the real-time status of other regions while predicting carbon emissions at the current time step, leading to more accurate predictions.

The pooling layer is used to reduce the dimensionality of node features and graph structures while preserving important information. This is particularly useful for graph-level regression tasks, where a fixed-size representation of the entire graph is required. The operation of the pooling layer is shown in Figure 6. After passing through the graph convolutional layer, $H^{(l)} \in \mathcal{R}^{N \times d}$ denotes the resulting node feature matrix. A mean pooling operation aggregates node features into a graph-level representation h_g:

$$h_{\mathcal{G}}^{mean} = \text{MeanPool}\left(H^{(l)}\right) = \frac{1}{N}\sum_{i=1}^{N} h_i^{(l)}. \tag{11}$$

Max pooling operations can be computed as:

$$h_{\mathcal{G}}^{max} = \text{MaxPool}\left(H^{(l)}\right) = \max_{i \in V} h_i^{(l)} \tag{12}$$

where $h_{\mathcal{G}}^{mean}, h_{\mathcal{G}}^{max} \in \mathcal{R}^d$. To construct a more informative and robust graph-level representation, this representation incorporates both average node information reflecting the system's overall carbon emission baseline and extreme information from key nodes that may dominate total carbon emissions. Therefore, the pooling layer must perform both mean pooling and max pooling simultaneously, enabling the fully connected layer to make more accurate and reliable predictions. Feature concatenation between mean pooling and max pooling is performed as follows:

$$h_{\mathcal{G}} = h_{\mathcal{G}}^{mean} \parallel h_{\mathcal{G}}^{max} \tag{13}$$

where $\parallel$ denotes the vector concatenation operation, $h_{\mathcal{G}} \in \mathcal{R}^{2d}$.

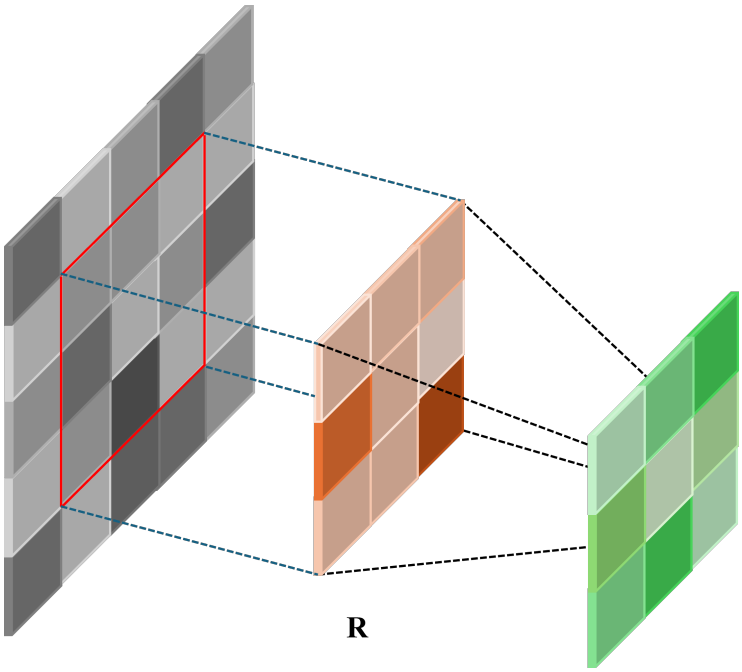

Figure 6. The operation of Pooling Layer.

Since the pooling layer representation $h_{\mathcal{G}}(t)$ consists of independent vectors and TCNs require sequence inputs of fixed time length, data transformation is necessary. This involves reconstructing $h_{\mathcal{G}}(t)$ into a sliding window format, enabling the model to preserve temporal dependencies while allowing TCNs to effectively capture dynamic patterns across images.

Given a temporal window length L, the input for the k-th sample is constructed as:

$$\mathbf{X}_k = \left[h_{\mathcal{G}}(k-L+1), h_{\mathcal{G}}(k-L+2), \ldots, h_{\mathcal{G}}(k)\right]^T \tag{14}$$

where $\mathbf{X}^{(k)} \in \mathcal{R}^{L\times 2d}$ represents the sequence of pooling representations from the past L time steps. For $k < L$, zero-padding is applied:

$$\mathbf{X}_k = \left[\underbrace{\mathbf{0}, \ldots, \mathbf{0}}_{L-k}, h_{\mathcal{G}}(1), \ldots, h_{\mathcal{G}}(k)\right]^T. \tag{15}$$

To match the input format of the TCN, the feature matrix is transposed as:

$$\mathbf{X}_k^{\text{TCN}} = \mathbf{X}_k^T \in \mathcal{R}^{2d\times L}, \tag{16}$$

where the final TCN input tensor $\mathcal{X}_{\text{TCN}} \in \mathcal{R}^{B\times 2d\times L}$, B denotes the batch size, $2d$ is the number of input channels, and L is the temporal length.

This construction establishes the mapping from the pooling output $h_{\mathcal{G}}$ to the TCN input $\mathcal{X}_{\text{TCN}}$, enabling the model to capture the temporal dynamics of graph-level representations.

3.2. Temporal Convolutional Layers

Graph pooling operations generate compact representations of the system at each time step, which capture the global carbon emission characteristics of the grid. However, these representations evolve dynamically as the system undergoes changes in load, generation structure, and inter-regional power flows. To model these sequential dependencies, a TCN is introduced after the pooling operations. The TCN architecture is particularly suitable for predicting carbon emissions because it combines two essential properties [25–28]:

1. causal convolutions that strictly preserve temporal order and avoid information leakage from future to past.

2. dilated convolutions that increase the receptive field and enable long-term dependency modeling.

A dilated causal convolution is illustrated as shown in Figure 7. Given a temporal input sequence $\mathcal{X}_{\text{TCN}}$, an l-th layer dilated causal convolution can be expressed as follows:

$$(Z^{(l)})_{c,t} = \sum_{c'=1}^{2d} \sum_{i=0}^{k-1} W_{c,c',i}^{(l)} \mathcal{X}_{c',t-m\cdot i} + b_c^{(l)} \tag{17}$$

where k represents the kernel size, m represents the dilation factor, c denotes the output channel, and t denotes the time step. W and b denote weights and biases. $\mathcal{X}_{c',\tau} = 0$ if $\tau < 0$, ensuring causality. This operation allows the model to capture both short-term and long-term dependencies by adjusting d across layers. Compared to recurrent networks, dilated convolutions can process sequences in parallel across time, significantly reducing training time while retaining the ability to model long-range dependencies.

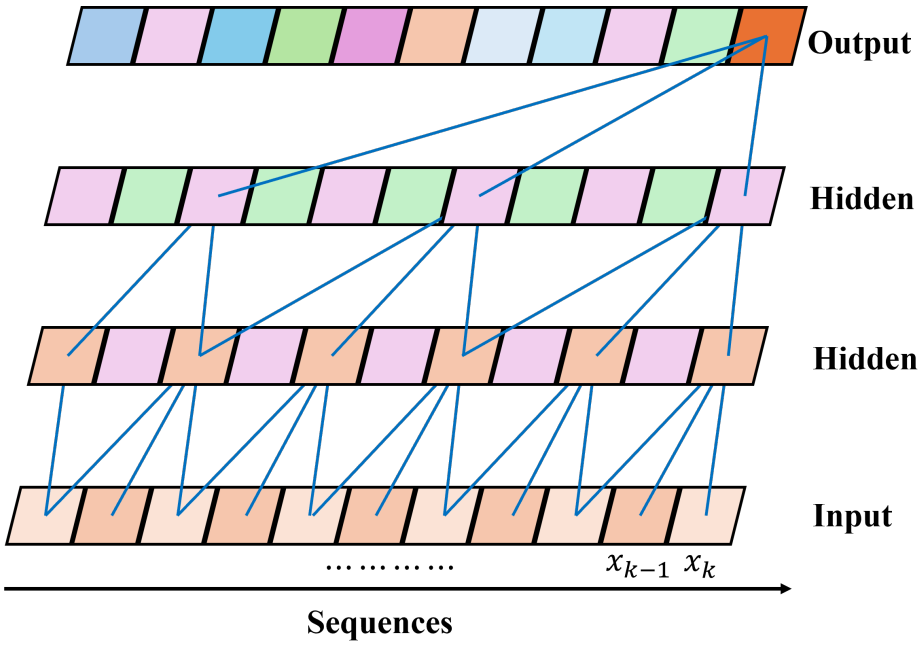

Figure 7. A dilation causal convolution to extract temporal features layer by layer.

Zero padding results in the output sequence containing extra padding elements at the end, which must be removed. To ensure that the output has the same length as the input, the convolution uses padding $p = (k-1)m$ and a cropping operator is applied:

$$\text{Chomp}_p(Z)[:,:,t] = Z[:,:,t], \quad t = 0,\ldots,L-1. \tag{18}$$

This step removes the extra padded elements and guarantees that the convolution is strictly causal.

Therefore, each TCN architecture requires stacking two dilated causal convolutions, each followed by a ReLU activation function and a dropout operation with probability ρ. The residual connection is formulated as:

$$Y = \text{ReLU}\big(R(\mathcal{X}) + F(\mathcal{X})\big) \tag{19}$$

where $F(\cdot)$ denotes stacked dilated convolution, $R(\cdot)$ represents 1×1 convolution. Residual connections stabilize the optimization process, enabling effective training of deeper

temporal convolutional neural networks. The structure of the final TCN block is shown in Figure 8.

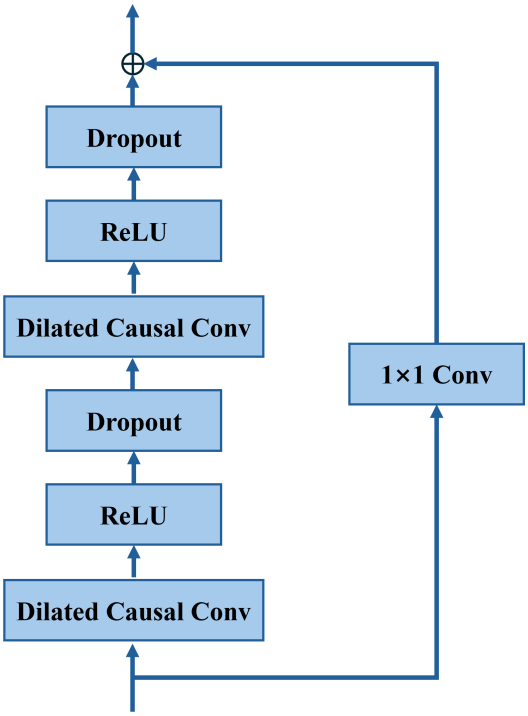

Figure 8. A causal dilation temporal convolution to extract temporal features layer by layer.

Multiple temporal blocks are stacked with exponentially increasing dilation factors $d_\ell = 2^\ell$ ($\ell = 0, \ldots, N-1$). The output of the stacked TCN is as follows:

$$\hat{\mathcal{Y}} = T_{N-1}\big(\cdots T_1(T_0(\mathcal{X}))\cdots\big) \tag{20}$$

where $T(\cdot)$ represents temporal block operations, the effective receptive field of the stacked TCN is as follows:

$$\mathcal{Q} = 1 + (k-1)\sum_{\ell=0}^{L-1} d_\ell = 1 + (k-1)(2^L - 1). \tag{21}$$

Note that $\mathcal{Q}$ quantifies the number of past steps that can be integrated into each output. This characteristic is crucial for predicting carbon emissions, as carbon emissions depend not only on instantaneous power generation and consumption but also on patterns formed through accumulation over hours or days.

The complexity of a single TCN layer is $O\big(L \cdot k \cdot 2d \cdot C_{\text{out}}\big)$. Unlike RNN-based models whose complexity grows with sequential steps due to recursive dependencies, TCN layers allow parallel computation across all time steps, enabling efficient training on large-scale datasets.

Recurrent neural network architectures excel at time series modeling, but suffer from gradient vanishing issues, resulting in high training costs and difficulty capturing ultra-long dependencies. Transformer models, while powerful, require quadratic attention complexity $O(T^2)$, making them less suitable for long-sequence power data. In contrast, Time Convolutional Networks (TCNs) achieve an ideal balance by integrating parallel processing, long-range dependency modeling, and computational efficiency. These advantages make TCNs particularly well-suited for predicting real-time regional carbon emissions—a domain where both timeliness and accuracy are critical.

The activation and layer-dimension settings were selected based on best practices reported in recent spatio-temporal forecasting literature [25–28]. ReLU activation provides stable gradient propagation, while three GCN layers and two TCN blocks yield sufficient receptive-field coverage for the IEEE 14-bus topology without over-parameterization. This configuration has proven effective in prior power-system applications [25,26].

Overall, the introduction of the TCN layer significantly enhances the model's ability to track the evolutionary patterns of carbon emissions in power systems. By combining causal convolutions with dilated convolutions and residual learning, this framework achieves high predictive accuracy, robustness against non-stationary fluctuations, and scalability for long-term sequence data. This design ensures effective real-time carbon emissions prediction in actual grid operations.

3.3. Training Method

The training of the proposed GCN-based hybrid model is formulated as an optimization problem aimed at minimizing the discrepancy between predicted and actual carbon emission values. The overall objective function is defined as the mean squared error (MSE) between the ground truth carbon emissions Y and the model predictions $\hat{\mathcal{Y}}$:

$$\mathcal{L}(\Theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{22}$$

where N denotes the number of samples in the training set, y_i is the actual carbon emission value of the i-th sample, $\hat{y}_i$ is the corresponding predicted value, and θ represents the set of all trainable parameters in the model.

To prevent overfitting and enhance generalization capabilities, L_2 weight regularization is incorporated into the loss function. The regularized loss function is defined as follows:

$$\mathcal{L}_{\text{reg}}(\Theta) = \mathcal{L}(\Theta) + \lambda\|\Theta\|_2^2, \tag{23}$$

where λ is the regularization hyperparameter controlling the strength of penalty imposed on large weights.

The model parameters are updated iteratively using the Adam optimizer, which combines the advantages of AdaGrad and RMSProp. The update rules for the first and second moment estimates are as follows:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\nabla_\Theta \mathcal{L}_{\text{reg}}(\Theta_t), \tag{24}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2)[\nabla_\Theta \mathcal{L}_{\text{reg}}(\Theta_t)]^2 \tag{25}$$

where m_t and v_t are estimates of the first moment and the second moment of the gradients, respectively, and β_1, $\beta_2 \in [0,1)$ are exponential decay rates. The parameters are then updated according to:

$$\Theta_{t+1} = \Theta_t - \eta\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{26}$$

where $\hat{m}_t = m_t/(1-\beta_1^t)$ and $\hat{v}_t = v_t/(1-\beta_2^t)$ are bias-corrected moment estimates, η is the learning rate, and ϵ is a small constant added for numerical stability.

To further improve training efficiency and avoid overfitting, we employ a learning rate scheduler that reduces the learning rate when the validation loss plateaus. The update rule is defined as:

$$\eta_{t+1} = \eta_t \cdot \gamma \cdot \mathbb{I}(\mathcal{L}_{\text{val}}(t) > \mathcal{L}_{\text{val}}(t-1)) \tag{27}$$

where γ is the decay factor, $\mathcal{L}_{\text{val}}$ is the validation loss, and $\mathbb{I}(\cdot)$ is the indicator function. Gradient backpropagation is performed through both temporal and graph convolutional layers. The gradient with respect to the weights of the GCN layer is computed as:

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \widetilde{A}^T \frac{\partial \mathcal{L}}{\partial H^{(l+1)}} \cdot \sigma'(Z^{(l)}) \cdot (H^{(l)})^T \tag{28}$$

where $Z^{(l)} = \widetilde{A} H^{(l)} W^{(l)}$, σ' denotes the derivative of the activation function, and $\widetilde{A}$ is the normalized adjacency matrix.

An early stopping strategy is adopted to terminate training when the validation loss does not improve for a predefined number of epochs. This ensures the model does not overfit while maintaining computational efficiency. The entire training process is summarized in Algorithm 1. This comprehensive training strategy ensures stable convergence, effective generalization, and high predictive accuracy for the proposed spatio-temporal model.

Algorithm 1 Training Procedure for GCN-based Model

Require: Training data $\mathcal{D}_{\text{train}}$, validation data $\mathcal{D}_{\text{val}}$
Ensure: Trained model parameters Θ
Initialize parameters Θ
while not converged **do**
  Sample a mini-batch from $\mathcal{D}_{\text{train}}$
  Forward pass to compute $\hat{\mathcal{Y}}$
  Compute loss $\mathcal{L}_{\text{reg}}(\Theta)$ using Equation (23)
  Backpropagate gradients
  Update parameters using Adam optimizer Equation (26)
  Compute validation loss $\mathcal{L}_{\text{val}}$
  if $\mathcal{L}_{\text{val}}$ does not improve for K epochs **then**
    Break
  end if
  Update learning rate using Equation (27)
end while

The following experiments will validate the effectiveness of the proposed GCN-based approach for predicting carbon emissions.

4. Experimental Studies on GCN-Based Approach

4.1. IEEE 14-Bus Systems Verification

The IEEE 14-bus system is a widely used standard test network in power system analysis. It consists of 14 buses, 5 generators, 11 loads, and 20 transmission lines, operating at voltage levels of 60–132 kV with a total installed capacity of approximately 1.1 GW. Despite its compact structure, the system incorporates diverse topological features including radial and loop configurations. This design captures the heavy load characteristics of urban distribution networks while preserving voltage-reactive power coupling phenomena at transmission levels.

Despite its scaled-down size, the IEEE 14-bus system fully preserves transmission-distribution coupling, multi-voltage levels, and the multi-source-grid-load structure. It flexibly maps the spatiotemporal distribution of fossil units, renewable energy sources, and loads. Its publicly available and validated topology and unit carbon emissions data provide a standardized, reproducible testing environment for constructing "node-branch" carbon flow tracking, dynamiccarbon emissions, and carbon emission trend prediction models [29]. The overall structure of the IEEE 14-bus electric power system is shown in Figure 9.

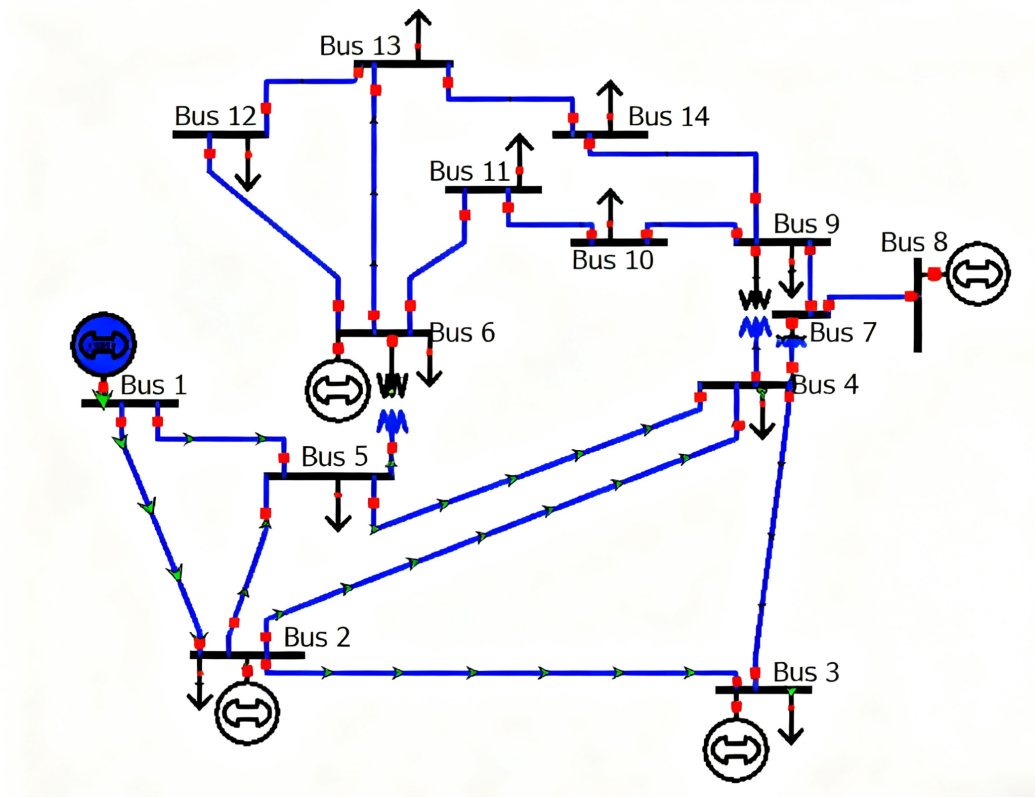

Figure 9. IEEE 14-bus electric power systems [30].

Each node in the system is characterized by a feature vector containing six attributes: active power load (P_d), reactive power load (Q_d), active power generation (P_g), reactive power generation (Q_g), voltage magnitude (V), and voltage angle (θ). The target variable is the total carbon emission (C_e) of the system at each time step, calculated as the sum of emissions from all generating nodes. Emissions are computed based on generator type and vary with load factor and time of day. At each time step t, each node input x_t contains the following data features:

$$x_t = \begin{bmatrix} P_d^T & Q_d^T & P_g^T & Q_g^T & V^T & \theta^T \end{bmatrix}^T \tag{29}$$

where $x_t \in \mathcal{R}^F$. The input feature matrix $X \in \mathcal{R}^{N \times F}$ and the output $Y \in \mathcal{R}$ are normalized using StandardScaler before training.

The dynamic load and generation profiles were obtained from the open-source IEEE 14-bus benchmark dataset. Time-series variations were created by sampling 2000 steps from load-generation profiles in the MATPOWER simulation environment under random perturbations of ±5 % to emulate daily fluctuations. Carbon-emission values at each time step were computed using generator-specific emission coefficients published by Ma et al. [7] and Luo et al. [12].

Remark 1. *To prevent data leakage and ensure a fair evaluation, the StandardScaler was fit only on the training set to compute the mean and standard deviation of each feature. The same scaling parameters were then applied to transform the validation and test sets, ensuring that no information from these sets influenced the normalization process.*

The GCN-based model consists of three GCN layers with hidden dimension 128, followed by a temporal convolutional network with channel sizes [64, 32]. The sequence length for TCN is set to 10 time steps. The model is trained using the Adam optimizer with a learning rate of 0.001 and early stopping based on validation loss. During the training phase, data is fed into the model, and Algorithm 1 is executed to produce the trained model. The simulation consists of 2000 time steps, with the final 300 used as the test set. The average power generation of each generation node is shown in Table 1. The system's total average power generation is 276.06 MW, and the total average load is 257.99 MW, indicating the system operates within a reasonable power balance range.

Table 1. Power Generation Statistics for Generating Nodes.

Node	Average Power Output (MW)
1	232.14
2	37.76
3	1.99
6	2.07
8	2.10

During the training phase, the loss function values for both the training and validation sets decrease as the number of training iterations increases. In the testing phase, the model's output values closely approximate the true values. To quantitatively evaluate the predictive accuracy of the proposed model, the following commonly used metrics are employed:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \quad \mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \quad \mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{30}$$

where y is the actual value, $\hat{y}$ is the predicted value, $\bar{y}$ is the mean of the actual values.

It should be noted that, although the model is designed for node-level carbon emission prediction, the evaluation in this specific case study is performed at the system level by aggregating the predicted emissions of all nodes into a total value. This system-level total is used to compute the performance metrics: RMSE, MAE, and R^2—enabling a straightforward and consistent comparison with baseline models.

To validate the effectiveness of the proposed model, comparative experiments are against three established baseline models on an identical test set. The experimental results, as illustrated in Figure 10, demonstrate the performance comparison across all methods. The selected benchmark models include support vector regression (SVR), ARIMA, and LSTM networks, representing classical machine learning, traditional time series analysis, and deep learning approaches, respectively. For comprehensive evaluation, three key metrics: RMSE, R^2, and MAE are computed for all models. The quantitative results are summarized in Table 2, while a comparative bar chart illustrating the performance across these metrics is presented in Figure 11.

Table 2. Comparison of key metrics across different models.

Model	RMSE ($kgCO_2$)	R^2	MAE ($kgCO_2$)
GCN-based	8566.9159	0.8431	6326.2168
SVR	15,918.2416	0.4584	13,854.6673
LSTM	10,963.1422	0.7431	8769.0145
ARIMA	8618.3974	0.8356	6390.3763

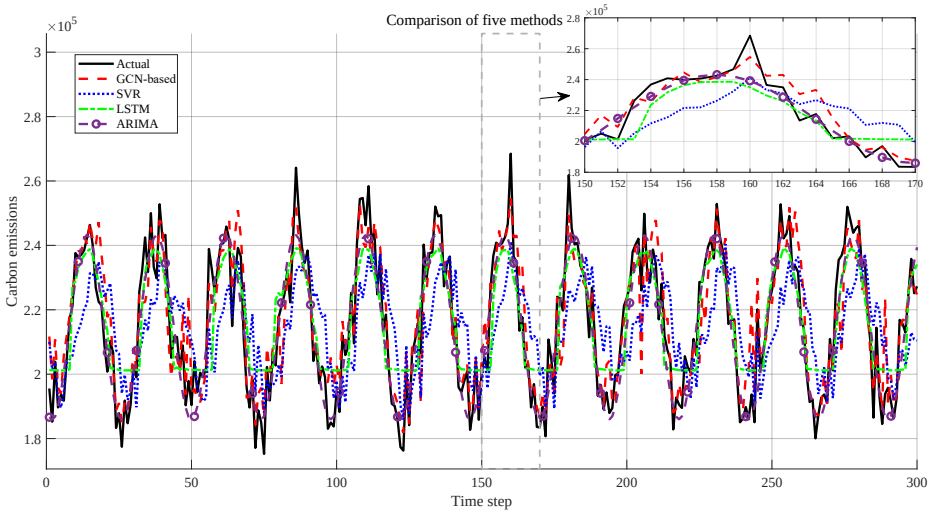

Figure 10. The comparison experimental results.

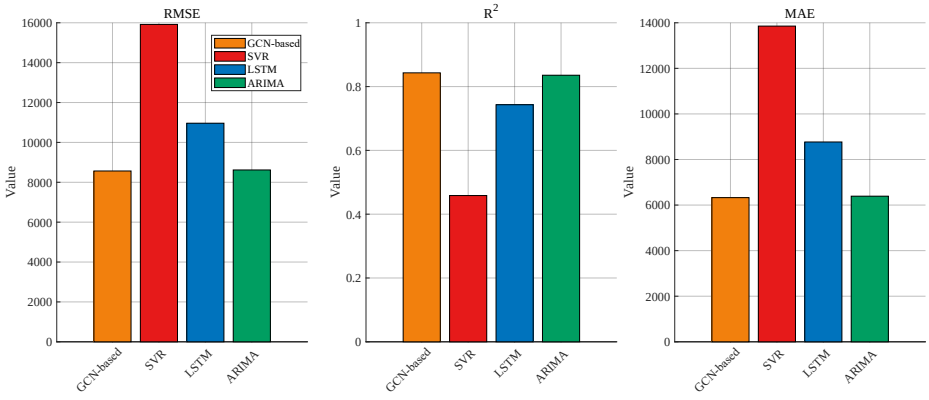

Figure 11. The key metrics bar charts for different models.

Based on the comparative results of key performance metrics presented in Table 2, the proposed model demonstrates superior performance in carbon emission prediction tasks. Specifically, the model achieves a RMSE of 8566.9159 $kgCO_2$, which is significantly lower than that of SVR and LSTM, and slightly better than that of ARIMA. In terms of R^2, the proposed model attains the highest value of 0.8431, substantially outperforming SVR and LSTM, and also marginally exceeding ARIMA. Furthermore, for MAE, the proposed model yields 6326.2168 $kgCO_2$, considerably lower than SVR and LSTM, and slightly superior to ARIMA. Collectively, these three key metrics confirm that the proposed model exhibits notable advantages in both prediction accuracy and stability, validating the effectiveness of its integrated graph convolutional and temporal convolutional architecture in capturing complex spatiotemporal dependencies.

To validate the effectiveness of the proposed GCN-based model for predicting carbon emissions in power systems, a progressive ablation experiment is designed. Using the complete model as a baseline, the TCN layer and pooling layer are sequentially removed

or modified to construct six variants. By comparing the performance of each variant across identical datasets and evaluation metrics, the contribution of each model component to prediction accuracy is quantitatively analysed, thereby clarifying their respective essentiality. Table 3 details the configuration of the ablation experiments.

Table 3. Configuration of ablation experiment models.

Model	Pooling Strategy	TCN Layer	GCN Layer
GCN-based	Dual Pooling	TCN	GCN
GCN-GAT	Dual Pooling	No	GAT
GCN-SAGE	Dual Pooling	No	SAGE
DualPooling	Dual Pooling	No	GCN
MaxPooling	Max Pooling	No	GCN
MeanPooling	Mean Pooling	No	GCN
NoPooling	No Pooling	No	GCN

In Table 3, the GCN-GAT model replaces the TCN layer with a graph attention network layer(GAT), while the GCN-SAGE model substitutes the TCN layer with a sampling and aggregation layer. The pooling layer operations categorize pooling into single max-pooling, mean pooling, and no pooling. The experimental results, as illustrated in Figure 12, demonstrate the performance comparison across all ablation experiment models.

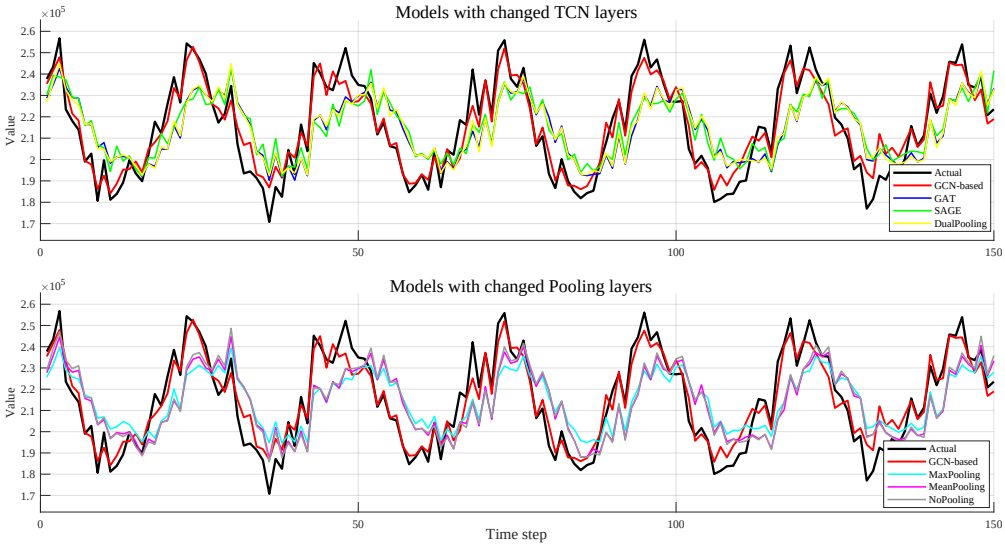

Figure 12. The results of ablation experiments.

The comparative results of the three key performance metrics in the ablation experiment are shown in Figure 13. The proposed GNN-based model achieves the lowest RMSE and MAE, while simultaneously attaining the highest R^2 value, significantly outperforming all ablation variants. This outstanding performance stems from the synergistic integration of mean and max pooling layers with the TCN, which effectively captures both the spatial graph structure and temporal dependencies inherent in carbon emission dynamics. In contrast, models employing a single aggregation mechanism exhibit higher errors due to incomplete graph representation; while removing the TCN leads to a significant performance decline, highlighting the critical role of temporal modeling. Employing alternative aggregators such as GAT or SAGE also results in a slight performance drop, indicating that the combination of standard GCN and TCN provides the most effective feature extraction and sequence modelling approach for this task.

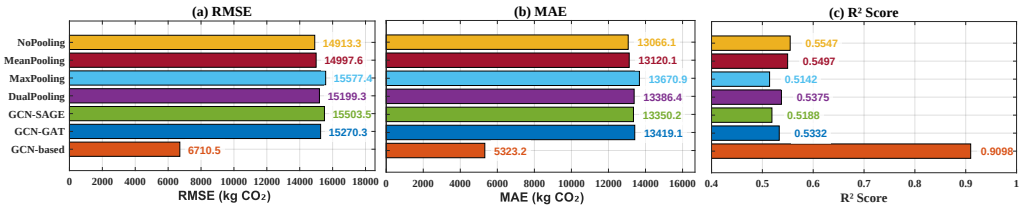

Figure 13. The results of ablation experiments.

4.2. Robustness Tests

In the real-world grid operating environment, the data acquisition and transmission process is fraught with uncertainty. To ensure the reliability in actual scenarios, robustness testing experiments are required. The proposed GCN-based model is evaluated under various challenging conditions, including data noise, missing data, and operational scenario variations. The corresponding results are systematically presented in Table 4 and Figure 14.

Table 4. The comparative results of the robustness experiments.

Model	RMSE ($kgCO_2$)	MAE ($kgCO_2$)	R^2
Baseline	9605.01	6556.59	0.8064
Gaussian Noise 0.05	9693.92	6622.94	0.8028
Gaussian Noise 0.1	9674.88	6628.84	0.8036
Gaussian Noise 0.2	9340.27	6501.82	0.8169
Uniform Noise 0.05	9561.43	6521.51	0.8081
Uniform Noise 0.1	9629.72	6613.13	0.8054
Uniform Noise 0.2	9672.01	6632.33	0.8037
Random Missing 0.1	9978.91	6867.45	0.7910
Random Missing 0.2	10,279.39	7431.31	0.7782
Random Missing 0.3	10,708.84	7634.64	0.7593
Block Missing 0.1	9602.47	6601.19	0.8065
Block Missing 0.2	9664.98	6791.65	0.8040
Block Missing 0.3	10,450.40	7508.43	0.7708
Feature Missing 0.1	11,060.08	7733.86	0.7433
Feature Missing 0.2	11,170.17	7771.10	0.7381
Feature Missing 0.3	10,454.12	7390.90	0.7706
Renewable 0.3	9470.38	6565.78	0.8119
Renewable 0.5	9413.37	6612.50	0.8143
Renewable 0.7	9385.58	6684.25	0.8155
Renewable 0.9	9389.30	6769.28	0.8154

The robustness against data noise is quantified by adding Gaussian and uniform noise to node features, formulated as:

$$X_{\text{noisy}} = X + \eta \cdot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0,1) \text{ or } \mathcal{U}(-1,1) \tag{31}$$

where η is the noise level. In the test, the values of η are 0.05, 0.1 and 0.2 respectively.

As shown in Table 4, the model's performance metrics exhibit only minor deviations across all tested noise levels. Notably, under Gaussian noise conditions with η = 0.2, the model slightly outperforms the baseline. This indicates that the model's feature extraction process possesses inherent smoothness and stability, enabling it to effectively filter random disturbances while maintaining performance without significant degradation.

The model is also tested under missing data scenarios, including random, block, and feature-specific missingness. The masking process is defined as:

$$X_{\text{missing}} = M \odot X, \quad M_{ij} \sim \text{Bernoulli}(1-p) \tag{32}$$

where p is the missing ratio and $\odot$ denotes element-wise multiplication. The elements in M satisfy a Bernoulli distribution. In the test, the values of p are 0.1, 0.2 and 0.3 respectively. Under three scenarios of random missing, block missing, and feature missing, model performance exhibited a reasonable decline with increasing missing rates. When the random missing rate $p = 0.3$, the RMSE rose to 10,708.84 $kgCO_2$, while R^2 decreased to 0.7593, indicating the model's sensitivity to high proportions of random missing. However, under block missing ($p = 0.1$) and feature missing ($p = 0.1$) conditions, the RMSE values were 9602.47 and 11,060.08 respectively, indicating the model exhibits greater tolerance towards locally continuous missing data than towards globally lost features. These findings indicate that the model partially compensates for local information loss through graph structure propagation mechanisms, yet its adaptability to system-level feature omissions remains subject to improvement.

Additionally, to evaluate the model's adaptability under different grid operating conditions, operational scenarios with renewable energy penetration rates varying from 30% to 90% are simulated. The integration of renewable energy is formulated as:

$$P'_g = P_g \cdot (1 - \beta) + P_{\text{renewable}}, \quad \beta \in [0.3, 0.9] \tag{33}$$

where P_g represents the original generator output, P'_g denotes the adjusted output, β indicates the renewable energy penetration rate, and $P_{\text{renewable}}$ represents the renewable energy generation. By simulating the gradual increase in renewable energy penetration rates, the model's capacity to adapt to evolving grid generation structures was assessed using Equation (33). Notably, as the β value increased, the model's predictive accuracy showed a slight improvement. This indicates that the proposed spatio-temporal model effectively captures the dynamic impact of variable renewable energy integration on carbon emissions, maintaining reliable performance even during significant shifts in generation composition.

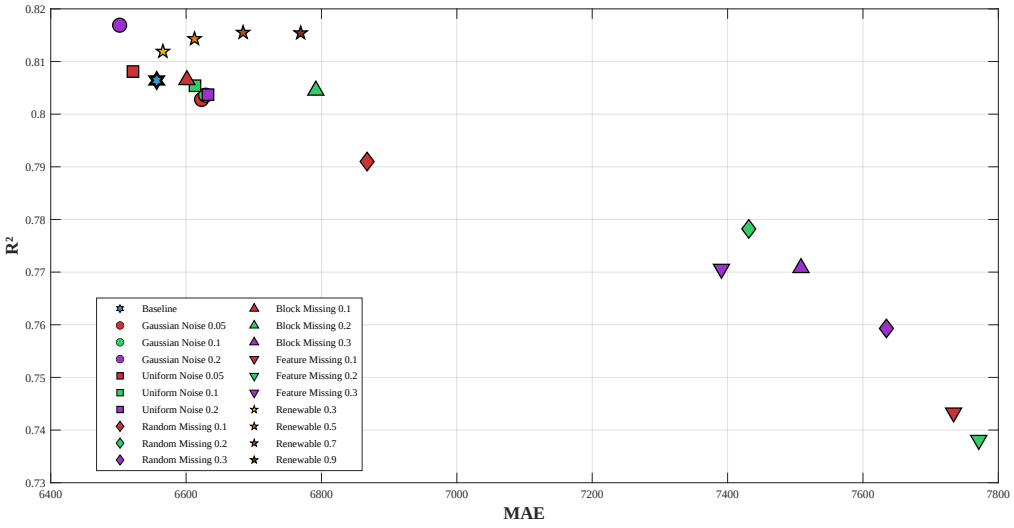

Figure 14. Model Performance under robustness test scenarios.

5. Conclusions

This paper proposes an innovative GCN-based hybrid framework for real-time prediction of regional carbon emissions. By integrating the structural advantages of graph neural networks with the temporal modeling capabilities of temporal recurrent neural networks, this model effectively addresses the inherent spatiotemporal complexity in power grid carbon emission forecasting. The model first constructs a graph structure representation of the power grid, enabling the GCN component to capture spatial dependencies by ag-

gregating information from neighboring nodes. This enriches feature representations for each node by leveraging contextual information across the entire grid. Concurrently, mean and max pooling operations are employed to enhance the model's ability to characterize baseline and extreme emission patterns at the graph level, thereby providing a more robust representation of the system's carbon emission state. On the temporal dimension, the TCN module employs extended causal convolutions to model long-range dependencies while preserving temporal causality. Residual connections within the TCN module further stabilize the training process and enable modeling of deep temporal relationships—crucial for capturing the cumulative nature of carbon emissions over time. Experimental validation on the IEEE 14-node system demonstrates the model's outstanding performance. Compared to traditional approaches, this framework integrates spatio-temporal modeling into a unified architecture, overcoming the limitations of purely temporal or spatial models. It efficiently handles variable-length sequences and scales to large scale multiregion power systems. Future research will focus on: incorporating renewable energy forecasting uncertainty, extending to multi-energy coupling systems, and exploring practical deployment solutions at regional grid control centers.

Despite its promising performance, the proposed model has certain limitations. First, it relies on accurate and up-to-date grid topology data, which may not always be available in real-time operations. Second, while the model scales well to the IEEE 14-bus system, its computational complexity may increase for larger networks with hundreds of nodes, necessitating further optimization for real-time deployment. Additionally, the current model does not explicitly account for uncertainties in renewable generation or multi-energy interactions, which are critical for practical carbon emission forecasting.

Future research will focus on:

- Integrating probabilistic forecasting techniques to handle renewable energy uncertainty, such as using Bayesian neural networks or Monte Carlo dropout within the TCN framework.
- Extending the model to multi-energy coupling systems by incorporating heat and gas network data, using coupled graph structures to capture cross-energy dependencies.
- Exploring model compression and distributed training strategies to enhance scalability for large-scale power grids.

Author Contributions: Conceptualization, Q.Z.; methodology, X.C.; software, X.C.; validation, X.C.; formal analysis, Q.J.; investigation, C.S.; resources, H.C.; data curation, C.S.; writing—original draft preparation, J.C.; writing—review and editing, H.C.; visualization, Q.J.; supervision, Q.Z.; project administration, Q.Z.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Authors Qian Zhao, Jianhua Chen, Qianwei Jia, and Cong Sun were employed by the company State Grid Jibei Electric Power Company Ltd., Beijing, China. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GCNs	Graph Convolutional Networks
TCNs	Temporal Convolutional Networks

LEAP	Long-range Energy Alternatives Planning
ARIMA	Auto-Regressive Integrated Moving Average
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
GNNs	Graph Neural Networks

References

1. Li, Y.; Yang, X.; Du, E.; Liu, Y.; Zhang, S.; Yang, C.; Zhang, N.; Liu, C. A review on carbon emission accounting approaches for the electricity power industry. *Appl. Energy* **2024**, *359*, 122681. [CrossRef]
2. Albuquerque, B.S.D.; Tostes, M.E.D.L.; Bezerra, U.H.; Carvalho, C.C.M.D.M.; Nascimento, A.L.L.D. Use of Distributed Energy Resources Integrated with the Electric Grid in the Amazon: A Case Study of the Universidade Federal do Pará Poraquê Electric Boat Using a Digital Twin. *Machines* **2024**, *12*, 803. [CrossRef]
3. Guo, Z.; Siew, W.H.; Li, Q.; Shi, W. On the Lightning Attachment Process of Wind Turbine–Observation, Experiments and Modelling. *Machines* **2025**, *13*, 704. [CrossRef]
4. Yang, L.; Mei, L.; Chen, Y.; Hao, Y.; Li, L.; Wu, J.; Mao, X. Prediction Method for Mechanical Characteristic Parameters of Weak Components of 110 kV Transmission Tower under Ice-Covered Condition Based on Finite Element Simulation and Machine Learning. *Machines* **2024**, *12*, 652. [CrossRef]
5. Li, P.; Wu, W.; Wang, X.; Xu, B. A Data-Driven Linear Optimal Power Flow Model for Distribution Networks. *IEEE Trans. Power Syst.* **2023**, *38*, 956–959.
6. Tian, P.; Jin, Y.; Xie, N.; Wang, C.; Huang, C. Power Flow Calculation for VSC-Based AC/DC Hybrid Systems Based on Fast and Flexible Holomorphic Embedding. *J. Mod. Power Syst. Clean Energy* **2024**, *12*, 1370–1382.
7. Ma, M.; Li, Y.; Du, E.; Jiang, H.; Zhang, N.; Wang, W.; Wang, M. Calculating Probabilistic Carbon Emission Flow: An Adaptive Regression-Based Framework. *IEEE Trans. Sustain. Energy* **2024**, *15*, 1576–1588. [CrossRef]
8. Zhang, X.; Zhu, H.; Cheng, Z.; Shao, J.; Yu, X.; Jiang, J. A review of carbon emissions accounting and prediction on the power grid. *Electr. Eng.* **2025**, *107*, 7561–7574. [CrossRef]
9. Wang, T.; Watson, J. Scenario analysis of China's emissions pathways in the 21st century for low carbon transition. *Energy Policy* **2010**, *38*, 3537–3546. [CrossRef]
10. McPherson, M.; Karney, B. Long-term scenario alternatives and their implications: LEAP model application of Panama's electricity sector. *Energy Policy* **2014**, *68*, 146–157. [CrossRef]
11. Maji, D.; Shenoy, P.; Sitaraman, R.K. Multi-Day Forecasting of Electric Grid Carbon Intensity Using Machine Learning. *ACM SIGENERGY Energy Inform. Rev.* **2023**, *3*, 19–33.
12. Luo, J.; Zhuo, W.; Liu, S.; Xu, B. The Optimization of Carbon Emission Prediction in Low Carbon Energy Economy Under Big Data. *IEEE Access* **2024**, *12*, 14690–14702. [CrossRef]
13. Lau, E.; Yang, Q.; Forbes, A.; Wright, P.; Livina, V. Modelling carbon emissions in electric systems. *Energy Convers. Manag.* **2014**, *80*, 573–581. [CrossRef]
14. Wang, H.; Li, B.; Khan, M.Q. Prediction of Shanghai Electric Power Carbon Emissions Based on Improved STIRPAT Model. *Sustainability* **2022**, *14*, 13068. [CrossRef]
15. Chen, J.; Zheng, L.; Che, W.; Liu, L.; Huang, H.; Liu, J.; Xing, C.; Qiu, P. A method for measuring carbon emissions from power plants using a CNN-LSTM-Attention model with Bayesian optimization. *Case Stud. Therm. Eng.* **2024**, *63*, 105334.
16. Leerbeck, K.; Bacher, P.; Junker, R.G.; Goranović, G.; Corradi, O.; Ebrahimy, R.; Tveit, A.; Madsen, H. Short-term forecasting of CO2 emission intensity in power grids by machine learning. *Appl. Energy* **2020**, *277*, 115527. [CrossRef]
17. Han, Z.; Cui, B.; Xu, L.; Wang, J.; Guo, Z. Coupling LSTM and CNN Neural Networks for Accurate Carbon Emission Prediction in 30 Chinese Provinces. *Sustainability* **2023**, *15*, 13934. [CrossRef]
18. Qiao, W.; Lu, H.; Zhou, G.; Azimi, M.; Yang, Q.; Tian, W. A hybrid algorithm for carbon dioxide emissions forecasting based on improved lion swarm optimizer. *J. Clean. Prod.* **2020**, *244*, 118612. [CrossRef]
19. Wei, X.; Xu, Y. Research on carbon emission prediction and economic policy based on TCN-LSTM combined with attention mechanism. *Front. Ecol. Evol.* **2023**, *11*, 1270248. [CrossRef]
20. Han, Y.; Hao, Y.; Feng, M.; Chen, K.; Xing, R.; Liu, Y.; Lin, X.; Ma, B.; Fan, J.; Geng, Z. Novel STAttention GraphWaveNet model for residential household appliance prediction and energy structure optimization. *Energy* **2024**, *307*, 132582.
21. Lopez-Garcia, T.B.; Domínguez-Navarro, J.A. Optimal Power Flow With Physics-Informed Typed Graph Neural Networks. *IEEE Trans. Power Syst.* **2025**, *40*, 381–393. [CrossRef]
22. Wu, T.; Zhang, Y.J.A.; Liu, Y.; Lau, W.C.; Xu, H. Missing Data Recovery in Large Power Systems Using Network Embedding. *IEEE Trans. Smart Grid* **2021**, *12*, 680–691. [CrossRef]
23. Wu, T.; Scaglione, A.; Arnold, D. Complex-Value Spatiotemporal Graph Convolutional Neural Networks and Its Applications to Electric Power Systems AI. *IEEE Trans. Smart Grid* **2024**, *15*, 3193–3207. [CrossRef]

24. Hossain, R.R.; Huang, Q.; Huang, R. Graph Convolutional Network-Based Topology Embedded Deep Reinforcement Learning for Voltage Stability Control. *IEEE Trans. Power Syst.* **2021**, *36*, 4848–4851. [CrossRef]
25. Chen, L.; Zhou, S.; Liang, X.; Lu, W.; Xia, M.; Weng, L.; Geng, J.; Liu, J. Predicting Power Dispatch for Unit Commitment Problems Using Graph-Temporal Convolutional Networks With Constrained Learning. *IEEE Trans. Ind. Inform.* **2025**, *21*, 5734–5745. [CrossRef]
26. Li, Y.; Song, L.; Zhang, S.; Kraus, L.; Adcox, T.; Willardson, R.; Komandur, A.; Lu, N. A TCN-Based Hybrid Forecasting Framework for Hours-Ahead Utility-Scale PV Forecasting. *IEEE Trans. Smart Grid* **2023**, *14*, 4073–4085.
27. Zhou, X.; Pang, C.; Zeng, X.; Jiang, L.; Chen, Y. A Short-Term Power Prediction Method Based on Temporal Convolutional Network in Virtual Power Plant Photovoltaic System. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 1–10. [CrossRef]
28. Wang, Y.; Chen, J.; Chen, X.; Zeng, X.; Kong, Y.; Sun, S.; Guo, Y.; Liu, Y. Short-Term Load Forecasting for Industrial Customers Based on TCN-LightGBM. *IEEE Trans. Power Syst.* **2021**, *36*, 1984–1997.
29. Ramos, A.C.S.; Freitas, F.D.; Zizzo, G. Economic Dispatch Problem Solution via Holomorphic Embedding Method on IEEE 14-Bus System for Different Loading Scenarios. *IEEE Trans. Ind. Appl.* **2024**, *60*, 2664–2672. [CrossRef]
30. Vazquez-Rodriguez, S.; Duro, R. A genetic based technique for the determination of power system topological observability. In Proceedings of the Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Lviv, Ukraine, 8–10 September 2003; pp. 48–52.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.