applied sciences

MDPI

Article

GCN–Informer: A Novel Framework for Mid-Term Photovoltaic Power Forecasting

Wei Zhuang [1,*], Zhiheng Li [1], Ying Wang [1], Qingyu Xi [1] and Min Xia [2]

[1] The School of Computer, Nanjing University of Information Science and Technology, Nanjing 210044, China; 20211221022@nuist.edu.cn (Z.L.); wyingnvzi@163.com (Y.W.); 202212490411@nuist.edu.cn (Q.X.)
[2] Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China; xiamin@nuist.edu.cn
* Correspondence: zw@nuist.edu.cn

Abstract: Predicting photovoltaic (PV) power generation is a crucial task in the field of clean energy. Achieving high-accuracy PV power prediction requires addressing two challenges in current deep learning methods: (1) In photovoltaic power generation prediction, traditional deep learning methods often generate predictions for long sequences one by one, significantly impacting the efficiency of model predictions. As the scale of photovoltaic power stations expands and the demand for predictions increases, this sequential prediction approach may lead to slow prediction speeds, making it difficult to meet real-time prediction requirements. (2) Feature extraction is a crucial step in photovoltaic power generation prediction. However, traditional feature extraction methods often focus solely on surface features, and fail to capture the inherent relationships between various influencing factors in photovoltaic power generation data, such as light intensity, temperature, and more. To overcome these limitations, this paper proposes a mid-term PV power prediction model that combines Graph Convolutional Network (GCN) and Informer models. This fusion model leverages the multi-output capability of the Informer model to ensure the timely generation of predictions for long sequences. Additionally, it harnesses the feature extraction ability of the GCN model from nodes, utilizing graph convolutional modules to extract feature information from the ‘query’ and ‘key’ components within the attention mechanism. This approach provides more reliable feature information for mid-term PV power prediction, thereby ensuring the accuracy of long sequence predictions. Results demonstrate that the GCN–Informer model significantly reduces prediction errors while improving the precision of power generation forecasting compared to the original Informer model. Overall, this research enhances the prediction accuracy of PV power generation and contributes to advancing the field of clean energy.

Keywords: photovoltaic power generation; informer; deep learning; graph convolution network

Citation: Zhuang, W.; Li, Z.; Wang, Y.; Xi, Q.; Xia, M. GCN–Informer: A Novel Framework for Mid-Term Photovoltaic Power Forecasting. *Appl. Sci.* **2024**, *14*, 2181. https://doi.org/10.3390/app14052181

Academic Editors: Sonia Leva, Emanuele Ogliari and Alessandro Niccolai

Received: 12 January 2024
Revised: 2 March 2024
Accepted: 3 March 2024
Published: 5 March 2024

1. Introduction

With the rapid economic and social development, both industrial and residential electricity consumption have been experiencing exponential growth. Furthermore, the excessive utilization of non-renewable energy sources such as fuel oil and coal has accelerated the depletion of energy resources and exacerbated the spread of greenhouse effects [1] on our planet. As a result, power generation systems based on renewable energy have emerged as a formidable force [2], gradually replacing fossil fuel-based power generation, hydroelectricity, and nuclear power, firmly establishing their position within the power grid [3].

The report from China’s National Energy Administration indicates that by the close of 2021, the country’s renewable energy installations had surged to a staggering 934 million kilowatts. Solar power, impressively, accounted for over 250 million kilowatts of this, making up 27.1% of the total [4]. Yet, beneath this rapid advancement lurks a challenge inherent to photovoltaic systems: their unpredictable stability and output cycles present significant hurdles to the seamless deployment of the power grid.

Precise forecasting in the photovoltaic power generation sector aims to provide a secure and reliable electricity usage environment for various clientele, catering to diverse power scenarios and consumption needs [5]. However, due to the unique nature of electrical energy, generated power cannot be directly stored over extended periods. Hence, a dynamic equilibrium between generation and consumption must be sought [6]. Given the uncertainties of renewable energy generation environments, predicting power output has emerged as a pivotal challenge. In recent years, substantial strides have been made in photovoltaic power prediction models, with forecast accuracy hinging on photovoltaic generation data. Consequently, for accurate photovoltaic power prediction, the reliability of the data themselves and the selection of their attributes directly influence the precision of the forecasts. Photovoltaic power prediction involves comprehensive consideration of natural factors, including meteorological conditions, temporal elements, weather patterns, regional context, and seasonal influences. It also encompasses operational characteristics of the photovoltaic power station, such as system performance, infrastructure development, and equipment conditions. By leveraging a predictive model and analyzing historical data, the interconnections within the data sources are explored. Striving to meet a certain level of accuracy, this process computes forthcoming power generation data under specific conditions.

In this era of modern technology, the advancement of information technology unfolds in a continuous and swift manner. The age of big data has dawned, and artificial intelligence has permeated the foundational frameworks of various industries. Models employed for photovoltaic power generation forecasting can be broadly categorized into two types: deep learning models and non-deep learning models.

A modeling and prediction framework is developed for photovoltaic power generation data in three regions, using a Random Forest (RF) algorithm optimized by Principal Component Analysis (PCA) and K-Means clustering. PCA and K-Means clustering [7] are employed to extract features that are similar to the prediction time points. The input data are then filtered to reduce interference. Additionally, an optimization algorithm is utilized to quickly select RF parameters, significantly mitigating errors caused by manual filtering factors. However, when addressing regression problems, RF does not provide continuous outputs, which can lead to overfitting when modeling data with certain noise patterns. To overcome this, a photovoltaic power output prediction model based on an empirical mode decomposition–support vector machine (EMD-SVM) is proposed [8]. Each decomposed component sequence and residual sequence is used to train respective Support Vector Machine (SVM) models for prediction. Furthermore, an artificial bee colony algorithm is employed to optimize the SVM model parameters. Finally, the outputs of each model are reconstructed to obtain the final prediction results. The predicted results demonstrate improved accuracy of the proposed model, and the computational speed of the optimized prediction model using the artificial bee colony algorithm is greatly enhanced. However, when dealing with large training samples, the quadratic programming algorithm in SVM consumes substantial machine memory and computation time, rendering it impractical for regression tasks.

Deep learning models simulate the human thought process and are capable of performing complex computational tasks. They have emerged as a popular technique for photovoltaic power prediction.

In the field of photovoltaic power generation prediction, recurrent neural networks (RNNs) have emerged as a leading AI model for accurate forecasting. As a specialized type of deep neural network, RNNs have also found widespread application in photovoltaic power generation prediction [9–12]. Each recurrent layer of an RNN maintains a hidden state, which serves as input for the next recurrent layer and updates as the time series progresses. This structure enables RNNs to handle dependencies in time-series data and comprehend temporal information within the sequences. By harnessing this capability, RNNs can capture patterns and trends in photovoltaic power generation, achieving more accurate predictions than traditional machine learning algorithms. However, one of the most significant issues facing RNNs is the problem of vanishing gradients, which can limit the model's ability to learn complex patterns over extended time frames. This challenge becomes increasingly apparent as the length of the time series increases, potentially constraining the predictive accuracy of RNNs in long-term forecasting scenarios.

In recent years, Long Short-Term Memory (LSTM) has emerged as a leading model for photovoltaic power prediction, addressing the temporal correlations between predicted and historical power that are often overlooked by non-deep learning models [13,14]. The ability of LSTM to capture long-term dependencies in sequential data has made it a suitable choice for photovoltaic power generation forecasting. Hossain et al. [15] combined historical solar irradiance data and the sky forecast type with LSTM for photovoltaic generation forecasting, achieving promising results. This approach utilizes additional meteorological data to enhance the predictive accuracy, further highlighting the versatility of LSTM in photovoltaic forecasting. De V. et al. [16] leveraged LSTM models for photovoltaic prediction in scenarios with limited data availability, showcasing the accuracy of LSTM models in photovoltaic forecasting. This study underscores the robustness of LSTM in scenarios where data scarcity is a challenge, making it a suitable choice for practical applications in photovoltaic power generation prediction. The accumulating evidence [17,18] points to the accuracy and effectiveness of LSTM in photovoltaic power prediction. Its ability to capture temporal patterns and long-term dependencies, combined with additional meteorological data, makes LSTM a promising candidate for enhancing photovoltaic power generation forecasting. The memory unit of an LSTM (Long Short-Term Memory) comprises a cell state and three gates: the input gate, forget gate, and output gate. At each time step, the LSTM receives the input for the current time step and the hidden state from the previous time step, updating the cell state and hidden state through a gated mechanism. This structure allows the LSTM to selectively retain and update information, maintaining effective memory capabilities when processing long time sequences. By stacking multiple LSTM layers, the network can further extract and represent complex features in the time series. However, LSTMs still suffer from the limitations of having too many parameters, leading to overfitting and an inability to effectively handle very long sequence inputs.

Currently, there has been a growing interest in utilizing Graph Convolutional Networks (GCNs) to address the spatio-temporal challenges in photovoltaic power prediction [19–23]. Zhang et al. [24] proposed a spatio-temporal graph neural network prediction method for regional photovoltaic electricity based on weather condition recognition. They employed Graph Convolutional Networks as the predictive model, enabling the capture of spatial features through graph convolutional models and temporal features through gated convolutional neural units. This approach resolves issues related to the inadequate consideration of time correlations in power output and physical connections between photovoltaic stations caused by cloud movement. Zhang et al. [25] introduced an optimal graph-structured approach that considers surrounding spatio-temporal correlations for short-term solar power forecasting. Based on complex network theory, they proposed a novel metric to evaluate graph structure connectivity, thereby enhancing the predictive capabilities of Graph Convolutional Network models. This method tackles the problem of information redundancy commonly encountered when modeling by indiscriminately using all station data without differentiation as seen in most existing approaches that consider spatial information from neighboring sites. While these two methods are innovative in addressing spatio-temporal challenges in photovoltaic power prediction, they do not fully resolve the issue of accurately accounting for sudden extreme weather events. In situations where unexpected weather phenomena, such as severe storms or rapid weather fluctuations, significantly impact photovoltaic power generation, these methods may still face challenges in providing precise predictions. The models primarily rely on historical data and may struggle to adapt to unanticipated, extreme weather conditions, which can lead to inaccuracies in power output forecasts.

While the models we mentioned above have made significant advancements in handling time-series data, they face challenges when applied directly to mid-term photovoltaic power prediction due to the focus of this study on long time-series processing. This gives rise to two major challenges:

(1) How do we address the computational challenges associated with LSTM models in parallel processing when applied to photovoltaic power prediction? Although LSTM models solve the problem of gradient vanishing during RNN backpropagation, they still face computational challenges in parallel processing. The depth of the network leads to high computational complexity and time consumption, making it difficult to meet real-time prediction requirements.

(2) How can we utilize GCN to more accurately extract hidden feature information among different features? Not only do we need to focus on the surface data of features but also we need to capture the relationship of mutual influence among data.

As a deep learning model, the Transformer has successfully tackled natural language processing (NLP) and computer vision (CV) tasks, with its core idea being the self-attention mechanism. Self-attention [26], which allows the model to focus on the importance of different positions within a sequence when processing sequential data, has become a crucial component of the Transformer model. This mechanism computes the correlation scores between each element in the sequence and all other elements, enabling the model to dynamically attend to relevant parts of the input sequence and thus improving its ability to capture information. Furthermore, the self-attention mechanism can capture long-range dependencies within the sequence by directly computing the correlation scores between elements, which is difficult to achieve with traditional recurrent neural networks and convolutional neural networks. However, self-attention also has some drawbacks. Firstly, its computational complexity increases with the length of the sequence, which can pose computational resource challenges when dealing with long sequences. Secondly, the self-attention mechanism ignores positional information in the sequence and requires additional positional encoding to compensate for this limitation. Finally, self-attention may over-focus on certain unimportant information, leading to a decline in model performance. However, due to its high time complexity and memory requirements, it cannot be directly applied to long time-series prediction. In response to these challenges, numerous scholars have contributed several methods to overcome these limitations, yielding promising results. Nikita Kitaev et al. [27] proposed the Reformer model, introducing Locality Sensitive Hashing (LSH) to enhance the attention module, thereby reducing the time complexity. They also introduced reversible residual layers to trade computational efficiency for memory consumption. Iz Beltagy et al. [28] introduced the Longformer model, capable of handling lengthy texts efficiently. This model addresses the issues associated with long sequences. Wang Sinong et al. [29] presented the Linformer model, which decomposes the self-attention matrix with singular values. They also proposed a novel self-attention mechanism that reduces the time complexity to linear. Haoyi Zhou et al. [30] introduced the Informer model, which not only reduces time complexity but also optimizes the input and output of long sequences. This model has demonstrated outstanding performance across various large-scale datasets.

As mentioned above, the Informer model is a relatively new approach for long time-series prediction, and its research still has significant limitations. To enhance the accuracy of photovoltaic power prediction, this study introduces a hybrid forecasting model. The performance of the GCN–Informer model is evaluated by comparing it with other prediction models. The main contributions of this research are as follows:

(1) This paper represents the first creative integration of the attention module in GCN and Informer. Each node in GCN is associated with query vectors and keyword vectors related to other nodes. Edges represent the connection strength between nodes. The strength of these connections is calculated through the dot product of query and key,

which can reflect the interaction level between nodes. This information can be used to build a graph where the edge weights are determined by attention scores. The aim is to extract more complex indirect relationships in weather features in photovoltaic power generation datasets more efficiently (such as a certain combination of temperature and humidity that may cause changes in photovoltaic power generation).

(2) By utilizing the decoder design of the Informer model, this paper can simultaneously consider information from multiple future time steps and generate predictions for these time steps in one shot, addressing the challenge of long sequence output in photovoltaic power forecasting and ensuring that the prediction results meet certain real-time requirements.

As shown below, the rest of the paper is structured as follows: Section 2 presents the methodology of Informer and GCN and the derivation of the relevant formulas. Section 3 performs the experimental preparation and presents the overall framework. In Section 4, a comparative analysis of the experimental results and errors is presented. Section 5 summarizes the paper.

2. Methodology

In this section, we first explain the theoretical support of our proposed model, and then introduce the technical details of the model.

2.1. Overall Framework

Our goal is to make the model learn better representations by making full use of local features.

In Figure 1, we can see how the GCN–Informer model is structured.

The GCN–Informer has two parts, the encoder layer and the decoder layer. In the encoder layer, this paper use ProbSparse self-attention to simplify the way the attention is calculated. The specific implementation is described in detail in Section 2.2.1. In the next down-sampling part, the input sequence is reduced to the original 1/2 through the self-attention distilling mechanism, which greatly improves the efficiency. The specific implementation process will be described in detail in Section 2.2.2. The main task of the encoder layer is to obtain a hidden layer representation of the input data and to ensure consistent dimensions. As shown in Figure 2, the EncoderStack mentioned in the paper is actually composed of multiple encoder and distillation layers. Each horizontal stack represents a separate encoder module. The upper stack is the main stack, which receives the entire input sequence, while the second stack takes half of the input, and the second stack throws away a layer of self-attention to the number of distillation layers, making the output dimensions of the two stacks aligned. Finally, the feature map connecting the two stacks is used as the output of the encoder. In the decoder layer, this paper uses the GCN module to extract the characteristics of adjacent contacts and the use of multi-output output together to solve the efficiency of long sequence prediction. Detailed description of the embodiments will be described in detail in Sections 2.2.3 and 2.3.

2.2. Informer

2.2.1. Probsparse Self-Attention

The ProbSparse self-attention is used in this paper. The previous self-attention is defined as a probabilistic nuclear smoothing method:

$$A(q_i, K, V) = \sum_j \frac{k(q_i, k_j)}{\sum\limits_l k(q_i, k_l)} v_j = E_{p(k_j|q_i)}[v_j] \tag{1}$$

where $p(k_j|q_i) = \frac{k(q_i,k_j)}{\sum\limits_l k(q_i,k_l)}$ and $k(q_i, k_j)$ select asymmetric exponential kernel exp $(\frac{q_i k_j^T}{\sqrt{d}})$.

The attention weight of the i-th query for all keys can be regarded as a probability distribution $p(k_j|q_i)$ and some significant attention weights will make $p(k_j|q_i)$ deviate from the uniform distribution. If $p(k_j|q_i)$ is close to a uniform distribution $q(k_j|q_i) = \frac{1}{L_K}$, then self-attention is redundant, which indicates that the i-th query is lazy.

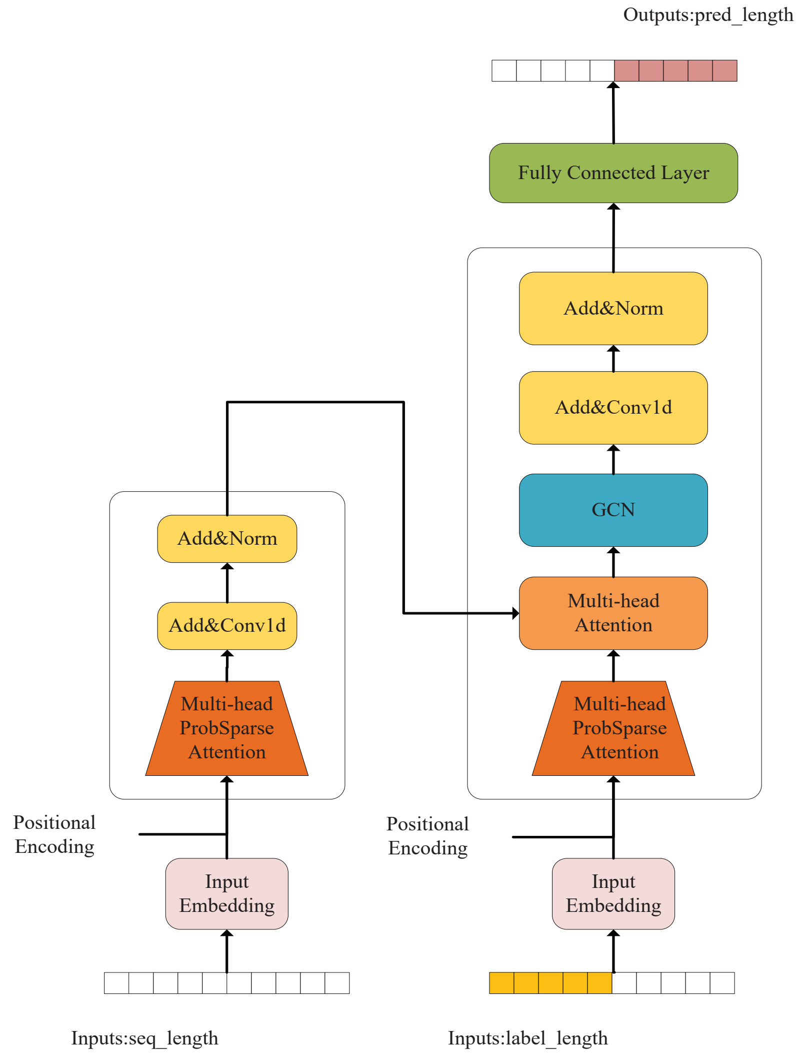


Figure 1. The structure of the GCN–Informer model.

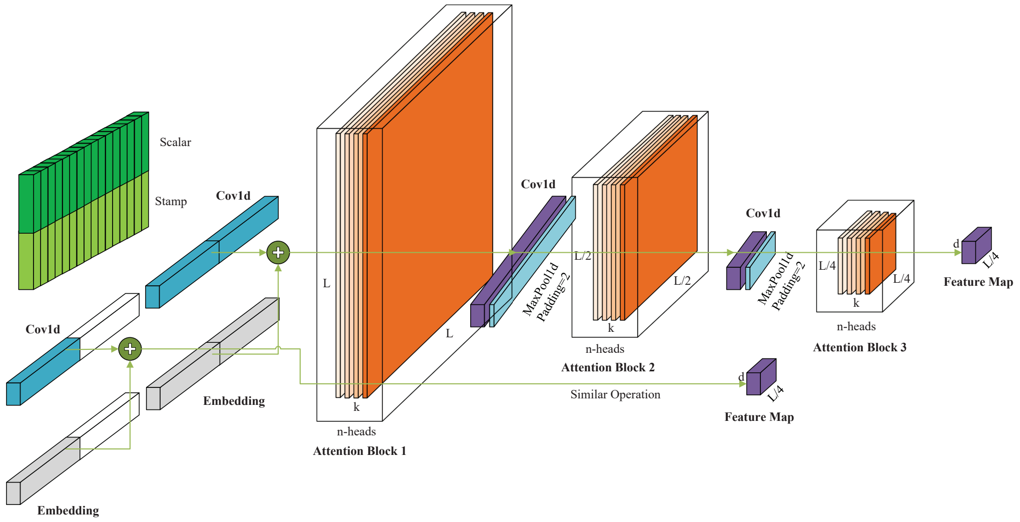


Figure 2. Diagram of EncoderStack in our model.

The difference between the distribution $p(k_j|q_i)$ of a query and the uniformly distributed $p(k_j|q_i)$ can be measured by KL divergence:

$$KL(q||p) = \ln \sum_{l=1}^{L_K} \exp(\frac{q_i k_l^T}{\sqrt{d}}) - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_l^T}{\sqrt{d}} - \ln L_K \tag{2}$$

By removing the constant, the sparsity measure of the i-th query can be defined as:

$$M(q_i, K) = \ln \sum_{j=1}^{L_K} \exp(\frac{q_i k_l^T}{\sqrt{d}}) - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_l^T}{\sqrt{d}} - \ln L_K \tag{3}$$

The process of ProbSparse self-attention is expressed:

$$A(Q, K, V) = soft \max(\frac{\hat{Q}K^T}{\sqrt{d_k}})V \tag{4}$$

This improved self-attention filters out the most important query, reduces the computational complexity, and solves the computational complexity of the square level of self-attention.

2.2.2. Self-Attention Distilling

According to the ProbSparse self-attention process introduced by the above algorithm, the attention result will inevitably produce redundancy. This paper uses self-attention distilling to solve the redundancy problem mentioned above. Specifically, the distillation operation from layer j to layer $j + 1$ is:

$$X_{j+1}^t = MaxPool(ELU(Conv1d([X_j^t]_{AB}))) \tag{5}$$

Therefore, the self-attention distillation operation is used to extract the main attention, which greatly shortens the input time dimension and the amount of network parameters, and solves the bottleneck of memory consumption caused by stacking multi-layer networks.

2.2.3. Multi-Output Output

The original decoder structure, which is stacking two identical multi-headed attention layers, is a dynamic decoding process that is iteratively output one by one. The difference is that this paper uses batch generation prediction to directly output multi-step prediction results in order to improve the speed of reasoning. The decoding process of the whole decoder abandons the dynamic decoding process and uses a forward process to decode the whole output sequence. In addition, the input fed into the decoder is also different. The formula is as follows:

$$X_{feedde}^t = Concat(X_{token}^t, X_0^t) \in R^{(L_{token}+L_y) \times d \bmod el} \tag{6}$$

where X_0^t represents the placeholder and $X_{token}^t \in R^{L_{token} \times d \bmod el}$ represents the start character.

This method is derived from dynamic decoding in NLP technology, which originally consists of a single character as the 'start character', but it is extended to a generative approach, namely, dynamically sampling a partial sequence from the input sequence close to the prediction target as the 'start Token'. Therefore, the problem in long sequence time-series forecasting (LSTF) is solved by calculating the output together with multiple outputs: the efficiency of predicting long-term output. This paper applies this method to solve the problem in Transformer where only one result can be output at a time, thus meeting the real-time requirement of prediction.

2.3. GCN

GCN, at its core, functions as a feature extractor, with the fundamental idea of updating its own nodes by aggregating the features of neighboring nodes. Traditional CNNs struggle to capture its spatial features, which are inherently irregular, and GCN proves to be an effective tool for this purpose. By constructing the adjacency matrix and degree matrix, coupled with the graph's inherent feature matrix, it becomes possible to extract the influence of other nodes on the target node. Taking this graph as an example, GCN can capture the mutual influence between adjacent nodes, a task that CNN networks find challenging to accomplish.

The feature vector at layer l associated with node i in the graph convolution network is represented by x_i^l. We apply a nonlinear transformation to the eigenvector x_j^l of all nodes j in the neighborhood node i, and activate x_i^{l+1} in the next layer. Therefore, the eigenvector x_i^{l+1} at the vertex i of the graph convolution network is represented:

$$x_i^{l+1} = f(x_i^l, \{x_i^l : j \sim i\}) \tag{7}$$

where $\{j \sim i\}$ represents the set of adjacent nodes centered on node i. In this article, node feature x and edge feature y are defined:

$$x_i^{l+1} = x_i^l + \mathrm{Re}\,LU(BN(W_1^l x_i^l + \sum_{j\sim i} \frac{\sigma(e_{ij}^l)}{\sum_{j'\sim i} \sigma(e_{ij'}^l) + \varepsilon} \odot W_2^l x_j^l)) \tag{8}$$

$$e_{ij}^{l+1} = e_{ij}^l + \mathrm{Re}\,LU\Big(BN\Big(W_3^l e_{ij}^l + W_4^l x_i^l + W_5^l x_i^l\Big)\Big) \tag{9}$$

where $W \in R^{h\times h}$, σ is the sigmoid function, ε is a very small value, ReLU is the activation function, and BN represents batch normalization. The final updating rule of the GCN is as follows:

$$H^{(l+1)} = \sigma\Big(\tilde{D}^{-\frac{1}{2}} Attention(Q,K) \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\Big) \tag{10}$$

where $H^{(l+1)}$ represents the feature matrix of the next layer's nodes. σ denotes the activation function. $H^{(l)}$ stands for the feature matrix of the current layer's nodes. $W^{(l)}$ signifies the weight matrix of the current layer.

Up to this point, each node in GCN is associated with a query vector and key vectors related to other nodes. Edges represent the strength of connections between nodes. The strength of these connections is computed by the dot product of query and key, and it can reflect the degree of interaction between nodes. This information can be utilized to construct a graph in which the edge weights are determined by attention scores.

After passing the dynamic adjacency matrix and node features as input to the GCN layer, the GCN layer will use this adjacency matrix to perform graph convolution operations, aggregating information from neighboring nodes. Since the adjacency matrix is constructed based on attention scores, the GCN layer considers the importance of different neighboring nodes when aggregating information. After processing by the GCN layer, each node will obtain an updated feature vector. These feature vectors not only contain the node's own information but also incorporate information from neighboring nodes, taking into account the importance of different neighbors. This information may include direct causal relationships (for example, increased irradiance leads to increased photovoltaic power generation) or more complex indirect relationships (such as a certain combination of temperature and humidity that may cause changes in photovoltaic power generation).

During training, we update the model's parameters through the backpropagation algorithm and gradient descent optimizer. This includes the parameters in the Informer model (such as the linear transformation weights for query and key vectors) and the parameters in the GCN layer (such as the weights and biases of the convolution kernels). Specifically, we first calculate the difference between the model's predicted values and the actual photovoltaic power generation data, using Mean Absolute Error (MAE) and Mean Squared Error (MSE) as loss functions. Then, we utilize the backpropagation algorithm to compute the gradients of the loss function with respect to the model's parameters. Finally, we use the gradient descent optimizer to update the model's parameters based on the calculated gradients. By repeatedly performing these steps (forward propagation, loss calculation, backpropagation, and parameter update), the model gradually learns effective ways to extract hidden features from photovoltaic power generation data and optimizes its performance during the training process.

This module solves the problem that the output of the encoder layer in the Informer model cannot be well extracted to the characteristics between adjacent contacts. After using this GCN module, the information output of the encoder layer can be more accurate, and the output after combining with the decoder layer can better fit the curve of the real data.

3. Experimental Preparation and Framework

3.1. Experimental Preparation

This paper applies the GCN–Informer model to the prediction of solar power generation. The study utilizes solar power data sampled every 5 min over the past decade in Australia, which is a publicly available dataset consisting of 966,771 time-series data. In addition, the dataset encompasses 12 feature values, including temporal characteristics, and one target value of active power generation. The feature values comprise Active Energy Delivered Received, Current Phase Average, Wind Speed, Weather Temperature Celsius, Weather Relative Humidity, Global Horizontal Radiation, Diffuse Horizontal Radiation, Wind Direction, Weather Daily Rainfall, Radiation Global Tilted, and Radiation Diffuse Tilted. Notably, the dataset contains a significant amount of missing values, and the specific handling methods will be detailed in the subsequent paragraphs.

3.2. Experimental Framework

According to Figure 3, the photovoltaic power generation prediction model is based on the following framework: data preprocessing, data splitting, model training, and model scoring.

Data preprocessing: The dataset exhibits some instances of missing values, prompting this study to employ interpolation techniques for their imputation. As illustrated in Figure 4, Pearson correlation analysis played a pivotal role in the feature selection process, enabling the removal of those features with weak correlations to the target prediction values. In the chart, if the absolute value of the Pearson coefficient between two features is less than 0.4, then we consider the correlation between these two features to be weak. Specifically, after observing the Pearson correlation coefficients between all the features and the prediction target, features displaying an absolute Pearson correlation coefficient below 0.4 were excluded from consideration, on the grounds of their limited relevance to the predictive target. Consequently, seven carefully chosen features were designated as inputs for the GCN–Informer model, with the actual power output serving as the target variable as showcased in Table 1. Following the comprehensive data cleaning procedures, an upsampling method was implemented to transform the original dataset, initially collected at 5 min intervals, into a dataset with 1 h intervals. During this consolidation process, feature values were aggregated using their respective mean values, while the predicted target variable was amalgamated through summation.

Data splitting: When employing the hold-out method combined with temporal splitting, we initially divide the dataset into three temporally consecutive and non-overlapping parts: training, validation, and test sets. This ensures that the model does not have access to future information during training and evaluation, preventing data leakage issues.

Specifically, we allocate 98% of the entire time series to the training set, which is used to train the model and learn the underlying patterns in the data. Of the remaining 2%, half (i.e., 1% of the entire dataset) is designated as the validation set. This set is utilized for tuning model parameters and hyperparameters during the training process and for obtaining initial performance estimates. The other half (also 1% of the dataset) serves as the test set, providing a final assessment of the model's performance and its ability to generalize to unseen data.

It is worth noting that while further subdivisions within each set could potentially be created using the hold-out method (e.g., through random sampling) for cross-validation purposes, in the context of time-series data, random sampling can disrupt the temporal dependencies. Therefore, when combining the hold-out method with temporal splitting, we prefer to maintain the temporal order within each subset to ensure that the model learns the correct time-series patterns.

By following this approach, we can effectively leverage the temporal characteristics of the time-series data while ensuring reliable and valid model training and evaluation processes.

Table 1. Input of GCN–Informer.

Input	Variable
x1	Timestamp
x2	Current Phase Average
x3	Weather Temperature Celsius
x4	Weather Relative Humidity
x5	Global Horizontal Radiation
x6	Diffuse Horizontal Radiation
x7	Radiation Global Tilted

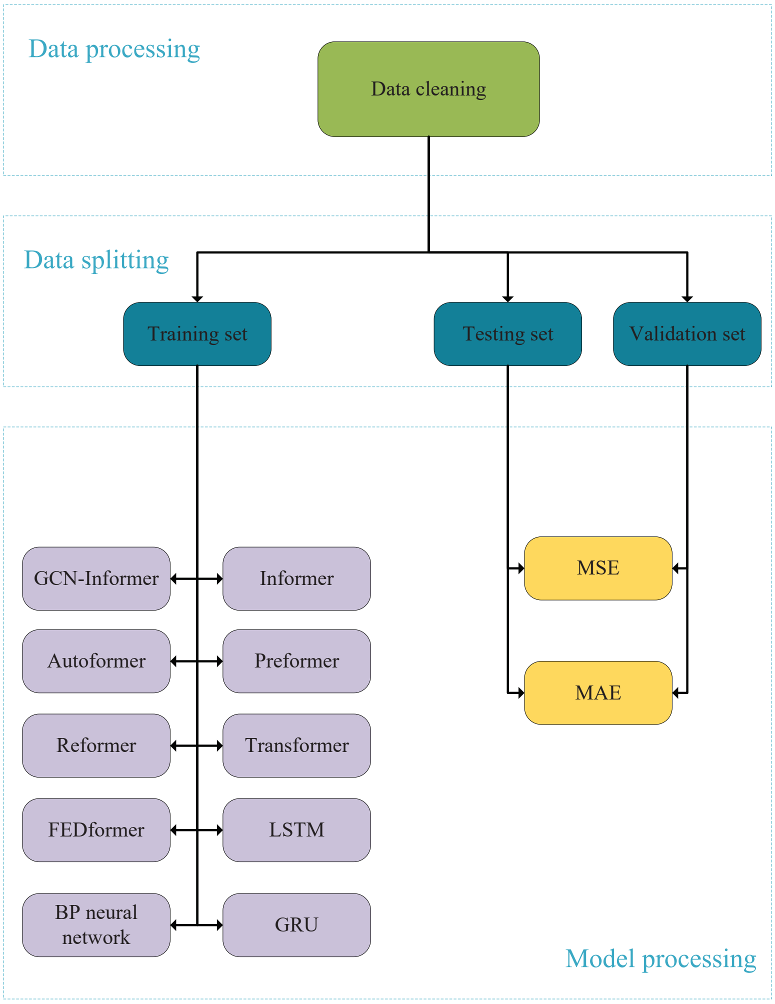
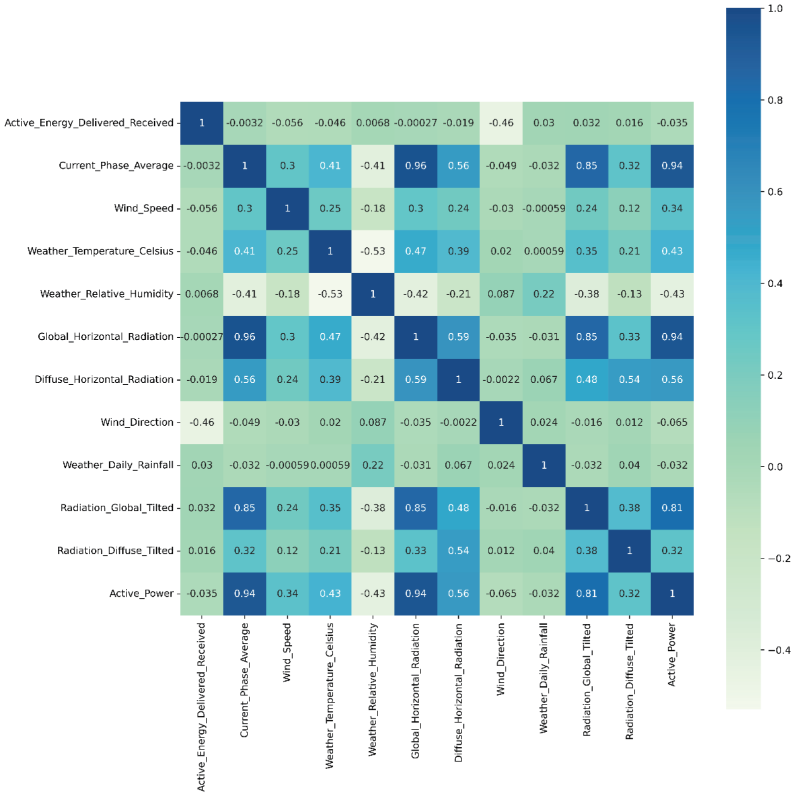


Figure 3. The framework of the model.



	Active_Energy_Delivered_Received	Current_Phase_Average	Wind_Speed	Weather_Temperature_Celsius	Weather_Relative_Humidity	Global_Horizontal_Radiation	Diffuse_Horizontal_Radiation	Wind_Direction	Weather_Daily_Rainfall	Radiation_Global_Tilted	Radiation_Diffuse_Tilted	Active_Power
Active_Energy_Delivered_Received	1	-0.0032	-0.056	-0.046	0.0068	-0.00027	-0.019	-0.46	0.03	0.032	0.016	-0.035
Current_Phase_Average	-0.0032	1	0.3	0.41	-0.41	0.96	0.56	-0.049	-0.032	0.85	0.32	0.94
Wind_Speed	-0.056	0.3	1	0.25	-0.18	0.3	0.24	-0.03	-0.00059	0.24	0.12	0.34
Weather_Temperature_Celsius	-0.046	0.41	0.25	1	-0.53	0.47	0.39	0.02	0.00059	0.35	0.21	0.43
Weather_Relative_Humidity	0.0068	-0.41	-0.18	-0.53	1	-0.42	-0.21	0.087	0.22	-0.38	-0.13	-0.43
Global_Horizontal_Radiation	-0.00027	0.96	0.3	0.47	-0.42	1	0.59	-0.035	-0.031	0.85	0.33	0.94
Diffuse_Horizontal_Radiation	-0.019	0.56	0.24	0.39	-0.21	0.59	1	-0.0022	0.067	0.48	0.54	0.56
Wind_Direction	-0.46	-0.049	-0.03	0.02	0.087	-0.035	-0.0022	1	0.024	-0.016	0.012	-0.065
Weather_Daily_Rainfall	0.03	-0.032	-0.00059	0.00059	0.22	-0.031	0.067	0.024	1	-0.032	0.04	-0.032
Radiation_Global_Tilted	0.032	0.85	0.24	0.35	-0.38	0.85	0.48	-0.016	-0.032	1	0.38	0.81
Radiation_Diffuse_Tilted	0.016	0.32	0.12	0.21	-0.13	0.33	0.54	0.012	0.04	0.38	1	0.32
Active_Power	-0.035	0.94	0.34	0.43	-0.43	0.94	0.56	-0.065	-0.032	0.81	0.32	1

Figure 4. The result of the Pearson's correlation.

Model processing: A comparison of the GCN–Informer performance with other models will be conducted in this experiment. To evaluate the model's performance, MAE and MSE are used in this paper. MAE represents the mean of the absolute error between the predicted value and the observed value. The MSE represents the mean of the sum of squares of the error corresponding to the predicted data and the original data. The results will be visualized, and the model performance will be better compared and evaluated:

$$MAE = \frac{1}{\mathrm{m}} \sum_{i=1}^{m} \|(y_i - \hat{y}_i)\| \tag{11}$$

$$MSE = \frac{1}{\mathrm{m}} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 \tag{12}$$

where y_i is the value of the original power generation, and $\hat{y}_i$ is the value of the predicted power generation.

3.3. Parameter Selection

As shown in Table 2, we conducted experiments using three different sets of data configurations for prediction. These configurations included setting the seq_len to 96 and label_len to 48, setting seq_len to 192 and label_len to 96, and setting seq_len to 144 and label_len to 72. Among these configurations, it was observed that the best predictive performance was achieved when the seq_len was set to 96 and label_len was set to 48.

The input features for the encoder and decoder layers were represented by enc_in and dec_in, respectively. The c_out variable represented the final prediction output. When evaluating different batch sizes, namely 32, 64, and 96, it was found that a batch size of 32 yielded the most optimal results in terms of prediction accuracy.

In order to ensure the reliability of the experiment, we chose to set a relatively large value for the train_epochs parameter. However, since there is an early stopping mechanism in place, not all 1000 epochs will be executed. To strike a balance between experiment reliability and time efficiency, we decided to run only 50 training epochs. This approach allowed us to obtain reliable results without sacrificing too much time.

Table 2. Parameters of GCN–Informer.

Input	Variable
seq_len	96
label_len	48
pred_len	168
enc_in	7
dec_in	7
c_out	1
d_model	512
n_heads	8
e_layers	2
d_layers	1
d_ff	2048
factor	5
dropout	0. 05
itr	50
train_epochs	1000
batch_size	32
patience	3
learning_rate	0.0001

4. Experiment and Analysis

4.1. Implementation Details and Benchmark Model

The task of predicting power generation is divided into two parts: encoder and decoder. The encoder section consists of an attention layer and a distillation layer, which work to extract relevant information from the input data. The attention layer aims to focus on key portions of the input while ignoring the rest while the distillation layer works to reduce the size of the input by eliminating redundant features.

As a complement to the encoder, the decoder consists of a multi-head self-attention layer and a GCN module. In the multi-head self-attention layer, this paper introduces a creative technique that uses the product of attention between query and key as the adjacency matrix for GCN, iteratively updating the values of query and key. The purpose of this step is to extract hide features between neighboring data points more effectively.

The initial parameters we set are as shown in Table 2. All models are implemented using the PyTorch library, which is a well-known machine learning framework for efficiently deploying deep neural networks. Overall, this architecture appears to be a promising approach to addressing the challenging task of predicting electricity generation.

This paper aimed to compare the performance of the GCN–Informer model with nine other deep learning and non-deep learning models in predicting power generation for a week using the same dataset. The objective was to evaluate each model's accuracy in making accurate predictions within the designated time frame. Through Table 3 and Figure 5, an overview of the accuracy levels of all the models is provided.

Table 3. The evaluation indicators for all models.

	Ours	Informer	Autoformer	Preformer	Reformer	Transformer	FEDformer	LSTM	BP Neural Network	GRU
MSE	0.159	0.165	0.177	0.34	0.169	0.164	0.31	0.504	0.703	0.473
MAE	0.199	0.218	0.24	0.347	0.2	0.222	0.374	0.622	0.759	0.695

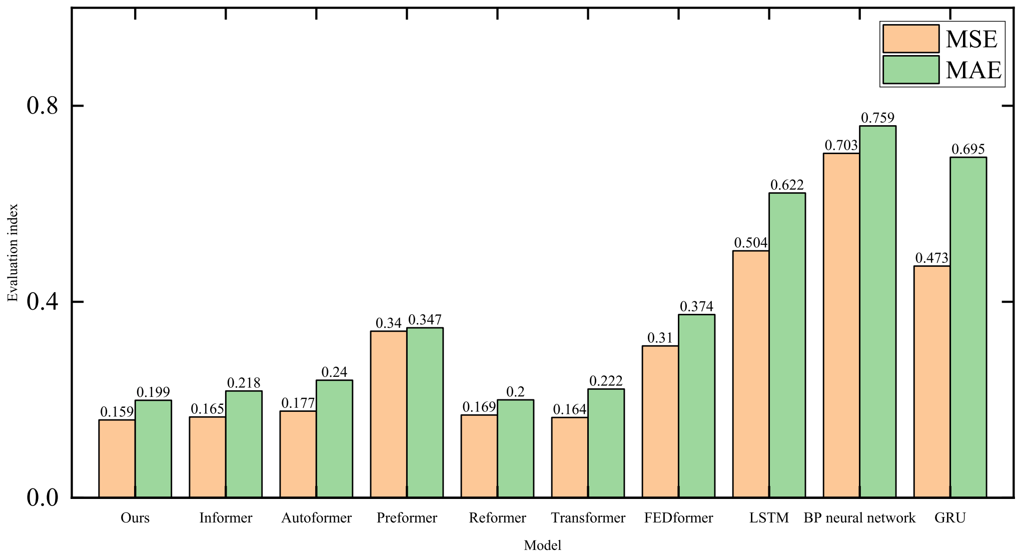


Figure 5. The evaluation indicators for all models.

Upon analyzing the results in Figure 6, it is evident that all models are capable of producing final predictions. However, the GCN–Informer model demonstrated significantly superior performance compared to the other models. In the first 24 h, all models exhibited remarkably similar predictive abilities to the actual values. However, as we approached the final three days, significant disparities emerged among the various models' forecasting capabilities. Not only did these differences evolve over time, but all models displayed fairly accurate predictions on sunny days. On cloudy days, however, the GCN–Informer model showcased a substantial advantage over the other models in terms of its predictive accuracy. Several factors can help explain this notable advantage.

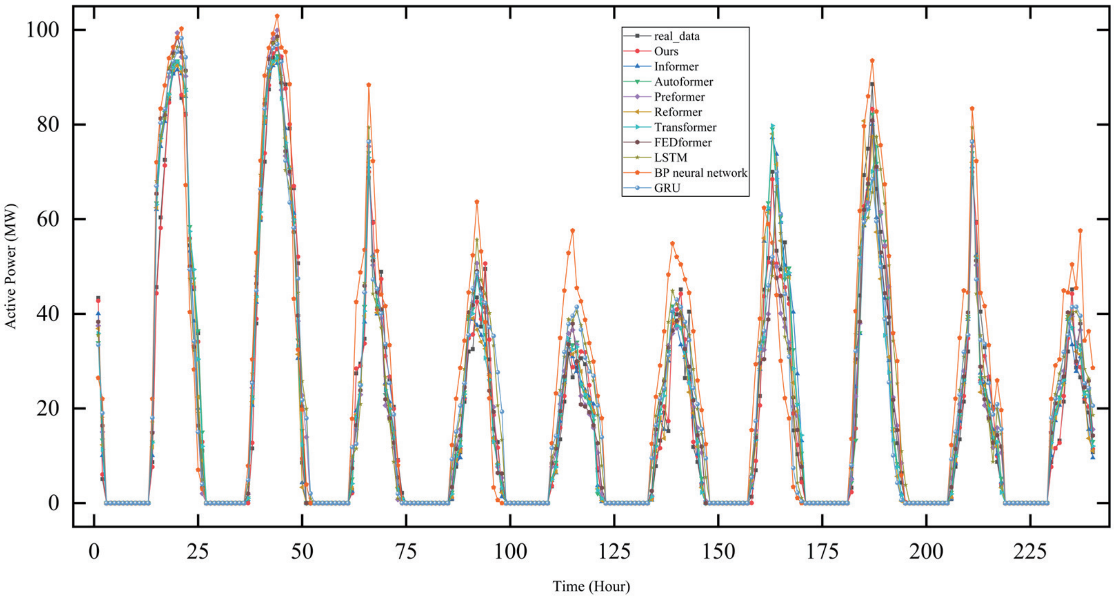


Figure 6. The predicted results for all models.

Firstly, the GCN–Informer model utilizes Graph Convolutional Networks (GCNs), which excel at capturing relationships between data points in a graph-like structure. Since power generation involves intricate interdependencies and interactions between various factors such as weather conditions, energy demand, and network infrastructure, the GCN–Informer model's ability to incorporate these complex relationships enhances its predictive capabilities. In contrast, traditional machine learning models may struggle to capture these intricate dependencies effectively.

Furthermore, the GCN–Informer model incorporates Informer, a popular architecture in deep learning, known for its exceptional performance in sequence modeling tasks. With the ability to capture long-term dependencies and learn from extensive historical data, Informer empowers the GCN–Informer model to understand complex temporal patterns inherent in power generation datasets. The utilization of Informer allows the GCN–Informer model to outperform other models that lack this advanced architecture.

The reasons why the GCN–Informer model achieves better performance than standard models like Transformer or Autoformer can be summarized as follows:

(1) Dynamic relationship modeling: Standard GCNs use static adjacency matrices to represent spatial relationships between nodes. However, in many applications, the relationships between nodes may be dynamically changing, especially when dealing with time-series data. By using the product of query and key from the Informer attention mechanism to construct the adjacency matrix, GCN–Informer is able to capture the dynamic relationships between nodes that change over time, thereby better adapting to the characteristics of the data.

(2) Combination of spatiotemporal features: The Transformer model captures temporal dependencies in sequences through its self-attention mechanism, but it does not directly consider spatial structural information. By combining the self-attention mechanism with GCN, GCN–Informer is able to consider both temporal and spatial features simultaneously. Specifically, the product of query and key reflects the importance of different time steps, and incorporating this information into the adjacency matrix of GCN enables the model to perform targeted feature aggregation in the spatial dimension as well.

Certainly, after demonstrating the excellent predictive capabilities of GCN–Informer, it is crucial to pay attention to the efficiency of our model. As the GCN–Informer model and Preformer proposed in this article both incorporate the GCN structure, their specific time complexity cannot be represented by O notation. Similarly, the time complexity of LSTM and GRU also depends on the feature dimensions and hidden state dimensions. Therefore, we provide the running time for all models to predict one week of power generation using the dataset mentioned in the article (time refers to the required time for one epoch, with other parameters consistent with the article). Our graphics card is an NVIDIA GeForce RTX 4070Ti. As shown in Table 4, although our proposed model is slower than other deep learning models, it is still within the same order of magnitude. If a NVIDIA GeForce RTX 4090 is used for calculation, the computation time should be faster.

Table 4. The computational complexity and running time for all models.

	Ours	**Informer**	**Autoformer**	**Preformer**	**Reformer**	**Transformer**	**FEDformer**	**LSTM**	**BP Neural Network**	**GRU**
Complexity	-	$O(nlogn)$	$O(nlogn)$	-	$O(nlogn)$	$O(n^2)$	$O(n)$	-	-	-
time (s)	228.86	86.49	91.3	223.78	181.86	203.5	90.45	302.75	417.33	278.98

In summary, the GCN–Informer model's superior performance can be attributed to its ability to capture complex relationships, exploit Informer architectures for sequence modeling, and effectively leverage both spatial and temporal information. These advantages enable it to surpass traditional machine learning models in accurately predicting power generation. The findings of this study have significant implications for various industries, particularly those involved in energy production and management. By providing more accurate and reliable power generation predictions, advanced deep learning models like the GCN–Informer model can assist these industries in optimizing their operations and making informed decisions.

4.2. The Practical Application in Photovoltaic Power Generation Forecasting

This section provides an in-depth exploration of the application of the GCN–Informer model for forecasting photovoltaic power generation. The conventional method for predicting photovoltaic power generation relies on the backpropagation (BP) neural network, which is plagued by challenges like local minima and sluggish convergence rates. Consequently, these issues lead to suboptimal identification accuracy and diminished reliability in existing prediction methodologies.

To address these limitations, we introduce the GCN–Informer model, which harnesses extensive power generation data and incorporates information from adjacent time points within the power grid. By capitalizing on the historical power grid data for photovoltaic power generation prediction, the GCN–Informer model brings about a substantial improvement in the dependability and precision of power generation forecasting. This innovation bolsters our ability to deliver more reliable and accurate predictions in the realm of power generation.

Furthermore, the GCN–Informer model exhibits strong versatility, as it can be interchangeably used with other deep learning models in the photovoltaic industry, highlighting its significant practicality. However, potential technical limitations may arise during industrial implementation, such as compatibility issues with legacy systems.

5. Conclusions

The growing demand for precise medium and long-term forecasts of photovoltaic power generation has surged in tandem with the swift advancement of new energy generation technologies. In light of this pressing demand, this study introduces an innovative approach that integrates both GCN and Informer methodologies into the realm of photovoltaic power generation prediction. This pioneering endeavor is poised to meet the increasing need for dependable energy forecasting in the evolving landscape of new energy power generation.

More precisely, the introduced approach is denoted as the GCN–Informer model. This model has been meticulously crafted to extract invaluable insights between successive time intervals, featuring both an encoder layer and a decoder layer. In the encoder layer, the incorporation of an attention distillation mechanism facilitates the coarse-grained capture of long-range dependencies among inputs, even in scenarios where memory resources are limited. On the other hand, the inclusion of a GCN module within the decoder layer bolsters the precision of feature extraction between adjacent time points. This strategic enhancement effectively addresses the issue of missing features in the original model, further elevating the overall performance and reliability of the GCN–Informer model.

The experimental results suggest that the GCN–Informer model may offer advantages in terms of accuracy compared to traditional machine learning and deep learning approaches. However, further investigation and validation may be warranted. Through the strategic integration of GCN and Informer methodologies, our proposed model excels in precisely forecasting photovoltaic power generation for medium-term horizons. At the same time, while meeting the real-time requirements of photovoltaic power generation prediction, the extraction of hidden information of different features also ensures the prediction accuracy of the model. This accomplishment not only meets but also exceeds the elevated demands set forth by the burgeoning landscape of new energy power generation. The GCN–Informer model stands as a testament to the power of advanced techniques in shaping the future of energy forecasting, ensuring the reliable and efficient utilization of photovoltaic power in the ever-evolving energy sector.

In future studies, we aim to develop an enhanced model for interval forecasting, leveraging the strengths of the GCN–Informer architecture. To better align with the realities of photovoltaic power generation in industrial and electrical sectors, we will introduce a tailored loss function. This custom metric will quantify the economic ramifications arising from discrepancies between predicted and actual power output, ultimately leading to more meaningful contributions in the field of photovoltaic forecasting.

Author Contributions: Conceptualization, W.Z. and Z.L.; methodology, Z.L.; validation, Z.L.; formal analysis, Y.W.; investigation, Q.X.; resources, W.Z.; data curation, Y.W.; writing—review and editing, W.Z., Z.L., Y.W. and Q.X.; visualization, Z.L.; supervision, M.X.; project administration, W.Z. and M.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Project of SGCC, the Research and application of data-driven intraday forward-looking scheduling technology for key transmission channels, grant number 5108-202318054A-1-1-ZN.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://github.com/Seb5Vet/MDPI (accessed on 12 January 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mai, T.; Sandor, D.; Wiser, R.; Schneider, T. *Renewable Electricity Futures Study. Executive Summary*; Technical Report; National Renewable Energy Lab. (NREL): Golden, CO, USA, 2012.
2. Board, C.N.E. *Canada's Energy Future 2016: Energy Supply and Demand Projections to 2040: Appendices*; National Energy Board: Calgary, AB, Canada, 2016.
3. Agency, I.E.; Birol, F. *World Energy Outlook 2013*; International Energy Agency: Paris, France, 2013.
4. Wang, J.; Zhang, S.; Zhang, Q. The relationship of renewable energy consumption to financial development and economic growth in China. *Renew. Energy* **2021**, *170*, 897–904. [CrossRef]
5. Yu, S.; Lu, T.; Hu, X.; Liu, L.; Wei, Y.M. Determinants of overcapacity in China's renewable energy industry: Evidence from wind, photovoltaic, and biomass energy enterprises. *Energy Econ.* **2021**, *97*, 105056. [CrossRef]
6. Aboagye, B.; Gyamfi, S.; Ofosu, E.A.; Djordjevic, S. Status of renewable energy resources for electricity supply in Ghana. *Sci. Afr.* **2021**, *11*, e00660. [CrossRef]
7. Liu, D.; Sun, K. Random forest solar power forecast based on classification optimization. *Energy* **2019**, *187*, 115940. [CrossRef]
8. Gao, X.M.; Yang, S.F.; Pan, S.B. Optimal parameter selection for support vector machine based on artificial bee colony algorithm: A case study of grid-connected pv system power prediction. *Comput. Intell. Neurosci.* **2017**, *2017*, 7273017. [CrossRef]
9. Lee, D.; Kim, K. Recurrent neural network-based hourly prediction of photovoltaic power output using meteorological information. *Energies* **2019**, *12*, 215. [CrossRef]
10. Li, G.; Wang, H.; Zhang, S.; Xin, J.; Liu, H. Recurrent neural networks based photovoltaic power forecasting approach. *Energies* **2019**, *12*, 2538. [CrossRef]
11. Abdel-Nasser, M.; Mahmoud, K. Accurate photovoltaic power forecasting models using deep LSTM-RNN. *Neural Comput. Appl.* **2019**, *31*, 2727–2740. [CrossRef]
12. Ahn, H.K.; Park, N. Deep RNN-based photovoltaic power short-term forecast using power IoT sensors. *Energies* **2021**, *14*, 436. [CrossRef]
13. Zhou, H.; Zhang, Y.; Yang, L.; Liu, Q.; Yan, K.; Du, Y. Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *IEEE Access* **2019**, *7*, 78063–78074. [CrossRef]
14. He, H.; Hu, R.; Zhang, Y.; Zhang, Y.; Jiao, R. A power forecasting approach for PV plant based on irradiance index and LSTM. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 9404–9409.
15. Hossain, M.S.; Mahmood, H. Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast. *IEEE Access* **2020**, *8*, 172524–172533. [CrossRef]
16. De, V.; Teo, T.T.; Woo, W.L.; Logenthiran, T. Photovoltaic power forecasting using LSTM on limited dataset. In Proceedings of the 2018 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia), Singapore, 22–25 May 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 710–715.
17. Gensler, A.; Henze, J.; Sick, B.; Raabe, N. Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 002858–002865.
18. Qing, X.; Niu, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **2018**, *148*, 461–468. [CrossRef]
19. Zhuang, W.; Fan, J.; Xia, M.; Zhu, K. A Multi-Scale Spatial-Temporal Graph Neural Network-Based Method of Multienergy Load Forecasting in Integrated Energy System. *IEEE Trans. Smart Grid* **2023**. . [CrossRef]
20. Ren, H.; Xia, M.; Weng, L.; Hu, K.; Lin, H. Dual Attention-Guided Multiscale Feature Aggregation Network for Remote Sensing Image Change Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 4899–4916. [CrossRef]

21. Wang, Z.; Xia, M.; Weng, L.; Hu, K.; Lin, H. Dual encoder-decoder network for land cover segmentation of remote sensing image. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *17*, 2372–2385. [CrossRef]
22. Ding, L.; Xia, M.; Lin, H.; Hu, K. Multi-level attention interactive network for cloud and snow detection segmentation. *Remote Sens.* **2023**, *16*, 112. [CrossRef]
23. Ji, H.; Xia, M.; Zhang, D.; Lin, H. Multi-Supervised Feature Fusion Attention Network for Clouds and Shadows Detection. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 247. [CrossRef]
24. Zhang, M.; Tao, P.; Ren, P.; Zhen, Z.; Wang, F.; Wang, G. Spatial-Temporal Graph Neural Network for Regional Photovoltaic Power Forecasting Based on Weather Condition Recognition. In Proceedings of the 10th Renewable Power Generation Conference (RPG 2021), Online, 14–15 October 2021; IET: London, UK, 2021; Volume 2021, pp. 361–368.
25. Zhang, M.; Zhen, Z.; Liu, N.; Zhao, H.; Sun, Y.; Feng, C.; Wang, F. Optimal graph structure based short-term solar PV power forecasting method considering surrounding Spatio-temporal correlations. *IEEE Trans. Ind. Appl.* **2022**, *59*, 345–357. [CrossRef]
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
27. Kitaev, N.; Kaiser, Ł.; Levskaya, A. Reformer: The efficient transformer. *arXiv* **2020**, arXiv:2001.04451.
28. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The long-document transformer. *arXiv* **2020**, arXiv:2004.05150.
29. Wang, S.; Li, B.; Khabsa, M.; Fang, H.; Ma, H.L. Self-attention with linear complexity. *arXiv* **2020**, arXiv:2006.04768.
30. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 11106–11115.

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.