energies

MDPI

Article

Vision Transformer-Based Photovoltaic Prediction Model

Zaohui Kang [1], Jizhong Xue [1], Chun Sing Lai [1,2,*], Yu Wang [1], Haoliang Yuan [1,*] and Fangyuan Xu [1,*]

[1] Department of Electrical Engineering, Guangdong University of Technology, Guangzhou 510006, China; 2112104017@mail2.gdut.edu.cn (Z.K.); 2112104485@mail2.gdut.edu.cn (J.X.); 2112204504@mail2.gdut.edu.cn (Y.W.)

[2] Brunel Interdisciplinary Power Systems Research Centre, Department of Electronic and Electrical Engineering, Brunel University London, London UB8 3PH, UK

* Correspondence: chunsing.lai@brunel.ac.uk (C.S.L.); haoliangyuan@gdut.edu.cn (H.Y.); xufangyuan@gdut.edu.cn (F.X.)

Abstract: Sensing the cloud movement information has always been a difficult problem in photovoltaic (PV) prediction. The information used by current PV prediction methods makes it challenging to accurately perceive cloud movements. The obstruction of the sun by clouds will lead to a significant decrease in actual PV power generation. The PV prediction network model cannot respond in time, resulting in a significant decrease in prediction accuracy. In order to overcome this problem, this paper develops a visual transformer model for PV prediction, in which the target PV sensor information and the surrounding PV sensor auxiliary information are used as input data. By using the auxiliary information of the surrounding PV sensors and the spatial location information, our model can sense the movement of the cloud in advance. The experimental results confirm the effectiveness and superiority of our model.

Keywords: photovoltaic prediction; visual transformer; auxiliary information

check for updates

Citation: Kang, Z.; Xue, J.; Lai, C.S.; Wang, Y.; Yuan, H.; Xu, F. Vision Transformer-Based Photovoltaic Prediction Model. *Energies* **2023**, *16*, 4737. https://doi.org/10.3390/en16124737

Academic Editor: Helena M. Ramos

Received: 8 May 2023
Revised: 9 June 2023
Accepted: 12 June 2023
Published: 15 June 2023

1. Introduction

When solar energy is used in the grid, the output of PV power generation is intermittent due to some meteorological factors, such as changes in solar radiation. Solar energy depends on local climatic conditions and cloud dynamics. This uncertainty affects the accuracy of PV predictions, as clouds blocking sunlight will lead to a sharp drop in light radiation intensity. The model cannot predict the situation at this time, resulting in a large difference between the model's prediction and the actual results, thereby reducing accuracy.

There are two primary PV prediction methods in current research: traditional machine learning methods and deep learning algorithms. Traditional machine learning methods generally include, but are not limited to, support vector machines, decision trees, random forests, hidden Markov model methods, etc. The feature extraction of most machine learning methods is independent of the network model. Known features or features [1] that experts in specific fields believe are important for completing specific tasks need to be manually extracted from the original data. In contrast, deep learning algorithms use an end-to-end approach, using the neural network layer to extract deep or abstract features from large complex datasets. Compared to traditional machine learning methods, most deep learning algorithms do not rely on hand-selected features. Deep learning algorithms extract relevant features from datasets in an unassuming manner, without requiring expert domain-specific knowledge [2].

Although many models have achieved good results in PV prediction, their performance in PV prediction is still insufficient. For example, previous PV prediction models could not perceive that cloud movement blocking the sun causes a rapid decline in power generation, which reduces the overall accuracy of the prediction. To overcome this problem, in this paper, we propose a PV prediction model based on the vision transformer (VIT) [3] model, which has been successfully applied in many computer vision tasks. In our model, we use the target PV sensor and surrounding PV sensors as input data. The PV sensor is located in Panyu, Guangdong Province, China. Since our target sensor is located inside these surrounding sensors, as shown in Figure 1, surrounding sensors can perceive the cloud movement in advance. We used the number 1 sensor as the target sensor and the remaining 8 as auxiliary sensors. The information collected by all sensors was utilized for training and testing. To effectively capture this advanced information, we adopted the multi-head self-attention (MSA) mechanism to exploit the auxiliary information from the surrounding PV sensors. Moreover, we considered the impact of the positional information of the surrounding PV sensors and the target PV sensor. In order to verify the validity of the model, we conducted a large number of comparative experiments and ablation experiments. Hence, the contributions of this paper are summarized as follows:

(1) We developed the VIT model for PV prediction, which utilizes the auxiliary information from the surrounding PV sensors to help the target sensor in anticipating the cloud movement in advance.

(2) Incorporating the geographic information of PV sensors into our model further enhances the prior knowledge needed to improve the PV prediction performance.

(3) Many comparative and ablation experiments confirm the effectiveness and superiority of our model.

The rest of the paper is organized as follows. Section 1 presents the literature review. Section 2 presents the methodology. Section 3 presents the results. Section 4 presents the conclusions and future prospects.

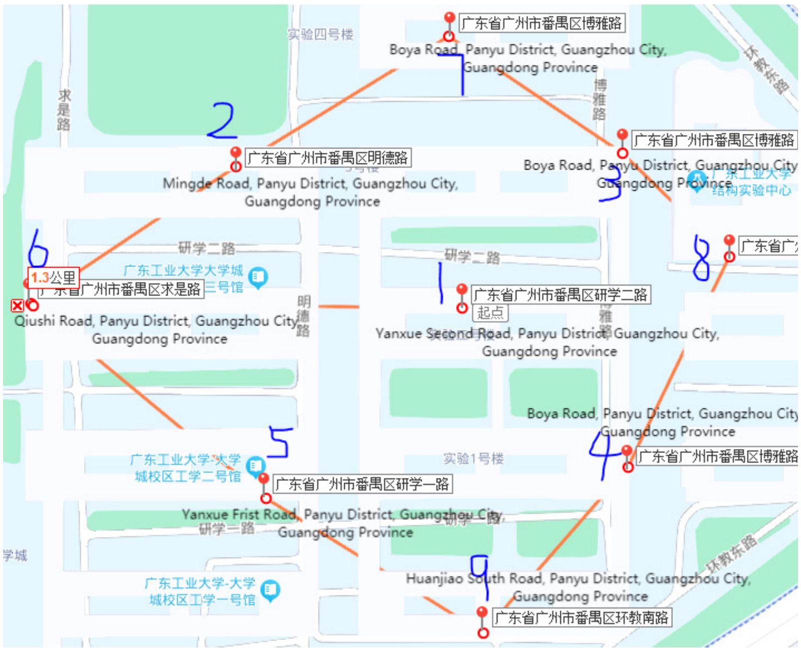

广东省广州市番禺区博雅路
Boya Road, Panyu District, Guangzhou City, Guangdong Province
实验四号楼
求是路
2
广东省广州市番禺区明德路
Mingde Road, Panyu District, Guangzhou City, Guangdong Province
博雅路
环教东路
广东省广州市番禺区博雅路
Boya Road, Panyu District, Guangzhou City, Guangdong Province
3
广东工业大学结构实验中心
6
1.3公里
广东省广州市番禺区求是路
Qiushi Road, Panyu District, Guangzhou City, Guangdong Province
研学二路
广东工业大学大学城三号馆
明德路
8
广东省广
1
广东省广州市番禺区研学二路
起点
Yanxue Second Road, Panyu District, Guangzhou City, Guangdong Province
Boya Road, Panyu District, Guangzhou City, Guangdong Province
5
4
广东省广州市番禺区博雅路
广东工业大学-大学城校区工学二号馆
实验1号楼
广东省广州市番禺区研学一路
Yanxue Frist Road, Panyu District, Guangzhou City, Guangdong Province
研学一路
学城
9
Huanjiao South Road, Panyu District, Guangzhou City, Guangdong Province
环教东路
广东工业大学-大学城校区工学一号馆
广东省广州市番禺区环教南路

Figure 1. The positional distribution map of the PV sensor, where 1 is the target sensor and the rest are the auxiliary sensors.

2. Related Works

2.1. Traditional Machine Learning for PV Prediction

Yang et al. proposed an autoregressive linear model, which was further extended to include vector autoregressive and vector autoregression models, based on the traditional autoregressive model [4]. Cavalcante et al. performed PV prediction by combining the vector autoregressive model with the minimum absolute shrinkage and selection operator framework [5]. Peder et al. used the autoregressive model with exogenous inputs to predict hourly values for solar PV power generation [6]. Zeng et al. proposed a radial basis function neural network-based model for short-term solar power prediction [7]. Hugo et al. compared the k-nearest neighbor, artificial neural network, and other solar PV power prediction models [8]. Bouzerdoum et al. proposed a SARIMA-SVM hybrid model for the time series forecasting of solar PV power generation [9]. Wu et al. combined the autoregressive integrated moving average model, SVM, artificial neural network, and fuzzy inference system to predict solar PV power generation [10]. The integration method is also popular in PV forecasting. Rana et al. integrated neural networks and support vector regression to make short-term predictions of PV power generation [11]. Asrari et al. proposed an artificial neural network to forecast solar PV power generation one hour in advance [12]. Shang et al. improved SVR and enhanced empirical mode decomposition for predicting solar PV power generation [13]. Behera et al. used the extreme learning machine to predict the PV power at intervals of 15 min, 30 min, and 60 min, respectively [14]. Eseye et al. applied a hybrid prediction model combining particle swarm optimization and SVM for the short-term power prediction of actual microgrid PV systems [15]. Although the above models have achieved good results in PV prediction, there are still deficiencies. The data used in machine learning methods need to be manually screened or supported by prior knowledge, which is very troublesome. Compared to machine learning methods, deep learning networks have good results in most cases.

2.2. Deep Learning for PV Prediction

Jeong et al. used a convolutional neural network (CNN) to extract spatiotemporal correlations by superimposing PV signals into images and reordering them based on their real-world locations [16]. In addition, Shih et al. introduced an attention mechanism to capture the spatial correlation among PV nodes [17]. Simeunovi et al. used the graph-convolutional transformer for PV prediction, which uses an attention mechanism [18]. Li et al. proposed a hybrid short-term PV power plant model, which combines a time-series generative adversarial network, K-medoids, and CNN-GRU [19]. To address the volatility and instability of PV power generation, Zhu et al. used a PV prediction model that combines the k-means technique with a long short-term memory (LSTM) network [20]. By using the attention layers of two LSTM neural networks, Zhou et al. found more important input features in PV prediction [21]. Qu et al. proposed a new hybrid model based on the gated recurrent unit to predict distributed PV power generation [22]. Basset et al. introduced a new deep learning architecture called PV-Net for short-term forecasting of day-ahead PV energy [23]. Perez et al. proposed an intra-day prediction model that does not require training or real-time data measurements [24]. Guermoui et al. used a novel decomposition method to decompose PV power into intrinsic functions and used extreme learning machines for prediction [25]. Korkmaz proposed a new CNN structure model called Solar-Net for the short-term prediction of PV output power under dissimilar weather seasons and conditions [26]. Sharma et al. applied a hybrid deep learning framework for PV prediction, which consists of a long short-term memory layer and maximum overlap discrete wavelet transform model [27]. Cannizzaro et al. proposed a new method for predicting solar radiation by combining variational mode decomposition, two CNNs, random forests, and LSTM networks [28]. The above deep network models have greatly improved the accuracy of PV prediction, but most of the network's input only uses historical PV information and local climate information. This makes it difficult for the network to perceive cloud movement information. Perceiving cloud movement information is very important for the PV prediction network.

2.3. Motivation

In order to solve the above problems, this paper develops a PV prediction model that is based on the vision transformer framework. The model uses the PV power information from both the target sensor and the auxiliary sensor as input, and integrates the geographic information matrix into the self-attention layer to allocate the information weight of information from the auxiliary sensor, so that the PV prediction network can perceive cloud movement information and improve prediction accuracy.

3. Materials and Methods

3.1. Model

The proposed model is shown in Figure 2. The input $\mathbf{X} \in R^{l\times w}$ of the model is the information of the PV sensor, where l is the length of time representing the sequence, and w is the number of PV sensors. The output $\mathbf{Y} \in R^{l\times 1}$ of the model is the PV prediction sequence of the target PV sensor. We add a learnable sequence $z_0^0 \in R^{l\times 1}$ in front of the input $\mathbf{X}$ for prediction. To fuse the sequence with the position embedding A_{pos}, we add a trainable linear projection A to map the input $\mathbf{X}$. The procedure of adding position embedding can be represented as follows:

$$Z_0 = [z_0^0, \mathbf{X}]^\top A + A_{pos}; A \in R^{l\times D}, A_{pos} \in R^{(w+1)\times D} \tag{1}$$

where $Z_0 \in R^{(w+1)\times D}$ represents the complete sequence after adding a learnable predictive token and position embedding. In the following, MSA is used to exploit the auxiliary information of the surrounding PV sensors, which is represented as follows:

$$Z'_\iota = MSA(LN(Z_{\iota-1})) + Z_{\iota-1}; \iota = 1\ldots N \tag{2}$$

$$Z_\iota = MLP(LN(Z'_\iota)) + Z'_\iota; \iota = 1\ldots N \tag{3}$$

where $LN(\cdot)$ is the layer normalization module, $MSA(\cdot)$ is the MSA module, and $MLP(\cdot)$ is the multi-layer perception (MLP) module. $Z'_\iota \in R^{(w+1)\times D}$ and $Z_\iota \in R^{(w+1)\times D}$ denote the ι-th middle variable and output variable. Through the N transformer encoder, the PV prediction of the target sensor is represented as Z_ι.

$$\mathbf{Y} = LN(z_N^0) \tag{4}$$

where z_N^0 represents the output state of our learnable prediction token z_0^0 in Z_0 after passing through the transformer encoder layer. $\mathbf{Y}$ is the output of the model, which is the PV prediction sequence of the target PV sensor.

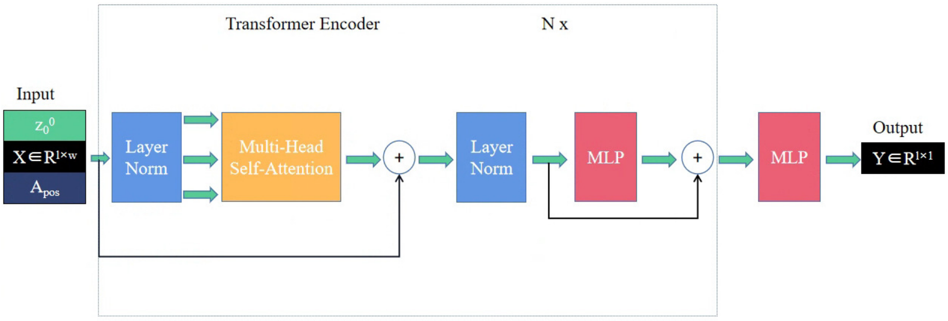


Figure 2. The structure of the network we propose. As shown in the figure, the input of the network consists of $\mathbf{X}$, A_{pos}, and P, where $\mathbf{X}$ represents the information sequence of the photovoltaic sensor, A_{pos} is the position encoding of $\mathbf{X}$, and z_0^0 is the learnable predictive token that we added. The input of the model is normalized by the layer norm in the transformer encoder.

3.2. Multi-Head Self-Attention

Self-attention [29] is a popular neural network module. For each sequence in the input $Z_0 \in R^{(w+1)\times D}$ of the transformer encoder, we generate three learnable weight matrices, denoted as $U_{QKV} = [W^Q, W^K, W^V]$, where $W^Q \in R^{D\times D_h}, W^K \in R^{D\times D_h}, W^V \in R^{D\times D_h}$. We calculate the weighted sum of all values V within the sequence.

$$[Q, K, V] = Z_0 U_{QKV}; U_{QKV} \in R^{D\times 3D_h} \tag{5}$$

where $Q \in R^{w\times D_h}, K \in R^{w\times D_h}, V \in R^{w\times D_h}$ are the three vectors obtained by multiplying the input sequence Z_0 and the corresponding three matrices in U_{QKV}.

$$E_1 = softmax(\frac{Q_1K_1^\top}{\sqrt{D_h}}); E_1 \in R^{(w+1)\times(w+1)} \tag{6}$$

$$SA_1(x) = E_1V_1 \tag{7}$$

where $SA(\cdot)$ is a self-attention computation. MSA denotes the splicing of multiple SA operations. We perform k self-attention operations in parallel to form k heads and map the splicing output of the k heads. In order to keep the number of calculations and parameters unchanged when changing k, D_h in (5) is usually set to D/k.

$$MSA(x) = [SA_1(x), SA_2(x), \ldots, SA_k(x)]\mathbf{U}_{msa} \tag{8}$$

where $\mathbf{U}_{msa} \in R^{D_h\times D}$ is a weight matrix.

3.3. Input Embedding and Position Embedding

We incorporate predictive input embedding and position embedding into the predictive input sequence of our model. As mentioned above, we add predictive input embedding and position embedding into the input sequence. When embedding, we not only add a learnable blank label, such as the VIT model, we add a geographic location information matrix for PV sensors. The geographic information matrix helps the model to predict at the beginning, allowing the model to adjust the parameters according to the geographical location of the PV sensor, to make the prediction more accurate. The definition of the geographic location information matrix is as follows (9) and (10).

$$S_{ij} = \begin{Bmatrix} d_i^j > d_k; 0 \\ d_i^j < d_k; 1 \end{Bmatrix}; i,j = 1\ldots w \tag{9}$$

$$S'_{ij} = \begin{Bmatrix} d_i^j > d_k; 1 \\ d_i^j < d_k; 0 \end{Bmatrix}; i,j = 1\ldots w \tag{10}$$

where S_{ij} denotes the value of row i and column j in the close-range geographic information matrix, S'_{ij} denotes the value of row i and column j in the long-range geographic information matrix. S_{ij} denotes the value of row i and column j in the short-range geographic information matrix. d_i^j denotes the distance from the i sensor to the j sensor, and d_k denotes half of the distance between the target PV sensor and the furthest PV sensor. In the experiment, we set four advanced times. When the prediction time was short, such as 60 s or 180 s, the short-range PV sensor information was relatively useful; hence, we increased the weight of the close-range PV sensor information. When the prediction time was long, such as 300 s or 600 s, the long-range PV sensor information was relatively useful, so the weight of the remote PV sensor information was increased.

$$A_{geo} = RELU(SO_s); \tag{11}$$

where $S \in R^{w\times w}$ is a geographic information matrix, which is mapped to the D dimension by $O_s \in R^{D\times D}$.

$$A_{pos} = A_{learn} + A_{geo} \tag{12}$$

where $A_{learn} \in R^{(w+1)\times D}$ is a learnable position embedding, and $A_{geo} \in R^{(w+1)\times D}$ is a geographic information matrix calculated by using the distance between each PV sensor. The initial dimension is $w \times w$, which has been mapped to the D dimension. In input X, we add a learnable predictive token. In order to keep the dimension consistent, we need to add its position coding and geographic information on A_{learn} and A_{geo}. Here, we add the initialized random value.

3.4. Loss Function

Our loss function uses the *MSE* mean squared error loss function, and the loss is propagated back to the model from the output of the model, adjusting the parameters of the model. *MSE* is a good measure of the mean error and the degree of variation in the evaluation data. The specific formula is as follows (13).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{13}$$

where Y_i is the truth value collected by the PV target sensor, and $\hat{Y}_i$ is the PV prediction value output of the model.

4. Results and Discussion

4.1. Experimental Detail

In order to prevent the results from being accidental due to the different initial values of each model, we repeated the following experiments to ensure the stability of the results and reduce uncertainty.

The flow chart of the prediction method is shown in Figure 3. Firstly, we used PV sensors for data acquisition, and then cleaned and normalized the data. Data cleaning includes deleting abnormal data and using the interpolation method to fill the values. We used the max–min normalization method. During both the training and testing processes, our model took as input the information from the target sensor, the auxiliary sensor, and the geographic information matrix. For the geographic information matrix description, please see Section 2.3.

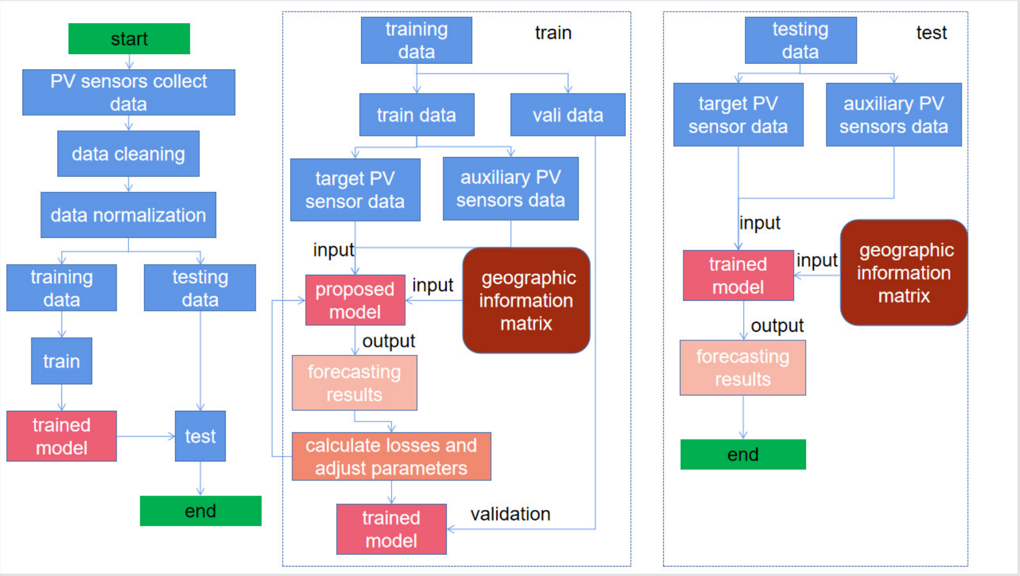


Figure 3. Methodology proposed for forecasting.

4.2. Datasets

We used our own PV sensor to collect data to make a PV dataset. The dataset contained data collected from seven PV sensors, every day, from 2019 to 2021, with a time resolution of 1 s. Seven PV sensors were placed in Panyu, Guangzhou, Guangdong Province, China. We used the PV sensor to collect data from Hebei Pingao Electronic Technology Co., Ltd. (Handan, China); this PV sensor can collect real-time optical radiance data with a sampling frequency of 1 s. Since the solar irradiance before sunrise and after sunset is negligible, we used data from between 9:00 a.m. and 3:00 p.m. Out of the missing data, there were about 2 million data points in our PV data that could be predicted. We disrupted the number of days on the total dataset and divided it into two datasets, on average (dataset1 and dataset2). In dataset1, we divide the dataset into the training set, validation set, and test set, according to the ratio of 7:1:2. The processing of dataset2 was the same as dataset1. During training, we disrupted the number of days but ensured the integrity and continuity of daily data.

4.3. Comparing the Model Descriptions, Training, and Settings

We selected four time series forecasting methods for comparison: BP [30], LSTMs [21], CNN [16] and RNN [31].

BP: BP neural networks can be divided into two parts, BP and neural networks. BP is short for backpropagation.

CNN: The CNN network has structural characteristics, such as local area connection, weight sharing, and downsampling. Weight sharing in a convolutional neural network makes its network structure more similar to that of a biological neural network.

RNN: RNN networks, specifically recurrent neural networks with LSTM units, efficiently handle sequence problems.

LSTMs: A combination forecasting model using the LSTM network, optimized by the ant lion optimization algorithm, is based on the ensemble empirical mode decomposition and K-means clustering algorithm.

We trained all models using the Adam optimizer [32], with the learning rate set at 0.0001; the learning rate was attenuated by the trained epoch number, which was 5×10^{-5}, 1×10^{-5}, 5×10^{-6}, 1×10^{-6}, 5×10^{-7}, 1×10^{-7}, and 5×10^{-8}; when the epoch numbers were 2, 4, 6, 8, 10, 15, and 20, the learning rate decayed sequentially, compared to the SGD optimizer. We found that the Adam optimizer worked well for various models in our setup. The batch_size was generally set to 32, which can be reduced when the memory is low, and has little impact on the prediction effect.

4.4. Experimental Results

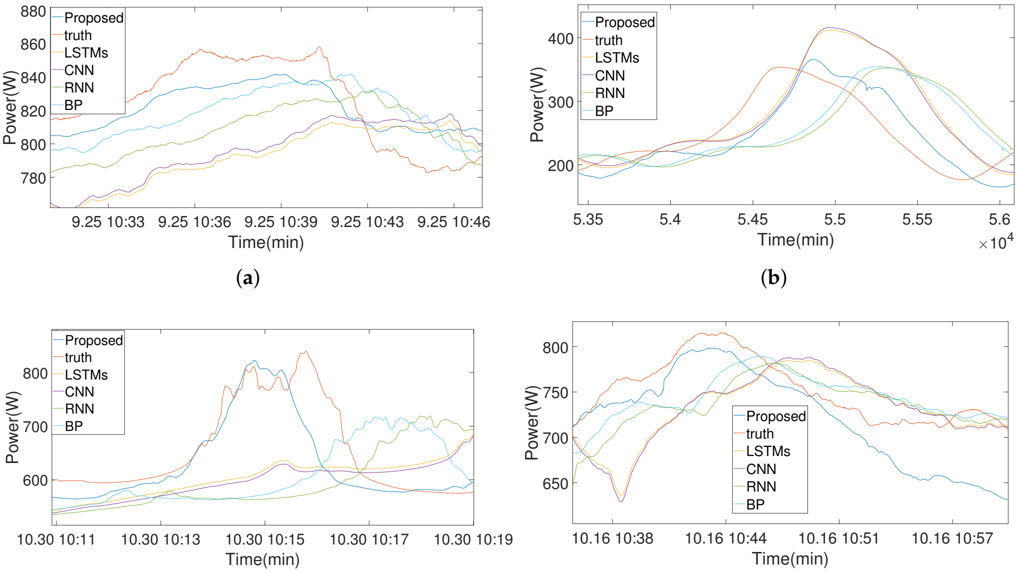
In order to ensure the stability and authenticity of the experimental results, we conducted several experiments on the two datasets. The experimental results are shown in Tables 1 and 2. As can be seen from the table, we list the mean and variance of the errors for each model on the two datasets. When the prediction time was 60 s, the error of our model was not much different from the comparison model. This is because the prediction time was short, the model increased the weight of the target sensor information and paid less attention to the auxiliary sensor information. However, as the prediction time increased, our model had much lower errors than the comparison model. In terms of stability, we used the standard deviation. It can be seen that our model has high stability. The effect of our model greatly improved after adding auxiliary information from surrounding PV sensors; in particular, as the delay time increased, our model exhibited a slower decline in prediction accuracy compared to the other models in the comparison. As the prediction times increased to 300 s and 600 s, there was a significant gap between the MSE and MAPE values in each model. The reason for this huge gap is that there was a gap among the models in their ability to assess the decline time of the PV forecast curve. As shown in Figure 4, one can see how each model predicts the curve at 300 s. When the real PV curve fell, our model reacted more quickly to sense the drop compared to other models. Other models require a delay of about 200–250 s to sense a drop. When the prediction time reached 600 s, as seen in Figure 5, the time gap between the models in sensing the decline of the real PV curve was greater. However, our model incorporates information from PV auxiliary sensors, reducing the time it takes for the model to sense the decline in the PV curve, and even anticipate it in advance. The x coordinates of all pictures represent Beijing time (UTC + 8:00), and the y coordinates represent power generation.

Table 1. The average MSE value and MSE standard deviation for the two datasets; the STD represents the standard deviation of MSE, and AMSE represents the average MSE (both the MSE and the standard deviation are calculated using denormalized values. The number of trials is 10).

Dataset	**Dataset1**								**Dataset2**							
Predict Time	**60 s**		**180 s**		**300 s**		**600 s**		**60 s**		**180 s**		**300 s**		**600 s**	
Metric	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**
our	8097.39	1648.91	13,349.34	1909.11	9022.29	1976.04	23,116.01	2065.71	5065.31	1523.42	10,032.25	1746.21	9479.44	1721.09	22,109.77	1999.65
CNN	9412.54	2254.74	18,892.89	3791.05	25,006.58	5862.31	30,065.47	5488.19	4268.51	1164.76	8013.54	1564.29	10,385.9	2011.54	28,647.01	2617.93
LSTMs	7932.59	2615.73	13,478.13	5002.18	13,303.78	5611.20	24,887.52	6138.17	4599.46	2207.84	9371.86	4617.81	11,670.73	5002.66	25,017.65	6002.55
RNN	14,178.05	2005.76	20,396.15	2716.71	23,097.42	4613.68	45,194.02	6061.74	5016.83	1871.56	11,538.06	2164.79	15,761.25	4509.69	42,174.05	7251.6
BP	8561.86	2571.92	13,880.32	4948.32	17,752.83	5032.25	42,652.12	6002.12	4765.14	2051.88	9287.19	2578.47	11,579.32	4076.21	46,521.82	5310.6

Table 2. The average MAPE value and MAPE standard deviation for the two datasets; the STD represents the standard deviation of MAPE, and AMAPE is the average MAPE (both MAPE and the standard deviation are calculated using denormalized values. The number of trials is 10).

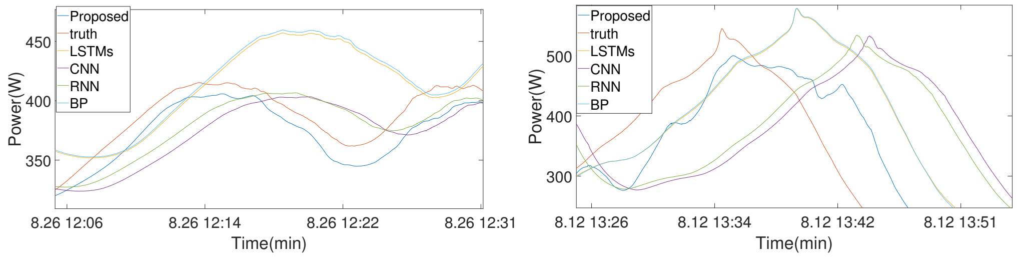
Dataset	**Dataset1**								**Dataset2**							
Predict Time	**60 s**		**180 s**		**300 s**		**600 s**		**60 s**		**180 s**		**300 s**		**600 s**	
Metric	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**
our	8.06	0.64	11.73	1.03	9.77	1.06	16.01	2.34	7.59	0.88	10.52	1.78	10.41	1.35	17.22	3.01
CNN	8.34	1.15	14.92	3.07	21.31	3.65	22.87	4.64	7.12	1.14	12.36	2.04	16.44	3.84	25.91	6.23
LSTMs	7.28	1.06	13.88	1.34	18.27	2.22	21.31	2.88	7.31	0.76	11.52	1.71	14.14	2.03	22.16	4.41
RNN	10.3	3.94	18.54	6.74	23.14	4.17	32.61	5.13	8.12	0.96	15.75	3.05	26.57	4.36	31.65	3.76
BP	8.41	0.96	16.43	3.61	20.88	4.52	33.76	5.21	9.03	1.01	19.33	4.04	27.43	4.11	30.76	5.06



(**a**) (**b**)

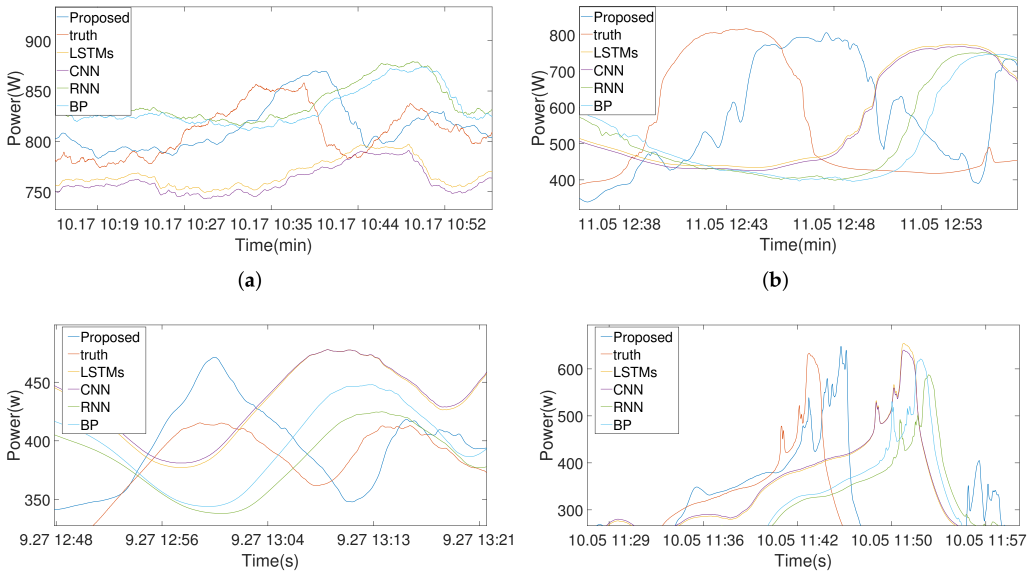
(**c**) (**d**)

Figure 4. *Cont.*



(**e**) (**f**)

Figure 4. (**a**–**f**) display the prediction curves of each model at a prediction time of 300 s.



(**a**) (**b**)

(**c**) (**d**)

Figure 5. (**a**–**d**) display the prediction curves of each model at a prediction time of 600 s.

4.5. Ablation Experiments

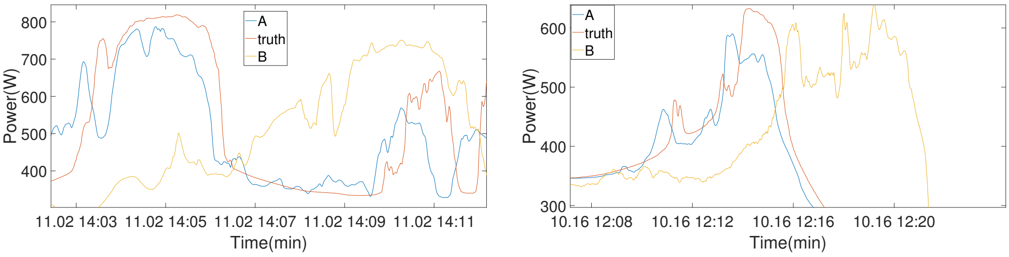
Verify the validity of the auxiliary information:In order to verify the effectiveness of our proposed fusion auxiliary sensor, we performed an ablation experiment and compared the network prediction results with auxiliary sensor information and the network prediction results without auxiliary sensor information. We placed the predicted comparison results in Tables 3 and 4. The model prediction curve results are shown in Figures 6 and 7. From the results, it can be seen that when the model does not add auxiliary information as input, the prediction accuracy will be greatly reduced, which is similar to other comparison models.

Table 3. The average MSE value and MSE standard deviation of the model with or without auxiliary information were included in the two datasets; the STD represents the standard deviation of MSE, and AMSE represents the average MSE (both MSE and the standard deviation are calculated using denormalized values; the number of trials is 10).

Dataset	**Dataset1**				**Dataset2**			
Predict Time	**300 s**		**600 s**		**300 s**		**600 s**	
Metric	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**
have auxiliary information	8817.62	1703.53	22,395.40	2022.12	10,096.62	1814.62	23,836.40	1981.49
no auxiliary information	11,284.16	3285.06	29,626.88	4271.47	10,928.16	3060.17	26,453.81	5001.76

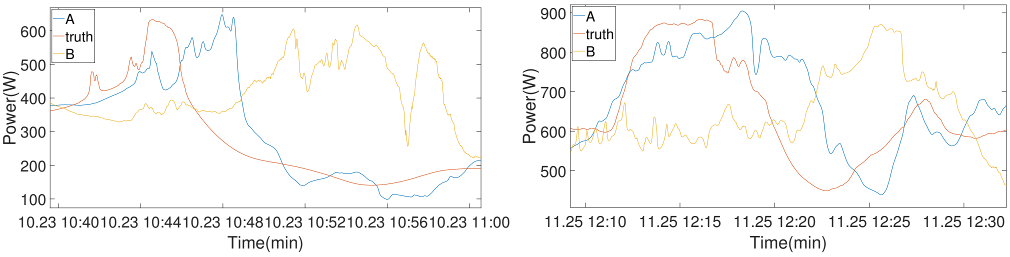
Table 4. The average MAPE value and MAPE standard deviation of the model with or without auxiliary information were included in the two datasets; the STD represents the standard deviation of MAPE, and AMAPE is the average MAPE (both MAPE and the standard deviation are calculated using denormalized values; the number of trials is 10).

Dataset	**Dataset1**				**Dataset2**			
Predict Time	**300 s**		**600 s**		**300 s**		**600 s**	
Metric	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**
have auxiliary information	8.31	0.92	16.51	2.09	10.31	1.03	19.51	2.13
no auxiliary information	12.61	1.78	23.82	3.23	18.61	1.98	24.82	2.79



(**a**) (**b**)

Figure 6. (**a**,**b**) show the model prediction curves with and without auxiliary information input, respectively, at a prediction time of 300 s (line A represents the prediction with auxiliary information and line B represents the prediction without auxiliary information).



(**a**) (**b**)

Figure 7. (**a**,**b**) show the model prediction curves with and without auxiliary information input, respectively, at a prediction time of 600 s (line A represents auxiliary information and line B without auxiliary information).

What if the model output is better? In the VIT model, the output of the model is a token added to the image's input sequence, which contains the classification information of the image. However, this method is not necessarily suitable for PV forecasting. In order to verify the effectiveness of this method, we used three outputs of the model for comparison, namely the output token, the mean sequence of the output time input sequence, and the output sequence obtained by maximum pooling of the time input sequence. The detailed comparison results are shown in Tables 5 and 6.

Table 5. The average MSE value and MSE standard deviation of different output modes in two datasets; the STD represents the standard deviation of MSE, and AMSE represents the average MSE (both MSE and the standard deviation are calculated using denormalized values; the number of trials is 10).

Dataset	**Dataset1**				**Dataset2**			
Predict Time	**300 s**		**600 s**		**300 s**		**600 s**	
Metric	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**	**AMSE**	**STD**
output token	9242.11	1627.49	21,305.40	2075.19	10,077.62	1474.81	23,024	2099.16
average output	10,284.16	1835.08	26,626.88	2267.71	11,204.16	1502.16	22,943.37	2076.93
maximum pooling output	9967.36	1727.59	24,112.7	2109.75	12,683.33	1536.67	21,670.19	1993.29

Table 6. The average MAPE value and the standard deviation of the model with or without auxiliary information were included in the two datasets; the STD represents the standard deviation of MAPE, and AMAPE is the average MAPE (both MAPE and the standard deviation are calculated using denormalized values; the number of trials is 10).

Dataset	**Dataset1**				**Dataset2**			
Predict Time	**300 s**		**600 s**		**300 s**		**600 s**	
Metric	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**	**AMAPE**	**STD**
output token	11.31	0.95	16.51	2.07	10.71	1.03	17.91	1.95
average output	11.61	0.93	18.23	2.36	11.96	1.26	18.01	1.98
maximum pooling output	12.74	1.15	18.67	2.17	12.23	1.18	17.21	1.92

The number of encoder layers used to extract features: In the experimental results, after adding the auxiliary information of the peripheral PV sensor, the advanced amount predicted by the model greatly improved, but it did not improve to the point of satisfaction. We assume that this could be due to insufficient layers in the feature extraction layer of the encoder. Thus, we set up this ablation experiment, hoping to find out the number of layers suitable for feature extraction; the experimental results are shown in Figures 8 and 9. In this experiment, we fixed the parameters of the model and then changed the number of layers in the encoder to prevent unfair results from different initial values.

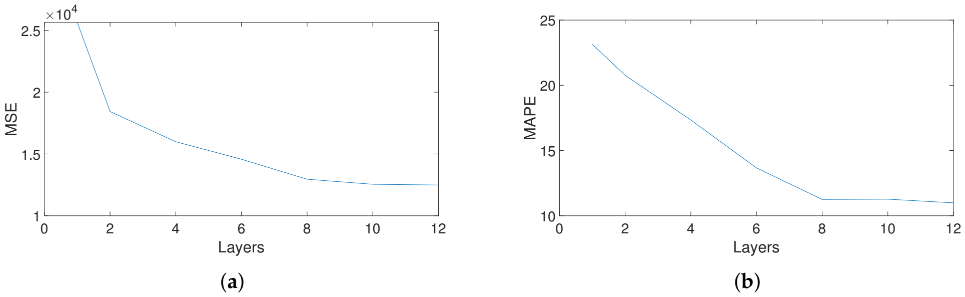


Figure 8. (**a**) The MSE and (**b**) the MAPE of our model on different encoder layers (prediction time 300 s; both MSE and MAPE are calculated using denormalized values).

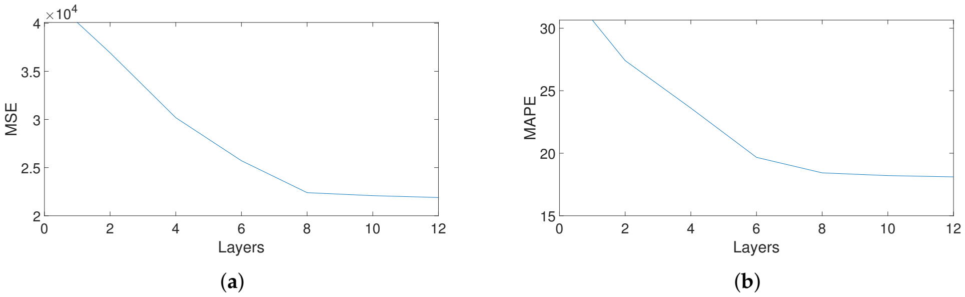


Figure 9. (**a**) The MSE and (**b**) MAPE of our model on different encoder layers (prediction time 600 s; both MSE and MAPE are calculated using denormalized values).

5. Conclusions

In this paper, the VIT model was improved to enable it to predict PV directly. In order to deal with the problem of reduced light radiation intensity caused by the occlusion of sunlight due to cloud motion, which leads to a decrease in prediction accuracy, auxiliary PV sensor information was added to the network input to effectively improve the accuracy of PV prediction; it was found from the actual prediction map that the network can perceive the cloud motion in advance and make responsive predictions. In the comparison experiments of each network, when the prediction time was 300 s, our MSE and MAPE were 9245.2 and 8.94%, respectively. Compared to the best network, our MSE was reduced by 18% and the accuracy improved by 4%. When the prediction time was 600 s, our MSE and MAPE were 23,763 and 16.45%, respectively. Compared to the best network, our MSE decreased by 11% and the accuracy increased by 4%.

Although these results are encouraging, there are still some shortcomings. For example, the network's ability to predict significant decreases in light radiation intensity is not yet at a satisfactory level. We hope to further explore the fusion between auxiliary PV sensor information and target PV sensor information in the future, to enhance the network's ability to predict more accurately and improve the overall accuracy of PV forecasting.

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

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available upon request from the corresponding authors.

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

Nomenclatures

l	time length of PV sequence
w	the number of PV sensors
$\mathbf{X} \in R^{l \times w}$	information sequence of PV sensor
$\mathbf{Y} \in R^{l \times 1}$	PV prediction sequence output by the model
$Z_0 \in R^{(w+1) \times D}$	model initial input
$z_0^0 \in R^{l \times 1}$	a learnable sequence
$Z_\iota \in R^{(w+1) \times D}$	the sequence after through the ι-th layer encoder layer
$A_{pos} \in R^{(w+1) \times D}$	position encoding of model input

$LN(\cdot)$	normalization
$\mathrm{SA}(\cdot)$	self-attention computation
$MSA(\cdot)$	multi-head self-attention
$MLP(\cdot)$	multi-layer perception
$U_{QKV} = [W^Q, W^K, W^V]$	the weight matrix of Q, K, V in self-attention
$S \in R^{w \times w}$	geographic information matrix

References

1. Deo, R. Machine learning in medicine. *Circulation* **2015**, *10*, 1920–1930. [CrossRef] [PubMed]
2. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *8*, 53. [CrossRef]
3. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Xiaohua, Z.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 ×16 Words: Transformers for Image Recognition at Scale. *arXiv* **2015**, arXiv:2010.11929.
4. Yang, C.; Thatte, A.A.; Xie, L. Multitime-Scale Data-Driven Spatio-Temporal Forecast of Photovoltaic Generation. *IEEE Trans. Sustain. Energy* **2015**, *6*, 104–112. [CrossRef]
5. Cavalcante, L.; Bessa, R.J. Solar power forecasting with sparse vector autoregression structures. In Proceedings of the 2017 IEEE Manchester PowerTech, Manchester, UK, 18–22 June 2017; pp. 1–6.
6. Bacher, P.; Madsen, H.; Aalborg Nielsen, H. Online short-term solar power forecasting. *Sol. Energy* **2009**, *83*, 1772–1783. [CrossRef]
7. Zeng, J.W.; Qiao, W. Short-term solar power prediction using an RBF neural network. In Proceedings of the 2011 IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 24–29 July 2011; pp. 1–8.
8. Pedro, H.; Coimbra, C.F. Assessment of forecasting techniques for solar power production with no exogenous inputs. *Sol. Energy* **2012**, *86*, 2017–2028. [CrossRef]
9. Bouzerdoum, M.; Mellit, A.; Massi Pavan, A. A hybrid model (SARIMA—SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant. *Sol. Energy* **2013**, *98*, 226–235. [CrossRef]
10. Wu, Y.K.; Chen, C.R.; Rahman, H.A. A Novel Hybrid Model for Short-Term Forecasting in PV Power Generation. *Int. J. Photoenergy* **2014**, *2014*, 569249. [CrossRef]
11. Rana, M.; Koprinska, I.; Agelidis, V.G. Univariate and multivariate methods for very short-term solar photovoltaic power forecasting. *Energy Convers. Manag.* **2016**, *121*, 380–390. [CrossRef]
12. Arash, A.; Thomas, X.W.; Benito, R. A Hybrid Algorithm for Short-Term Solar Power Prediction—Sunshine State Case Study. *IEEE Trans. Sustain. Energy* **2017**, *8*, 582–591.
13. Shang, C.F.; Wei, P.C. Enhanced support vector regression based forecast engine to predict solar power output. *Renew. Energy* **2018**, *127*, 269–283. [CrossRef]
14. Behera, M.K.; Majumder, I.; Nayak, N. Solar photovoltaic power forecasting using optimized modified extreme learning machine technique. *Eng. Sci. Technol. Int. J.* **2018**, *21*, 428–438. [CrossRef]
15. Eseye, A.T.; Jianhua, Z.; Dehua, Z. Short-term photovoltaic solar power forecasting using a hybrid Wavelet-PSO-SVM model based on SCADA and Meteorological information. *Renew. Energy* **2018**, *118*, 357–367. [CrossRef]
16. Jeong, J.; Kim, H. Multi-Site Photovoltaic Forecasting Exploiting Space-Time Convolutional Neural Network. *Energies* **2019**, *12*, 4490. [CrossRef]
17. Shih, S.Y.; Sun, F.K.; Lee, H.Y. Temporal pattern attention for multivariate time series forecasting. *Mach. Learn.* **2018**, *12*, 1–21. [CrossRef]
18. Simeunović, J.; Schubnel, B.; Alet, P.J.; Carrillo, R.E. Spatio-Temporal Graph Neural Networks for Multi-Site PV Power Forecasting. *IEEE Trans. Sustain. Energy* **2021**, *13*, 1210–1220. [CrossRef]
19. Li, Q.; Zhang, X.Y.; Ma, T.J.; Liu, D.G.; Wang, H.; Hu, W. A Multi-step ahead photovoltaic power forecasting model based on TimeGAN, Soft DTW-based K-medoids clustering, and a CNN-GRU hybrid neural network. *Energy Rep.* **2022**, *3*, 10346–10362. [CrossRef]
20. Zhu, K.; Fu, Q.; Su, Y.X.; Yang, H. A photovoltaic power forecasting method based on EEMD-Kmeans-ALO-LSTM. In Proceedings of the 2022 7th Asia Conference on Power and Electrical Engineering (ACPEE), Hangzhou, China, 15–17 April 2022; pp. 251–256.
21. Zhou, H.X.; Zhang, Y.J.; Yang, L.F.; Liu, Q.; Yan, K.; Du, Y. Short-Term Photovoltaic Power Forecasting Based on Long Short Term Memory Neural Network and Attention Mechanism. *Energy Rep.* **2019**, *7*, 78063–78074. [CrossRef]
22. Qu, Y.P.; Xu, J.; Sun, Y.Z.; Liu, D. A temporal distributed hybrid deep learning model for day-ahead distributed PV power forecasting. *Appl. Energy* **2021**, *304*, 117704. [CrossRef]
23. Abdel-Basset, M.; Hawash, H.; Ripon, K. Chakrabortty and Michael Ryan. PV-Net: An innovative deep learning approach for efficient forecasting of short-term photovoltaic energy production. *J. Clean. Prod.* **2021**, *303*, 127037. [CrossRef]
24. Pérez, E.; Pérez, J.; Segarra-Tamarit, J.; Beltran, H. A deep learning model for intra-day forecasting of solar irradiance using satellite-based estimations in the vicinity of a PV power plant. *Sol. Energy* **2021**, *218*, 652–660. [CrossRef]
25. Guermoui, M.; Bouchouicha, K.; Bailek, N.; Boland, J.W. Forecasting intra-hour variance of photovoltaic power using a new integrated model. *Energy Convers. Manag.* **2021**, *245*, 114569. [CrossRef]

26. Korkmaz, D. SolarNet: A hybrid reliable model based on convolutional neural network and variational mode decomposition for hourly photovoltaic power forecasting. *Appl. Energy* **2021**, *300*, 117410. [CrossRef]
27. Sharma, N.; Mangla, M.; Yadav, S.; Goyal, N.; Singh, A.; Verma, S.; Saber, T. SolarNet: A sequential ensemble model for photovoltaic power forecasting. *Comput. Electr. Eng.* **2021**, *96*, 107484. [CrossRef]
28. Cannizzaro, D.; Aliberti, A.; Bottaccioli, L.; Macii, E.; Acquaviva, A.; Patti, E. Solar radiation forecasting based on convolutional neural network and ensemble learning. *Expert Syst. Appl.* **2021**, *181*, 115167. [CrossRef]
29. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Aidan, N. Gomez and Lukasz Kaiser and Illia Polosukhin. Attention is All you Need. *arXiv* **2017**, arXiv:1706.03762.
30. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
31. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent Neural Network Regularization. *arXiv* **2014**, arXiv:1409.2329.
32. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.

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.