Mathematical Biosciences and Engineering

MBE, 21(1): 1167–1185.
DOI: 10.3934/mbe.2024049
Received: 08 September 2023
Revised: 12 November 2023
Accepted: 27 November 2023
Published: 25 December 2023

http://www.aimspress.com/journal/MBE

Research article

Photovoltaic power prediction based on dilated causal convolutional network and stacked LSTM

Chongyi Tian[1], Longlong Lin[1], Yi Yan[1,]*, Ruiqi Wang[2], Fan Wang[1] and Qingqing Chi[2]

[1] Shandong Key Laboratory of Intelligent Buildings Technology, School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan 250101, China
[2] State Grid Shandong Integrated Energy Services CO., LTD, Jinan 250101, China

* **Correspondence:** Email: yanyi19@sdjzu.edu.cn.

Abstract: Due to the crucial role of photovoltaic power prediction in the integration, scheduling and operation of intelligent grid systems, the accuracy of prediction has garnered increasing attention from both the research and industry sectors. Addressing the challenges posed by the nonlinearity and inherent unpredictability of photovoltaic (PV) power generation sequences, this paper introduced a novel PV prediction model known as the dilated causal convolutional network and stacked long short-term memory (DSLSTM). The methodology begins by incorporating physical constraints to mitigate the limitations associated with machine learning algorithms, thereby ensuring that the predictions remain within reasonable bounds. Subsequently, a dilated causal convolutional network is employed to extract salient features from historical PV power generation data. Finally, the model adopts a stacked network structure to effectively enhance the prediction accuracy of the LSTM component. To validate the efficacy of the proposed model, comprehensive experiments were conducted using a real PV power generation dataset. These experiments involved comparing the predictive performance of the DSLSTM model against several popular existing models, including multilayer perceptron (MLP), recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU), stacked LSTM and stacked GRU. Evaluation was performed using four key performance metrics: Mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE) and R-squared (R2). The empirical results demonstrate that the DSLSTM model outperforms other models in terms of both prediction accuracy and stability.

Keywords: photovoltaic (PV) prediction; dilated causal convolution network (DCCN); stacked LSTM network; deep learning; hybrid deep learning

1. Introduction

The extensive utilization of fossil fuels has given rise to profound energy crisis challenges and greenhouse gas emission issues. Consequently, energy conservation and emission reduction has emerged as a topic of common concern for all countries in the present world [1]. Photovoltaic (PV) power generation is known as a clean, safe, and sustainable renewable energy source [2], which has received special attention from investors and researchers because of its low cost of use, long lifetime, no greenhouse gas emissions [3], easy accessibility, low maintenance difficulty, abundant resources, and fixed payback period [4]. However, the fluctuating and stochastic nature of solar irradiance has introduced complexities to short-term scheduling within power systems, thereby incurring substantial supplementary costs for power suppliers [5]. Therefore, accurate forecasting of PV power generation is extremely important and is of great benefit to both power suppliers and power systems. Power suppliers need to obtain accurate information about PV power generation in order to set up specialized commercial offers to reduce production costs and increase profits; power systems can mitigate the negative impacts caused by uncertainty in PV power generation and ensure its stable and reliable operation [6].

According to the classification criteria of reference [7], existing PV power generation prediction methods can be classified into four main categories: Physical models, statistical models, machine learning models and deep learning models.

The most distinctive feature of the physical model is that it does not require any historical data [8] and it is based on solar irradiance and a complex set of mathematical equations describing the physical state of the PV system [4]. Under stable weather conditions, a physical model can achieve an acceptable level of predictive accuracy. However, its precision cannot be guaranteed when there are significant weather fluctuations. In the literature [9,10], several typical physical models are presented. In addition physical models can only produce meteorological data values after 6 hours, thus limiting its applicability for very short-term forecasting purposes [11].

Statistical models use purely mathematical equations to create a mapping relationship between historical and target forecast data to predict future PV generations [12]. These are data-driven approaches with the advantages of easy modeling and inter-regional generalizability. As a result, various statistical methods have been widely used in recent years to forecast PV power generation, including regression methods (RM) [13], autoregressive moving average models (ARMA) [14] and its improved version seasonal autoregressive integrated moving average model (SARIMA) [15]. However, the volatility and non-periodicity of the PV power generation time series may undermine the prerequisites required for the application of statistical methods, such as a substantial amount of suitable and high-quality input data, as well as an appropriate input time range.

Machine learning (ML) models, which have evolved from the foundation of statistical models, have been used in recent years in various fields of science and engineering, including the prediction of PV power generation. It follows the process of preparing data, training algorithms, generating models, and then making and refining predictions [16]. ML models frequently used in PV power generation prediction include artificial neural networks (ANN) [17], support vector machines (SVM) [18] and extreme learning machine (ELM) [19].

The above models are shallow networks and are suitable for handling small-scale datasets. With the explosive growth of data volume, these methods are not able to mine effective features from massive data, so deep learning can be used to address this bottleneck. Deep learning (DL) models have been used to solve many research areas and have been successfully applied to PV power generation [20]. Deep learning is capable enough to extract, transform and model the intricate features inherent in the PV power generation time series and provide not only more effective but also superior prediction results than traditional models. LSTM, as one of the most important deep learning techniques, has been frequently applied in related work [2].

Literature [21] compared LSTM with the persistence algorithm, linear least squares regression algorithm and back propagation algorithm using a dataset collected in the island of Santiago, Cape Verde, as an input, and the experimental results showed that the LSTM algorithm performs better in terms of prediction accuracy and generalization ability. Although a single LSTM unit outperforms compared to traditional physical and ML models, its prediction accuracy is still limited by the simple network structure. Literature [22] first uses the Bayesian optimization algorithm to optimize the hyperparameters of the deep neural network, which solves the problem of unidirectional data transfer of LSTM and achieves bidirectional propagation of historical and future information. However, the BiLSTM network is still essentially a simple RNN model, which is capable of effectively extracting historical useful information of the time series, but has a weak feature extraction capability for the input data. Literature [23] uses CNNto extract the features of the influencing factors of the input data and uses BiLSTM for time series prediction. However, in order to achieve longer time-size convolution, an extremely deep network of convolutional layers is required. In addition, most of the above prediction models are based only on a large amount of data, ignoring its real-world issues or physical implications, thereby potentially yielding irrational predictive outcomes.

Therefore, based on the above studies, this paper proposes a PV prediction model based on dilated causal convolution network (DCCN) and stacked LSTM (SLSTM) in conjunction with the physical constraints. The main contributions of our work can be summarized as follows:

- The introduction of the basic characteristics of PV power plants as physical constraints ensures the rapidity of subsequent model training and the reasonableness of model output;
- Employing temporal convolutional network for feature extraction can fully exploit the spatial features of PV historical data and improve the prediction accuracy of the subsequent model;
- The stacked network structure is used for training. Compared with the original LSTM model, the superiority of SLSTM is that it can more fully learn the temporal features in the PV sequences, thus elevating the prediction accuracy of the model.

The rest of the paper is described as follows: Section 2 gives details of the theoretical background of the proposed prediction model; Section 3 presents the experimental results and discussions, comparing the designed model against various others; Section 4 summarises the work of this paper and provides an outlook for future work.

2. Materials and methods

2.1. Overview

The overall framework of the proposed dilated causal convolution network-stacked LSTM model (DSLSTM) is shown in Figure 1. The structure of the method consists of a dilated causal convolutional network for feature extraction and a SLSTM layer. It consists of three main steps: Data preprocessing, training of the DL network and PV power prediction.

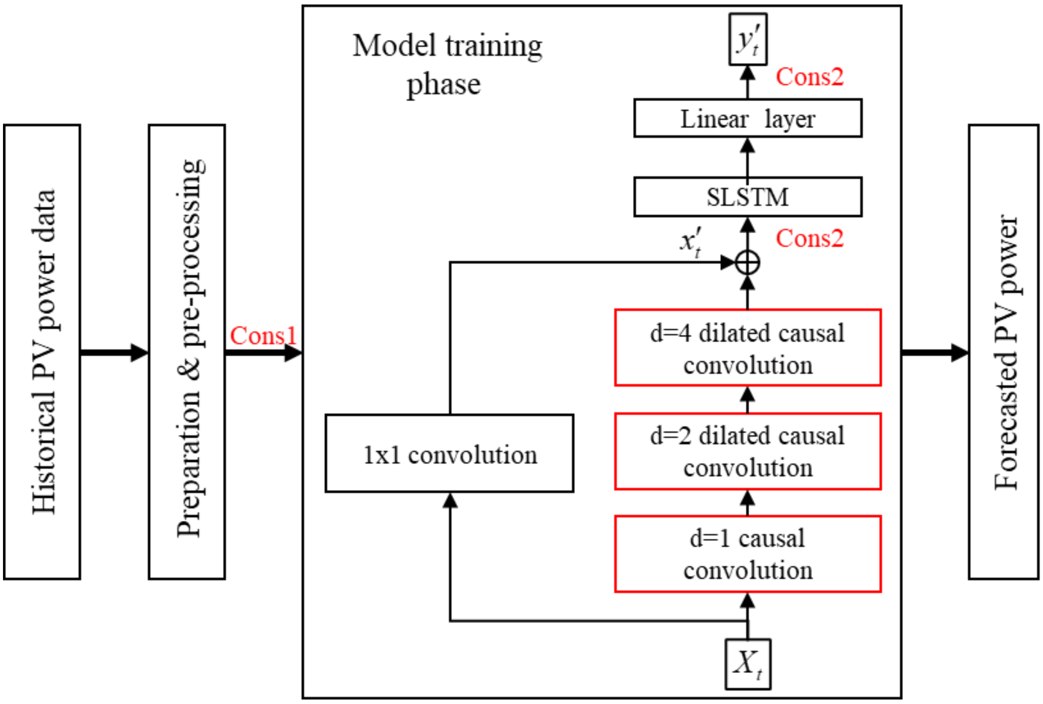

Figure 1. The proposed DSLSTM model. The PV forecasting task was divided into three steps: Input pre-processing, model training, and output prediction.

2.2. Physical constraint

In this subsection, physical constraints are initially extracted from the domain knowledge and physical laws of PV and then integrated into the construction of the DSLSTM model. The aim is to overcome the limitations of DL algorithms, which often yields predictions based solely on extensive data, potentially leading to results that do not adhere to the physical laws of the real world. This includes scenarios such as negative power generation during daytime hours (05:00 to 19:00) and positive power generation during nighttime hours (19:00 to 05:00), which may not align with the actual physical constraints.

Specifically, there are two types of constraints in the structure of DSLSTM (denoted as Cons1 and Cons2 in Figure 1). The first constraint, known as the data cropping module, is designed based on physical knowledge or general knowledge of PVs [17]. Its purpose is to eliminate physically unreasonable predictions during the training and testing processes by cropping the data. For instance, it addresses the occurrence of positive power generation during the nighttime hours of PVs (from 19:00 on the first day to 05:00 on the following day). The original dataset contained data points at 15-minute intervals, totaling 96 points per day. Given that the photovoltaic power generation remains at 0 before 5:00 in the morning and after 19:00 in the evening, data for the training of the model was only selected from the period 05:00 to 19:00 each day. Consequently, the data scale reduced from 35,040 data points per year to 20,440 data points per year, representing a decrease of 41.7%. This approach not only prevents occurrences of positive power generation during nighttime but also reduces the model training time.

The second constraint, called the data filtering module, is used to limit the output of the model to a reasonable range during training and testing. It is designed based on knowledge of the natural science of PV to eliminate physically unreasonable predictions such as negative power generation. According to the laws of physics, the generation value of PV should be greater than zero in reality, so the output of the model should be positive. Therefore, the predicted output of DSLSTM y_t^p should be constrained by Eq (1):

$$y_t^p = \mathrm{ReLU}(f(x_1, x_2, ..., x_n)), \tag{1}$$

where ReLU() represents the rectified linear unit function. When the input to the ReLU function is less than 0, the ReLU function returns 0; when the input is greater than 0, the original value of the input is returned. x_i represents the ith input and f() represents the neural network.

The inclusion of the above two constraints ensures the rapidity of the DSLSTM model training and the reasonableness of the output.

2.3. Feature extraction networks based on dilation causal convolutional networks

2.3.1. Causal convolution

The principle of causal convolution is that the current state cannot contain future information. In other words, the output at time step t + 1 is only correlated with the previous time steps, i.e., t, t-1, ..., t-n (as illustrated in Figure 3, where n = 14 in this paper). This approach effectively prevents the issue of information leakage caused by regular convolution.

A significant drawback of causal convolution lies in its requirement for extremely deep networks or exceedingly large filters to achieve convolutions over longer time spans, both of which were not initially entirely practical. Therefore, the introduction of the dilation factor, denoted as ‘d’, serves to enlarge the receptive field and accept a broader range of input information.

2.3.2. Dilation convolution

A simple causal convolution can only recall a history that is linearly related to the depth of the network. This makes causal convolution difficult to apply to tasks that require longer time sizes. In this paper, dilation convolution is used to increase the sense field. Dilation convolution is equivalent to introducing a fixed-length interval between two neighboring elements in the convolution kernel of ordinary convolution. When the dilation factor is equal to 1, the dilation convolution is the ordinary convolution. The use of larger dilation factor represents a larger receptive field. The structure of the dilation causal convolution is shown in the following figure.

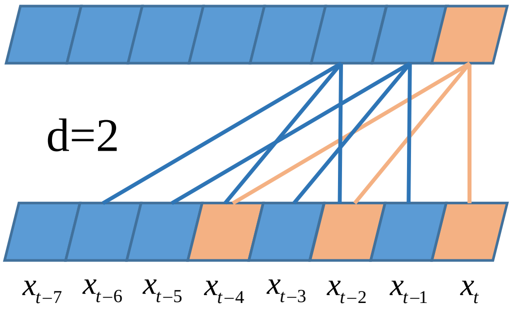

Figure 2. The architecture of dilation convolution.

2.3.3. Residual connection

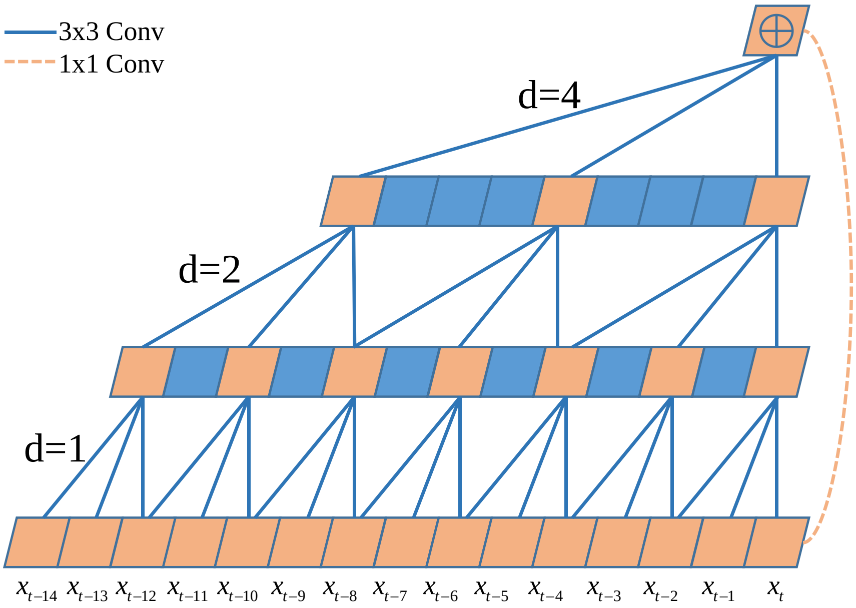

Figure 3. The architecture of proposed DCCN.

With the increase of convolutional layers, there is a risk of losing crucial feature information, so the residual connection that can realize the transfer of data across layers is introduced. At the same time, the residual connection can also effectively alleviate the gradient disappearance or gradient explosion problem that exists in the deep network. The structure of the feature extraction module proposed in this paper is shown in Figure 3, which consists of two parts: The left section is the dilation causal convolution, while the right section employs a 1 × 1 convolution for residual connections. The 1 × 1 convolution ensures matching tensor shapes when elements are added together [24]. The formula for the residual connection is shown in Eq (2):

$$\mathrm{DCCN}(x) = \mathrm{DCC}(x) + \mathrm{Conv}(x)\,, \tag{2}$$

where x denotes the input, $\mathrm{Conv}(x)$ denotes the output of the 1 × 1 convolution, $\mathrm{DCC}(x)$ denotes the output of the dilation causal convolution and $\mathrm{DCCN}(x)$ denotes the output of the dilated causal convolutional network.

2.4. RNNs

Different architectures for DL, such as RNNs and deep neural networks (DNNs), have been used in many application areas and have produced superior to most data modeling techniques. A distinctive aspect separating RNNs from DNNs lies in the fact that RNNs generate not only outputs but also hidden states. These hidden states, subsequently coupled with input data at the subsequent timestamp, are employed to adapt network weights, thereby giving rise to the construction of deep learning architectures. The hidden states have the capacity to retain prior information and deploy it in subsequent training phases [24].

RNN is a powerful and robust neural network that uses existing time series data to predict future data for a specific length of time. In RNN, the output of the previous timestamp as well as the internal state of the current timestamp will be fed into the RNN unit. Consequently, the current output of the model depends not only on the current inputs, but also bears the influence of the previous states. Therefore, RNN is very promising in processing the historical PV data characterized by sequential characteristics.

2.4.1. LSTM

Although RNN can effectively extract temporal information from temporal data, RNN encounters issues of vanishing and exploding gradients during training on lengthy sequences. To overcome the limitations of RNN, Hochreiter and Schmidhuber [25] proposed LSTM architecture. LSTM adds forget gate, input gate, update gate and output gate to RNN. As the name suggests, the forget gate determines the percentage of long-term memory that is forgotten, the input gate determines what percentage of the current moment can be passed to the amount of cell state at the current moment, the update gate is used to update the cell state and the output gate produces the output at the current moment. These four main components of the LSTM will work and interact with each other in a special way. The internal structure of the LSTM is illustrated in Figure 4.

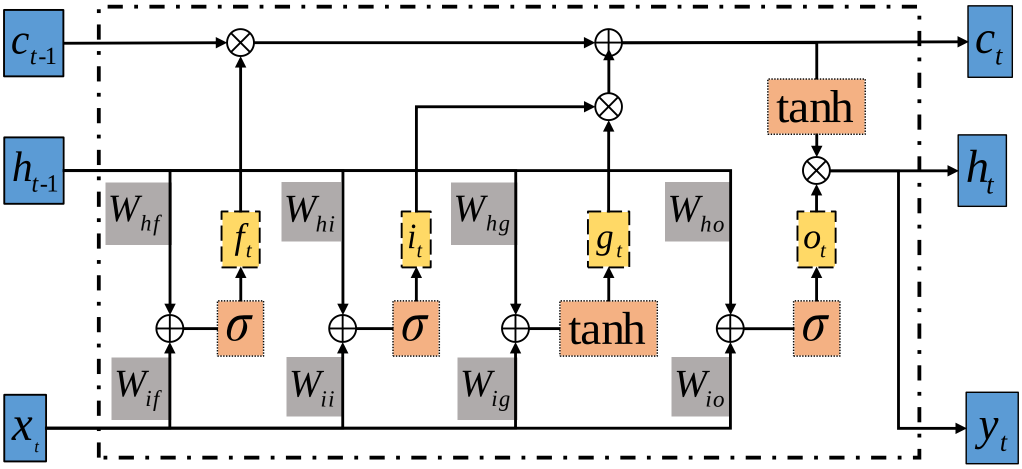

Figure 4. The unfold structure of the LSTM unit.

The forget gate determines the proportion by which the previous timestep cell state, serving as long-term memory, is to be forgotten. The forget gate consists of the hidden state of the previous moment and the input of the current moment, which is finally obtained by the activation function. The process of calculating the forget gate f_t is shown in Eq (3):

$$f_t = \sigma(W_{if} x_t + W_{hf} h_{t-1} + b_f), \tag{3}$$

where x_t is the input of the current moment, W_{if} is the weight of the current input, h_{t-1} denotes the hidden state functioning as short-term memory from the previous timestep, W_{hf} represents the weight of the hidden state in the previous moment, and b_f is the bias of the forget gate. The symbol σ denotes the activation function and its outputs take on the range of [0,1], where 0 means complete forgetting and 1 means complete retention.

The input gate determines the proportion of short-term information at the current moment that can be updated into long-term memory, and the process of calculating the input gate i_t is described in Eq (4):

$$i_t = \sigma(W_{ii} x_t + W_{hi} h_{t-1} + b_i), \tag{4}$$

where W_{ii}, W_{hi} and b_i stand for the weight matrix of the current input of the input gate, the weight matrix of the hidden state at the previous moment and the bias of the input gate, respectively.

The update gate is used to control the update of the candidate cell state and the candidate cell state g_t is obtained by the tanh activation function, which takes the value in the range of [–1,1]; the calculation is shown in Eq (5):

$$g_t = \tanh(W_{ig} x_t + W_{hg} h_{t-1} + b_g), \tag{5}$$

where W_{ig}, W_{hg} and b_g, respectively, denote the weight matrix of the current input of the candidate cell state, the weight matrix of the hidden state at the previous moment and the bias of the candidate cell state.

The cell state at the current moment is jointly determined by the forget gate and the input gate. The calculation is shown in Eq (6):

$$c_t = f_t * c_{t-1} + i_t * g_t, \tag{6}$$

The output gate is used to control the output of a cell state and transfer that state to the next cell. The process of calculating the output gate o_t value is shown in Eq (7):

$$o_t = \sigma(W_{io} x_t + W_{ho} h_{t-1} + b_o), \tag{7}$$

where W_{io} represents the weight matrix of the output gate, W_{ho} is the weight of the hidden state in the previous moment and b_o stands for the bias of the output gate.

Upon calculating the forget gate, input gate, update gate, output gate and candidate cell state, LSTM will calculate the output as well as update the hidden state with the following formula:

$$y_t = o_t * \tanh(c_t), \tag{8}$$

$$h_t = y_t. \tag{9}$$

2.4.2. SLSTM

Deep network architectures have demonstrated strong capabilities in dealing with complex nonlinear feature representations [26]. Research indicates that although a single LSTM unit can solve the gradient vanishing and explosion problems in RNN models, its prediction accuracy is still limited by the simple network structure [27]. Therefore, by increasing the stacking depth of the LSTMs, the features of the input sequences can be better learnt, consequently improving the network performance.

The structure of SLSTM is shown in Figure 5. It consists of multiple LSTM layers, where the input of the first LSTM layer is the original data and the output of the last LSTM layer is the prediction result. The inputs of the other intermediate LSTM layers come from the outputs of their previous LSTM layer, and the outputs are used as inputs to the latter LSTM layer. As with the original LSTM, the hidden states and cell states in the stacked LSTM are also passed to the next moment. The difference is that the dimension is increased in this stacked structure.

While stacking multiple LSTM layers significantly enhances the network's capacity to learn from long sequences, excessive layer stacking should be avoided. An increase in the number of layers can lead to slower model update iterations, reduced convergence effectiveness and exponential growth in temporal and memory costs during training. This can make the model susceptible to issues such as gradient vanishing. Therefore, in this paper we choose to adopt a stacked LSTM module consisting of three LSTM layers.

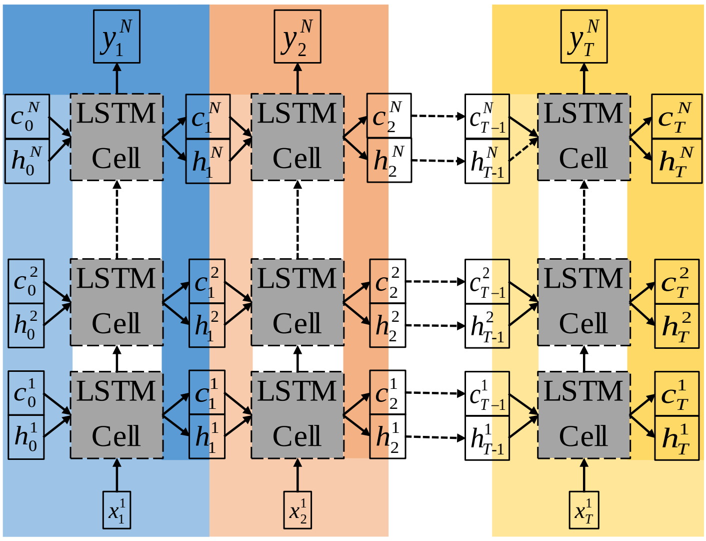

Figure 5. The architecture of SLSTM.

3. Experimental setup

3.1. *Experimental dataset*

The PV data utilized in this study is from Gaoyou, Jiangsu with the PV power plant positioned at 32 degrees, 58 minutes, 31 seconds north latitude and 119 degrees, 36 minutes, eight seconds east longitude, boasting an installed capacity of 10MW. The selected data used for model training and validation spanned from January 1, 2017, 00:00:00 to December 31, 2017, 23:45:00, with a temporal resolution of 15 minutes and 8:2 split ratio for the training and testing sets. As anticipated, solar irradiance is generally higher between 11:00 and 14:00, corresponding to elevated PV power generation during daylight hours. Comparatively, power generation is notably lower during the early morning hours (05:00 to 11:00) and the afternoon (14:00 to 19:00). Notably, power generation remains at zero during the night (19:00 to 05:00 the following day) due to the absence of solar irradiance.

Recent studies have used multivariate datasets consisting of meteorological data or other environmental variables to improve the performance of prediction models [2,5,15,16]. However, in many cases, such as in short-term studies, the addition of these variables has little effect on the accuracy of the predictions due to their small variation over a short period of time (i.e., 15 minutes) [28]. Nonetheless, additional variables complicate the model and slow down the training process. Consequently, this study exclusively considers historical PV generation data as the model input to validate the superiority of the proposed DSLSTM model.

The normalization of data can eliminate the effect of magnitude and dimensions, and overcome the problem of overflow of individual data during the training process. Common normalization techniques are max-min normalization, mean normalization and Z-Score normalization. Considering that the PV generation data is all positive, this study uses max-min normalization that scales the PV data between [0,1]. The formulation for this normalization process is presented as Eq (10):

$$x = \frac{x' - x^{\min}}{x^{\max} - x^{\min}}, \tag{10}$$

where x' and x are the original and normalized values of PV, respectively, $x^{\min}$ denotes the minimum value of PV data and $x^{\max}$ denotes the maximum value of PV data.

3.2. Evaluation indicators

In this paper, we employ four performance evaluation metrics to assess the predictive outcomes: Coefficient of determination (R2), mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE) [18,21,24]. These metrics are used to quantify the accuracy and performance of the predictive model. The MAE value represents the average magnitude of prediction errors, quantifying the average absolute difference between predicted values and actual values; the MSE reflects the average Euclidean distance between the predicted and actual values, and these two metrics are often used to gauge the overall performance of predictive models. RMSE is a widely used method in evaluating prediction errors, which defines the degree of error that exists between the prediction and the actual result, and is usually more sensitive to large deviations between measurements and predictions. Smaller values of MAE, MSE and RMSE indicate better predictive performance. The R2 reflects the correlation between inputs and outputs and is frequently used to assess the fit quality of regression models. In regression models, the closer the sum of squared residuals is to zero, the closer the R2 value is to 1, indicating higher precision in the model's predictions. It is worth noting that due to scenarios where both actual and predicted values are zero within this study, the mean absolute percentage error (MAPE) and symmetric mean absolute percentage error (SMAPE) were not adopted as evaluation metrics. Below are the specific formulas for calculating these four metrics used to measure predictive model performance:

$$MAE\left(y^A, y^P\right) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i^A - y_i^P\right|, \tag{11}$$

$$MSE\left(y^A, y^P\right) = \frac{1}{n}\sum_{i=1}^{n}(y_i^A - y_i^P)^2, \tag{12}$$

$$RMSE\left(y^A, y^P\right) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i^A - y_i^P)^2}\ , \tag{13}$$

$$R2\left(y^A, y^P\right) = 1 - \frac{SS_{res}\left(y^A, y^P\right)}{SS_{tot}\left(y^A, y^P\right)}, \tag{14}$$

where $\bar{y}^A$ and $\bar{y}^P$ denote the average value of the actual output and the average value of the predicted output, respectively; y_i^A and y_i^P denote the true value and the predicted value at the i moment, respectively; n denotes the length of the training samples; SS_{res} stands for the sum of squares of the residuals and SS_{tot} represents the total sum of squares of the real data, which are expressed in the following formulas:

$$SS_{res}\left(y^A, y^P\right) = \sum_{i=1}^{n}(y_i^A - y_i^P)^2\ , \tag{15}$$

$$SS_{tot}\left(y^A, y^P\right) = \sum_{i=1}^{n}(y_i^A - \bar{y}^A)^2\ . \tag{16}$$

In the following section, a case study will be conducted to validate the feasibility and effectiveness of the proposed methodology using real-world PV datasets.

3.3. Simulation setup and hyper-parameter selection

All experiments were conducted on a desktop computing workstation running on an Intel(R) Core(TM) i5-9500F central processing unit (CPU) @ 3.00GHz, NVIDIA GeForce GTX 1060 GPU, 16GB DDR4 RAM, and the operating system is Windows 10 Professional. The network proposed in this paper is built under Python 3.10.9, Pytorch 1.12.1. The Adam optimizer with weight decay set to 0.0001 and a learning rate of 0.0001 is used for optimization and iterative training of the networks through backpropagation.

The choice of hyperparameters of the model is essential for the correct training of the model. The main hyperparameters of the model are set as Table 1.

Table 1. Hyperparameters of the proposed DSLSTM model.

Hyper-Parameters	Qty
Input size (lag)	15 (225 min)
Batch size	8
Kernel size of Conv1d	1 × 3
Dilation step of Conv1d	2
Layers of Conv1d	3
Layers of LSTM	3
Dropout rate of LSTM	0.2
Hidden size of LSTM	64

4. Discussion

4.1. LSTM hidden layer tuning

Theoretically, the more the number of LSTM hidden layers, the stronger the curve fitting ability of the predicted data. However, as the number of layers increases, the neural network structure becomes more and more complex, the training time becomes longer and longer, it is easy to appear overfitting phenomenon and the generalization ability becomes worse. In this paper, we compare the prediction effect when the number of hidden layers is one, two, three, four, five, six, seven, respectively, and at the same time, in order to avoid chance, we randomly do ten experiments and take the average of the best two of them as the most results. The results are shown in Figure 6.

In Figure 6, although the single-layer LSTM prediction results deviate from the actual value, the training time is small; the prediction results of the two, three, four and five-layer LSTM have significant improvement, while the three-layer LSTM has the smallest prediction error; as for the six or seven layers, with the increase of the number of layers, the training time of the LSTM model increases significantly. The possible reason for this is that the network structure is getting more and more complex to appear the phenomenon of overfitting and the generalization ability becomes worse. Taken together, the three-layer LSTM model is optimal for prediction.

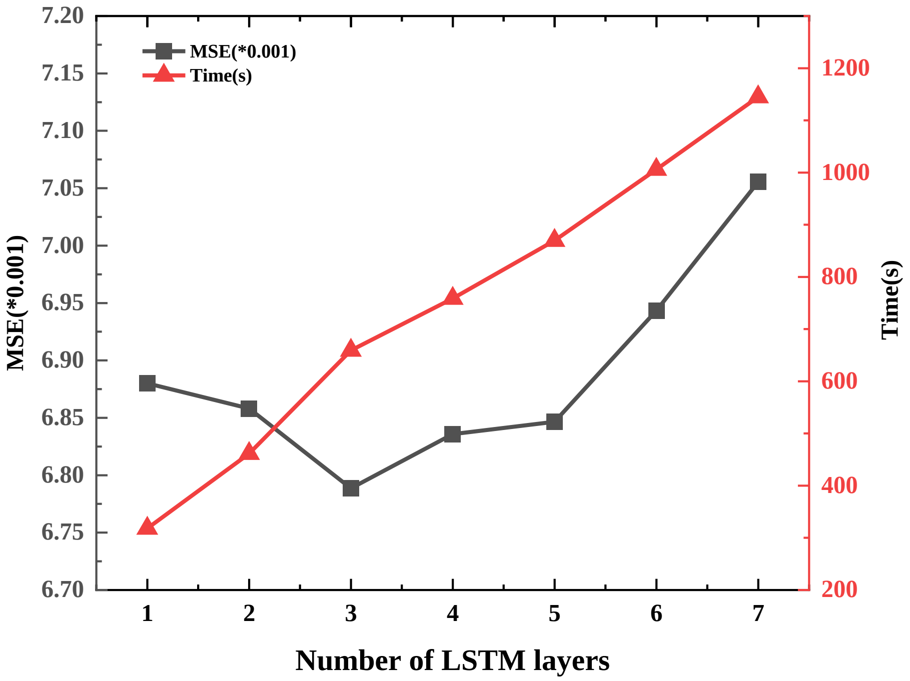

Figure 6. Prediction error and training time for different layers of LSTMs.

4.2. Comparison and analysis of model prediction results

To validate the effectiveness of DSLSTM, the proposed predictive model is compared with MLP, RNN, GRU, LSTM, SGRU, SLSTM, DSGRU and DSLSTM. In order to strictly control the variables, the same dataset is used as input, and to ensure the accuracy of the experimental results and avoid the influence of chance factors, the average value is taken as the prediction result after several experiments on the test set. The effectiveness of the application of various prediction algorithms is shown in Table 2 below.

Table 2. The numerical metrics of the proposed DSLSTM and benchmark models.

	MAE	MSE	RMSE	R2
MLP	4.531	66.575	8.159	0.898
RNN	4.297	65.321	8.082	0.899
GRU	4.414	65.948	8.121	0.901
LSTM	4.154	63.264	7.954	0.903
SGRU	4.263	63.482	7.967	0.903
SLSTM	4.145	62.287	7.892	0.905
DSGRU	3.872	62.400	7.899	0.905
DSLSTM	3.728	59.094	7.687	0.910

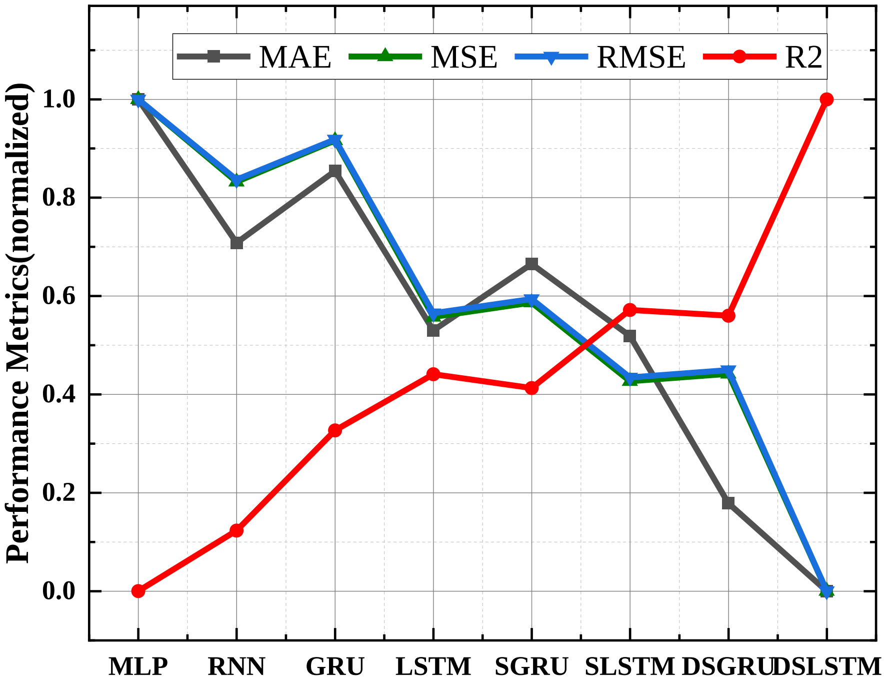

Figure 7. The performance metrics of the proposed DSLSTM and benchmark models on the real dataset.

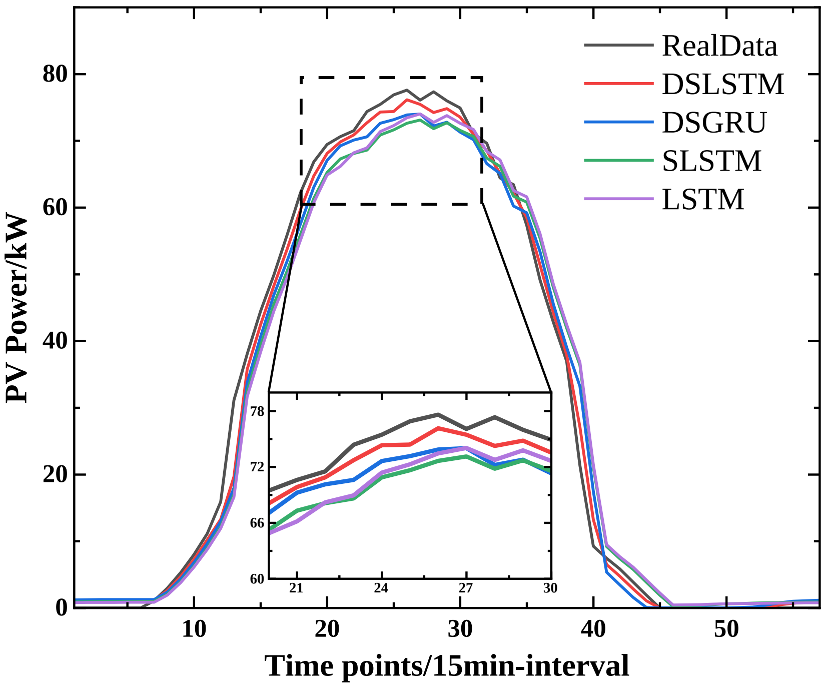

Figure 8. The forecasting results of the proposed DSLSTM and benchmark models on the real dataset.

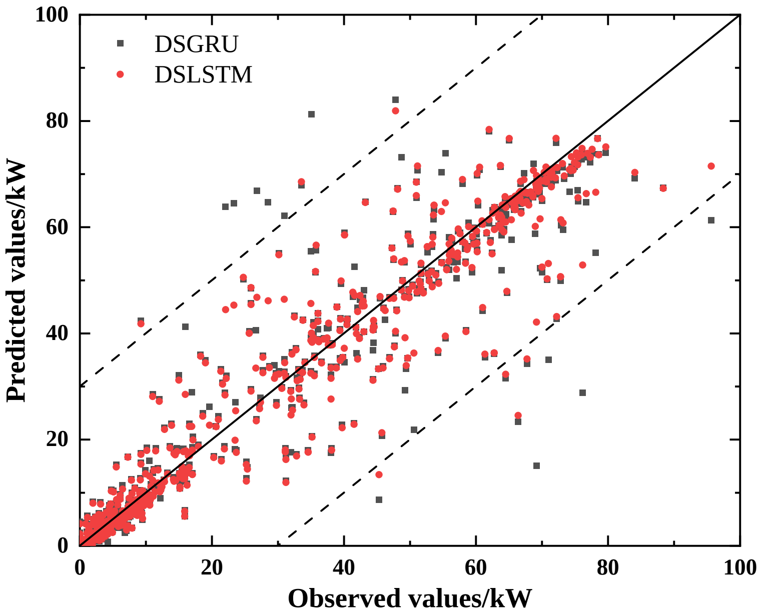

Figure 9. Scatter plots of observed and forecasted PV power generation using DSLSTM and DSGRU

First and foremost, concerning individual models, it is evident from Table 2 that, compared to MLP, RNNs demonstrate superior adaptability and learning capabilities in time-series prediction tasks due to their capacity to retain previous information and incorporate it into current output computation. Moreover, an analysis of Table 2 reveals that, although GRU simplifies LSTM computations and reduces parameters, it falls short of effectively controlling data flow; thus, leading to inferior performance compared to LSTM, especially in scenarios involving sizable datasets. This assertion is corroborated by the comparative results between LSTM and GRU.

Second, we can infer that the performance of stacked models surpasses that of individual models. Single models, constrained by their simplistic network structures, benefit from increased depth through stacked networks, thereby enhancing the learning of distinctive features within input sequences and, consequently, improving network performance. This assertion finds support in the lower MSE of SLSTM by 1.5% and SGRU by 3.7% compared to LSTM and GRU, respectively.

Furthermore, it can be concluded that composite models exhibit significantly superior performance compared to stacked models. Observing Figure 7, it is apparent that the proposed DSLSTM and DSGRU models demonstrate lower MAE , MSE and RMSE and higher R-squared (R2) values. Specifically, compared to the stacked models without the expanded causal convolution network, DSLSTM achieves a reduction of 10, 5.13 and 2.6% in MAE, MSE and RMSE, respectively, while increasing R2 by 0.55%. Similarly, in the case of DSGRU, the MAE, MSE and RMSE decrease by 9.17, 1.7 and 0.86%, respectively, with a corresponding increase of 0.19% in R2. Figure 8 illustrates that prediction models incorporating the expanded causal convolution network exhibits superior performance, primarily owing to the network's ability to capture holistic feature information from historical data, thereby facilitating more accurate PV output predictions. This highlights the significance of feature extraction capabilities.

Finally, compared to DSGRU, DSLSTM demonstrates a 3.7, 5.3 and 2.7% reduction in MAE, MSE, and RMSE evaluation metrics, respectively, along with a 0.6% increase in R2. Figure 9 also indicates that the prediction deviation of the DSLSTM model is less than that of DSGRU, underscoring the ability of the proposed model to offer more precise and reliable PV predictions, thus exhibiting promising practical application prospects.

Naturally, the DSLSTM network proposed in this paper demands more time for training. However, in practical applications, our focus remains on prediction time (testing time), while training can be completed during offline and idle periods.

5. Conclusions

The prediction of PV power generation has been extremely important in the development of the entire PV industry. This article presents an innovative deep learning-based framework to address the short-term prediction challenges inherent in PV power generation. Through experimental simulations and analytical examinations, the following conclusions have been drawn:

1) The introduction of the fundamental physical constraint properties of PV power plants ensures the rapidity of the subsequent model training and the reasonableness of the model prediction output.

2) For the huge dataset, the use of DCCN for feature extraction can fully exploit the relevant features to the PV historical data, thereby enhancing the prediction accuracy of the model.

3) The adoption of a SLSTM model for training presents an advantage over the conventional LSTM model due to its intricate network architecture, which more comprehensively captures the patterns of variation within the solar sequence; thus, enhancing the predictive accuracy of the model.

Through comparative analysis with various alternative models, it becomes evident that the proposed DSLSTM model outperforms in all performance metrics. From the comprehensive results, this indicates that the proposed DSLSTM model possesses excellent overall performance, thereby demonstrating substantial feasibility for practical applications.

In future work, the anticipated direction will involve an in-depth anticipation of various decomposition algorithms to improve the accuracy of short-term PV power forecasting. In addition, migration learning will be considered to enhance the practicality of the model in response to the insufficient amount of data from PV power plants.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Y. Dai, Y. Wang, M. Leng, X. Yang, Q. Zhou, LOWESS smoothing and random forest based GRU model: a short-term photovoltaic power generation forecasting method, *Energy*, **256** (2022), 124661. https://doi.org/10.1016/j.energy.2022.124661
2. H. Zhou, Y. Zhang, L. Yang, Q. Liu, K. Yan, Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism, *IEEE Access*, **7** (2019), 78063–78074. https://doi.org/10.1109/ACCESS.2019.2923006
3. A. Khanlari, A. Sözen, C. Sirin, A. D. Tuncer, A. Gungor, Performance enhancement of a greenhouse dryer: analysis of a cost-effective alternative solar air heater, *J. Clean. Prod.*, **251** (2020), 119672. https://doi.org/10.1016/j.jclepro.2019.119672
4. U. K. Das, K. S. Tey, S. Mehdi, S. Mekhilef, M. Y. I. Idris, W. Van Deventer, et al., Forecasting of photovoltaic power generation and model optimization: a review, *Renew. Sustain. Energy Rev.*, **81** (2018), 912–928. https://doi.org/10.1016/j.rser.2017.08.017
5. Y. Liu, Y. Liu, H. Cai, J. Zhang, An innovative short-term multihorizon photovoltaic power output forecasting method based on variational mode decomposition and a capsule convolutional neural network, *Appl. Energy*, **343** (2023), 121139. https://doi.org/10.1016/j.apenergy.2023.121139
6. S. Lin, C. Li, F. Xu, D. Liu, J. Liu, Risk identification and analysis for new energy power system in China based on d numbers and decision-making trial and evaluation laboratory (DEMATEL), *J. Clean. Prod.*, **180** (2018), 81–96. https://doi.org /10.1016/j.jclepro.2018.01.153
7. R. Ahmed, V. Sreeram, Y. Mishra, M. D. Arif, A review and evaluation of the state-of-the-art in PV solar power forecasting: techniques and optimization, *Renew. Sustain. Energy Rev.*, **124** (2020), 109792. https://doi.org/10.1016/j.rser.2020.109792

8. H. Wang, Y. Liu, B. Zhou, C. Li, G. Cao, N. Voropai, E. Barakhtenko, Taxonomy research of artificial intelligence for deterministic solar power forecasting, *Energy Convers. Manage.*, **214** (2020), 112909. https://doi.org/10.1016/j.enconman.2020.112909
9. A. Dolara, S. Leva, G. Manzolini, Comparison of different physical models for PV power output prediction, *Sol. Energy*, **119** (2015), 83–99. https://doi.org/10.1016/j.solener.2015.06.017
10. D. Koster, F. Minette, C. Braun, O. Oliver, Short-term and regionalized photovoltaic power forecasting, enhanced by reference systems, on the example of Luxembourg, *Renew. Energy*, **132** (2019), 455–470. https://doi.org/10.1016/j.renene.2018.08.005
11. F. Wang, Z. Zhen, C. Liu, Z. Mi, B. Hodge, M. Shafie-Khah, et al., Image phase shift invariance based cloud motion displacement vector calculation method for ultra-short-term solar PV power forecasting, *Energy Convers. Manage.*, **157** (2018), 123–135. https://doi.org/10.1016/j.enconman.2017.11.080
12. H. Sharadga, S. Hajimirza, R. S. Balog, Time series forecasting of solar power generation for large-scale photovoltaic plants, *Renew. Energy*, **150** (2020), 797–807. https://doi.org/10.1016/j.renene.2019.12.131
13. J. Müller, E. Trutnevyte, Spatial projections of solar PV installations at subnational level: accuracy testing of regression models, *Appl. Energy*, **265** (2020), 114747. https://doi.org/10.1016/j.apenergy.2020.114747
14. J. Boland, M. David, P. Lauret, Short term solar radiation forecasting: island versus continental sites, *Energy*, **113** (2016), 186–192. https://doi.org/10.1016/j.energy.2016.06.139
15. M. Bouzerdoum, A. Mellit, A. Massi Pavan, A hybrid model (SARIMA–SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant, *Sol. Energy*, **98** (2013), 226–235. https://doi.org/10.1016/j.solener.2013.10.002
16. X. Luo, D. Zhang, X. Zhu, Deep learning based forecasting of photovoltaic power generation by incorporating domain knowledge, *Energy*, **225** (2021), 120240. https://doi.org/10.1016/j.energy.2021.120240
17. J. Liu, W. Fang, X. Zhang, C. Yang, An improved photovoltaic power forecasting model with the assistance of aerosol index data, *IEEE T. Sustain. Energy*, **6** (2015), 434–442. https://doi.org/10.1109/TSTE.2014.2381224
18. M. Pan, C. Li, R. Gao, Y. Huang, H. You, T. Gu, et al., Photovoltaic power forecasting based on a support vector machine with improved ant colony optimization, *J. Clean. Prod.*, **277** (2020), 123948. https://doi.org/10.1016/j.jclepro.2020.123948
19. P. Tang, D. Chen, Y. Hou, Entropy method combined with extreme learning machine method for the short-term photovoltaic power generation forecasting, *Chaos Solitons Fractals*, **89** (2016), 243–248. https://doi.org/10.1016/j.chaos.2015.11.008
20. Y. Ju, J. Li, G. Sun, Ultra-short-term photovoltaic power prediction based on self-attention mechanism and multi-task learning, *IEEE Access*, **8** (2020), 44821–44829. https://doi.org/10.1109/ACCESS.2020.2978635
21. X. Qing, Y. Niu, Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM, *Energy*, **148** (2018), 461–468. https://doi.org/10.1016/j.energy.2018.01.177
22. X. Guo, Y. Mo, K. Yan, Short-term photovoltaic power forecasting based on historical information and deep learning methods, *Sensors*, **22** (2022). https://doi.org/10.3390/s22249630

23. Y. He, Q. Gao, Y. Jin, F. Liu, Short-term photovoltaic power forecasting method based on convolutional neural network, *Energy Rep.*, **8** (2022), 54–62. https://doi.org/10.1016/j.egyr.2022.10.071
24. S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, preprint, arXiv: 1803.01271.
25. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
26. A. Ajami, M. Daneshvar, Data driven approach for fault detection and diagnosis of turbine in thermal power plant using independent component analysis (ICA), *Int. J. Electr. Power Energy Syst.*, **43** (2012), 728–735. https://doi.org/10.1016/j.ijepes.2012.06.022
27. J. Li, F. Guo, A. Sivakumar, Y. Dong, R. Krishnan, Transferability improvement in short-term traffic prediction using stacked LSTM network, *Trans. Res. Emerging Technol.*, **124** (2021), 102977. https://doi.org/10.1016/j.trc.2021.102977
28. Z. Fazlipour, E. Mashhour, M. Joorabian, A deep model for short-term load forecasting applying a stacked autoencoder based on LSTM supported by a multi-stage attention mechanism, *Appl. Energy*, **327** (2022), 120063. https://doi.org/10.1016/j.apenergy.2022.120063