

A Stacked GRU-RNN-based Approach for Predicting Renewable Energy and Electricity Load for Smart Grid Operation

Abstract—Predictions of renewable energy (RE) generation and electricity load are critical to smart grid operation. However, the prediction task remains challenging due to the intermittent and chaotic character of RE sources, and the diverse user behavior and power consumers. This paper presents a novel method for the prediction of RE generation and electricity load using improved stacked gated recurrent unit-recurrent neural network (GRU-RNN) for both uni-variate and multi-variate scenarios. First, multiple sensitive monitoring parameters or historical electricity consumption data are selected according to the correlation analysis to form the input data. Second, a stacked GRU-RNN using a simplified GRU is constructed with improved training algorithm based on AdaGrad and adjustable momentum. The modified GRU-RNN structure and improved training method enhance training efficiency and robustness. Third, the stacked GRU-RNN is used to establish an accurate mapping between the selected variables and RE generation or electricity load due to its self-feedback connections and improved training mechanism. The proposed method is verified by using two experiments: prediction of wind power generation using multiple weather parameters and prediction of electricity load with historical energy consumption data. The experimental results demonstrate that the proposed method outperforms state-of-the-art methods of machine learning or deep learning in achieving an accurate energy prediction for effective smart grid operation.

Index Terms—Stacked GRU-RNN, Renewable energy prediction, Electricity load prediction, Smart grid.

I. INTRODUCTION

With the integration of the latest information and communication technology (ICT), the traditional grid is transforming rapidly into a Smart Grid (SG). SG is aiming to provide the consumers with a reliable, economical, sustainable, secure, and efficient energy supply by monitoring, protecting, and optimizing the production, distribution, and consumption of electric energy [1]. The predictions of both energy production and the electricity load play a vital role in a balanced and secure grid scheduling and operation [2]. However, accurate prediction of energy production and load is still a challenging task. On one side, the prediction of energy production faces new challenges due to the large-scale integration of renewable energy (RE) sources such as solar and wind energy [3]. Power generation from renewable sources has continuously grown over the past decade because of the evident benefits of RE to the sustainability of energy and environment. In 2019, the cumulative capacities of wind energy and solar energy in Europe had reached up to 205GW and 131.9GW respectively, which represented about 18% of the electricity the EU-28 consumed [4], [5]. The shares of renewables in the net electricity generation are over 46% and 49% in Germany and Spain, the two leading countries in RE usage in EU. However, the intermittent and random nature of RE sources brings

significant uncertainties in energy production [6]. On the other side, the increased diversity of the behavior of the users and the consumers connected to smart devices, electric vehicles and so on, has increased the fluctuation of the electricity load [7].

In the past decade, extensive studies have been conducted on the prediction of RE production and electricity load [8]. For various energy management purposes, prediction with different forecasting times horizons are considered such as short-term, medium-term, and long-term, where short-term forecasting is used for scheduling energy flow among power sources, loads and storage devices, while medium-term and long-term forecastings are mainly responsible for price settlement, load dispatch and maintenance scheduling, respectively [9]. With the increased amount of monitoring data from the SG and the advanced data mining technologies, data-driven approaches in predicting energy production and consumption have gained considerable attention [10]. Typical input data include selected weather forecast data and historical energy usage data. Data-driven energy prediction methods can be classified into two categories: statistical methods and machine learning (ML) methods [11]. The statistical methods aim to build mathematical relations between input and output data. Various statistical approaches have been investigated including autoregressive moving average [12], Kalman filter [13], and Markov chain model [14]. Compared with statistical methods, ML methods have produced more promising results due to the stronger ability in mapping the complex nonlinear input-output relation [15]. Typical ML methods include the support vector machine, random forest, and fuzzy logic [16].

More recently, deep learning, with its superior capability in discovering the inherent nonlinear features instead of using handcraft features, has achieved remarkable performance in many prediction applications including energy production and load forecasting [17]. Torres et al. proposed a deep feed-forward neural network-based approach for solar power forecasting using data from multiple sources (solar power and weather forecast data) [18]. Hong et al. developed a hybrid deep learning-based method for wind power prediction where the convolutional neural network was used to extract features of volatile wind power time-series [19]. However, the temporal correlation of the time series data was not considered by using either the fully connected neural networks or convolutional neural network (CNN). With its unique self-feedback connections, recurrent neural network (RNN) has outperformed the regular DNN and CNN in time-series prediction [20]. However, RNN has the problem of gradient vanishing, which prevents it from learning long-term dependency [21]. As an enhanced variant of the classical RNN, long-short term memory-based RNN (LSTM-RNN) is constructed to address the gradient vanishing problem, and it has been applied in energy prediction [22]. To further improve modeling

performance based on LSTM-RNN, gated recurrent unit-based RNN (GRU-RNN) has been developed since 2014. However, the computational cost of GRU-RNN can still be high for long sequential data [23]. Further investigation in reducing model parameters and improving training efficiency is needed to develop accurate and efficient methods for the prediction of renewable energy generation and electricity load with increased uncertainties from both supply and demand.

The present paper proposes a novel prediction method for RE production and electricity load based on an improved stacked GRU-RNN with reduced model parameters and a new training mechanism. The experimental results on wind energy and electricity load predictions confirm the superior prediction performance of the proposed method compared with existing machine learning and deep learning methods. The main contributions of the paper are as follows:

- 1) A new framework for both the prediction of renewable energy and electricity load is proposed by using improved stacked GRU-RNN. The developed approach can perform accurate energy prediction using time-series data with monitoring parameters.
- 2) Modified GRU-RNN structure is used with fewer model parameters. The improved GRU-RNN reduces the model complexity which saves the computational cost and requires less training data.
- 3) The training algorithm of the stacked GRU-RNN is improved by adding a novel adjustable momentum scheme. The adaptive learning rate used in the proposed method results in a more effective training process.
- 4) The proposed method can deal with both multi-variate and uni-variate cases. The effectiveness of the method is demonstrated by conducting two experiments including multi-variate wind energy prediction and uni-variate electricity load prediction.

The rest of the paper is organized as follows. Section II briefly introduces classical LSTM-RNN. The proposed method is given in Section III. Section IV presents two cases using two open-access datasets consists of wind power generation in Germany and electricity load in Spain. The conclusions and possible future work are given in Section V.

II. INTRODUCTION TO LSTM-RNN

By incorporating neurons with self-feedback connections, RNN achieves the unique capability of modeling time-series data, when compared with other types of deep learning models such as CNN, deep belief network (DBN), and stacked autoencoder (SAE). LSTM-RNN was developed to address the gradient vanishing problem when finding long-term temporal correlations in time series. As shown in **Fig. 1**, LSTM-RNN is constructed by using LSTM units to replace the traditional hidden neurons. Each LSTM unit is composed of an input gate (\mathbf{i}_t), a forget gate (\mathbf{f}_t), an output gate (\mathbf{o}_t), and a memory cell (\mathbf{c}_t). The operations in LSTM-RNN can be expressed by

$$\mathbf{g}_t = \sigma(\mathbf{W}_{ih} \cdot \mathbf{x}_t + \mathbf{W}_{hh} \cdot \mathbf{h}_{t-1} + \mathbf{b}_h) \quad (1)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ii} \cdot \mathbf{x}_t + \mathbf{W}_{hi} \cdot \mathbf{h}_{t-1} + \mathbf{p}_i \odot \mathbf{c}_{t-1} + \mathbf{b}_i) \quad (2)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{if} \cdot \mathbf{x}_t + \mathbf{W}_{hf} \cdot \mathbf{h}_{t-1} + \mathbf{p}_f \odot \mathbf{c}_{t-1} + \mathbf{b}_f) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{io} \cdot \mathbf{x}_t + \mathbf{W}_{ho} \cdot \mathbf{h}_{t-1} + \mathbf{p}_o \odot \mathbf{c}_t + \mathbf{b}_o) \quad (4)$$

$$\mathbf{c}_t = \mathbf{i}_t \odot \mathbf{g}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1} \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \varphi(\mathbf{c}_t) \quad (6)$$

$$\sigma(t) = 1/(1 + e^{-t}) \quad (7)$$

$$\varphi(t) = (e^t - e^{-t}) / (e^t + e^{-t}) \quad (8)$$

where \mathbf{x}_t is the input sample at time t ; \mathbf{h}_t is the hidden output at time t ; \mathbf{h}_{t-1} is the hidden output at the previous instant; σ is the activation functions of the three gates; φ represents the activation functions of the input and output layers; \odot denotes the dot product operation; \mathbf{W}_{ih} , \mathbf{W}_{ii} , \mathbf{W}_{if} , \mathbf{W}_{io} denote the weights that connect the input layer and the LSTM unit; \mathbf{W}_{hh} , \mathbf{W}_{hi} , \mathbf{W}_{hf} , \mathbf{W}_{ho} denote the self-connection weights between the current time t and the previous time $t-1$; \mathbf{b}_h is the bias of the input layer, \mathbf{b}_i , \mathbf{b}_f , \mathbf{b}_o are the biases of the three gates in the LSTM unit; and \mathbf{p}_i , \mathbf{p}_f , \mathbf{p}_o are the weights connecting the peepholes and the three gates. Through training, these weights and biases are optimized by the stochastic gradient descent (SGD) algorithm to minimize the mean square error (MSE) between the actual and predicted output values.

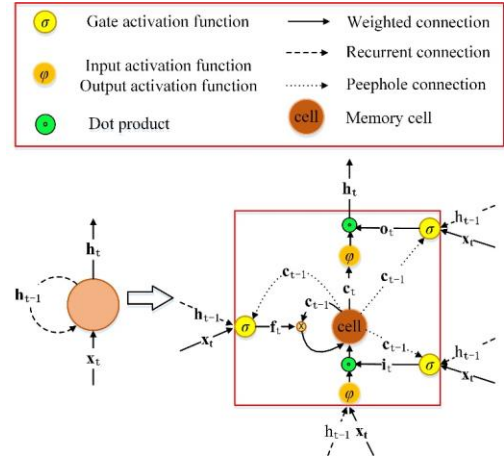


Fig. 1. The model architecture of LSTM-RNN.

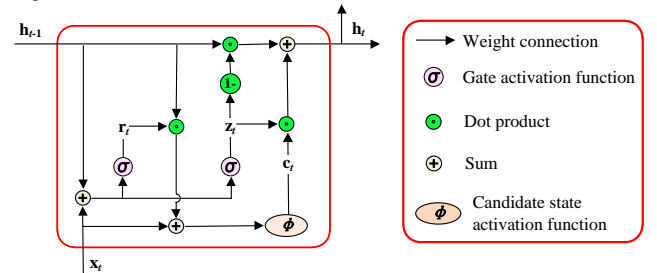


Fig. 2. The model architecture of a classic GRU-RNN.

III. THE PROPOSED METHOD

A. Stacked Novel GRU-RNN

Fig. 2 shows the model architecture of a classic GRU, which is formed by only two gates called reset gate and update gate. The reset gate (\mathbf{r}_t) is used to decide whether all or part of \mathbf{h}_{t-1} (previous) is taken into consideration. The purpose of the update gate (\mathbf{z}_t) is to decide how much of \mathbf{h}_t (current) is updated based on the state of the candidate layer (\mathbf{c}_t). For a time-series sample set $\{\mathbf{x}_t\}$, the calculation procedures of GRU are

$$\mathbf{z}_t = \sigma(\mathbf{V}_{xz} \mathbf{x}_t + \mathbf{U}_{hz} \mathbf{h}_{t-1} + \mathbf{b}_z) \quad (9)$$

$$\mathbf{r}_t = \sigma(\mathbf{V}_{xr} \mathbf{x}_t + \mathbf{U}_{hr} \mathbf{h}_{t-1} + \mathbf{b}_r) \quad (10)$$

$$\mathbf{c}_t = \varphi(\mathbf{V}_{xc} \mathbf{x}_t + \mathbf{U}_{hc} (\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_c) \quad (11)$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \mathbf{c}_t \quad (12)$$

where \mathbf{V}_{xz} , \mathbf{V}_{xr} , and \mathbf{V}_{xc} are the weights connecting the input layer and the update gate, reset gate and candidate layer, respectively; \mathbf{U}_{hz} , \mathbf{U}_{hr} , and \mathbf{U}_{hc} are the self-connection weights between the current time t and the previous time $t-1$; \mathbf{b}_z , \mathbf{b}_r , and \mathbf{b}_c are the biases of the update gate, reset gate, and the candidate layer in the GRU unit, respectively.

Training an effective GRU-RNN with long sequential data is still time-consuming. To tackle this problem, we utilize a gate-variant of the classic GRU-RNN. Specifically, in our developed GRU-RNN, each gate is computed using only the previous hidden state and the bias, which largely reduces the total number of the trained parameters. The gate-variant of the proposed GRU-RNN can be expressed by:

$$\mathbf{z}_t = \sigma(\mathbf{U}_{hz} \mathbf{h}_{t-1} + \mathbf{b}_z) \quad (13)$$

$$\mathbf{r}_t = \sigma(\mathbf{U}_{hr} \mathbf{h}_{t-1} + \mathbf{b}_r) \quad (14)$$

In practice, the time-series signals with multiple parameters often show highly nonlinear and non-stationary characteristics. Therefore, to perform accurate and reliable mapping, multiple GRU-RNN layers could be stacked to construct a stacked GRU-RNN. As shown in Fig. 3, by introducing a regression layer at the top, the stacked GRU-RNN can be used for prediction, defined as:

$$\hat{\mathbf{y}}_t = \sigma(\mathbf{U}_{yh} \mathbf{h}_t + \mathbf{b}_y) \quad (15)$$

where $\hat{\mathbf{y}}_t$ represents the predicted output after the regression layer; \mathbf{U}_{yh} represents the weight for the regression layer; \mathbf{b}_y represents the bias for the regression layer; and the regression layer also employs σ as the activation function.

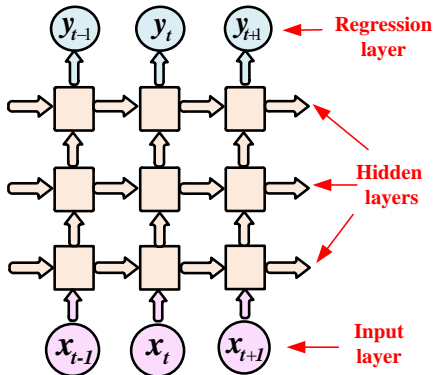


Fig. 3. Stacked GRU-RNN for prediction using time series.

B. Modification of the training algorithm

Like the classical RNN and LSTM-RNN, the GRU-RNN typically uses the SGD algorithm with momentum to adjust the weights and biases by minimizing the MSE between the actual outputs and predicted outputs, expressed as

$$E = \sum_{t=1}^T (\mathbf{y}_t - \hat{\mathbf{y}}_t)^2 / 2T \quad (16)$$

$$\boldsymbol{\theta}_{q+1} = \boldsymbol{\theta}_q - \eta \cdot (\partial E / \partial \boldsymbol{\theta}_q) + \beta \cdot (\boldsymbol{\theta}_q - \boldsymbol{\theta}_{q-1}) \quad (17)$$

where E is the MSE function, $\boldsymbol{\theta}_q$ represents the parameter set at the q th iteration consisting of different weights and biases given in Eqs.(11-15); \mathbf{y}_t denotes the actual output; T is the total number of training samples; and η is the learning rate. In the basic training algorithm, the learning rate and momentum both stay the same during the training process, which is inadequate to effectively adjust all the weights and biases of the GRU-RNN. Despite AdaGrad strategy improves the training performance to some degree by adjusting the learning rate in real-time, the momentum still stays the same, which may still lead to slow convergence and overfitting.

In the present paper, to overcome the limitation of AdaGrad and further enhance the training performance, a new modified training algorithm using AdaGrad and adjustable momentum is developed to enable an adaptive learning rate and momentum at different gradients during the training process. The adaptive learning principle is expressed as follows:

$$\boldsymbol{\theta}_{q+1} = \boldsymbol{\theta}_q - \eta_q \cdot (\partial E_q / \partial \boldsymbol{\theta}_q) + \beta_q \cdot (\boldsymbol{\theta}_q - \boldsymbol{\theta}_{q-1}) \quad (18)$$

$$\eta_q = \eta_0 / \sqrt{\sum_{i=1}^q g_i^2 + \lambda} \quad (19)$$

$$g_q = \partial E_q / \partial \boldsymbol{\theta}_q \quad (20)$$

$$\beta_q = e^{-\kappa \cdot \|g_q\|} \quad (21)$$

where Eq.(19) represents the adjustable momentum; E_q , η_q , g_q , β_q represent the MSE, learning rate, gradient, momentum factor and trained parameter set at the q th iteration, respectively; λ represents a small positive number (default 1e-8); κ is the initial coefficient of adjustable momentum term; and η_0 is the initial learning rate.

C. Flowchart of the proposed method

This paper presents a novel method for the prediction of renewable energy generation and electricity load using improved stacked GRU-RNN. The flowchart of the proposed method is shown in Fig. 4. The main steps are as follows.

Step 1: Collect the historical time-series data containing the monitoring parameters of the energy production and electricity load.

Step 2: Select the sensitive monitoring parameters according to the correlation analysis between the parameters and output.

Step 3: Divide the samples into training and testing sets.

Step 4: Build a stacked GRU-RNN model with the proposed GRU structure and the improved training algorithm. Train the model with training samples to establish an accurate mapping between the selected monitoring parameters and the output.

Step 5: Verify the prediction performance of the stacked GRU-RNN model using the testing samples.

IV. EXPERIMENTAL EVALUATION

To evaluate the effectiveness of the proposed approach for both the prediction of renewable energy and electricity load, two experiments are conducted: wind energy prediction using multiple weather parameters and electricity load prediction using historical load data.

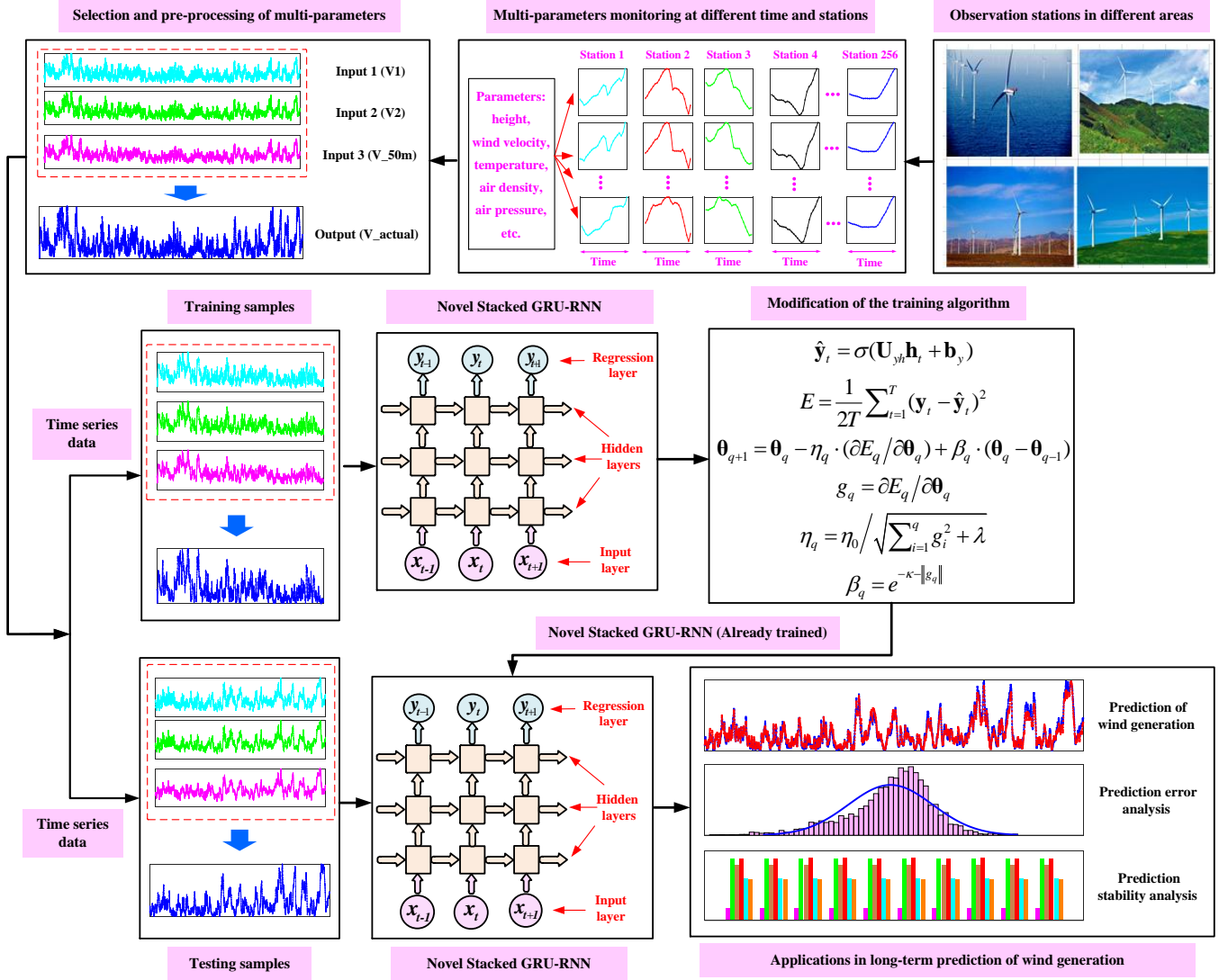


Fig. 4. The overall framework of the proposed method.

TABLE I
DESCRIPTIONS OF THE INPUT AND OUTPUT VARIABLES

Descriptions	Three-dimensional input variables			One-dimensional output variable
	V1	V2	V_50m	DE_wind_generation_actual
Physical meanings	Velocity at height h1 (2 meters above displacement height)	Velocity at height h2 (10 meters above displacement height)	Velocity at 50 meters above the ground	Actual wind generation
Units	[m/s]	[m/s]	[m/s]	[MW]
Names of files	Weather_data_GER_2016.csv	Weather_data_GER_2016.csv	Weather_data_GER_2016.csv	Solar wind energy production DE.csv
Locations in files	Column E	Column F	Column G	Column D

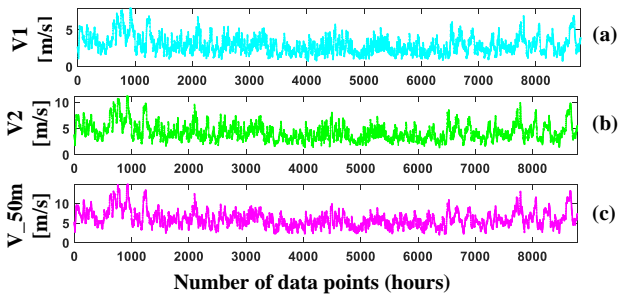


Fig. 5. The average values of the three input variables during the entire year: (a) V1; (b) V2; (c) V_50m.

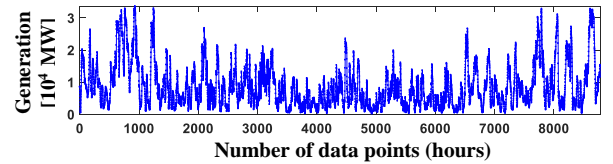


Fig. 6. The output variable for the entire year.

A. Experiment one: Wind energy prediction

1) Data description and experimental setup

In this experiment, the dataset containing the hourly actual wind energy generation for 2016 in Germany together with the hourly weather conditions at 256 locations in the country is

used [24]. The weather data contains 6144 (256*24) data points for 11 monitoring variables every day, including the height above the ground, wind velocity at different heights, temperature, air density, air pressure, and so on. In this study, the average values of the monitoring parameters from the 256 locations are calculated.

To determine the input variables for the prediction task, correlation analysis is performed. The sensitive input variables can be decided by calculating the correlation coefficients among the average values of the monitoring parameters and the actual wind generation, defined as

$$r = \frac{N \sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)}{\sqrt{N \sum_i (x_i)^2 - (\sum_i x_i)^2} \sqrt{N \sum_i (y_i)^2 - (\sum_i y_i)^2}} \quad (22)$$

where x_i and y_i represent the average values of a monitoring parameter and the actual wind generation values, respectively; and N is the total data number. The correlation coefficients between V1, V2, V_50m, and the actual wind generation values are 0.8773, 0.9131, and 0.9503, respectively, which are larger compared with the remaining eight monitoring parameters. Thus, V1, V2, and V_50m are selected as the model inputs. More details about the three input variables are presented in **Table I**. The average values of the three input variables during the entire year (from 01/01/2016 to 12/31/2016) are shown in **Fig. 5**. **Fig. 6** shows the actual wind generation of the whole year, where the horizontal axis indicated the total data point number 8784 (24*366).

As shown in **Fig.5** and **Fig.6**, the three input variables and the one output variable are all highly nonlinear and complex non-stationary signals, all containing 8784 time-series data points, denoted as $\{x_t^{(1)}\}_{t=1}^{8784}$, $\{x_t^{(2)}\}_{t=1}^{8784}$, $\{x_t^{(3)}\}_{t=1}^{8784}$ and $\{y_t\}_{t=1}^{8784}$, respectively. Here, the prediction task is to establish the nonlinear mapping between the three input variables and the one output variable. Each sample is set as $\{\mathbf{x}_t\} = \{x_t^{(1)}, x_t^{(2)}, x_t^{(3)}\} \rightarrow \{y_t\}$. All the data are normalized to the scale of [0, 1] before feeding into the model.

Here, two indices are selected to quantitatively and comprehensively evaluate the prediction performance: root mean square error (RMSE) and mean absolute error (MAE). The definition of the two indices are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{T} \sum_{i=1}^T (y_i - \hat{y}_i)^2} \quad (23)$$

$$\text{MAE} = \frac{1}{T} \sum_{i=1}^T |y_i - \hat{y}_i| \quad (24)$$

where y_i is the actual wind generation value (Normalized form), \hat{y}_i is its predicted value, and T is the total number of the predicted values (the testing samples). In this experiment, the total numbers of the training and testing samples are 5000 and 3784, respectively.

2) Comparisons with shallow models, SAE, and DBN

In this part, the proposed method is compared with least squares support vector regression (LSSVR), extreme learning machine (ELM) and wavelet neural network (WNN), and two

types of state-of-the-art deep learning models (improved SAE and improved DBN) proposed in recently published papers [25, 26], to demonstrate its feasibility and superiority in wind energy prediction. To construct the stacked GRU-RNN, the size of the input and output layers are selected directly based on the dimensions of the input and output data, which is 3 and 1 here. The number of hidden layers N_H and the number of neurons in each layer N_L are determined by grid search where $N_H \in \{2,3,4,5\}$ and $N_L \in \{5,10,15,20\}$. Model with the structure of "3-15-10-1" achieves the best result. The initial learning rate, initial coefficient of adjustable momentum term, and iteration number are 0.0009, 0.01, and 150, respectively. For the comparative methods, their hyperparameters are also decided by grid search with predetermined ranges. The hyperparameter selection results of the different models are as follows. In LSSVR, radial basis function kernel is used, the regularization parameter and kernel parameter are 1 and 5, respectively. In ELM, sigmoid is the activation function of the hidden layer with 12 neurons. In WNN, Morlet wavlet is used as the activation function with 7 hidden neurons, learning rate, momentum and iteration number are 0.01, 0.95, and 200, respectively. In improved SAE, the architecture is "3-10-7-1" with sigmoid as the activation function, sparsity penalty factor, sparsity coefficient, weight decay coefficient, learning rate, momentum and iteration number are 5, 0.06, 0.003, 0.001, 0.90, 100, respectively. In improved DBN, the structure is selected as "3-12-9-1", learning rate, momentum and iteration number are 0.1, 0.9 and 100, respectively.

Fig. 7 shows the average predicted results (Normalized form) of the testing samples using different methods. To ensure the stability of the predicted results, all the methods are repeatedly tested ten times. Furthermore, the histograms with normal distribution fits of the average predicted errors given by different methods are compared in **Fig. 8**, in which the horizontal axis represents the average predicted errors, and the vertical axis represents the frequency number (total frequency number is 3784). From **Fig. 8**, the average predicted errors given by the proposed method are closer to the normal distribution when compared with the other five methods. The statistical information shown in **Fig. 8** is listed in **Table II**. According to **Table II**, the numbers of the predicted errors within ± 0.02 , ± 0.05 and ± 0.10 are 1326, 2715, and 3524, respectively, which are all larger than the other methods.

Table III records the two evaluation indices and run time of different methods (Intel Core i7-8550U CPU, 16 GB Memory, Matlab R2016b). According to Table III, the RMSE and MAE given by the proposed method (Method 1) are 0.0526 and 0.0393, respectively, which are both smaller than the other five methods (Methods 2-6). However, the average run time of the proposed method is about 32.54s, which is more than the other five methods (10.51s, 16.28s, 2.04s, 27.57s, and 29.89s).

Through analysis and comparison of the test results, the proposed method is more effective and robust than the other five methods for wind energy prediction with multiple monitoring parameters. The main reason for that is the proposed method is based on the improved GRU-RNN, which is suitable and effective in modeling time-series data due to the presence of self-feedback connections and the improved training mechanism.

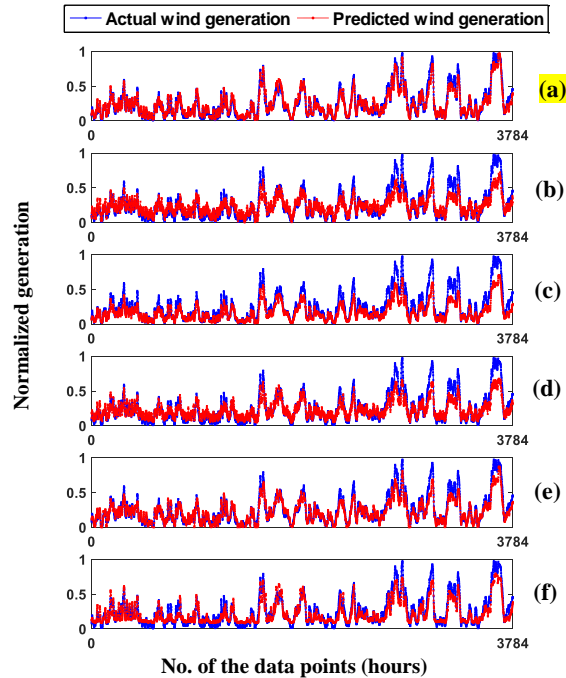


Fig. 7. Average predicted results (Normalized form) of ten trials: (a) Method 1; (b) Method 2; (c) Method 3; (d) Method 4; (e) Method 5; (f) Method 6.

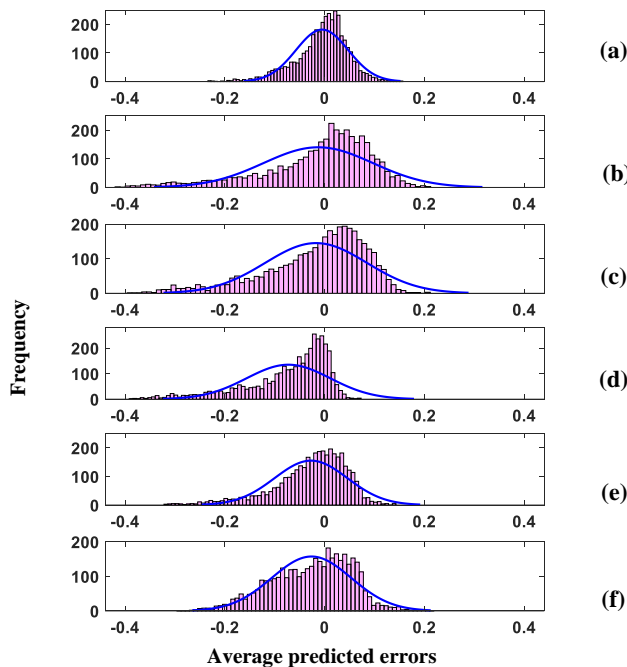


Fig. 8. Distributions of the average predicted errors of ten trials: (a) Method 1; (b) Method 2; (c) Method 3; (d) Method 4; (e) Method 5; (f) Method 6.

TABLE II
STATISTICAL ERROR INFORMATION IN COMPARATIVE EXPERIMENT 1

Prognosis methods	Range of predicted errors		
	± 0.02	± 0.05	± 0.10
Method 1 (Proposed method)	1326	2715	3524
Method 2 (LSSVR)	674	1591	2715
Method 3 (WNN)	664	1588	2784
Method 4 (ELM)	1080	1942	2750
Method 5 (SAE proposed in [25])	976	2236	3225
Method 6 (DBN proposed in [26])	741	1723	2938

TABLE III

EVALUATION INDICES AND COMPUTING TIME IN COMPARATIVE EXPERIMENT 1

Prognosis methods	RMSE	MAE	Run time (s)
Method 1 (Proposed method)	0.0526	0.0393	32.54
Method 2 (LSSVR)	0.1104	0.0816	10.51
Method 3 (WNN)	0.1033	0.0616	16.28
Method 4 (ELM)	0.1109	0.0767	2.04
Method 5 (SAE proposed in [25])	0.0779	0.0561	27.57
Method 6 (DBN proposed in [26])	0.0839	0.0662	29.89

3) Comparisons with basic and state-of-art stacked RNNs

In this part, another three state-of-the-art methods based on RNNs are compared, including Method 7 (hierarchical GRU published in [27]), Method 8 (SAE-LSTM published in [28]), and Method 9 (Basic stacked RNN), with the proposed method to demonstrate the effectiveness.

As before, all the methods are repeatedly tested ten times. The average predicted results on the testing samples are shown in Fig. 9. Fig. 10 plots the error histograms with normal distribution fits of all the methods. The errors produced by the proposed method are found to be closer to the normal distribution when compared with the other types of RNNs. The statistical information of the prediction errors is presented in Table IV. The numbers of samples for predicted errors within ± 0.02 and ± 0.04 provided by the comparative methods (Methods 7-9) are all smaller than the proposed method. Table V lists the evaluation indices where the proposed method achieves the best results. Besides, the run time of the proposed method is less compared with the other three methods (44.79s, 47.82s, and 53.37s). Based on Comparative Experiments 1 and 2, it can be found that the proposed method achieved much higher prediction accuracy than shallow models. Compared with state-of-art RNN models, it needs significantly less run time and has slightly higher and more robust prediction performance.

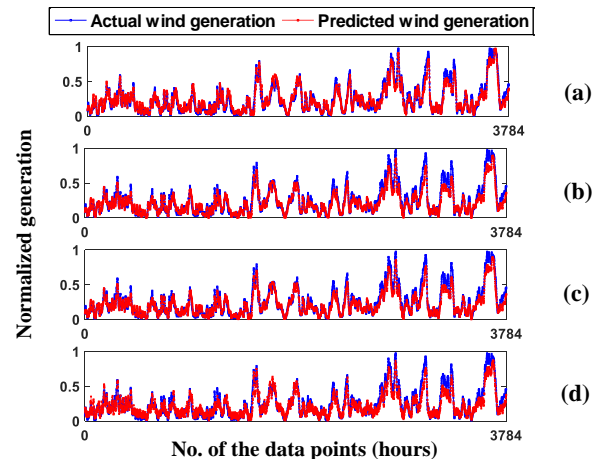
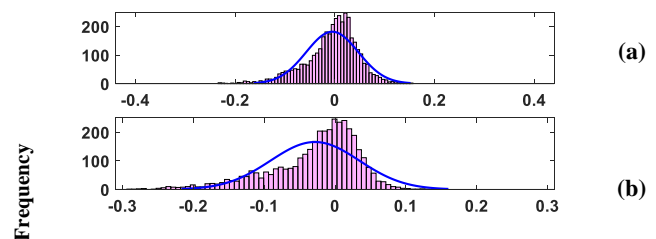


Fig. 9. Average predicted results (Normalized form) of ten trials: (a) Method 1; (b) Method 7; (c) Method 8; (d) Method 9.



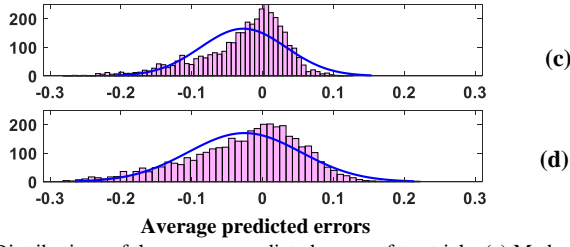


Fig. 10. Distributions of the average predicted errors of ten trials: (a) Method 1; (b) Method 7; (c) Method 8; (d) Method 9.

TABLE IV

STATISTICAL ERROR INFORMATION IN COMPARATIVE EXPERIMENT 2		
Prognosis methods	Range of predicted errors	
	± 0.02	± 0.04
Method 1 (Proposed method)	1326	2370
Method 7 (Hierarchical GRU proposed in [27])	1305	2253
Method 8 (SAE-LSTM proposed in [28])	1330	2271
Method 9 (Basic stacked RNN)	847	1548

TABLE V

EVALUATION INDICES AND COMPUTING TIME IN COMPARATIVE EXPERIMENT 2

Prognosis methods	RMSE	MAE	Run time (s)
Method 1 (Proposed method)	0.0526	0.0393	32.54
Method 7 (Hierarchical GRU in [27])	0.0689	0.0482	44.79
Method 8 (SAE-LSTM in [28])	0.0662	0.0467	47.82
Method 9 (Basic stacked RNN)	0.0838	0.0637	53.37

B. Experiment two: Electricity load prediction

In this experiment, the proposed novel stacked GRU-RNN is used for the short-term prediction of the electricity load. The data used here is the hourly load data from 01/01/2015 to 12/31/2018 (a total of 1461 days with missing data on a few days) in Spain [29]. Here, to verify both the prediction accuracy and model generalization, we selected recorded data of two periods with different lengths and cover different seasons. One is from day 649 to day 748 (100 days, from Oct. 2016 to Jan. 2016), and the other is from day 749th to day 948th (200 days, from Jan. 2017 to Jul. 2017). In the first period, the first 99 days ($24 \times 99 = 2376$ data points) are used for training the proposed method, and the rest 1 day (24 data points) used for the prediction, as shown in Fig. 11. In the second period, the first 197 days are used for training and the remaining 3 days (72 data points) are used for prediction, as shown in Fig. 12.

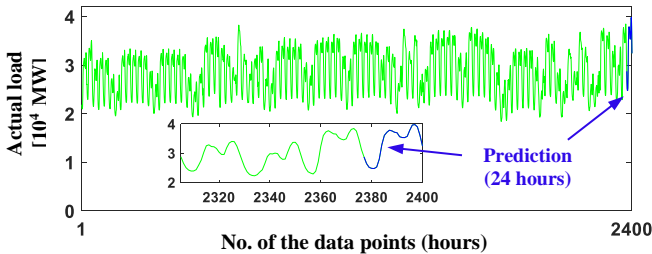


Fig. 11. The actual electric power loads during the 649th day to the 748th day.

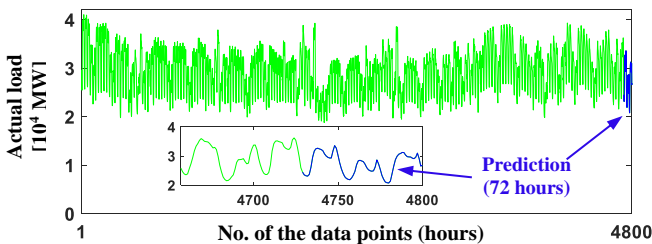


Fig. 12. The actual electric power loads during the 749th day to the 948th day.

Considering the correlation between the same hours on different days, and the weekly energy consumption pattern, 24 stacked GRU-RNNs are used to learn the relationship within the same hours, with the regression step set as 7. Construction of data sample for the two periods are the same. Take the first period as an example, the specific format of the data sample is $\{x_t^{(H)}, x_{t+1}^{(H)}, \dots, x_{t+7}^{(H)}\} \rightarrow \{x_{t+8}^{(H)}\}$, where H is the hour index, i.e., 1, 2, ..., 24; and t is the index of the time-series. The first training sample is $\{x_1^{(1)}, x_2^{(1)}, \dots, x_7^{(1)}\} \rightarrow \{x_8^{(1)}\}$; the last training sample is $\{x_{92}^{(1)}, x_{93}^{(1)}, \dots, x_{98}^{(1)}\} \rightarrow \{x_{99}^{(1)}\}$; and the one testing sample is $\{x_{93}^{(1)}, x_{94}^{(1)}, \dots, x_{99}^{(1)}\} \rightarrow \{\hat{x}_{100}^{(1)}\}$, where $\hat{x}_{100}^{(1)}$ is the prediction value for 01:00. Accordingly, the short-term predictions for the other 23 hours of the day can be obtained..

Fig. 13 and Fig. 14 plot the prediction results of the first and second test periods using the proposed methods and the other four deep learning methods, respectively. The RMSE and MAE of the proposed method are both significantly smaller than those of the other four methods, as listed in Table VI. The comparison results confirm the superiority of the proposed method for short-term prediction of electricity load. The proposed method can make accurate predictions for different lengths of period and with different seasons, although more training samples (longer periods) could slightly improve the prediction accuracy.

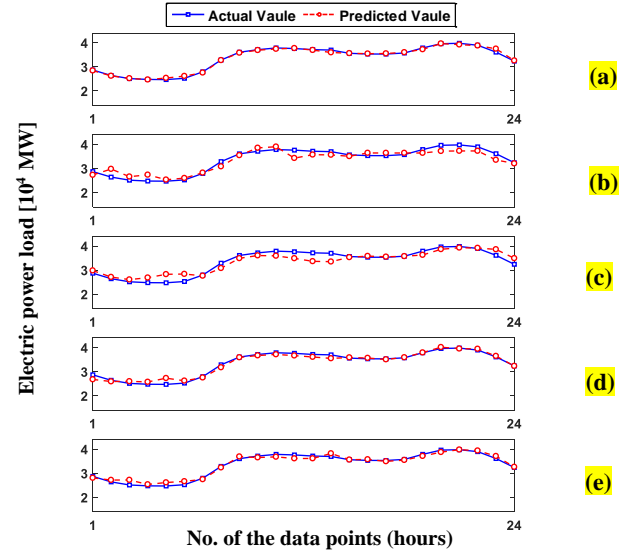
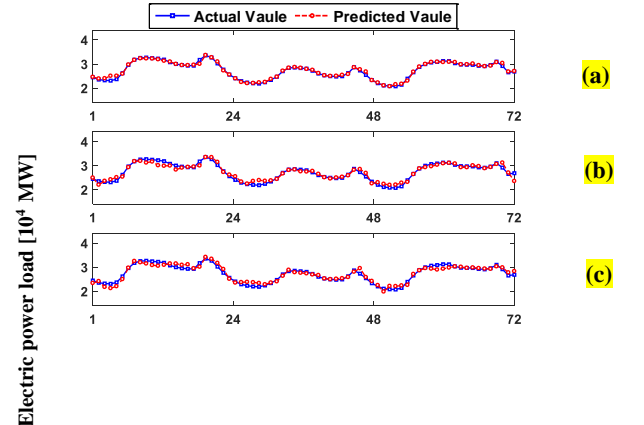


Fig. 13. Average predicted results (Original form) from the 649th day to 748th day: (a) Method 1; (b) Method 5; (c) Method 6; (d) Method 7; (e) Method 8.



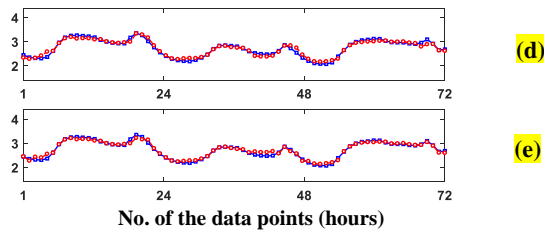


Fig. 14. Average predicted results (Original form) from the 749th day to 948th day: (a) Method 1; (b) Method 5; (c) Method 6; (d) Method 7; (e) Method 8.

TABLE VI

PREDICTION OF ELECTRICITY LOAD UNDER DIFFERENT TEST PERIODS				
Prognosis methods	From the 649th day to the 748th day (100 days)		From the 749th day to the 948th day (200 days)	
	RMSE	MAE	RMSE	MAE
Method 1 (Proposed)	507.2	351.1	484.2	327.8
Method 5 ([26])	1441.7	1206.2	992.2	768.0
Method 6 ([27])	1618.8	1324.7	1042.3	873.6
Method 7 ([28])	920.7	697.3	843.4	666.2
Method 8 ([29])	848.2	663.5	697.0	524.8

V. CONCLUSIONS

To achieve accurate predictions of renewable energy and electricity load for effective smart grid management, a novel method based on improved stacked GRU-RNN and multiple monitoring parameters was presented in this paper. Correlation analysis was first conducted, if necessary, to determine the most sensitive parameters as the model input. The improved GRU-RNN reduced the model complexity by using fewer parameters which saved the computational cost and requires less training data. In the present study, to improve the training efficiency, AdaGrad and adjustable momentum were combined to modify the training algorithm with an adaptive learning rate. The constructed modified GRU-RNN was then used to establish an accurate mapping between the monitoring parameters and the renewable energy generation or the electricity load. Both multi-variate and uni-variate cases could be catered by the developed prediction method. Experimental results on real wind energy generation and electricity load predictions demonstrated the feasibility and superiority of the proposed approach by comparing it with other state-of-art data-driven prediction methods. However, there still exist some limitations in this method. For example, some hyperparameters are determined by experience; the training time is much longer than shallow models; training samples of considerable length are needed. Possible future work includes how to improve the prediction performance by further improving the training strategy and investigating additional monitoring parameters; how to precisely model and accurately predict the energy generation from multiple renewable energy sources; and how to detect anomalies in electricity load caused by extreme cases.

REFERENCES

[1] A. Naz, M. U. Javed, N. Javaid, T. Saba, M. Alhussein and K. Aurangzeb, "Short-term electric load and price forecasting using enhanced extreme learning machine optimization in smart grids," *Energies*, vol. 12, no. 5, 2019.

[2] M. H. K. Tushar, A. W. Zeineddine and C. Assi, "Demand-Side Management by Regulating Charging and Discharging of the EV, ESS, and Utilizing Renewable Energy," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 117-126, 2018.

[3] G. Notton et al., "Intermittent and stochastic character of renewable energy sources: Consequences, cost of intermittence and benefit of forecasting," *Renew. Sust. Energ. Rev.*, vol. 87, pp. 96-105, 2018.

[4] I. Komušanac, Wind energy in Europe in 2019: Trends and statistics. 2020.

[5] A. Jäger-Waldau, I. Kougias, N. Taylor, and C. Thiel, "How photovoltaics can contribute to GHG emission reductions of 55% in the EU by 2030," *Renew. Sust. Energ. Rev.*, vol. 126, 109836, 2020.

[6] M. Aghamohamadi, A. Mahmoudi, and M. Haque, "Two-stage Robust Sizing and Operation Co-optimization for Residential PV-battery Systems Considering the Uncertainty of PV Generation and Load," *IEEE Trans. Ind. Informat.*, 2020, doi: 10.1109/TII.2020.2990682.

[7] G. Nalçaci, A. Özmen, and G. W. Weber, "Long-term load forecasting: models based on MARS, ANN and LR methods," *Cent. Eur. J. Oper. Res.*, vol. 27, no. 4, pp. 1033-1049, 2019.

[8] Z. Jianhua, "Energy management system, generation and demand predictors: a review," *IET Generation, Transmission & Distribution*, vol. 12, no. 3, pp. 519-530, 2018.

[9] J. Ma and X. Ma, "A review of forecasting algorithms and energy management strategies for microgrids," *Systems Science & Control Engineering*, vol. 6, no. 1, pp. 237-248, 2018.

[10] K. Amasyali and N. M. El-Gohary, "A review of data-driven building energy consumption prediction studies," *Renew. Sust. Energ. Rev.*, vol. 81, pp. 1192-1205, 2018.

[11] D. van der Meer, G. R. C. Mouli, G. M.-E. Mouli, L. R. Elizondo, and P. Bauer, "Energy Management System With PV Power Forecast to Optimally Charge EVs at the Workplace," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 311-320, 2018.

[12] S. Moonchai and N. Chutsagulprom, "Short-term forecasting of renewable energy consumption: Augmentation of a modified grey model with a Kalman filter," *Applied Soft Computing*, vol. 87, 105994, 2020.

[13] R. H. M. Zargar and M. H. Y. Moghaddam, "Development of a Markov-Chain-Based Solar Generation Model for Smart Microgrid Energy Management System," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 736-745, 2020.

[14] K. Y. Bae, H. S. Jang, and D. K. Sung, "Hourly Solar Irradiance Prediction Based on Support Vector Machine and Its Error Analysis," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 935-945, 2017.

[15] C. Voyant et al., "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569-582, 2017.

[16] W. Zou, C. Li, and P. Chen, "An Inter Type-2 FCR Algorithm Based T-S Fuzzy Model for Short-Term Wind Power Interval Prediction," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 4934-4943, 2019.

[17] H. Wang, Z. Lei, X. Zhang, B. Zhou and J. Peng, "A review of deep learning for renewable energy forecasting," *Energ. Convers. Manage.*, vol. 198, 111799, 2019.

[18] J. Torres, A. Troncoso, I. Koprinska, Z. Wang, and F. Martínez-Álvarez, "Big data solar power forecasting based on deep learning and multiple data sources," in *Expert Systems*, vol. 36, no. 4, p.e12394, 2019.

[19] Y.-Y. Hong and C. L. P. P. Rioflorida, "A hybrid deep learning-based neural network for 24-h ahead wind power forecasting," *Applied Energy*, vol. 250, pp. 530-539, 2019.

[20] M. Xia, X. Zheng, M. Imran, and M. Shoaib, "Data-driven prognosis method using hybrid deep recurrent neural network," *Applied Soft Computing*, vol. 93, p. 106351, 2020.

[21] A. Rahman, V. Srikumar, and A. D. Smith, "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks," *Applied Energy*, vol. 212, pp. 372-385, 2018.

[22] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841-851, 2019.

[23] R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, 2017, pp. 1597-1600.

[24] Open Power System Data, available online: <https://data.open-power-system-data.org> (Accessed: 2020-06-02)

[25] C. Sun, M. Ma, Z. Zhao, S. Tian, R. Yan and X. Chen, "Deep Transfer Learning Based on Sparse Autoencoder for Remaining Useful Life Prediction of Tool in Manufacturing," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2416-2425, 2019.

[26] R. Jiao, K. Peng, J. Dong and C. Zhang, "Fault monitoring and remaining useful life prediction framework for multiple fault modes in prognostics," *Reliab. Eng. Syst. Safe.*, vol. 203, 107028, 2020.

- [27] X. Li, H. Jiang, X. Xiong and H. Shao, "Rolling bearing health prognosis using a modified health index based hierarchical gated recurrent unit network", *Mech. Mach. Theory*, vol. 133, pp. 229-249, 2019.
- [28] J. Guo, Z. Lao, M. Hou, C. Li and S. Zhang, "Mechanical fault time series prediction by using EFMSAE-LSTM neural network", *Measurement*, 2020, doi: <https://doi.org/10.1016/j.measurement.2020.108566>.
- [29] Hourly energy demand generation and weather, available online: <https://www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather>(Accessed: 2020-06-08).