Pattern Sequence Neural Network for Solar Power Forecasting

Yang Lin[1(✉)], Irena Koprinska[1(✉)], Mashud Rana[2], and Alicia Troncoso[3]

[1] School of Computer Science, University of Sydney, Sydney, NSW, Australia
ylin4015@uni.sydney.edu.au, irena.koprinska@sydney.edu.au
[2] Data61, CSIRO, Sydney, Australia
mdmashud.rana@data61.csiro.au
[3] Data Science and Big Data Lab, University Pablo de Olavide, Seville, Spain
atrolor@upo.es

Abstract. We propose a new approach for time series forecasting, called PSNN, which combines pattern sequences with neural networks. It is a general approach that can be used with different pattern sequence extraction algorithms. The main idea is to build a separate prediction model for each pattern sequence type. PSNN is applicable to multiple related time series. We demonstrate its effectiveness for predicting the solar power output for the next day using Australian data from three data sources - solar power, weather and weather forecast. In our case study, we show three instantiations of PSNN by employing the pattern sequence extraction algorithms PSF, PSF1 and PSF2. The results show that PSNN achieved the most accurate results.

Keywords: Solar power forecasting · Pattern sequence similarity · Neural networks

1 Introduction

The use of solar photovoltaic (PV) systems is rapidly growing due to their improved efficiency and continued cost reduction, and the advantages of solar energy. However, it is challenging to integrate large amounts of electricity produced by PV systems into the power grid and maintain a stable electricity supply. The produced solar power is highly variable as it depends on meteorological factors such as solar irradiance, ambient temperature, clouds, dust and wind. This necessitates the development of methods for accurate PV power prediction, to ensure reliable electricity supply of grid-connected PV systems.

Different approaches for PV power forecasting have been proposed, based on statistical methods such as linear regression and autoregressive moving average [1] and machine learning methods such as Neural Networks (NNs) [1,2] and support vector regression [3,4]. Recently, the application of Pattern Sequence-based Forecasting (PSF) [5] methods has been studied for solar power forecasting in [6,7], showing promising results. PSF assigns a cluster label to each day and

T. Gedeon et al. (Eds.): ICONIP 2019, CCIS 1143, pp. 727–737, 2019.
https://doi.org/10.1007/978-3-030-36802-9_77

then uses a nearest neighbour approach to find similar sequences of days to the target sequence and make a prediction for the new day. One of PSF's distinct characteristics is that it predicts all values for the next day simultaneously (e.g. all half-hourly PV values for the next day), as opposed to predicting them iteratively or building a separate prediction model for each value, as the majority of the other prediction methods.

While the standard PSF algorithm uses only one time series (the time series of interest, e.g. PV data), two PSF extensions utilizing data from multiple related time series (e.g. PV, weather and weather forecast data) have been proposed in [6] and evaluated for solar power forecasting. The results showed that both extensions PSF2 and PSF1 were more accurate than the standard PSF algorithm. However, there is still an opportunity for further improvement. To make the final prediction, the PSF algorithms simply take the average of the values of the relevant days from the matched sequences. In this paper we propose to better utilise the information from the matched sequences and build a classifier to produce the final prediction. In particular, we investigate if it is possible to combine the advantages of PSF and NNs and improve the performance. The contributions of this paper are as follows:

1. We propose a novel approach combining pattern sequences with NNs, called Pattern Sequence Neural Network (PSNN). It takes as an input a sequence of cluster labels, extracts pattern sequences of different types and builds a separate NN prediction model for each of them. It combines the efficient pattern sequence extraction and similarity matching of the PSF algorithms with the advantages of NNs for modelling complex and nonlinear relationships.
2. PSNN is a general approach that can be used with different clustering and cluster sequence extraction algorithms, and can be applied to multiple related time sequences. In our case study for solar power forecasting, we show three instantiations of the PSNN approach by employing the PSF, PSF1 and PSF2 algorithms.
3. We evaluate the performance of PSNN on a Australian dataset, which includes data from three sources (PV solar, weather and weather forecast), for two years. Our results show that PSNN was the most accurate method.

2 Data

As a case study, we consider the task of simultaneously predicting the PV power output for the next day at half-hourly intervals. Given: (1) a time series of PV power output up to day d: $PV = [PV_1, \ldots, PV_d]$, where PV_i is a vector of half-hourly PV power output for day i, (2) a time series of weather vectors for the same days: $W = [W_1, \ldots, W_d]$, where W_i is a weather vector for day i, and (3) a weather forecast vector for the next day d+1: WF_{d+1}, our goal is to forecast PV_{d+1}, the half-hourly PV power output for day d+1.

2.1 Data Sources and Feature Sets

We use PV and weather data for two years - from 1 January 2015 to 31 December 2016 (731 days). Table 1 summarizes the data sources and extracted feature sets.

Table 1. Data sources and feature sets

Data source	Feature set	Attribute information
PV data	PV$\in \Re^{731\times 20}$	Daily: half-hourly solar power between 7am and 5pm
Weather data 1	W1$\in \Re^{731\times 14}$	(1–6) Daily: min and max temperature, rainfall, sunshine hours, max wind gust and average solar irradiance; (7–14) At 9am and 3pm: temperature, relative humidity, cloudiness and wind speed
Weather data 2	W2$\in \Re^{731\times 4}$	Daily: min and max temperature, rainfall and solar irradiance. W2 is a subset of W1.
Weather forecast data	WF$\in \Re^{366\times 4}$	Daily: min and max temperature, rainfall and average solar irradiance

Solar PV Data. This data was collected from a rooftop PV plant located at the University of Queensland in Brisbane, Australia, and is available from http://www.uq.edu.au/solarenergy/.

Weather Data. The corresponding weather data was collected from the Australian Bureau of Meteorology, http://www.bom.gov.au/climate/data/, from a weather station close to the PV plant. There are three sets of weather features: W1, W2 and WF. W1 includes the full set of 14 weather features. W2 is a subset of W1 and includes only 4 features which are frequently used in weather forecasts and available from meteorological bureaus.

The weather forecast feature set WF is obtained by adding 20% Gaussian noise to the W2 data. This is done since the weather forecasts were not available retrospectively for previous years. When making predictions for the days from the test set, the WF set replaces W2 as the weather forecast for these days.

2.2 Data Preprocessing

The raw PV data was measured every 1 min and was aggregated to 30-min intervals by taking the average value of the interval. All data was normalised.

There was a small percentage of missing values (0.82% in the PV and 0.02% in the weather data) which were replaced by using a nearest neighbour method, applied firstly to the weather data and then to the PV data as in [6].

3 Pattern Sequence Forecasting Methods

3.1 PSF

PSF [5] is a forecasting method combining clustering and sequence matching. Consider the PV power time series data $PV = (PV_1, ..., PV_n)$, where PV_i is the D dimensional vector of the PV power output for day i, $PV \in \Re^{n\times D}$. PSF firstly employs the k-means algorithm to cluster all vectors PV_i from the training data into k_1 clusters and labels them as $C_1, \ldots, C_{k_1}$, see Fig. 1a.

To make a prediction for a new day d+1, PSF extracts a sequence of w consecutive days, starting from the previous day and going backwards. This sequence is defined in terms of cluster labels and called a *target sequence*. It then matches the target sequence with the previous days to extract a set of equal sequences ES, finds the post-sequence day for each of them and obtains the final prediction by averaging the PV vectors of these post-sequence days. For example, in Fig. 1a the PV power prediction for day d+1 is the average of the PV vectors for days 4 and 69.

The window size w and the number of clusters k_1 are hyperparameters of the PSF algorithm and are optimised using 12-fold cross validation.

3.2 PSF1

PSF1 [6] is an extension of PSF, utilising data from more than one source. While PSF uses a single data source for clustering and sequence matching (PV data for our case study), PSF1 uses additional data source - weather forecast for our case study. It firstly clusters the training set days based on the W2 data into k_2 clusters with labels $C_1, \ldots, C_{k_2}$, see Fig. 1b.

To make a prediction for a new day d+1, PSF1 obtains the cluster label for this day by using its weather forecast vector WF, comparing it with the cluster centroids and assigning it to the cluster of the closest centroid. It then extracts a target sequence of w consecutive days from day d+1 backwards and including d+1, matches this sequence with the previous days and finds a set of equal sequences ES. The final prediction is obtained by taking the average of the PV vectors of the last days in each ES. For example, in Fig. 1b the PV power prediction for day d+1 is the average of the PV vectors for days 4 and 68.

3.3 PSF2

PSF2 [6] is an extension of PSF1 using two additional data sources: weather data and weather forecast data. It clusters the days from the training set in two different ways: using the weather data (k_1 clusters with labels $C_1, ..., C_{k_2}$) and weather forecast data (k_2 clusters with labels $K_1, \ldots, K_{k_2}$), see Fig. 1c.

The prediction for the new day d+1 is computed using the following steps, see Fig. 1c. First, a target sequence of w consecutive days from day d backwards and including day d is extracted based on the weather data and matched to find the set of equal sequences ES. Second, the cluster label K_x for day d+1 is obtained based on the weather forecast data. Third, the cluster labels of the post-sequence days for all ES are checked and if they are not the same as K_x, these sequences are excluded from ES. The final prediction for d+1 is formed by taking the average of the post-sequence days for each ES. For example, in Fig. 1c the PV power prediction for day d+1 is the average of the PV vectors for days 4 and 69. Note that day 72 is not included in the final prediction as its cluster label is K_3, which is different than the cluster label K_2 of day d+1.

4 Pattern Sequence Neural Network

Table 2 summarizes the proposed PSNN approach. The key idea is to identify the matched sequences as in PSF, but then instead of taking the average of the relevant days as in PSF, learn the relationship between the previous and next day for the set of relevant days using an NN classifier.

Table 2. The proposed PSNN approach

Step	Description
1	Cluster the days and generate a sequence of cluster labels
2	Find pattern sequences relevant to the task
3	Generate a training set for each type of pattern sequence
4	Aggregate training sets if the number of examples is too small
5	Build a prediction model for each pattern sequence type and use it to make a prediction for the new day

4.1 Clustering and Pattern Sequence Extraction

The first step of the proposed PSNN approach involves clustering the days and generating a sequence of cluster labels, where each day is assigned a cluster label. This can be done by employing any clustering algorithm suitable for the task, using an appropriate feature vector representation for the day. For example, in our solar power case study, we use the k-means algorithm to cluster the days based on their PV vector (in the PSNN-PSF instantiation of PSNN), W2 weather data (in PSNN-PSF1) and both the W1 and W2 data (in PSNN-PSF2).

The second step involves extracting pattern sequences that are relevant for the task. This can be done in different ways, by employing different algorithms. In our case study we use the PSF, PSF1 and PSF2 algorithms.

4.2 Generating Training Sets and Building NN Prediction Models

The next steps of PSNN involve identifying the unique pattern sequence types, creating a training set and building a separate NN prediction model for each of them. As shown in Fig. 1, the NN model takes as an input the PV and W1

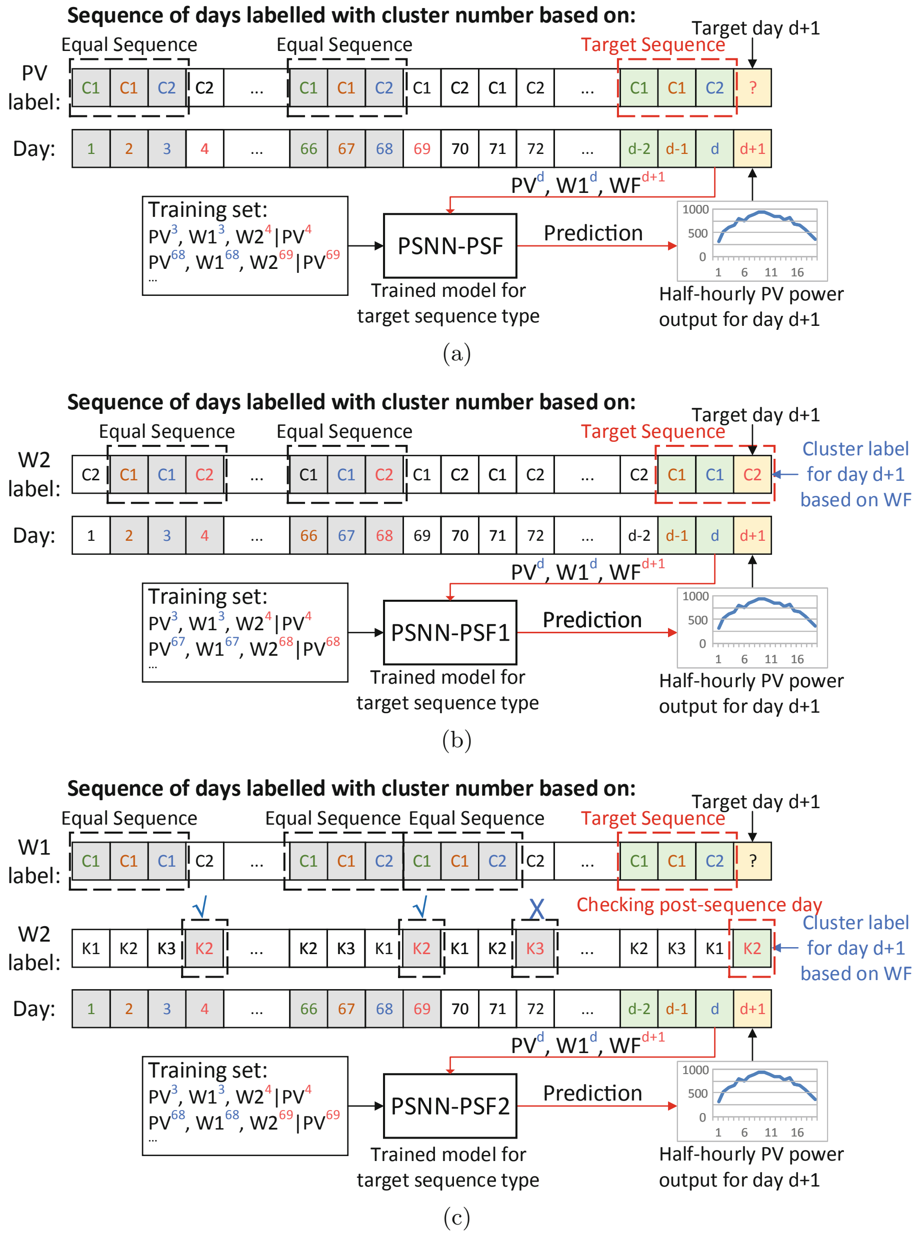

Fig. 1. The proposed PSNN approach when used in conjunction with PSF, PSF1 and PSF2 pattern sequence methods: (a) PSNN-PSF, (b) PSNN-PSF1, (c) PSNN-PSF2

features for the previous day, and the weather forecast (W2 features) for the next day, and predicts the PV data for the next day. Using information from all three data sources for the NN models was found to be beneficial in [6].

The number of unique sequences Si depends on the number of clusters and window size. For example, for PSF it is k_1^w, where k_1 is the number of clusters and w is the window size. If some pattern sequences appear less frequently, the training set for them may be too small, not allowing to build an accurate prediction model. To deal with this issue, we set a required minimum number of training examples L_{min}, expressed as a percentage of the total number of training examples, and aggregate the training sets of the pattern sequences not satisfying this condition to create a combined NN model for them.

Specifically, we apply the following aggregation rule. We identify all unique sequences Si with less than L_{min} training examples. We iteratively merge the training sets of the sequences with the same last cluster label, then the same second last cluster label and so on, until the L_{min} condition is satisfied. Thus, the aggregation rule considers the similarity between the unique pattern sequences, attempting to merge sets based on the similarity of the most recent days first.

As an example, Table 3 shows the application of the aggregation rule for PSF for $k_1 = 2$, $w = 2$ and $L_{min} = 20\%$. There are four initial sequences and two of them (S2 and S3) do not satisfy the L_{min} condition. By applying the aggregation rule, the training sets of the sequences with the same last cluster label will be aggregated, i.e. (1) S1 and S3, and (2) S2 and S4. The new training sets now satisfy the L_{min} condition, and hence two NN models will be trained - one for each of the aggregated sets.

Table 3. Aggregation of training sets - example

Initial sequences	Training set size	Aggregation rule	Final sequences
S1: C1-C1	N1: 127 (35%)	N1+N3 $\geq L_{min}\%\surd$	S1$_{new}$ = S1 or S3 = C1-C1 or C2-C1
S2: C1-C2	N2: 38 (11%) $\times$	N2+N4 $\geq L_{min}\%\surd$	S2$_{new}$ = S2 or S4 = C1-C2 or C2-C2
S3: C2-C1	N3: 38 (11%) $\times$		
S4: C2-C2	N4: 160 (44%)		

4.3 Prediction for New Days

To forecast the PV power for a new day $d + 1$, PSNN forms a target sequence depending on the pattern sequence algorithm used (PSF, PSF1 or PSF2 in our case study) and then employs the trained NN classifier for this sequence to make the prediction. For example, for PSNN-PSF1 (see Fig. 1b), the target sequence is formed by concatenating the WF cluster label of day $d+1$ with the W2 cluster labels of the previous $w-1$ days. For PSNN-PSF2 (see Fig. 1c), the target sequence is constructed by concatenating the WF cluster label of day $d+1$ with the W1 cluster labels of the previous $w-1$ days. The cluster label of $d+1$ is obtained by calculating the distance to the cluster centroids, for the clustering of the training data. Next, the pre-trained NN model for the specific target pattern sequence is used to predict the PV power for day $d+1$, taking as an input the PV and W1 vectors of day d and the WF vector of day $d+1$.

5 Experimental Setup

The PV power and corresponding weather data was split into two subsets: training and validation (the first year) and test (the second year).

Hyperparameter Tuning for PSF Models. For the PSF models and the PSF part of PSNN, the first year was used to determine the hyperparameters (number of clusters k_1 and k_2 and sequence size w) using 12-fold cross validation with grid search, consistent with the original PSF algorithm [6]. The grid search for w included values from 1 to 5. The best number of clusters was selected by varying k_1 and k_2 from 1 to 10 and evaluating three cluster quality indexes (Silhouette, Dunn and Davies-Bouldin) as described in [6].

The selected best hyperparameters were: PSF and PSF1: $k_1 = 2$, $w = 2$; PSF2: $k_1 = 2$, $k_2 = 2$, $w = 1$. The PSF parts of PSNN used the same hyperparameters as the corresponding PSF models.

Table 4. Best hyperparameters for the NN models

Model	Hidden neurons	Learning rate	L2 λ	Batch size	Epochs
NN	[25]	0.0005	0.0015	64	900
PSNN-PSF (1)	[35,25]	0.001	0.0008	64	900
PSNN-PSF (2)	[35,25]	0.001	0.001	256	500
PSNN-PSF1 (1)	[35,25]	0.003	0.0008	256	500
PSNN-PSF1 (2)	[35]	0.001	0.0015	256	900
PSNN-PSF2 (1)	[25]	0.0005	0.0008	64	500
PSNN-PSF2 (2)	[25]	0.001	0.0005	64	500
PSNN-PSF2 (3)	[30]	0.001	0.0005	256	900
PSNN-PSF2 (4)	[30]	0.0005	0.0015	64	900

Hyperparameter Tuning for NN Models. For the NN part of PSNN, the tuning of the hyperparameters was done using 5-fold cross validation with grid search. The NN training was done using the Adam optimisation algorithm. The options for the hyperparameters were: hidden layer size: 1 layer with 25, 30 and

35 neurons, 2 layers with 35 and 25, 40 and 30 neurons, respectively; learning rate: 0.0005, 0.001, 0.003, 0.005, 0.01, 0.1 and 0.3; L2 regularization parameter λ: 0.0005, 0.0008, 0.001 and 0.0015; batch size: 64 and 256, and number of epochs: 500 and 900. The activation functions were ReLu for the hidden layers and linear for the output layer, and the weight initialization mode was set to normal.

The best hyperparameters for the NN models are listed in Table 4, and were used for the evaluation on the test set. The number in brackets in the first column denotes the model built for the corresponding pattern sequence type. For example, after the aggregation, there were two pattern sequences for PSNN-PSF: PSNN-PSF (1) and PSNN-PSF (2). L_{min} was set to 20%.

Persistence Model. As a baseline used for comparison, we developed a persistence prediction model B_{per} which considers the PV power output of day d as the forecast for day d+1.

Evaluation Measures. To evaluate the performance on the test set, we used the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE).

6 Results and Discussion

Table 5 shows the MAE and RMSE results of all models. We also conducted a pair-wise comparison between the prediction models for statistical significance of the differences in accuracy (MAE and RMSE, at point level) using the Wilcoxon signed-ranked test with $p < 0.05$.

Table 5. Accuracy of all methods

Model	MAE (kW)	RMSE (kW)
B_{per}	124.80	184.29
PSF	117.15	149.77
PSF1	115.55	147.72
PSF2	109.89	141.50
NN	79.46	117.65
PSNN-PSF	77.17	**106.33**
PSNN-PSF1	**77.10**	107.04
PSNN-PSF2	77.37	109.09

The main results can be summarized as follows:

- The best prediction model is PSNN-PSF1 in terms of MAE and PSNN-PSF in terms of RMSE. This shows the effectiveness of the PSNN approach in combining pattern sequences with NNs.

- All PSNN models outperform their corresponding PSF models. More specifically, PSNN-PSF outperforms PSF, PSNN-PSF1 outperforms PSF1 and PSNN-PSF2 outperforms PSF2, and the differences are statistically significant.
- All PSNN models also outperform the NN model but only the difference between PSNN-PSF and NN is statistically significant.
- Comparing the PSNN models, we can see that all three models perform similarly, with PSNN-PSF1 and PSNN-PSF performing slightly better than PSNN-PSF2. The pair differences between the three PSNN models are statistically significant except for PSNN-PSF1 vs PSNN-PSF2.
- Among the PSF models, PSF2 is the most accurate, followed by PSF1 and PSF, and the pair differences are statistically significant. This finding is consistent with [6], showing that the PSF extensions utilizing more than one data source are beneficial.
- All prediction models outperform the persistence baseline B_{per}.

We conducted additional analysis of the performance of PSNN, NN and PSF, comparing the characteristics of the days for which they performed well. We found that PSNN-PSF performs well for sunny days without rain. On the other hand, NN is the best model for rainy days and days following humid and windy days. A possible explanation is a sudden weather change, in which case the similarity of the sequence of previous days, used in PSF and PSNN but not in NN, is less important. This deserves further investigation.

7 Conclusion

In this paper, we present PSNN - a new approach combining pattern sequences with neural networks. It is a general approach that can be used with different clustering and cluster sequence extraction algorithms. It takes as an input a sequence of cluster labels, extracts pattern sequences of different types and builds a separate NN prediction model for each of them. PSNN can be applied to multiple complementary time series. In our case study for solar power forecasting, we show three instantiations of the PSNN approach by employing the pattern sequence extraction algorithms PSF, PSF1 and PSF2. We evaluate the performance of PSNN on Australian data for two years, from three sources (solar, weather and weather forecast). Our results show that PSNN was the most accurate method, with PSNN-PSF1 and PSNN-PSF obtaining the highest accuracy. All PSNN versions outperformed the PSF methods, and the differences were statistically significant. They also outperformed the NN model used for comparison but not all differences were statistically significant. Hence, we conclude that both PSNN and NN are promising approaches for solar power forecasting.

In future work we plan to investigate: (i) the use of other clustering algorithms in the PSF part, which may capture better the characteristics of the time series compared to the currently used k-means algorithm, (ii) seasonal changes and if building seasonal prediction models as in [8] can improve the accuracy, and (iii) the application of PSNN to other time series forecasting tasks.

References

1. Pedro, H.T., Coimbra, C.F.: Assessment of forecasting techniques for solar power production with no exogenous inputs. Solar Energy **86**, 2017–2028 (2012)
2. Rana, M., Koprinska, I., Agelidis, V.: Forecasting solar power generated by grid connected PV systems using ensembles of neural networks. In: International Joint Conference on Neural Networks (IJCNN) (2015)
3. Shi, J., Lee, W.J., Liu, Y., Yang, Y., Wang, P.: Forecasting power output of photovoltaic system based on weather classification and support vector machine. IEEE Trans. Ind. Appl. **48**, 1064–1069 (2012)
4. Rana, M., Koprinska, I., Agelidis, V.G.: 2D-interval forecasts for solar power production. Solar Energy **122**, 191–203 (2015)
5. Martínez-Álvarez, F., Troncoso, A., Riquelme, J.C., Aguilar Ruiz, J.S.: Energy time series forecasting based on pattern sequence similarity. IEEE Trans. Knowl. Data Eng. **23**, 1230–1243 (2011)
6. Wang, Z., Koprinska, I., Rana, M.: Solar power forecasting using pattern sequences. In: Lintas, A., Rovetta, S., Verschure, P.F.M.J., Villa, A.E.P. (eds.) ICANN 2017. LNCS, vol. 10614, pp. 486–494. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68612-7_55
7. Torres, J., Troncoso, A., Koprinska, I., Wang, Z., Martínez-Álvarez, F.: Big data solar power forecasting based on deep learning and multiple data sources. Expert Syst. **36**(4), e12394 (2019)
8. Koprinska, I., Rana, M., Agelidis, V.G.: Yearly and seasonal models for electricity load forecasting. In: International Joint Conference on Neural Networks (IJCNN), pp. 1474–1481 (2011)