Solar Power Forecasting Based on Pattern Sequence Similarity and Meta-learning

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 consider the task of simultaneously predicting the solar power output for the next day at half-hourly intervals using data from three related time series: solar, weather and weather forecast. We propose PSF3, a novel pattern sequence forecasting approach, an extension of the standard PSF algorithm, which uses all three time series for clustering, pattern sequence extraction and matching. We evaluate its performance on two Australian datasets from different climate zones; the results show that PSF3 is more accurate than the other PSF methods. We also investigate if a dynamic meta-learning ensemble combining the two best methods, PSF3 and a neural network, can further improve the results. We propose a new weighting strategy for combining the predictions of the ensemble members and compare it with other strategies. The overall most accurate prediction model is the meta-learning ensemble with the proposed weighting strategy.

Keywords: Solar power forecasting · Pattern sequence similarity forecasting · Neural networks · Meta-learning ensemble

1 Introduction

Renewable energy production and utilization is rapidly growing, providing numerous economic, environmental and social benefits. This is encouraged by government policies, with many countries setting renewable energy targets. Solar energy produced by photovoltaic (PV) systems is one of the most promising renewable options. However, unlike traditional energy sources such as gas, oil and coal, solar energy is variable as it is weather-dependent. This makes its integration into the electricity grid more difficult and requires forecasting of the generated solar power, to balance supply and demand during power dispatching and to ensure the stability and efficient operation of the electricity grid.

Both statistical and machine learning methods have been applied for solar power forecasting. Linear regression, exponential smoothing and auto-regressive

I. Farkaš et al. (Eds.): ICANN 2020, LNCS 12396, pp. 271–283, 2020.
https://doi.org/10.1007/978-3-030-61609-0_22

moving average are classical statistical methods for modelling time series data [1]. Previous studies [2,3] have shown that they are promising for solar power forecasting especially when a single time series is available, but have limitations for modelling noisy and nonlinear time series [4]. Machine learning methods such as Neural Networks (NNs) [2,3,5,6], nearest neighbour [2,3] and support vector regression [7], have also been applied and often shown to provide more accurate predictions than statistical methods.

Recently, the application of Pattern Sequence-based Forecasting (PSF) [8] methods has been studied for solar power forecasting [9–11]. PSF assigns a cluster label to each day and then uses a nearest neighbour approach to find sequences of days which are similar to the target sequence. 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, as most of the other methods. While the standard PSF algorithm uses only one time series (the time series of interest, PV data), two PSF extensions using multiple related time series (weather and weather forecast data) have been proposed in [9] and evaluated for solar power forecasting. The results showed that both PSF1 and PSF2 were more accurate than the standard PSF. However, there is still an opportunity for further improvement, since even the best performing PSF2 method does not fully utilize all three available data sources for constructing the pattern sequences. In this paper, we propose PSF3, a new pattern sequence forecasting algorithm, which utilizes multiple related time series for clustering, sequence extraction and pattern matching.

In addition, we investigate if a dynamic meta-learning ensemble, combining PSF3 and NN, can further improve the performance. Motivated by [12], we employ meta-learners to predict the errors of the ensemble members for the new day and assign higher weights to the more accurate ones. However, unlike [12] which combines ensemble members of the same type (NNs) and generates diversity by using random examples and feature sampling, we create a heterogeneous ensemble combining the predictions of PSF3 and NN, to generate natural diversity, and we also utilize data from multiple data sources (PV, weather and weather forecast) not only PV data. In addition, we propose a new weighting strategy, which increases the difference in contribution between the most and least accurate ensemble members, and show the effectiveness of this strategy.

In summary, the contributions of this paper are as follows:

1. We propose PSF3, a novel pattern sequence forecasting approach, which uses all three related time series (PV, weather and weather forecast) for pattern sequence extraction, matching and forecasting. We evaluate the performance of PSF3 on two Australian datasets, from different climate zones, each containing data for two years from three different sources. Our results show that PSF3 outperformed the other PSF methods.
2. We investigate if a dynamic meta-learning ensemble (MLE), combining PSF3 and NN, can further improve the results. We propose a new log weighting strategy for combining the ensemble member predictions and show its effectiveness compared to linear and softmax strategies. The most accurate

prediction model was the meta-learning ensemble with the log weighting strategy, outperforming the single methods it combines, all other PSF methods and a persistence baseline model used for comparison.

2 Task and Data

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: $PV = [PV_1, ..., PV_d]$, 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: $W = [W_1, ..., W_d]$, 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 collected data from two Australian PV plants located in different climate zones: humid subtropical (Brisbane) and hot desert (Alice Springs). The two datasets are referred as the University of Queensland (UQ) and Sanyo, and contain both PV and weather data. The data sources and extracted features for each dataset are shown in Table 1 and Table 2 respectively.

Solar PV Data. The PV data for the UQ dataset was collected from a rooftop PV plant located at the University of Queensland in Brisbane[1]. The Sanyo dataset was collected from the Sanyo PV plant in Alice Springs[2]. Both datasets contain data for two years - from 1 January 2015 to 31 December 2016 (731 days).

Weather Data. The weather data for the UQ dataset was collected from the Australian Bureau of Meteorology[3]. There are three sets of weather features - W1, W2 and WF, see Table 1 and Table 2.

W1 includes the full set of collected weather features - 14 for the UQ dataset and 10 for the Sanyo dataset. The 10 features are common for both datasets but UQ contains four additional features (daily rainfall, daily sunshine hours and cloudiness at 9 am and 3 pm) which were not available for Sanyo.

W2 is a subset of W1 and includes only 4 features for the UQ dataset and 3 features for the Sanyo dataset. These features are frequently used in weather forecasts and are 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.

[1] https://solar-energy.uq.edu.au/.
[2] http://dkasolarcentre.com.au/source/alice-springs/dka-m4-b-phase.
[3] https://www.bom.gov.au/climate/data/.

Table 1. UQ dataset - data sources and feature sets

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

Table 2. Sanyo dataset - data sources and feature sets

Data source	Feature set	Description
PV data	PV$\in \Re^{731\times 20}$	Daily: half-hourly solar power from 7 am to 5 pm
Weather data 1	W1$\in \Re^{731\times 10}$	(1–4) Daily: min temperature, max temperature, max wind gust and average solar irradiance; (5–10) At 9 am and 3 pm: temperature, relative humidity and wind speed
Weather data 2	W2$\in \Re^{731\times 3}$	Daily: min temperature, max temperature and solar irradiance. W2 is a subset of W1
Weather forecast data	WF$\in \Re^{731\times 3}$	Daily: min temperature, max temperature and average solar irradiance

2.2 Data Pre-processing

The raw PV data was measured at 1-min intervals for the UQ dataset and 5-min intervals for the Sanyo dataset and was aggregated to 30-min intervals by taking the average value of every 30-min interval. There was a small percentage of missing values - for the UQ dataset: 0.82% in the PV power and 0.02% in the weather data; for the Sanyo dataset: 1.98% in the PV power and 4.85% in the weather data. These missing values were replaced as in [13] by a nearest neighbour method, applied firstly to the weather data and then to the PV data. Both the PV and weather data were normalised between 0 and 1.

3 Pattern Sequence Forecasting Methods: PSF, PSF1 and PSF2

PSF [8] utilizes a single data source for clustering and sequence matching - the time series of interest, which is the PV data in our case. It uses k-means to cluster the days in the training data based on their PV vectors into k_1 clusters with labels $C_1, ..., C_{k_1}$. To make a prediction for a new day d+1, PSF extracts a target sequence of w consecutive days, starting from the previous day and going backwards. This sequence of cluster labels is matched with the previous days to find the set of equal sequences ES. The final prediction is obtained by averaging the PV vectors of the post-sequence days for each sequence in ES.

PSF1 [9] is an extension of PSF, which utilizes the W2 data for clustering and sequence matching. It clusters the training set days based on the W2 data into k_2 clusters with labels $C_1, ..., C_{k_2}$. To make a prediction for a new day d+1, PSF1 firstly obtains the cluster label for this day using its WF vector. It then extracts a target sequence of cluster labels for 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 for each sequence in ES.

PSF2 [9] is an extension of PSF utilizing two of the related time sequences for clustering and pattern matching - W1 and W2. It clusters the days from the training set using W1 (k_1 clusters with labels $C_1, ..., C_{k_1}$) and W2 (k_2 clusters with labels $K_1, ..., K_{k_2}$). The prediction for the new day d+1 is computed as follows. A target sequence of cluster labels for w consecutive days from day d backwards and including day d is extracted based on W1 and matched to find the set of equal sequences ES. The cluster label K_x for day d+1 is obtained based on WF. Then, the cluster label of the post-sequence day for all sequences in ES is checked and if it is not 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 all sequences in ES.

In summary, in PSF and PSF1 the clustering, sequence extraction and pattern matching are done using only one of the related time series, while PSF2 uses two of them. The proposed PSF3 algorithm builds upon PSF2, but utilizes all three related time series (PV, W1 and W2) for clustering, sequence extraction and pattern matching. We investigate if this approach improves performance.

4 Proposed Methods

4.1 Pattern Sequence Forecasting: PSF3

PSF3 is an extension of PSF, which utilizes all related time series (PV, weather and weather forecast) in the clustering, pattern extraction and matching phases.

PSF3 firstly employs the k-means algorithm to cluster the days from the training data separately based on their PV, W1 and W2 vectors. Specifically, it clusters the days (i) based on the W1 weather data in k_1 clusters with labels

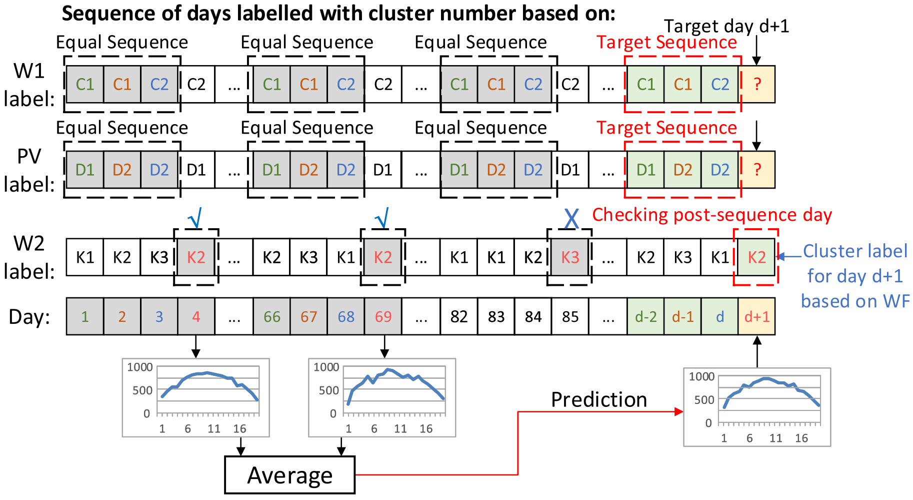

Fig. 1. The proposed PSF3 method

$C_1, ..., C_{k_1}$, (ii) based on the PV data in k_2 clusters with labels $D_1, ..., D_{k_2}$ and (iii) based on the W2 weather forecast data in k_3 clusters with labels $K_1, ..., K_{k_3}$, as shown in Fig. 1. Hence, each day from the training data is assigned three cluster labels, one for each of the three clusterings.

To make a prediction for a new day d+1 from the test set, PSF3 firstly assigns cluster labels to the previous days if they were not part of the training set. This is done by comparing these days with the cluster centroids and assigning them to the cluster of the closest centroid, for each data source separately. Then, the prediction for the new day d+1 is computed using the following steps:

1. A target sequence of w consecutive days from day d backwards, including day d, is extracted based on the W1 weather data. This sequence consists of the cluster labels. It is matched with the previous days to find the set of equal sequences ES.
2. The same process is repeated for the PV data - a target sequence of w consecutive days from day d backwards, including day d, is extracted based on the PV data and matched to find the set of equal sequences ES_{PV}.
3. The sequences included in ES and ES_{PV} are compared, to find a subset for which both the W1 and PV cluster label sequence matches. The non-matching sequences are excluded from ES. Only ES is used for further analysis, and not ES_{PV}.
4. The cluster label K_x for day d+1 is obtained based on the weather forecast data.
5. The cluster label of the post-sequence day for each sequences in ES is checked and if it is not K_x, these sequences are excluded from ES.
6. The final prediction of the PV power for day d+1 is obtained by taking the average of the PV powers of the post-sequence days for all sequences in ES.

For example, in Fig. 1, the PV power prediction for day $d+1$ will be the average of the PV vectors for days 4 and 69. Note that day 85 is not included in the final prediction - the sequence ending with day 84 matches the target sequence in terms of both W1 and PV, but the post-sequence days do not match in terms of W2 - the cluster label of day 85 is K_3, while the cluster label of day $d+1$ is K_2, i.e. the matching condition in step 5 is not satisfied.

The window size w and the number of clusters $k1$, $k2$ and $k3$ are hyperparameters of the PSF3 algorithm, selected using 12-fold cross-validation with grid search as described in Sect. 5.

4.2 Meta-Learning Ensemble: MLE

We also investigate if a dynamic meta-learning ensemble can further improve the performance of PSF3. The motivation behind dynamic ensembles is that the different ensemble members have different areas of expertise and as the time series changes over time, the accuracy of the ensemble members also changes. By using a suitable criterion, we can select the most appropriate weighted combination of ensemble members for the new example.

A meta-learning ensemble (called EN-meta), combining only NNs and employing random example and random feature sampling, was used in [12], demonstrating good performance for univariate solar power forecasting (PV data only). In this paper, we propose to build a heterogeneous ensemble combining the predictions of the two best single models - PSF3 and NN. PSF3 and NN are representatives of different machine learning paradigms. Heterogeneous ensembles have been shown to perform well in various applications [12,14,15], by providing natural diversity and complementary expertise. Another important difference with EN-Meta [12] and the other ensemble methods [14,15], is that we utilize data from different data sources and related time series (PV, weather and weather forecast), not only univariate data (PV).

The main idea of our Meta-Learning Ensemble (MLE) is to predict the error of each ensemble member for the new day and based of this error to determine the contribution of the ensemble member in the final prediction.

Building MLE involves three steps: 1) training ensemble members, 2) training meta-learners, 3) calculating the weights of the ensemble members for the new day and predicting the new day.

Training Ensemble Members. The first step is training the ensemble members, PSF3 and NN, to predict the PV power for the next day. PSF3 performs clustering separately on the PV, W1 and W2 data and then pattern matching, see Sect. 4.1. NN also uses all three data sources but directly, without clustering, to build a prediction model. It constructs training data as follows: the feature vector consists of the PV and W1 weather data for day d and the weather forecast data W2 for the next day $d+1$: $[\mathrm{PV}_d; \mathrm{W1}_d; \mathrm{W2}_{d+1}]$, and the associated target output is the PV vector for the next day: PV_{d+1}.

Training Meta-learners. The second step is training the meta-learners. There is one meta-learner for each ensemble member and we used NN models as meta-learners. Each meta-learner takes the same input as above ($[\mathrm{PV}_d; \mathrm{W1}_d; \mathrm{W2}_{d+1}]$), but learns to predict the error of its corresponding ensemble member (PSF3 or NN) for the next day. We used the Mean Absolute Error (MAE) as an error. For example, the meta-learner for the NN ensemble member ML_{NN} will have as an input $[\mathrm{PV}_d; \mathrm{W1}_d; \mathrm{W2}_{d+1}]$ and as a target output MAE_{d+1} of the NN ensemble member. Similarly, the meta-learner of the PSF3 ensemble member ML_{PSF3} will take the same input but learn to predict MAE_{d+1} of the PSF3 ensemble member. To create a training set for a meta-learner, we first obtain the predictions of its corresponding ensemble member for all training examples and then calculate the MAEs, which are the target outputs.

Weight Calculation and Final Prediction. The third step involves calculating the weights of the ensemble members for the new day by converting the predicted errors into weights and calculating the final weighted average prediction.

The rationale behind MLE is that different ensemble members have different areas of expertise and their performance changes over time. It assumes that the error of an ensemble member could be predicted based on its past performance and uses this error to weight the contributions of ensemble members when making the final prediction - the ensemble members which are predicted to be more accurate will be given higher weights.

Two weight calculation strategies were investigated in [12]: linear and softmax nonlinear. The linear strategy decreases the weight of an ensemble member linearly as its error increases:

$$w_i^{d+1} = \frac{1 - e_{norm,i}^{d+1}}{\sum_{j=1}^{S}(1 - e_{norm,j}^{d+1})}$$

where w_i^{d+1} is the weight of ensemble member E_i for predicting day $d + 1$, $e_{norm,i}^{d+1}$ is the predicted error of E_i by its corresponding meta-learner, normalized between 0 and 1, and S is the number of ensemble members.

The nonlinear strategy in [12] computes a softmax function of the negative of the predicted error:

$$w_i^{d+1} = \frac{exp(-e_i^{d+1})}{\sum_{j=1}^{S} exp(-e_j^{d+1})} = softmax(-e_i^{d+1})$$

where e_i^{d+1} is the predicted error of E_i by its corresponding meta-learner and exp denotes the exponential function.

The use of $1 - e_{norm,i}^{d+1} and - e_i^{d+1}$ above is necessary for the inverse relationship between forecasting error and weight (lower errors resulting in higher weights and vice versa), and the denominator ensures the weights of all ensemble members sum to 1.

We propose a new nonlinear weight calculation strategy, the log weighting, where the weights are calculated as follows:

$$w_i^{d+1} = \frac{ln(\frac{e_i^{d+1}}{\sum_{m=1}^{S} e_m^{d+1}})^{-1}}{\sum_{j=1}^{S} ln(\frac{e_j^{d+1}}{\sum_{m=1}^{S} e_m^{d+1}})^{-1}}$$

In summary, we take the natural logarithm of the inverse of the ensemble member error, normalised over all ensemble members. The denominator ensures that the weights of all ensemble members sum to 1.

Compared to the linear and softmax strategies, the log strategy increases the difference between the weights of the more accurate and less accurate ensemble members. Hence, it increases the contribution of the most accurate ensemble members and decreases the contribution of the less accurate ones in the final prediction.

5 Experimental Setup

All prediction models were implemented in Python 3.6 using scikit-learn 0.22.1 and Keras 2.3.1 libraries. For both datasets, the PV power and corresponding weather data were split into two equal subsets: training and validation (the first year) and test (the second year). Cross-validation with grid search was used for tuning of the hyperparameters as described below.

Table 3. Hyperparameters for the PSF models

UQ dataset		Sanyo dataset	
Method	Hyperparameters	Method	Hyperparameters
PSF	$k_1 = 2$, $w = 2$	PSF	$k_1 = 2$, $w = 1$
PSF1	$k_1 = 2$, $w = 2$	PSF1	$k_1 = 2$, $w = 2$
PSF2	$k_1 = 2$, $k_2 = 2$, $w = 1$	PSF2	$k_1 = 2$, $k_2 = 2$, $w = 2$
PSF3	$k_1 = 2$, $k_2 = 2$, $k_3 = 2$, $w = 1$	PSF3	$k_1 = 2$, $k_2 = 2$, $k_3 = 2$, $w = 1$

5.1 Tuning of Hyperparameters

For the PSF models, the first year was used to determine the hyperparameters (number of clusters k_1, k_2 and k_3, and sequence size w) by using 12-fold cross-validation with grid search, consistent with the original PSF algorithm [8]. The grid search for w included values from 1 to 5. The best number of clusters was selected by varying k_1, k_2 and k_3 from 1 to 10 and evaluating three cluster quality indexes (Silhouette, Dunn and Davies-Bouldin) as described in [9]. The selected best hyperparameters are listed in Table 3.

For the NN models, the tuning of the hyperparameters was done using 5-fold cross-validation with grid search on the first year data. The training algorithm was the mini-batch gradient descent with Adam optimization. The tunable hyperparameters and options considered were: hidden layer size: 1 layer with 25, 30, 35 and 40 neurons, 2 layers with 25 and 20, 30 and 20, 35 and 25, 40 and 30 neurons; learning rate: 0.0005, 0.001, 0.003, 0.005, 0.01, 0.1 and 0.3; L2 regularization λ: 0.0005, 0.0008, 0.001 and 0.0015; batch size: 64 and 256, and fixed number of epochs 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 selected hyperparameters for the NN models based on the cross-validation performance are listed in Table 4. After the best parameters were selected, a new NN model is built using the whole first year data and then evaluated on the test set.

For the NN meta-learners, we followed the same procedure as above but used early stopping instead of the maximum number of epochs. The selected hyperparameters are also listed in Table 4.

Table 4. Hyperparameters for the NN models

Model	Hidden layer size	Learning rate	L2 λ	Batch size	Epochs
NN (UQ dataset)	[25]	0.0005	0.0015	64	900
NN (Sanyo dataset)	[35, 25]	0.001	0.0015	64	900
NN meta-learner (UQ dataset)	[25]	0.0015	0.0001	64	505
NN meta-learner (Sanyo dataset)	[35]	0.001	0.0015	64	103

5.2 Baseline and Evaluation Measures

As a baseline for comparison, we developed a persistence prediction model B_{per}. It simply predicts the PV power output of the previous day as the PV power output for the next day, i.e. $\hat{PV}_{d+1} = PV_d$.

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

6 Results and Discussion

Table 5 shows the MAE and RMSE results of all models for UQ and Sanyo datasets. The main results can be summarized as follows:

- The proposed PSF3 is the most accurate PSF method, followed by PSF2, PSF1 and PSF. This shows the effectiveness of the proposed pattern matching algorithm, utilising all available data sources.

Table 5. Accuracy of all models

Method	UQ dataset		Sanyo dataset	
	MAE (kW)	RMSE (kW)	MAE (kW)	RMSE (kW)
B_{per}	124.80	184.29	0.75	1.25
PSF	117.15	149.77	0.77	1.07
PSF1	115.55	147.72	0.70	0.98
PSF2	109.89	141.50	0.69	0.98
PSF3	106.11	138.39	0.65	0.95
NN	81.23	115.76	0.49	0.74
MLE (linear)	81.26	111.47	0.51	0.74
MLE (softmax)	81.09	**109.73**	0.52	0.76
MLE (log)	**78.67**	110.74	**0.48**	**0.72**

- The accuracy ranking of the other PSF methods (PSF2, PSF1 and PSF) is consistent with [9], confirming that the extensions PSF1 and PSF2 outperformed the standard PSF algorithm.
- From the single models, NN is the most accurate model, outperforming all PSF models. This finding is also consistent with [9]. Unlike the PSF models, NN does not require clustering and pattern matching; it uses directly the data from the three sources to construct a feature vector and build a prediction model. Compared to the PSF models, however, it requires significantly more time for training and parameter tuning.
- The overall most accurate prediction model is MLE (log), the proposed meta-learning ensemble with the log weighting strategy. This shows the benefit of combining the two best models, NN and PSF3, using meta-learning and the advantage of using the log weighting strategy.
- All three ensemble models are considerably more accurate than the PSF methods in all cases but only the ensemble MLE (log) was consistently more accurate than NN on both datasets. This again shows the advantage of using log weighting strategy to increase the influence of the more accurate ensemble members, compared to the linear and softmax strategies.
- All models outperform the persistence baseline B_{per}, except for one case - PSF on the Sanyo dataset for MAE.
- We further analysed the performance of the best models, PSF3 and NN, by comparing the characteristics of the days for which they performed well. NN is better than PSF3 at forecasting days with slightly higher PV power and temperature on both datasets, and days with higher humidity for the Sanyo dataset.

7 Conclusions

We considered the task of simultaneously predicting the PV solar power for the next day at half-hourly intervals using data from three related time series:

PV, weather and weather forecast. We proposed PSF3, a novel pattern sequence forecasting approach, an extension of the standard PSF algorithm, which uses all three time series for clustering, pattern sequence extraction and matching. We evaluated its performance on two Australian datasets, from different climate zones (humid subtropical and hot desert), each containing data for two years. Our results show that PSF3 was more accurate than the other PSF methods. We also investigated if a dynamic meta-learning ensemble combining the two best methods, PSF3 and NN, can further improve the results. For this ensemble, we proposed and evaluated a new log weighting strategy for combining the predictions of the ensemble members and showed that it was more effective than linear and softmax strategies. The overall most accurate prediction model was the meta-learning ensemble with the proposed weighting strategy; it outperformed both PSF3 and NN, the other PSF models and the persistence baseline.

References

1. Trull, Ó., García-Díaz, J.C., Troncoso, A.: Application of discrete-interval moving seasonalities to Spanish electricity demand forecasting during Easter. Energies **12**, 1083 (2019)
2. Pedro, H.T., Coimbra, C.F.: Assessment of forecasting techniques for solar power production with no exogenous inputs. Sol. Energy **86**, 2017–2028 (2012)
3. Chu, Y., Urquhart, B., Gohari, S.M., Pedro, H.T., Kleissl, J., Coimbra, C.F.: Short-term reforecasting of power output from a 48 MWe solar PV plant. Sol. Energy **112**, 68–77 (2015)
4. Inman, R.H., Pedro, H.T., Coimbra, C.F.: Solar forecasting methods for renewable energy integration. Prog. Energy Combust. Sci. **39**, 535–576 (2013)
5. 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)
6. Chen, C., Duan, S., Cai, T., Liu, B.: Online 24-h solar power forecasting based on weather type classification using artificial neural networks. Sol. Energy **85**, 2856–2870 (2011)
7. Rana, M., Koprinska, I., Agelidis, V.G.: 2D-interval forecasts for solar power production. Sol. Energy **122**, 191–203 (2015)
8. 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)
9. Wang, Z., Koprinska, I., Rana, M.: Solar power forecasting using pattern sequences. In: Proceedings of the International Conference on Artificial Neural Networks (ICANN) (2017)
10. 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**, e12394 (2019)
11. Lin, Y., Koprinska, I., Rana, M., Troncoso, A.: Pattern sequence neural network for solar power forecasting. In: Proceedings of the International Conference on Neural Information Processing (ICONIP) (2019)
12. Wang, Z., Koprinska, I.: Solar power forecasting using dynamic meta-learning ensemble of neural networks. In: Proceedings of the International Conference on Artificial Neural Networks and Machine Learning (ICANN) (2018)

13. Wang, Z., Koprinska, I.: Solar power prediction with data source weighted nearest neighbors. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN) (2017)
14. Cerqueira, V., Torgo, L., Pinto, F., Soares, C.: Arbitrated ensembles for time series forecasting. In: Proceedings of the European Conference on Machine Learning and Principles of Knowledge Discovery in Databases (ECML-PKDD) (2017)
15. Galicia, A., Talavera-Llames, R., Troncoso, A., Koprinska, I., Martínez-Álvarez, F.: Multi-step forecasting for big data time series based on ensemble learning. Knowl.-Based Syst. **163**, 830–841 (2019)