IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 16, 2020, accepted October 22, 2020, date of publication October 27, 2020, date of current version November 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034100

Solar Power Forecasting Based on Domain Adaptive Learning

HANMIN SHENG[1], (Member, IEEE), BIPLOB RAY[2], (Member, IEEE), KAI CHEN[1], (Member, IEEE), AND YUHUA CHENG[1], (Senior Member, IEEE)

[1]Department of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Centre for Intelligent Systems (CIS), School of Engineering and Technology (SET), Central Queensland University, Rockhampton, QLD 4701, Australia

Corresponding author: Kai Chen (kaichen@uestc.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC1505203, and in part by the National Natural Science Foundation of China under Grant 61903066.

ABSTRACT Solar power forecasting is critical to ensure the safety and stability of the power grid with high photovoltaic power penetration. Machine learning methods are compelling in solar forecasting. These methods can capture the complex coupling relationship between different meteorological factors without physical modeling. Most of the existing machine learning based forecasts follow the batch learning manner. Once the training is completed, the structure and parameters of the model are usually no longer adjusted. However, the climate is complex and dynamic. It is difficult for a fixed model to adapt to the climate characteristics of different regions or periods. Therefore, an online domain adaptive learning approach is proposed in this paper. Knowledge can be selectively accumulated or forgotten in its iterative process. As weather changes, the model can dynamically adjust its structure to adapt to the latest weather conditions. Unlike existing adaptive iterative methods, the proposed adaptive learning approach does not rely on the labels of the test data in the updating process. Experiments show that this method can effectively track changes in data distribution and obtain reliable prediction results.

INDEX TERMS Solar power forecasting, adaptive learning, neural networks, ensemble learning.

NOMENCLATURE

Symbols and abbreviations	Description
Y	Solar irradiation
X	Meteorological variables
X^{Origin}	Original variables
X^{Norm}	Centralized variables
X^{Train}	Training input
X^{Test}	Test input
X^{Val}	Validation input
e^{Val}	Absolute error with respect to the validation set
γ	Mean of absolute error
f	Forecasting model
$\widehat{Y}$	Forecasted output
μ_j	Mean value of j^{th} feature
φ	Mapping from the original space to the new space
L	Smoothness loss function
ε	Performance of sub-model
t	Iteration step
T	Maximum iteration step
ξ	Weight adjustment threshold
Z	Normalization factor
ω	Sample weight
ς	Adjustment rate
K	Number of neighborhood samples
q	Dimension of smooth embedding space
d	Dimension of meteorological variables
Q	Number of sub-models selected
I^{clr}	Theoretical clear sky irradiance
PV	Photovoltaic
NWP	Numerical weather prediction
ANN	Artificial neural networks
BPNN	Backpropagation neural network
SVM	Support vector machine
ELM	Extreme learning machine
GPR	Gaussian process regression
GMR	Gaussian mixture regression

The associate editor coordinating the review of this manuscript and approving it for publication was Dongbo Zhao.

AdaBoost	Adaptive boosting
NREL	National Renewable Energy Laboratory
RMSE	Root mean square error
MBE	Mean bias error
MAE	Mean absolute error

I. INTRODUCTION

Solar energy is clean and sustainable [1]. Due to the decline in cost, photovoltaic (PV) power generation has developed rapidly. PV power generation has intermittent and stochastic nature due to external factors that cannot be controlled [2]. Therefore, accurate PV power generation forecasts are critical to ensure the balance between power generation and electricity consumption [3]. Solar power is the main factor affecting PV power generation; much effort has been devoted to solar power forecasting. Existing methods can be roughly divided into three categories: the numerical weather prediction (NWP), the image-based technique, and the statistical methods. NWP methods generate solar irradiance with hundreds of meteorological parameters [4]; It is a versatile approach for 6-48-h-ahead forecasting [5]. However, NWP model is based on dynamic equations of the atmospheric states computed with a fixed resolution, usually at the regional and national scale, not site-specific. Imagery-based forecasting methods mainly study the influence of clouds on the sun rays. Sky cameras [6], shadow cameras [7], and satellite images [8] are the primary image sources. Such methods usually have high fidelity in short-term forecasts.

The statistical methods discover the relationships and patterns related to solar power through data mining—no need to handle the complex coupling mechanisms between different climate characteristics. Traditional statistical methods include time series treatments [9] and the persistence models [10]. The persistence models assume the clear sky index remains constant for a short period. The times series treatments are based on data consistency. These methods usually have a good predictive performance in clear weather, and the prediction error will be higher in rapidly changing weather.

As a branch of statistical-based methods, machine learning has become increasingly important in solar power forecasting because of its advantages in nonlinear fitting. Commonly used machine learning prediction methods include artificial neural networks (ANN), support vector machines (SVM), extreme learning machines (ELM), ensemble learning, and Gaussian process regression (GPR). ANNs [11] are comprised of flexible structures. If many neurons in the hidden layers are allowed, the model will have a strong nonlinear fitting capability. Still, flexible and complex structures will cause over-fitting problems, especially for small sample scenarios. A SVM based solar power forecasting approach has been proposed in [12]. Compared with ANNs, it has better small sample generalization ability. ELM [13] has received much attention in recent years. This algorithm has advantages in learning speed. However, the stability of ELM is controversial. Ensemble methods use multiple sub-learners to obtain better predictive performance [14]; diversity of sub-models is leveraged. Evaluating the prediction of an ensemble typically requires more computation than evaluating the prediction of a single model [15]. A Gaussian process regression based solar power forecasting algorithm has been reported in [16]. This method provides the posterior probability of the prediction result while making predictions. Some researchers also used deep learning for solar power prediction [17]. Deep learning has a strong nonlinear fitting capability, but at the same time, it has higher requirements for data volume. An overview of machine learning-based solar forecasting approaches has been provided in the literature [5].

Existing machine learning based solar power forecasts are mainly based on batch learning manner. Once the training is completed, the structure and parameters of the model are no longer updated. However, it is difficult for a fixed model to adapt to the climate characteristics of different regions and different time periods. The research of adaptive learning is valuable in the field of solar power forecasting. On the one hand, an adaptive learning model can continuously accumulate knowledge and adapt to changes in the environment. On the other hand, if the prediction model is applied in a new area, the model can be updated based on the existing knowledge, instead of giving up all the prior experience, which reduces the dependence on data volume.

Some adaptive learning methods have been proposed. A dynamic performance evaluation strategy has been introduced in [18]. During its update process, the model reconstructs the training set based on a synoptic event. A disadvantage is that it cannot inherit the information obtained from the previous training process. An adaptive combination strategy based on Bayesian theory has been reported in [19], multiple sub-models are combined with a Bayesian model averaging technique. Each sub-model is weighted based on training samples; higher weights are assigned to better performing predictors. If the distribution of the training set and the test set are different, the reliability of the model will be affected. [20] has established an adaptive learning hybrid model, linear models work together with a backpropagation neural network (BPNN). When a new group of data is acquired, the model determines whether to add the data to the previous dataset and retrain the BPNN model. This method cannot inherit the experience gained from previous iterative steps. Moreover, it depends on the labels of the test data. Its updating rate is restricted by the forecast horizon. A novel online adaptive learning framework is proposed in this paper. Unlike the above-mentioned adaptive strategies, the proposed method can inherit the knowledge obtained from the previous training process. Knowledge can be incrementally accumulated or selectively forgotten. More importantly, this method avoids the use of test data labels. The following contributions are made:

- A domain adaptive learning approach is proposed to realize knowledge accumulation and selective forgetting function.

- A smooth embedding space is constructed, where the mapped sample has a consistent mapping relationship with its neighbor samples.
- Sub-models are weighted based on the performance of neighborhood points, avoiding the dependence on test set samples during iteration.

The rest of the paper is organized as follows, in section II, the smooth embedding space and the domain adaptive learning framework is introduced in detailed. Experiments are detailed in section III, which is followed by conclusion in section IV.

II. MODEL CONSTRUCTION

A. SMOOTH EMBEDDING SPACE

A smooth embedding space is constructed in this sub-section, where the neighboring samples have similar mapping relationships. In this way, test data can be replaced by its neighboring points for knowledge selection. In other words, if neighboring samples have similar mapping relationships ($f : \boldsymbol{x} \rightarrow y$), the labels corresponding to adjacent points in the feature space should have similar values. To this end, a smooth loss function is proposed to evaluate the mapping relationships' consistency for a specific feature space. Suppose all the training and testing data are projected to a new feature space by mapping function $\varphi \in \mathbb{R}^{n\times q}$. the smoothness loss function L is represented as equation (1).

$$\arg\min_{\varphi} L = \frac{\sum_{i=1}^{n}\left(y|x_i\varphi^T - \frac{1}{K}\sum_{k=1}^{K} y_{i,k}^{neighbor}|x_{i,k}^{neighbor}\varphi^T\right)}{\sum_{i=1}^{n_1}\left\|x_i - x_{i,j}^{neighbor}\varphi^T\right\|} \tag{1}$$

where x_i is the input feature of i^{th} sample, $x_{i,j}^{neighbor}$ is the neighboring points of x_i, K is the maximum number of neighbor points selected. $x_i\varphi^T$ is the projection of sample x_i in the new space and $y|x_i\varphi^T$ is its corresponding label. By minimizing the loss function L, the adjacent samples in the space will have similar label values. The sample should present a smooth transition state. Define the original inputs $X^{Origin} \in \mathbb{R}^{n\times d}$. The mean of each feature has the form

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{2}$$

X^{Origin} can be centralized by subtracting the mean from each sample, the element in its i^{th} row and j^{th} column can be characterized as X^{Norm}

$$x_{i,j}^{Norm} = x_{i,j}^{Origin} - \mu_j \tag{3}$$

Calculate the eigenvectors of X^{Norm}. φ is a combination of the eigenvectors. When smoothness loss function L takes the minimum value, φ at this time is the mapping function from the original space to the smooth embedding space.

B. DOMAIN ADAPTIVE LEARNING

In subsection II. A, a smooth embedding space is constructed. On this basis, a domain adaptive learning algorithm is proposed in order to track the changes in data distribution. First, the training dataset is divided by sliding windows, knowledge learned from different sliding windows is selectively used in the prediction process. The Adaptive Boosting (AdaBoost) learning strategy [21] is adopted, which can generate a variety of sub-models while avoiding overfitting. Suppose we have training dataset with n elements, noted as $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x_i \in \mathbb{R}^{d\times 1}$ is the input feature, $y_i \in \mathbb{R}$ is the label. Weak learners are generated as follows:

(1) Define the length of the sliding window as m. At step s, sliding window contains data $\left(x_{1+m*(s-1)}, y_{1+m*(s-1)}\right), \ldots, (x_{m*s}, y_{m*s})$.

(2) Call the learning algorithm f. Single hidden layer neural network is used as the base learner in this paper. The number of hidden layer nodes is set to a random integer between $[2d, 5d]$.

(3) Each sample is assigned with a weight ω_i. In AdaBoost regression, the iteration step is denoted by t, $t \in [1, T]$.

(4) At the initial stage, weights are evenly distributed among all the samples, $\omega_i^{s=1,t=1} = 1/m$. In the iterative process, $\omega_i^{s,t}$ is adjusted according to the estimation error. The error of the instance i at step t can be calculated as Equation (4).

$$e_t(i) = \left(f^{s,t}(x_i) - y_i\right)^2 \tag{4}$$

The PV panels cannot generate electricity at night, if the relative error is used, its denominator might be zero. Therefore, variance is used in equation (4) instead of the relative error.

(5) Evaluate the performance of $f^{s,t}(x)$ according to equation (5).

$$\varepsilon_t = \sum \omega_{t,i}, \quad i|e_t(i) > \xi \tag{5}$$

where ε_t in an index evaluates the model's performance at step t . ξ is an error threshold. As illustrated in Fig. 1, the red bubbles represent the samples with large estimation errors and the blue bubbles represent samples with smaller estimation errors. If the estimation error is higher than the threshold ξ, its weight will be enhanced in the next iteration.

(6) Sample weights are adjusted based on the rules presented in Equation (6).

$$\omega_{t+1,i} = \frac{\omega_{t,i}}{Z_t} \times \begin{cases} \varepsilon_t^{\varsigma} & \left|f^{s,t}(x_i) - y_i\right| \le \xi \\ \varepsilon_t^{-\varsigma} & \left|f^{s,t}(x_i) - y_i\right| > \xi \end{cases} \tag{6}$$

where $\varepsilon_t \in [0, 1]$, ς is a hyper-parameter used to control the adjustment rate. When ε_t is close to 1, it means that most samples have a prediction error larger than ξ, if ε_t is close to 0, it shows that the model fits well with most instances. Z_t is a normalization factor chosen such that Equation (7) can be satisfied.

$$\sum_{i=1}^{n} \omega_{t+1,i} = 1 \tag{7}$$

So far, multiple sub-models can be obtained. The model's output is a weighted combination of sub-models. Intuitively, the sub-models with better performance in the target domain should be assigned with higher weights. Test data is essential to evaluate the performance of the sub-models. However, the label of the test data is not available for the training phase. A domain adaptive learning strategy is proposed to solve this problem. As introduced in section II. A, all the data is projected to a smooth embedding space, the adjacent samples have similar mapping relationships; therefore, the test data can be replaced by its neighbors for knowledge screening. The implementation is as follows:

(1) Obtain the mapping φ according to Equation (1), all the training samples and test samples are mapped to the low dimensional smooth embedding space, marked as X^{Train} and X^{Test}. $x_i^{Train}, x_i^{Test} \in \mathbb{R}^d$, $X^{\text{Train}}\varphi^T, X^{\text{Test}}\varphi^T \in \mathbb{R}^q$, q is the dimension of the new space, $q \ll d$.

(2) Normalize the data, search the k nearest neighbors for each test sample. Let $\left(x_{(1)}^{Train}, y_{(1)}^{Train}\right), \left(x_{(2)}^{Train}, y_{(2)}^{Train}\right), \ldots, \left(x_{(m)}^{Train}, y_{(m)}^{Train}\right)$ be a reordering of the training data as presented in Equation (8).

$$\left\|x_{(1)}^{Train} - x_i^{Test}\right\| \leq \left\|x_{(2)}^{Train} - x_i^{Test}\right\| \cdots \leq \left\|x_{(m)}^{Train} - x_i^{Test}\right\| \tag{8}$$

(3) The validation dataset is composed of the selected nearest k samples, shown as Equation (9).

$$X^{Val} = \left\{x_{(1)}^{Val}, x_{(2)}^{Val}, \ldots, x_{(k)}^{Val}\right\} \tag{9}$$

(4) In Fig. 1, the orange bubble indicates the test data, and the blue bubbles present the neighbors of the test data, which are used as the validation dataset. Absolute error of each validation sample can be calculated according to Equation (10).

$$e_i^{Val} = \left(f^{s,t}\left(x_i^{Val}\right) - y_i^{Val}\right)^2 \tag{10}$$

where $f^{s,t}$ is the model learned from the sliding window s at iteration t. The performance of the model can be characterized by the mean value of absolute error, noted as γ_t.

$$\gamma_t = {}^1\!/_k \sum_{i=1}^{k} e_i^{Val} \tag{11}$$

(5) A large γ_t indicates the model has a large error. At this time, the model should be given a low weight. The output of the adaptive model is given by Equation (12).

$$\widehat{Y} = \frac{\sum_t \log\left(\frac{1}{\gamma}\right) f_t(x)}{\sum_t \log\left(\frac{1}{\gamma}\right)} \tag{12}$$

By continuously choosing the knowledge that consistent with the target domain, the model can track the changes in data distribution adaptively.

As shown in Fig. 1, the sample points with large errors (sample 1, 2 of the base model 1 and sample 3, 4 of the base model 2) will have higher weights in the iteration process. The weights of sub-models are decided by the estimation errors of the neighboring points of the test data. The pseudo code of the domain adaptive learning model is shown in Algorithm 1.

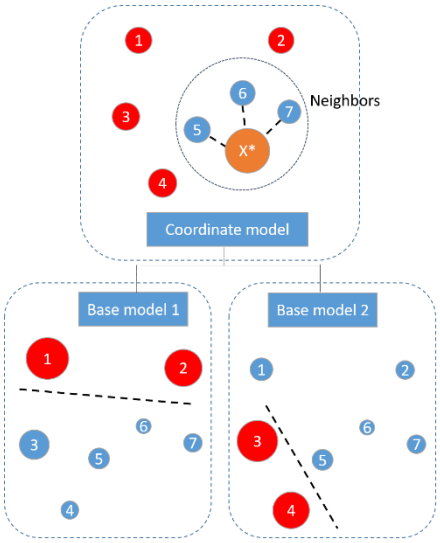

FIGURE 1. The two-layer structure of adaptive ensemble learning.

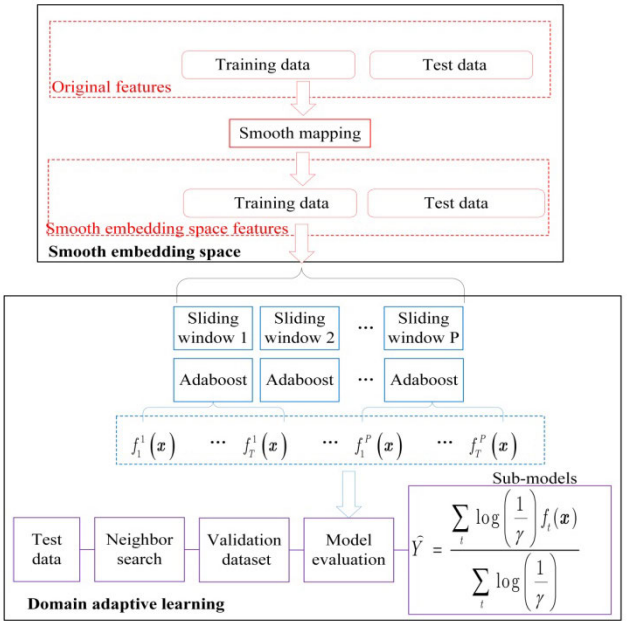

FIGURE 2. Block diagram of domain adaptive learning.

The block diagram of domain adaptive learning is shown in Fig. 2. Data is updated with sliding windows. Multiple sub-models are generated through the AdaBoost strategy. The sub-models are dynamically integrated according to their fitness in the target domain. The fitness evaluation of the sub-model does not depend on the label of test data. In domain adaptive learning, the test data and the training data are mapped to the smooth embedding space. The neighborhood points of the test data are used to evaluate the adaptability of the sub-models. The knowledge learned is represented by

Algorithm 1 Domain Adaptive Learning Algorithm

Input

- Given training input $(x_i, y_i), i = 1, \ldots, n$. where $x_i \times y_i \in \mathbb{R}^d \times \mathbb{R}^1$.
- Learning algorithm f.
- Length of sliding window m.
- Number of AdaBoost iterations T.
- Number of hidden nodes N ($N \in [2d, 5d]$).
- Hyperparameter ς.
- Number of neighborhood searches k.
- Number of sub-models to be selected Q.

Output

- Forecasting value $\widehat{Y}$ for input X^{Test}.

Algorithm:

1. Data normalization in Eq. (3).
2. Matrix decomposition.
3. Select a combination φ from the feature vectors to minimize the smoothing loss function in Eq. (1).
4. Map the training data to the smooth feature space through $x\varphi$.
5. **for** s = 1: *floor* (n/m), select training subset $(x_i, y_i), i = (s-1) \times m + 1, \ldots, s \times m$.
6. Initialize the weight vector $D_i^{s=1,t=1} = 1/m$for all i.
7. **for** t < T do
8. Call the i^{th} learner f_t and provide it with the weight $D_i^{s,t}$ of each sample.
9. Obtain the model $f^{s,t}(x) \rightarrow y$ through training.
10. Compute the forecasting error rate in Eq (4).
11. Adjust the weight of each sample in Eq (6).
12. **end for**
13. Adjust Z_t such that $\sum_{i=1}^{n} \omega_{t+1,i} = 1$.
14. **end for**
15. Map X^{Test} to the smooth embedding space.
16. Sort the distance of the neighbors according to formula (8).
17. Choose the k nearest samples.
18. Calculate the error of each sub-model in Eq (10).
19. Select Q sub-models that consistent with the distribution of the validation dataset, and use them to build an ensemble model.
20. Make prediction in equation (12).
21. **Return** $\widehat{y}$.

the structure and parameters of sub-models. As the number of sub-models increases, knowledge can be accumulated. If the weight of a sub-model is close to zero in the iterative process, the knowledge it represents is selectively forgotten.

III. EXPERIMENT

A. DETAIL OF THE REAL-WORLD DATASETS

Three real-world datasets from the National Renewable Energy Laboratory (NREL) [22] and E. E. P. Ltd., Australia [23] are used:

- The first dataset is provided by NREL, collected at 39.742° North, 105.169° West, Solar Radiation Research Laboratory, United States. Which span the whole duration of the year 2019 sampled at 1-min intervals.
- The second dataset is provided by NREL, collected at 39.911° North, 105.235° West, Flatirons Campus, United States, from 1st January 2015 to 31st December 2018. This dataset is with 1-min interval.
- The third dataset is collected by E. E. P. Ltd. [23] from a large scale PV station in Townsville, Australia (19.2500° South, 146.7670° East 1st. January 1991 to 31st December 2015). Solar irradiance is collected every hour.

The NREL solar radiation research laboratory and the Flatiron campus have a continental climate, the average solar irradiation is higher in summer and weaker in winter. Townsville has a subtropical climate. This experiment uses meteorological features: solar irradiation, temperature, wind speed and humidity. In addition to the directly measured meteorological features, time and extra-terrestrial radiation are also used as the model's input. The experiment has used time sequence input: 60 min, 65 min, 70 min, 1day, 2 days.

The domain adaptive learning has two main steps: smooth embedding space construction and domain adaptive learning. The former ensures that adjacent samples in the space have similar mapping relationships, and the latter ensures that the model adapts to the latest data distribution. The smoothness performance before and after feature mapping is tested in sub-section II.B, and the effectiveness of the domain adaptive learning is verified in sub-section II.C. Several similar algorithms are introduced for comparison.

B. CONSTRUCTION OF SMOOTHNESS EMBEDDING SPACE

The smoothing the embedding space is a feature space where the adjacent samples have similar mapping relationships, that is, a combination of features with a small smoothing loss function. The data provided by the Solar Radiation Research Laboratory is used in this experiment. The data from 1st January to 31st January, 2019 is used as the training set, and the data from 1st February to 28th February is used as the test set. Meteorological variables include solar irradiation, ambient temperature, time, extraterrestrial radiation, wind speed, and humidity. Smooth embedding spaces of different dimensions are constructed. The smoothness indicators of different feature combinations are shown in table I.

Among the commonly used meteorological variables, solar irradiation, time, and extraterrestrial radiation have a low smoothness loss function value. In contrast, the smoothness loss function of temperature, humidity, and wind speed are higher. As the forecasting horizon increases, the smoothness index gradually deteriorates. Compared with the original meteorological features, the mapped features have better smoothness performance, and these features are decoupled from each other. In actual applications, the operator can choose different smooth mapping spatial dimensions

TABLE 1. Smooth index of different features.

Feature		Smoothness index train	Smoothness index test
Solar irradiation	60-min-ahead	48.9618	64.5118
	65-min-ahead	51.7331	68.5657
	120-min-ahead	85.2287	108.6343
	1-day-ahead	46.1377	70.7842
	2-day-ahead	48.9840	69.3753
	3-day-ahead	49.0547	63.3229
Ambient temperature	60-min-ahead	130.3931	170.5067
	65-min-ahead	131.9256	171.9215
	120-min-ahead	138.1978	178.8751
	1-day-ahead	131.8802	165.2405
	2-day-ahead	129.5421	169.0150
	3-day-ahead	128.9489	166.2390
Time	Predict moment	39.3424	51.6458
Extraterrestrial radiation	Predict moment	35.6037	54.0977
Wind speed	60-min-ahead	142.4923	171.2667
	65-min-ahead	144.3949	175.1757
	120-min-ahead	147.1348	175.7655
	1-day-ahead	143.2361	171.5767
	2-day-ahead	145.5398	173.5893
	3-day-ahead	145.4378	175.7462
Humidity	60-min-ahead	128.9532	171.8085
	65-min-ahead	129.6318	172.5701
	120-min-ahead	133.9793	175.2225
	1-day-ahead	132.3753	171.0260
	2-day-ahead	129.0767	168.0861
	3-day-ahead	131.8195	169.3117
Solar irradiation, ambient temperature and wind speed		44.0021	55.4883
Temperature, wind speed and Humidity		104.2524	133.2012
One dimensional smooth embedding space		39.6102	52.4122
Three dimensional smooth embedding space		22.7384	34.5475
Four dimensional smooth embedding space		22.1446	35.0911
Six dimensional smooth embedding space		20.8994	32.6910

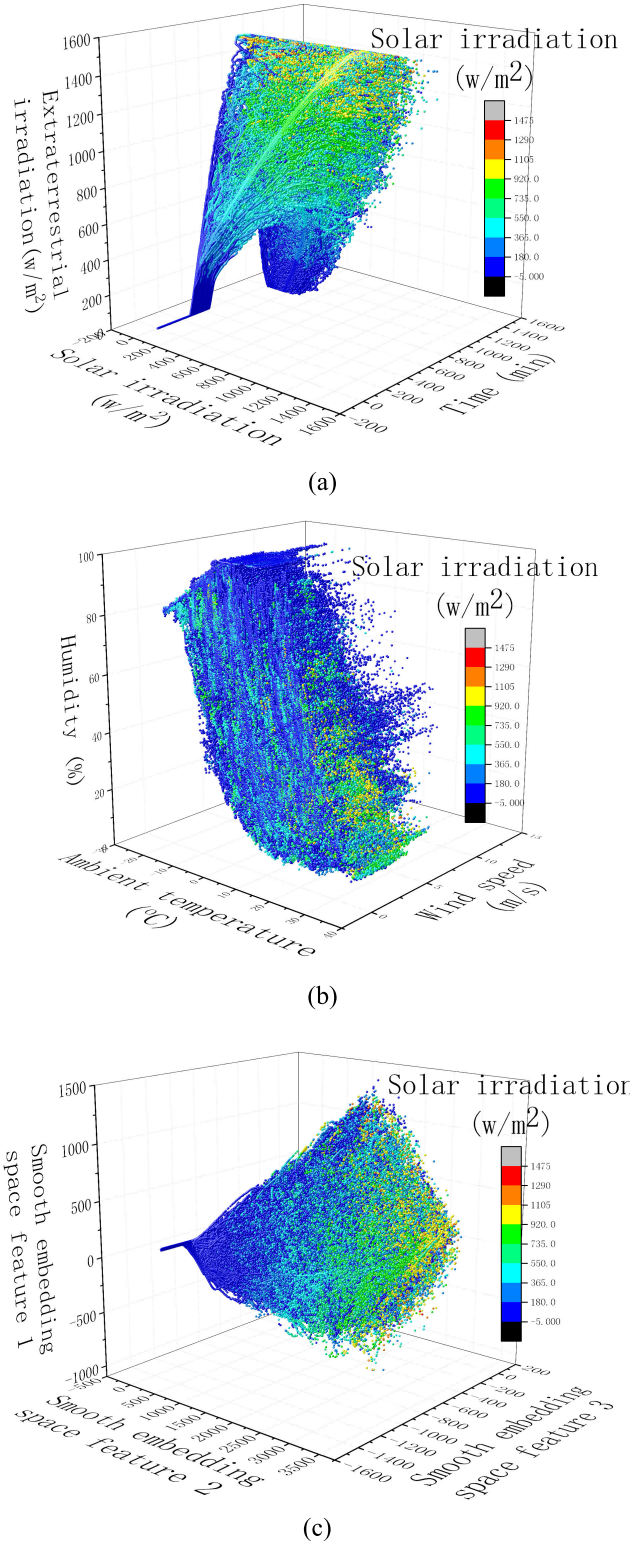

FIGURE 3. The smoothness of label changes: (a) features with high smoothness indexes, (b) features with low smoothness indexes, and (c) smooth embedding feature space.

according to the amount of information that needs to be retained. It can also be seen from the table I that the training dataset's smoothness indexes and the test dataset are highly consistent. This shows the constructed smooth embedding space can be generalized from the training set to the test set. Smoothness index is not just a mathematical definition, and physically it characterizes the smoothness of the label transition in the feature space, as shown in Fig. 3.

Fig. 3(a) is plotted with the feature of solar irradiation, time, and extraterrestrial radiation. Fig. 3(b) is plotted with the feature of temperature, humidity and wind speed. Fig. 3(c) shows the distribution of labels in the three-dimensional smooth embedding space. In comparison, the features used in Fig. 3(c) has a better smoothness performance. Intuitively, if the label is distributed randomly, a slight disturbance in the input can cause a significant difference in the output. The model is prone to overfit in this case. It is worth noting that the Smoothness index is defined based on the Euclidean spatial distance, and the comparison of the smoothness index of different feature combinations needs to be performed in the same dimension.

TABLE 2. Parameter settings of domain adaptive learning.

Parameter	Value
AdaBoost iteration number	50
Activation function	*Tansig*
Optimization function	Momentum gradient descent
Learning rate	0.01
Maximum epochs	500
Optimization goal	0.001
Performance evaluation function	SSE
Number of neighborhood points	5

C. SOLAR POWER FORECASTING EXPERIMENT

In this paper, a single hidden layer neural network is used as the base learner. A variety of parameter configurations are tested. The parameter settings of the domain adaptive learning model are shown in the Table 2.

Some well-known data-driven solar power forecasting methods are used for comparison (ANN [24], Gaussian mixture regression (GMR) [16], SVM [25], ELM [13] and clear-sky persistence model [26]). The model structure and parameter settings follow the parameter settings in the reference. The ANN model has a single hidden layer structure with 30 hidden nodes. Competitive swarm optimization approach is used for optimization. Radial basis function is the activation function and root mean average error (RMSE) is adopted as the fitness function. The GPR model has a squared exponential covariance function, *10-fold* cross-validation is implemented for parameter selection. The SVM model uses the radial basis kernel function, *10-fold* cross-validation is utilized for parameter tuning. The ELM has a single hidden layer structure. The following hidden layer node numbers are tested: 220, 260, 300, and 410, and the number with the best performance is used as the ELM comparison model. All of these algorithms have been fine-tuned. The clear-sky model assumes the clearness index k_t remains unchanged during the forecast interval. $k_t$is the ratio of the global horizontal irradiance to that extraterrestrial irradiance, It can be expressed mathematically as Equation (13).

$$k_{t+\Delta t} = k_t = \frac{GHI_t}{I_t^{\text{clr}}} \tag{13}$$

where Δt is the forecast horizon, I_t^{clr} is the clear sky irradiance at time t. k_t is the clear sky index at time t, represents the ratio of the actual solar irradiance to the theoretical solar irradiance under the clear sky condition. For the persistence model, the predicted solar irradiation can be calculated as Equation (14).

$$GHI_{t+\Delta t}^{\text{pers}_k} = k_t I_{t+\Delta t}^{\text{Clr}} \tag{14}$$

where I^{Clr} is the theoretical clear sky solar irradiance derived using Equation (15) [27].

$$I^{Clr} = 951.39 \times (\cos\theta)^{1.15} \tag{15}$$

The forecasting time horizon is 60 min in this paper. Mean bias error (MBE), mean absolute error (MAE), and RMSE

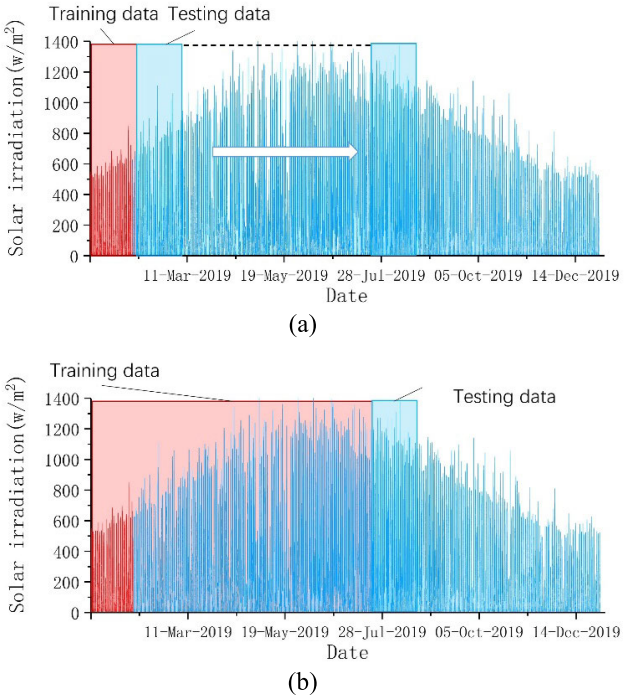

FIGURE 4. Data division of the Solar Radiation Research Laboratory:(a) initial state; (b) sliding process.

are currently the most popular metrics [3], the experiment considers these options.

$$MBE = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{Y}(x_i) - Y(x_i)\right) \tag{16}$$

$$MAE = \frac{1}{\mathrm{N}}\sum_{i=1}^{N}\left|\widehat{Y}(x_i) - Y(x_i)\right| \tag{17}$$

$$RMSE = \sqrt{\frac{1}{\mathrm{N}}\sum_{i=1}^{N}\left(\widehat{Y}(x_i) - Y(x_i)\right)^2} \tag{18}$$

where N is the number of training data. The first experiment used the dataset provided by the solar radiation research laboratory. In the initial stage, the data from 1st Jan. to 31st Jan. is used as the training dataset. A sliding window is used to select the test dataset. In the iteration process, the window slides forward, and the sliding interval is 30 days. Each time the sliding window moves, the previous step's test dataset is added to the training dataset. The division of training set and test set are shown in Fig. 4. The red window represents the training data, and the blue window denotes the test data. The forecasting performance of different machine learning methods are shown in Fig. 5.

Table 3 shows the average value of MAE, RMSE, and MBE. The prediction error increases gradually in the first six months, peaks in June, and then gradually decreases. The seasonal differences lead to the changes in data distribution and eventually bring the periodic change in error. The forecast

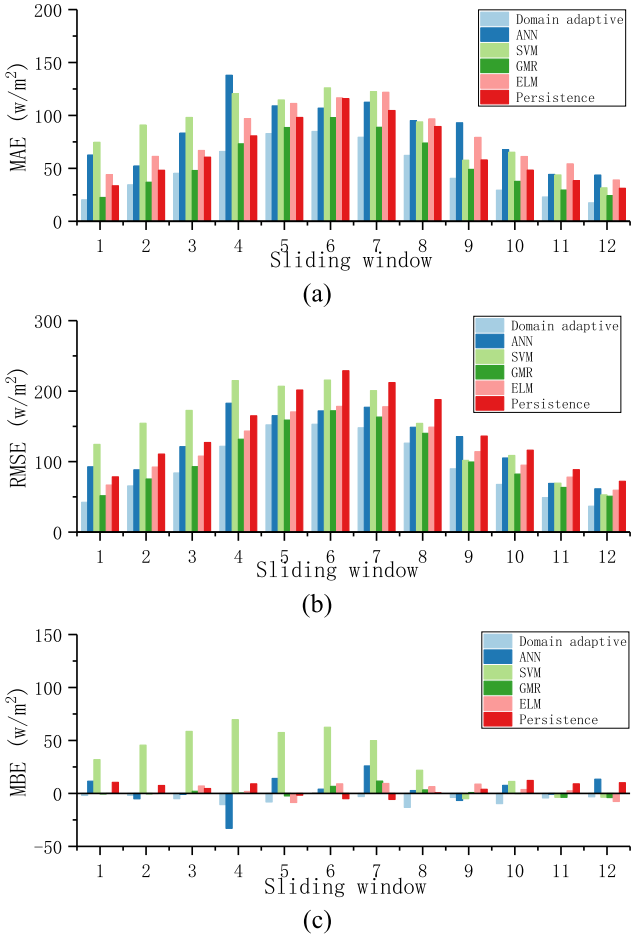

FIGURE 5. Data distribution and models' performance (Solar Radiation Research Laboratory dataset): (a) MAE, (b) RMSE, and (c) MBE.

TABLE 3. Performance comparison (solar radiation research laboratory dataset).

Algorithm	MAE(w/m^2)	RMSE(w/m^2)	MBE(w/m^2)
Domain adaptive learning	48.8530	94.6843	-5.3984
ANN	84.0367	126.6187	2.7501
SVM	86.6476	148.0981	33.0502
GMR	55.8532	106.7835	1.1504
ELM	79.1386	119.3766	2.6953
Persistence model	67.2620	143.7483	4.5844

error is related to the model performance and is affected by seasonal factors.

Among several comparison algorithms, GMR has better prediction accuracy. The clear-sky persistence model has also achieved reliable prediction results in this experiment. Domain adaptive learning continuously selects and reorganizes historical information in the forecasting process. It can be seen from the results that although the advantage in MBE is not apparent. Domain adaptive learning is more stable in

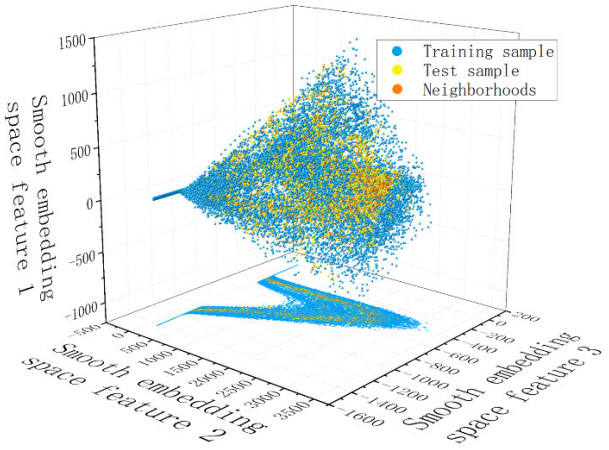

FIGURE 6. Distribution of training set, test set and neighborhood samples.

terms of MAE and RMSE compared with the comparison algorithms. In the next experiment, the mechanism of domain adaptive learning will be illustrated in the experiment experiments.

The key to the domain adaptive method is the knowledge updating mechanism. It is realized by neighborhood sample evaluation. This practice a compromise way because the labels of the test data samples are unavailable in the training stage. The feasibility of this means is verified in the following experiment. The data of September is used as the test dataset. Its neighbor points can be searched in the smooth embedding space, as shown in Fig. 6. In the iterative process, 50 sub-models are generated based on the AdaBoost algorithm. The test dataset and its neighborhood points are put into these models. Root mean square error (RMSE) of the model is shown in figure Fig. 7. It can be seen from the results that the test set and the neighborhood points have different errors on these sub-models, but the trend is highly consistent. Therefore, when the sample label of the test set is not available, it can be replaced by the neighborhood point in the historical data..The data from 1st Sep. to 30th Sep. is used to demonstrate how the area adaptation algorithm works.

Knowledge obtained in the learning process is included in different sub-models. The accumulation of knowledge can be reflected in the growth of the sub-model number. Not all the sub-models are equally important in the forecasting process. For the AdaBoost ensemble learning, sub-models with smaller cross-validation errors are assigned with higher weights. The purpose of the domain adaptive learning is to adjust the weights based on the model's fitness to the target domain. The representation of model fitness depends on the target domain samples. If the test data is used to evaluate the model's fitness, one must wait until the actual label value is obtained. Instead, this paper uses the test set's neighborhood points in the smooth embedding space as the validation data.

The following experiment is to verify the substitution effect of neighborhood points on the test dataset. In this experiment, 50 sub-models are selected (generated from the ninth sliding

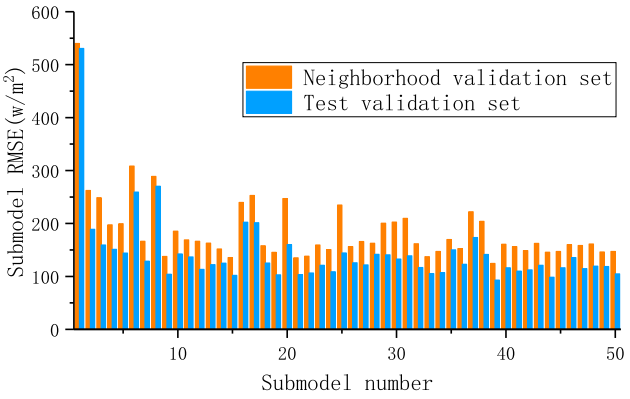

FIGURE 7. Errors of neighborhood points and test points in different sub-models.

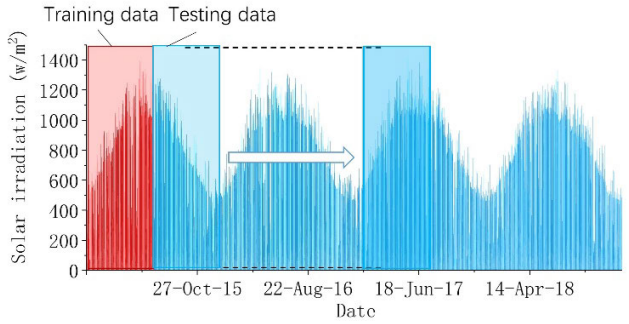

FIGURE 8. Data division of Flatirons campus dataset.

TABLE 4. Performance comparison(flatirons campus dataset).

Algorithm	MAE(w/m^2)	RMSE(w/m^2)	MBE(w/m^2)
Domain adaptive learning	87.6907	140.3673	1.9778
ANN	121.7609	172.9699	6.0445
SVM	103.7532	159.8781	-3.1797
GMR	102.4603	159.2625	-5.3557
ELM	116.7895	175.2956	0.8243
Persistence model	133.2884	232.5613	-52.9914

window). The neighborhood points and the test samples are used as the validation dataset. If the neighborhood points and the test data have similar effects, the error trends of different sub-models should be consistent. RMSE of the different sub-models is shown in Fig. 7. It can be seen from the results that the prediction errors of the test dataset and the neighborhood samples have the same trend in these sub-models. The best perform sub-models for the neighboring samples also happened to be the best perform sub-models for the test data. In the following two sets of experiments, data provided by Flatirons Campus and E. E. P. Ltd. are used for further verification.

Fig. 8 shows the data division of the training set and test set in the Flatirons campus dataset. In this experiment, the sliding

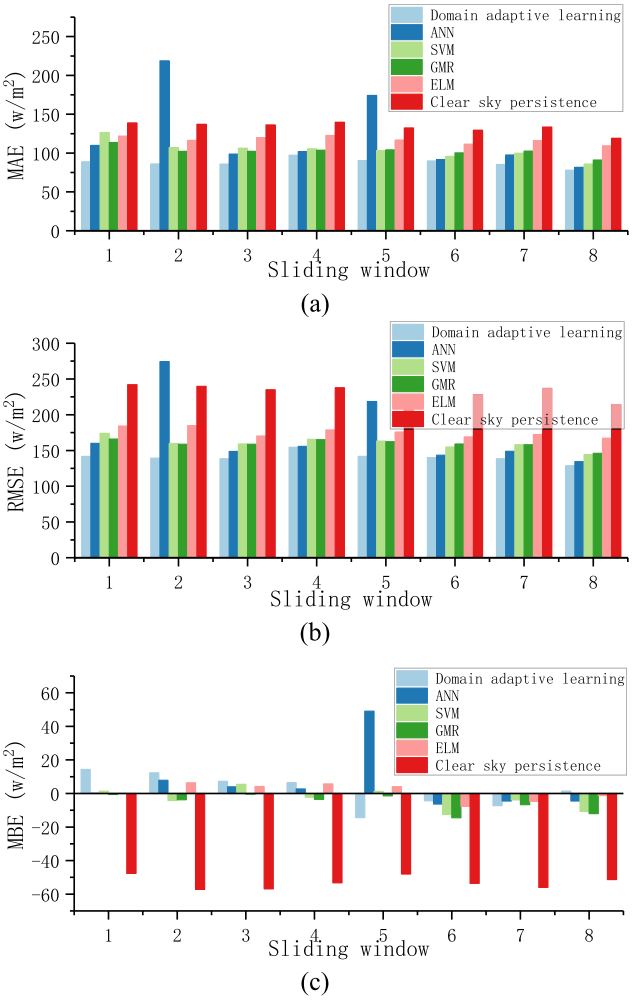

FIGURE 9. Model performance comparison (Flatirons campus dataset) : (a) MAE, (b) RMSE, and (c) MBE.

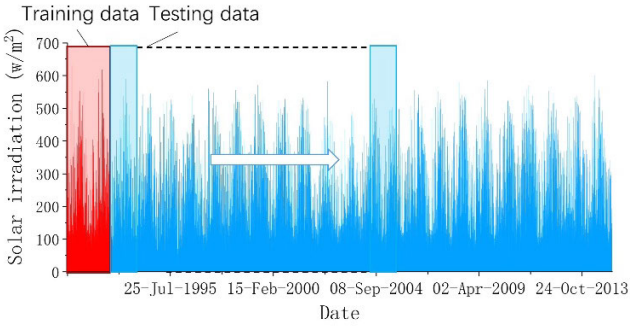

FIGURE 10. Data division of E. E. P. Ltd. dataset.

window's span is 180 days, and each step generates 50 new sub-models. The forecasting performance is shown in Fig. 9 and Table 4, the performance of ANN is not stable, and SVM and GMR have achieved relatively better prediction results. Due to the data's poor stability, the prediction error of the persistence model in this experiment is relatively high. In gen-

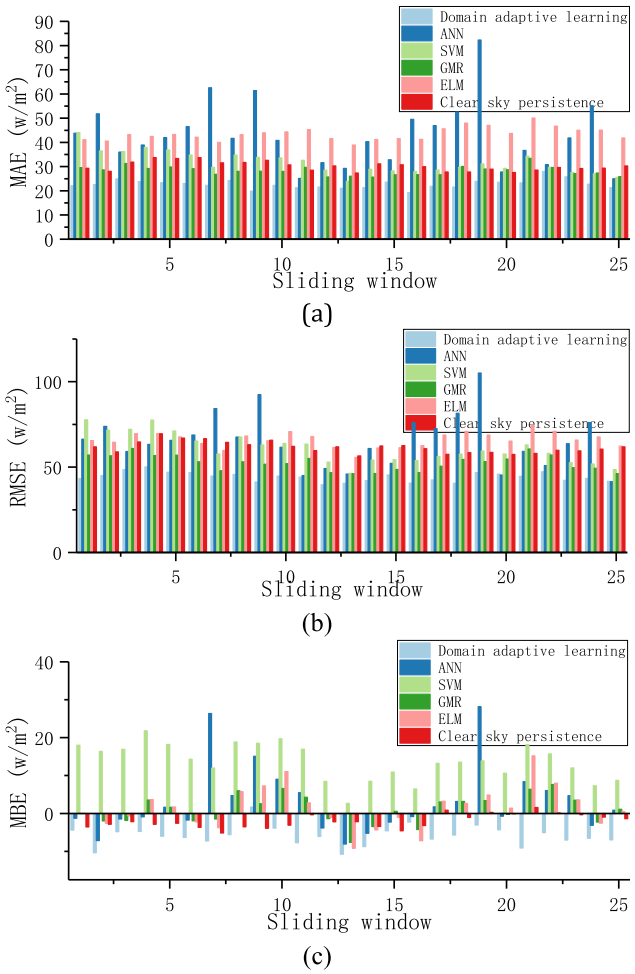

FIGURE 11. Performance comparison (E. E. P. Ltd. dataset): (a) MAE, (b) RMSE, and (c) MBE.

eral, domain adaptive learning has shown better performance in MAE, RMSE, and MBE.

Fig. 10 shows the division of training set and test set in E. E. P. Ltd. dataset. The length of the sliding window is 365 days. MAE, RMSE. Fig. 11 and Table 5 shows the performance comparison of different models. It can be seen that the domain adaptive learning model has significant advantages in MAE RMSE, but the benefit it is not prominent in MBE. The experiment also shows that when the time axis is stretched to the 25-year range, the data distribution has a cyclical change pattern over the years. The error peaks in the 7^{th} sliding window and the 19^{th} sliding window, and the error trough is at the 14^{th} sliding window. Compared with the previous two experiments, forecasting models have lower MAE, RMSE, and MBE value in this experiment.

There are several reasons why the forecasting models in this experiment have lower error rates. First, the increase in the data volume can reduce the risk of overfitting. Second, the region has sufficient solar irradiation and clear weather. Moreover, each sample used in this experiment is the average of the data in one hour, which leads to better data stability.

TABLE 5. Performance comparison(E. E. P. Ltd. dataset).

Algorithm	MAE(w/m^2)	RMSE(w/m^2)	MBE(w/m^2)
Domain adaptive learning	22.74674	44.21007	-5.86547
ANN	42.93479	65.1494	3.150263
SVM	31.64432	60.7089	13.67036
GMR	28.27839	52.4752	1.07478
ELM	43.48772	66.02163	1.48506
Persistence model	30.15786	61.604552	-2.02088

Even if the model used is the same, the model's performance is different when different data sets are used. Therefore, the comparison should be performed under the same data conditions.

IV. CONCLUSION

A domain adaptive learning approach is proposed in this paper. Knowledge can be selectively accumulated or forgotten in its iterative process to respond to the climate changes in solar power forecasting. Unlike the existing adaptive photovoltaic power forecasting methods, this algorithm avoids the use of test labels. Instead, the neighboring points in the smooth embedding space are used to judge the fitness of the sub-models. Experimental results show that the neighbors have similar knowledge selecting performance with the actual test set samples. Knowledge can be selectively updated or forgotten through domain adaptive learning. Comparative experiments show that the proposed model has a distinct advantage in terms of RMSE and MAE. In future work, the proposed domain adaptive learning can be extended to the transfer learning task; knowledge suitable for the target domain can be extracted from different data domains. When the training data of the target location is insufficient, the data from other regions can be used as a supplement.

REFERENCES

[1] T. Khalili, S. Nojavan, and K. Zare, "Optimal performance of microgrid in the presence of demand response exchange: A stochastic multi-objective model," *Comput. Electr. Eng.*, vol. 74, pp. 429–450, Mar. 2019.

[2] T. Khalili, A. Bidram, S. Nojavan, and K. Jermsittiparsert, "Selection of Cost-Effective and Energy-Efficient Storages with Respect to Uncertain Nature of Renewable Energy Sources and Variations of Demands," in *Integration of Clean and Sustainable Energy Resources and Storage in Multi-Generation Systems*. Cham, Switzerland: Springer, 2020, pp. 15–27. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-42420-6_2

[3] D. Yang, S. Alessandrini, J. Antonanzas, F. Antonanzas-Torres, V. Badescu, and H. G. Beyer, "Verification of deterministic solar forecasts," *Sol. Energy*, vol. 2020, no. 1, pp. 1–18, 2020.

[4] B. Wolff, J. Kühnert, E. Lorenz, O. Kramer, and D. Heinemann, "Comparing support vector regression for PV power forecasting to a physical modeling approach using measurement, numerical weather prediction, and cloud motion data," *Sol. Energy*, vol. 135, pp. 197–208, Oct. 2016.

[5] G. M. Yagli, D. Yang, and D. Srinivasan, "Automatic hourly solar forecasting using machine learning models," *Renew. Sustain. Energy Rev.*, vol. 105, no. 1, pp. 487–498, May 2019.

[6] Z. Zhen, S. Pang, F. Wang, K. Li, Z. Li, H. Ren, M. Shafie-khah, and J. P. S. Catalao, "Pattern classification and PSO optimal weights based sky images cloud motion speed calculation method for solar PV power forecasting," *IEEE Trans. Ind. Appl.*, vol. 55, no. 4, pp. 3331–3342, Jul. 2019.

[7] P. Kuhn, S. Wilbert, C. Prahl, D. Garsche, D. Schüler, T. Haase, L. Ramirez, L. Zarzalejo, A. Meyer, P. Blanc, and R. Pitz-Paal, "Applications of a shadow camera system for energy meteorology," *Adv. Sci. Res.*, vol. 15, pp. 11–14, Feb. 2018.
[8] R. Marquez, H. T. C. Pedro, and C. F. M. Coimbra, "Hybrid solar forecasting method uses satellite imaging and ground telemetry as inputs to ANNs," *Sol. Energy*, vol. 92, pp. 176–188, Jun. 2013.
[9] V. Kushwaha and N. M. Pindoriya, "A SARIMA-RVFL hybrid model assisted by wavelet decomposition for very short-term solar PV power generation forecast," *Renew. Energy*, vol. 140, pp. 124–139, Sep. 2019.
[10] D. Yang, "Choice of clear-sky model in solar forecasting," *J. Renew. Sustain. Energy*, vol. 12, no. 026101, pp. 1–10, 2020.
[11] X. Huang, J. Shi, B. Gao, Y. Tai, Z. Chen, and J. Zhang, "Forecasting hourly solar irradiance using hybrid wavelet transformation and elman model in smart grid," *IEEE Access*, vol. 7, pp. 139909–139923, 2019.
[12] H.-T. Yang, C.-M. Huang, Y.-C. Huang, and Y.-S. Pai, "A weather-based hybrid method for 1-Day ahead hourly forecasting of PV power output," *IEEE Trans. Sustain. Energy*, vol. 5, no. 3, pp. 917–926, Jul. 2014.
[13] Y. Han, N. Wang, M. Ma, H. Zhou, S. Dai, and H. Zhu, "A PV power interval forecasting based on seasonal model and nonparametric estimation algorithm," *Sol. Energy*, vol. 184, pp. 515–526, May 2019.
[14] X. Zhang, Y. Li, S. Lu, H. F. Hamann, B.-M. Hodge, and B. Lehman, "A solar time based analog ensemble method for regional solar power forecasting," *IEEE Trans. Sustain. Energy*, vol. 10, no. 1, pp. 268–279, Jan. 2019.
[15] C. Pan and J. Tan, "Day-ahead hourly forecasting of solar generation based on cluster analysis and ensemble model," *IEEE Access*, vol. 7, pp. 112921–112930, 2019.
[16] A. Rohani, M. Taki, and M. Abdollahpour, "A novel soft computing model (Gaussian process regression with k-fold cross validation) for daily and monthly solar radiation forecasting (part: I)," *Renew. Energy*, vol. 115, pp. 411–422, Jan. 2018.
[17] K. Wang, X. Qi, and H. Liu, "A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network," *Appl. Energy*, vol. 251, Oct. 2019, Art. no. 113315.
[18] Y. S. Manjili, R. Vega, and M. M. Jamshidi, "Data-Analytic-Based adaptive solar energy forecasting framework," *IEEE Syst. J.*, vol. 12, no. 1, pp. 285–296, Mar. 2018.
[19] M. Q. Raza, N. Mithulananthan, and A. Summerfield, "Solar output power forecast using an ensemble framework with neural predictors and Bayesian adaptive combination," *Sol. Energy*, vol. 166, pp. 226–241, May 2018.
[20] Y. Wang, Y. Shen, S. Mao, G. Cao, and R. M. Nelms, "Adaptive learning hybrid model for solar intensity forecasting," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1635–1645, Apr. 2018.
[21] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *J.-Jpn. Soc. Artif. Intell.*, vol. 14, nos. 771–780, p. 1612, 1999.
[22] U. S. D. O. Energy, "Solar Radiation Research Laboratory dataset," Nat. Renew. Energy Lab., Denver, CO, USA, 2019. [Online]. Available: https://midcdmz.nrel.gov/
[23] E. E. P. Ltd, "Townsville PV power generation dataset," E. E. P. Ltd, Fadden, ACT, Australia, 2015. [Online]. Available: http://www.exemplary.com.au/solar_climate_data/solar-radiation-data.php
[24] Z. Yang, M. Mourshed, K. Liu, X. Xu, and S. Feng, "A novel competitive swarm optimized RBF neural network model for short-term solar power generation forecasting," *Neurocomputing*, vol. 397, pp. 415–421, Jul. 2020.
[25] S. Buhan, Y. Özkazanç, and I. Çadirci, "Wind pattern recognition and reference wind mast data correlations with NWP for improved wind-electric power forecasts," *IEEE Trans. Ind. Informat.*, vol. 12, no. 3, pp. 991–1004, Jun. 2016.
[26] R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra, "Solar forecasting methods for renewable energy integration," *Prog. Energy Combustion Sci.*, vol. 39, no. 6, pp. 535–576, Dec. 2013.
[27] J. Adnot, B. Bourges, D. Campana, and R. Gicquel, "Utilisation de courbes de fréquences cumulées d'irradiation solaire globale pour le calcul des installations solaires," *Ener. Solaire*, vol. 1, no. 1, pp. 9–40, 1979.

HANMIN SHENG (Member, IEEE) received the B.Sc. and Ph.D. degrees in electrical engineering from Southwest Jiaotong University, Chengdu, China, in 2011 and 2017, respectively. He was a Visiting Scholar with the Energy Research Institute, Nanyang Technological University, Singapore, where he worked on microgrid and intelligent control. He is currently a Lecturer with the University of Electronic Science and Technology of China, Chengdu.

BIPLOB RAY (Member, IEEE) received the Ph.D. degree from Deakin University, Australia, in 2015. He is currently a Senior Lecturer in ICT with Central Queensland University, with a background mix of research, academic, and industry experience. He has more than 35 international journal and conference publications. He has worked/working with the Australia Federal Government and industry-funded research projects. His high-quality research work has been recognized by peers and cited extensively. He received the Vice-Chancellor's Award for Outstanding Research in 2019. He has been serving as a keynote/plenary speaker, the organizing chair, a PC member, and a Reviewer for several conferences, since 2012. He served as a guest editor, an editorial board member, and a Reviewer for reputable journals.

KAI CHEN (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2008, 2011, and 2015, respectively. He is currently an Associate Professor with the School of Automation Engineering, University of Electronic Science and Technology of China. His research interests include renewable energy, precision measurement, and solar photovoltaic power generation.

YUHUA CHENG (Senior Member, IEEE) received the Ph.D. degree in measurement techniques and automation devices from Sichuan University, Chengdu, China, in 2007. He is currently a Professor and the Dean of the School of Automation Engineering, University of Electronic Science and Technology, Chengdu. His current research interests include intelligent control, structural health monitoring, renewable energies, and application of power electronics.