Adaptive Learning Hybrid Model for Solar Intensity Forecasting

Yu Wang, *Member, IEEE*, Yinxing Shen, Shiwen Mao, *Senior Member, IEEE*, Guanqun Cao, and Robert M. Nelms, *Fellow, IEEE*

***Abstract*—Energy management is indispensable in the smart grid, which integrates more renewable energy resources, such as solar and wind. Because of the intermittent power generation from these resources, precise power forecasting has become crucial to achieve efficient energy management. In this paper, we propose a novel adaptive learning hybrid model (ALHM) for precise solar intensity forecasting based on meteorological data. We first present a time-varying multiple linear model (TMLM) to capture the linear and dynamic property of the data. We then construct simultaneous confidence bands for variable selection. Next, we apply the genetic algorithm back propagation neural network (GABP) to learn the nonlinear relationships in the data. We further propose ALHM by integrating TMLM, GABP, and the adaptive learning online hybrid algorithm. The proposed ALHM captures the linear, temporal, and nonlinear relationships in the data, and keeps improving the predicting performance adaptively online as more data are collected. Simulation results show that ALHM outperforms several benchmarks in both short-term and long-term solar intensity forecasting.**

***Index Terms*—Artificial neural network (ANN), genetic algorithm back propagation neural network, local linear estimation (LLE), online adaptive learning, solar intensity forecasting.**

NOMENCLATURE

Notation

Symbol	Description
Y	solar intensity
X	meteorological variables
X_p	the pth meteorological variables
$\vec{X}$	vector of meteorological variables
β_p	coefficient of the pth meteorological variable
$\vec{\beta}$	vector of meteorological variable coefficients
ϵ_l	error of linear model
$\vec{X}(t)$	vector of meteorological variables at time t
$\vec{\beta}(t)$	vector of meteorological variable coefficients at time t
$Y(t)$	the actual value of solar intensity at time t
$\hat{Y}(t)$	the predicted solar intensity at time t
$\tilde{Y}(t)$	corrected predicted solar intensity at time t
$\epsilon_l(t)$	the error of linear model at time t
$\hat{\vec{\beta}}(t)$	estimated value of $\vec{\beta}(t)$
$\epsilon_h(t)$	the error of hybrid model at time t
$E(\cdot)$	nonlinear part of prediction model
$\vec{\beta}'(t)$	derivative of $\vec{\beta}(t)$
h	bandwidth for LLE
$\hat{h}$	estimated value of h
$K(\cdot)$	kernel function
n	number of total observations
$\hat{\vec{\beta}}_h(t)$	estimated value of $\vec{\beta}(t)$ under the bandwidth$h$
$\hat{\vec{\beta}}'_h(t)$	derivative of $\hat{\vec{\beta}}_h(t)$
$\Lambda(t)$	covariance function
$([\vec{X}], [Y])$	training data set
ω	weights of GABP training by $[\vec{X}_o]$
θ	thresholds of GABP
e	mean square error of GABP

Manuscript received August 31, 2017; revised November 28, 2017; accepted December 17, 2017. Date of publication January 3, 2018; date of current version April 3, 2018. This work was supported in part by the National Science Foundation (NSF) of China under Grant 51607087, in part by the NSF under Grant DMS-1736470 and Grant CNS-1702957, and in part by the Wireless Engineering Research and Education Center at Auburn University, Auburn, AL, USA. Paper no. TII-17-2042. *(Corresponding author: Yu Wang.)*

Y. Wang and Y. Shen are with Department of Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210007, China (e-mail: yuwang15@nuaa.edu.cn; xubanqiu@163.com).

S. Mao and R. M. Nelms are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: smao@ieee.org; nelmsrm@auburn.edu).

G. Cao is with the Department of Mathematics and Statistics, Auburn University, Auburn, AL 36849, USA (e-mail: gzc0009@auburn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2789289

I. INTRODUCTION

IN RECENT years, smart grid (SG) has become an irreversible tendency in many countries all over the world. The advanced techniques from many fields, including industrial informatics, power electronics, and automatic control make SG a sustainable power grid, which integrates more renewable energy sources, such as solar and wind [1], [2]. Because of the intermittency of renewable power generation, energy management is thus very important to improve the reliability, efficiency, and utility of a SG [3]–[5]. It is mentioned in [5] that energy management efficiency can be greatly improved if the renewable energy generation can be predicted more accurately [6]–[8]. Thus, predicting renewable energy generation in the SG has attracted great interests [9], [10], mainly focusing on predicting solar power for their wide range of utilization.

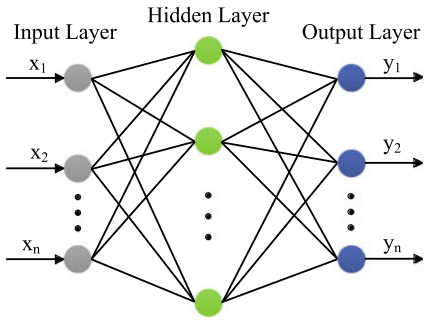

Fig. 1. Structure of a typical three-layer BPNN.

Solar power generation from solar panels are proportional to solar intensity, power generated per unit area. Therefore, predicted solar power can be acquired by predicting solar intensity, which is related to meteorological variables. Many recent works focus on the meteorological-data-based solar intensity forecasting problem by presenting different methods [2], [11]. Sharma *et al.* [2] provides acceptable predicting results using SVM regression, and Silva *et al.* [11] proposes the Hybrid Fuzzy Inference System algorithm as solar intensity forecast mechanism but it lacks a deep analysis of the solar power generation and weather data.

Learning techniques are also used to predict solar intensity, capturing the relationships between solar intensity and the meteorological variables. Artificial neural network (ANN) [12]–[14] is also a commonly used learning algorithm for complex function approximation, and it has a group of members, including the radial basis function neural network (RBFNN), back propagation neural network (BPNN), and the self-organizing neural network (SONN).

A typical three-layer BPNN is shown in Fig. 1, which is composed of one input layer, one hidden layer, and one output layer. BPNN can capture the relationships quickly, which makes it a better choice over many other ANNs, such as RBFNN with slower action. Besides, the computing loads can be reduced through optimization [15]. This allows it to work with a simpler structure comparing to other ANNs using deep learning, such as SONN. So BPNN can theoretically approximate any function at arbitrary precision [16] under a relatively simple mechanism. However, it is easy for BPNN to fall into the local minimum problem during training process. Fortunately, genetic algorithm (GA) provides a suitable way to solve the problem, which forms GA back propagation neural network (GABPNN, abbreviated as GABP). Although GABP is famous for function approximation, the relationship between solar intensity and meteorological variables is too complicated to capture all the linear, nonlinear, even temporal relations. Therefore, if the linear and temporal factors can be departed from the data, GABP can focus on the remaining nonlinear relationship, and its performance could be further improved as well. In this case, a basic three-layer structure is preferred for faster training and it avoids the complicated process of searching for a suitable structure of the GABP.

On the other hand, because of the highly complicated relationship between solar intensity and meteorological variables, it is not possible to capture all the linear and nonlinear relations based on limited amount of data for any method. Therefore, a model capable of online adaptive learning would be highly desirable in predicting solar intensity as more data are collected. Motivated by this, we start from the basic MLR, because it shows some linearity between solar intensity and meteorological variables. We then present a hybrid forecasting model integrating a time-series local linear model and a three-layer GABP, capturing the linearity, temporal, and nonlinear nature of the data, respectively. Based on this hybrid model, we further propose an innovative ALHM, which performs variable selections, and learns adaptively from the new data and, thus, increases the predicting accuracy to a very high level.

The main contribution of this paper is the proposal of the ALHM for meteorological-data-based solar intensity forecasting. First, it is based on the integration of the time-varying multiple linear model and a simple structure GABP, which dig out useful information inside the meteorological data and filter out the redundant data. The time-varying multiple linear model captures the linear relationships and time-varying features, and the three-layer GABP learns the nonlinear relationships in the data with faster training and searching. These two methods are good complement to each other to guarantee satisfactory predictions. Also, it is capable of online and adaptive learning which improves the predicting performance. This superior quality makes it possible to provide more accurate predictions as more data collected, even the initial training data size is limited. Furthermore, our model can also be adaptive to other data-based forecasting problem, which is not restricted to the place and time scale, such as predicting wind power, power grid load, traffic volumes, and stock prices.

The remainder of this paper is organized as follows. We present statistical formulation and several forecasting models in Section II. We propose the adaptive learning online hybrid algorithm in Section III. Performance evaluation is presented in Section IV. Section V concludes this paper.

II. Model Construction

In this section, we first introduce the basic linear regression model for predictions from the original statistical formulation and the general time-varying multiple linear model and learning model are then derived from this, which can improve forecasting precision from the temporal and nonlinear properties of the data, respectively. Based on these models, we finally propose the ALHM. Construction of ALHM is shown in Fig. 2.

A. Statistical Formulation

In the meteorological-data-based solar intensity forecasting problem, solar intensity is considered to be connected to several meteorological variables such as temperature, humidity, and precipitation. However, the connection appears to be very complicated in most weather conditions. Therefore, the problem is normally formulated statistically as follows:

$$Y|\vec{X} \sim P\left(\cdot, f\left(\vec{X}\right)\right) \tag{1}$$

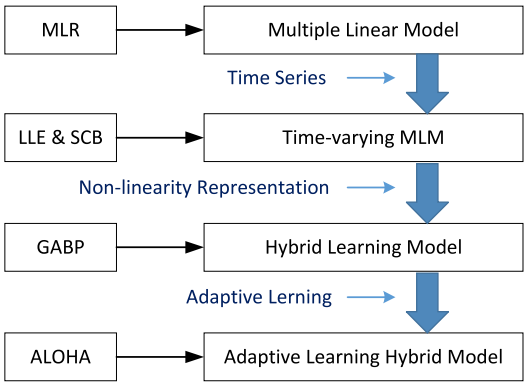

Fig. 2. Construction of the adaptive learning hybrid model (ALHM).

where $P(\cdot,\theta)$ represents a stochastic increasing family of function with parameter θ as the covariate of $\vec{X}$, and $f(\cdot)$ is an unknown smooth function. Unfortunately, the analytic expression of $f(\cdot)$ is too complex to acquire in the solar intensity forecasting problem, where solar intensity Y and meteorological variables $\vec{X}$ are linked together through $f(\cdot)$, written as $Y = f(\vec{X})$. Therefore, an alternative option is to explore suitable models which approximate $f(\cdot)$ as much as possible, given a dataset including meteorological variables and solar intensity.

B. Multiple Linear Model

In traditional statistical analysis, the MLR model is a fundamental method to quickly generate a linear function between variables and response. It is quite effective in representing a linear relationship, but is not possible to represent any nonlinearity. Because it shows fairly strong linearity between solar intensity and some meteorological variables, the overall performance of MLR in predicting solar intensity is acceptable and even good for some sunny days. The MLR model is written as follows:

$$Y = \vec{X}^T \vec{\beta} + \epsilon_l \tag{2}$$

where $\vec{X} = (1, X_1, \ldots X_{p-1})^T$ and $\vec{\beta} = (\beta_0, \ldots \beta_{p-1})^T$ are both $p \times 1$ vectors, Y represents the solar intensity, $X_1, \ldots, X_{p-1}$ are meteorological variables, $\vec{\beta}$ are coefficients, and ϵ_l is error term in the linear model.

C. Time-Varying Multiple Linear Model

MLR is constructed using the spatial linearity between solar intensity and weather variables, however, the meteorological data are often recorded in time series and it usually shows a strong correlation between two adjacent data points in a time series. For example, the solar radiation at noon is closely related to that at 11:00 in the same day. This relation grows stronger as time intervals get shorter. Therefore, processing the weather data as time series accords with the natural generations. Based on MLM in (2), we can write the time-varying MLM model in time series as follows:

$$Y(t) = \vec{X}^T(t)\vec{\beta}(t) + \epsilon_l(t), t \in R \tag{3}$$

where solar intensity Y, meteorological variables $\vec{X}$, coefficients $\vec{\beta}$, and the error ϵ_l in (2) are transformed to continuous time series by adding time indicator (t). It contains both spatial and temporal features. Based on this, we further propose the local linear model to acquire the estimated coefficients $\hat{\vec{\beta}}(t)$ and make variable selections in Section III.

D. Hybrid Learning Model

Although the time series has been added to the MLR model, the nonlinear relationships in the weather data are not reflected well. To further improve the accuracy, a new function $E(\cdot)$ is added to (3) to construct a new model as

$$Y(t) = \vec{X}^T(t)\vec{\beta}(t) + E\left(\vec{X}^T(t), \vec{X}^T(t)\vec{\beta}(t)\right) + \epsilon_h(t) \tag{4}$$

where $t \in R$ and $E(\vec{X}^T(t), \vec{X}^T(t)\vec{\beta}(t))$ represents part of the modeling error $\epsilon_l(t)$ from (3), which can be revealed by some learning model, and the remaining error in the model is $\epsilon_h(t)$, which is expected to be less than $\epsilon_l(t)$ in (3). $\vec{X}^T(t)\vec{\beta}(t)$ here is the predicted solar intensity from the previous time-varying linear model. $E(\cdot)$ is a function of $\vec{X}^T(t)$ and $\vec{X}^T(t)\vec{\beta}(t)$, which means the model learns the nonlinear relationship from both weather variables and the predicted solar intensity. As introduced in Section I, GABP is a good option to get a precise $E(\cdot)$. We present the utilization of GABP in Section III-C. Although GABP is able to reveal any function in any dataset, it becomes more difficult to reach the real function when more data and relationships are included, because the structure of GABP becomes more complicated. By separating the nonlinear part of the error, it not only reduces the complexity of the data, but also takes full advantage of the linear relationship, which can be strong sometimes in the meteorological data (see Section IV). Therefore, $E(\cdot)$ here is taken as the error caused by nonlinear relationship.

E. Adaptive Learning Hybrid Model

In the data-based solar intensity forecasting problem, we have built the model exploring different characteristics of the data, including linearity, time series, and nonlinearity. It is also very important for the model to keep learning online as more data collected, so that the model becomes more and more accurate. The predicting model should also be adaptive for different places and different time scales, because different places have different climates with different related function between weather variables and solar intensity and in different applications, it requires predictions on multiple time scales from minutes to day ahead. For short-time predictions, the data size needs to be controlled as well to guarantee the computing time shorter than the time interval. With the ability of adaptive online learning, it is possible to reduce the forecasting errors gradually in any type of data for different time scales. To capture the above characteristics, we thus propose the adaptive learning online hybrid algorithm in Section III-D to realize online adaptive learning.

III. Adaptive Learning Online Hybrid Algorithm

In this section, we present the forecasting algorithms solving the functions in the models stated above. We begin from

proposing local linear estimation (LLE) followed by simultaneous confidence bands (SCB) construction and variable selection. Then, we present GABP and propose the adaptive learning online hybrid algorithm.

A. Local Linear Estimation

Because MLR is a well-developed model, we begin from proposing the LLE to solve the time-varying coefficients $\vec{\beta}(t)$ in the time-varying MLM.

It is noted that in (3), $\vec{\beta}(t)$ is a continuous-time function. Therefore, for t_i close to t, we have $\vec{\beta}(t_i) = \vec{\beta}(t) + (t_i - t)\vec{\beta}'(t)$, and thus for any time t_i close to t , we can rewrite (3) to have the local linear model as [17]

$$Y(t_i) = \vec{X}^T(t_i)\left(\vec{\beta}(t) + (t_i - t)\vec{\beta}'(t)\right) + \epsilon(t_i), t_i \in t \pm h \tag{5}$$

where the bandwidth h is the size of the local neighborhood. This model divides the time series into periods and creates linear models using local data. This way, we treat the data as a continuous-time series, and exploit the strong correlations between close time periods in weather data. Then, we present the least squares method for linear regression to identify the time-varying coefficients $\vec{\beta}(t)$. Because a closer neighbor would have a stronger effect, while a further neighbor weaker, we need to add weights on the terms. Usually a kernel function $K(\cdot)$ is assigned to each point, which is a symmetric density function defined on $[-1, 1]$ [17]. Here, we use a popular Epanechnikov kernel

$$K(a) = \begin{cases} 3(1-a^2)/4, & \text{if } |a| \le 1 \\ 0, & \text{if } |a| > 1 \end{cases}$$

which decays fast for remote data point. Then, we have the following weighted least squares problem to solve:

$$\min : \sum_{t_i \in t \pm h} \left(Y(t_i) - \vec{X}^T(t_i)\left(\vec{\beta}(t) - (t_i - t)\vec{\beta}'(t)\right)\right) K\left(\frac{t_i - t}{h}\right). \tag{6}$$

At each time t, we solve for coefficients $\hat{\vec{\beta}}_h(t)$ and $\hat{\vec{\beta}}'_h(t)$ under the bandwidth h. Suppose the total number of observations is n, we can pick t_i simply as $t_i = i/n$, $1 \le i \le n$, and denote $Y(t_i)$ as y_i and $\vec{X}(t_i)$ as $\vec{x}_i$. From [18], we can solve (6) by calculating the following matrices $\mathbf{S}_k(t)$ and $\mathbf{R}_k(t)$:

$$\mathbf{S}_k(t) = \sum_{i=1}^{n} \vec{x}_i \vec{x}_i^T \left(\frac{t_i - t}{h}\right)^k K\left(\frac{t_i - t}{h}\right)/(nh) \tag{7}$$

$$\mathbf{R}_k(t) = \sum_{i=1}^{n} \vec{x}_i y_i \left(\frac{t_i - t}{h}\right)^k K\left(\frac{t_i - t}{h}\right)/(nh) \tag{8}$$

where $k = 0, 1, 2, \ldots$. We, then, have

$$\begin{pmatrix} \hat{\vec{\beta}}_h(t) \\ h\hat{\vec{\beta}}'_h(t) \end{pmatrix} = \begin{pmatrix} \mathbf{S}_0(t) & \mathbf{S}_1^T(t) \\ \mathbf{S}_1(t) & \mathbf{S}_2(t) \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{R}_0(t) \\ \mathbf{R}_1(t) \end{pmatrix}. \tag{9}$$

To solve problem (6) for the complete model using (7) to (9), we need to fix bandwidth h and h is the bandwidth which determines the size of data used to estimate for a local linear model at time t. If h is too small, many useful points are not included and the relevant information would miss which may cause a huge error. While, if it is too large, more remote points are included and much unnecessary information would follow, which increases the computation complexity and reduce the smoothness of the model at the same time. Therefore, it is important to choose a proper h.

For constant bandwidth selection considered in the model, we adopt the generalized cross-validation (GCV) technique [19]. Similar to the coefficients $\vec{\beta}$ estimated from the observed data $[\vec{X}]$ and $[Y]$. Thus, a square hat matrix $\mathbf{H}(h)$ exist for $\hat{\vec{Y}} = \mathbf{H}(h)\vec{Y}$ [20], depending on the bandwidth. Then, we can choose the bandwidth h by

$$\hat{h} = \arg\min \left\{ \frac{\left|\hat{\vec{Y}} - \vec{Y}\right|^2}{n(1 - \text{tr}\{\mathbf{H}(h)\}/n)^2} \right\} \tag{10}$$

where tr$(\cdot)$ is the trace of the matrix.

Therefore, we can get the estimated $\hat{Y}(t)$ as

$$\hat{Y}(t_i) = \vec{X}^T(t_i)\left(\hat{\vec{\beta}}_{\hat{h}}(t) + (t_i - t)\hat{\vec{\beta}}'_{\hat{h}}(t)\right), t_i \in t \pm \hat{h}. \tag{11}$$

B. SCB and Variable Selection

In linear regression, confidence intervals indicate the possible coverage of the coefficient $\hat{\beta}$ under a typical probability. Similarly, we can construct the coverage for the time-varying coefficients $\hat{\beta}(t)$, which turns to be a band through time, the SCB. The mathematical details to construct SCB is shown in the Appendix.

The SCB provides a dynamic and comprehensive view on $\vec{\beta}(t)$. In a linear regression, the confidence interval provides a measure of the overall quality of the regression line [20]. In local linear model, the SCB illustrates the overall pattern of $\vec{\beta}(t)$ and the accuracy of the model. Confidence bands with smaller width imply a better model with smaller variability, while very wide confidence bands are limited in using. Note that the SCB is constructed under a complete analysis on the continuous-time assumption, which is not merely the connections of the pointwise confidence intervals. More importantly, the SCB indicate whether the coefficients $\vec{\beta}(t)$ are truly time-varying or not. Specifically, if a horizontal line is covered by the SCB of a $\beta_k(t)$, we accept the hypothesis that $\beta_k(t)$ is constant and not time-varying. This provides a good way to select time-varying variables. Normally, the variables with narrow SCBs, which does not cover any horizontal line, are preferred and kept in the model. The selected variables are then used in (5) to make predictions.

C. GA Back Propagation Neural Network

It is discussed in Section II-D that $E(\cdot)$ is required to compensate the error capturing the nonlinear relations between the previous predicted errors and the meteorological data. Here, we apply GABP to acquire a satisfactory $E(\cdot)$.

As stated in Section I, BPNN can theoretically approximate any nonlinear function at arbitrary precision under a three-layer structure. The major drawback of BPNN is that it sometimes ends in local optimum during iterative calculations. The GA is then used to solve this local minimum problem. GA is a parallel stochastic searching method, initially proposed to simulate the natural genetic mechanism in biological evolution theory, can reach the optimal value quickly and precisely. It is thus a good complement to BPNN. By optimizing the initial weights and thresholds of BPNN through GA, it forms the algorithm of GABP [21].

Considering both the nonlinear representation and the computation speed, we choose three-layer GABP with single hidden layer as our learning model. Given training data, GABP learns the function $E(\cdot)$ in (4), and corrects the predicted solar intensity to be $\tilde{Y}(t)$ by

$$\tilde{Y}(t) = \hat{Y}(t) + E\left(\vec{X}(t), \hat{Y}(t)\right) \tag{12}$$

where $\hat{Y}(t)$ is predicted solar intensity from time-varying multiple linear model (TMLM).

D. Adaptive Learning Online Hybrid Algorithm

Based on (12), we update the predicted solar intensity to $\tilde{Y}(t)$ from $\hat{Y}(t)$. However, it cannot guarantee the predicting errors to reduce for new data. Therefore, we further propose the adaptive learning online hybrid algorithm (ALOHA) to improve the forecasting accuracy as more data collected.

For a group of new data at time t, the predictions are made using (11) and (12) for HLM and TMLM, respectively, we can write the predicting errors from these two models as

$$\epsilon_l(t) = \left|\hat{Y}(t) - Y(t)\right| \tag{13}$$

$$\epsilon_h(t) = \left|\tilde{Y}(t) - Y(t)\right| \tag{14}$$

where $\hat{Y}(t)$ and $\epsilon_l(t)$ are the predicted solar intensity and predicting error in TMLM, respectively, and $\tilde{Y}(t)$ and ϵ_h are those in HLM, respectively. The effect of GABP can be demonstrated by comparing $\epsilon_h(t)$ with $\epsilon_l(t)$. If $\epsilon_h(t) < \epsilon_l(t)$, it means that the trained GABP successfully reduces the predicted error, and thus, GABP does not need to learn this new data further; if $\epsilon_h(t) \geq \epsilon_l(t)$, it means that the current GABP fails to reveal the nonlinearity in the new data and even causes larger predicting error and, thus, the data are added to the previous dataset to train GABP again. This allows GABP to make better predictions next time when similar data are encountered in future. In this way, the predicting performance of the model improves gradually as more data collected.

Besides learning from new data, it is also important to avoid the new data from lowering the overall predicting performance.It is thus necessary to filter out the new data which may cause worse overall predicting performance. The mean square error (MSE) is a good way to quantify the overall predicting performance of a model, defined as

$$e = \frac{1}{n}\sum_{i=1}^{n}\left(\epsilon_l(i) - \hat{\epsilon_l}(i)\right)^2 \tag{15}$$

where ϵ_l is the actual error of TMLM, $\hat{\epsilon_l}$ is the error from GABP, and n is number of the whole historical training data.

Denote the initial training data as $([\vec{X}], [Y])_o$, and record the trained GABP's weights as ω_o, thresholds as θ_o, and the MSE of the predictions as e_o on $([\vec{X}], [Y])_o$. For a new data, if $\epsilon_h(t) \geq \epsilon_l(t)$, it is added to $([\vec{X}], [Y])_o$ forming a new training data, denoted as $([\vec{X}], [Y])_n$, and the original ω_o and thresholds θ_o are taken as the initials of the new training. New GABP's weights ω_n, thresholds θ_n, and MSE e_n of the predictions on $([\vec{X}], [Y])_n$ are obtained after the training. e_n is then compared to the old one e_o. If $e_n \geq e_o$, it means the MSE of predictions increases and the overall predicting performance decreases. Thus, the new data are filtered out, and the previous $([\vec{X}], [Y])_o$ remains as the historical training data, and ω_o, θ_o, and e_o remain the same. If $e_n < e_o$, it means that with new data added, the MSE of predictions decreases, and the overall predicting performance improves. Therefore, the historical training data are updated as $([\vec{X}], [Y])_n$, and GABP is updated accordingly to be ω_n and θ_n, and the new MSE is e_n. For every new data, follow the above steps to guarantee the overall predicting accuracy increases until the limit of the model, where GABP digs out all the possible nonlinearity in the meteorological data.

By establishing the above two restrictions, it selects only beneficial data, so that the GABP evolves to be better and better. The steps of ALOHA are described in Algorithm 1, and the

Algorithm 1: The Adaptive Learning Online Hybrid Algorithm for Solar Intensity Predictions.

1. Based on the historical data $([\vec{X}], [Y])_o$, choose proper bandwidth $\hat{h}$ from (10) and get the coefficients $\hat{\vec{\beta}}(t)$ from (9) to construct the TMLM by (11);
2. Use $([\vec{X}], [Y])_o$ to train GABP, obtain HLM by (12) and record its weights ω_o and the thresholds θ_o. Find the predicting MSE e_o using (15);
3. For a new group of meteorological data $\vec{X}(t)$ at time t, get the predicted solar intensity $\hat{Y}(t)$ from (11), and then acquire the corrected $\tilde{Y}(t)$ from (12);
4. Obtain the errors from two models $\epsilon_l(t)$ and $\epsilon_h(t)$ from (13) and (14). If $\epsilon_h(t) > \epsilon_l(t)$, go to the next step; else jump to Step 8;
5. Use ω_o and θ_o as initials to train GABP again on the data $([\vec{X}], [Y])_n$ by adding $\vec{X}(t)$, $Y(t)$ to $([\vec{X}], [Y])_o$;
6. Obtain the new predicting MSE e_n. If $e_n < e_o$, go to the next step; else jump to Step 8;
7. Update $([\vec{X}], [Y])_o$, ω_o, θ_o and e_o with $([\vec{X}], [Y])_n$, ω_n, θ_n and e_n respectively;
8. Keep $([\vec{X}], [Y])_o$, ω_o, θ_o and e_o unchanged, and repeat from Step 3 until all new data are processed.

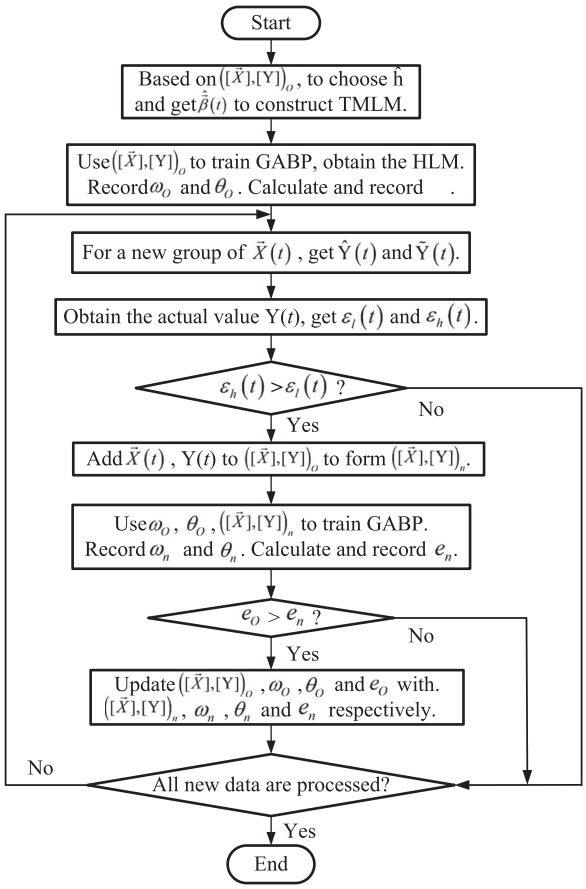

Fig. 3. Flowchart of ALOHA.

flowchart is depicted in Fig. 3. Through the combination of MLR, LLE, and GABP, ALOHA first generates acceptable predictions on solar intensity given any historical data in any time scales. Then, as more data collected, ALOHA is able to make more precise predictions with the help of the online adaptive GABP learning, and the more data it learns, the better ALOHA predicts on the solar intensity. It is thus very desirable for predicting power generation from solar energy and many other renewable energy resources.

IV. Performance Evaluation

In this section, we present the trace-driven simulation results from different models and algorithms proposed in previous sections, and compare the performances of them for meteorological-data-based solar intensity forecasting. Then, we compare performances between the model ALHM and two benchmarks for different time scales.

A. Data Description

We apply the proposed methods using the real data from the UMASS Trace Repository [22], which records the solar intensity in watts/m^2, and the data of several meteorological metrics from January 2015 to February 2017. It recorded the meteorological data every 5 min. Many meteorological variables were observed in details. We consider five main variables, including temperature, humidity, dew point, wind speed, and precipitation. We predict daily solar intensity averaged on the day time when the solar intensity is nonzero, based on the averaged meteorological variables. It is note that the model we propose can be applied to any time scale, our major results are based on daily predictions, because daily power generation forecasting is necessary for power management, especially for some power grids with a large part of power generation from renewable energy resources. We also evaluate our model under the shortest time scale of 5 min, compared to two other benchmarks in Section IV-E.

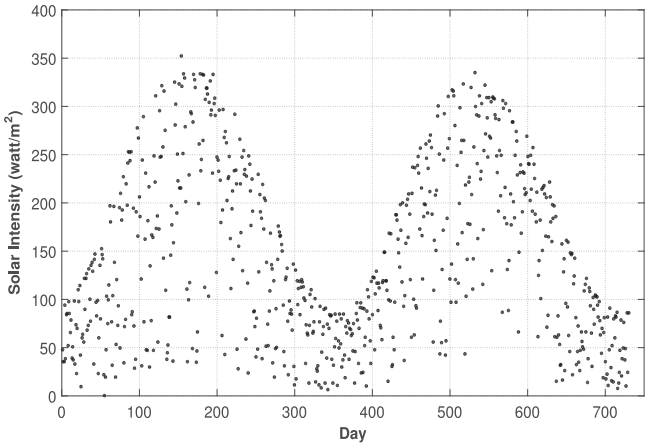

Fig. 4. Daily solar intensity for 2015 and 2016.

The data are departed into two parts, one of which is used as the initial training data to build forecasting models, and the other is used to test the forecasting accuracy. Here, we pick the data of two years from January, 2015 to December, 2016 as the training data, and the data of January and February in 2017 as the test data. The daily solar intensity from 2015 to 2016 is plotted in Fig. 4, which shows an apparent seasonal pattern. It is also interesting to see a similar pattern for daily observations, and similar patterns can also be found for other meteorological variables. Fig. 4 also shows a strong correlation between two consecutive days, which offers a basis for applying LLE in building TMLM.

The proposed models are constructed based on the data, and the performances of them on predicting daily solar intensity are compared. Then, we compare the outstanding ALHM on two benchmarks for both daily and 5-min predictions.

B. Predicting Performance of TMLM

1) TMLM construction: We first write the TMLM model

$$\hat{y}_i = \hat{\beta}_0(i/n) + \sum_{p=1}^{5} \hat{\beta}_p(i/n) x_{p,i}, \text{ for } i = 1, \ldots, n \qquad (16)$$

where $\hat{y}_i$ is the predicted solar intensity, $x_{p,i}, p = 1, \ldots, 5$ represent the series of temperature in Fahrenheit, humidity in percentage, dew point in Fahrenheit, wind speed in miles per hour, and precipitation in inches respectively, and $n = 731$ in the training data; $\hat{\beta}_0(\cdot)$ is the intercept, and $\hat{\beta}_p(\cdot)$ are the associated coefficients for $x_{p,i}$. The meteorological variables are centered on their averages so that the intercept $\hat{\beta}_0(\cdot)$ can be interpreted

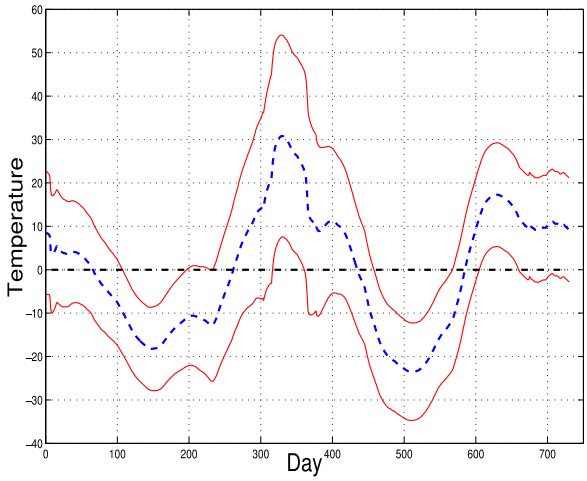

Fig. 5. SCB of 95% for temperature.

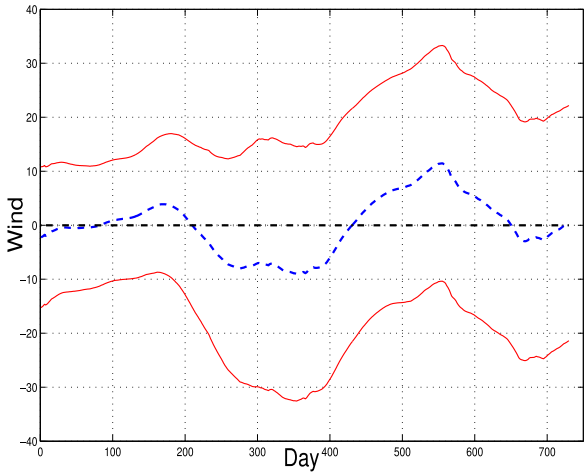

Fig. 6. 95% SCB for Wind.

as the expected solar intensity and we quantify the forecasting accuracy of a model by mean absolute percentage error (MAPE) and root mean square error (RMSE):

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i} \times 100\% \tag{17}$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{18}$$

where y_i is the real solar intensity at i.

2) Variable selection by SCB: According to Algorithm 2 presented in Appendix, we select the bandwidth $h = 0.25$ and get the coefficients $\beta_p(\cdot)$. We show the SCB for temperature, wind, and precipitation in Figs. 5 to 7. Other SCBs are similar to that for temperature, and thus not depicted here. In each figure, the middle thick solid curve is the estimated series for the variable; the upper and lower solid curves are the envelops for the SCB for each variable. Based on the SCBs, we can test whether a coefficients is significantly associated with the solar

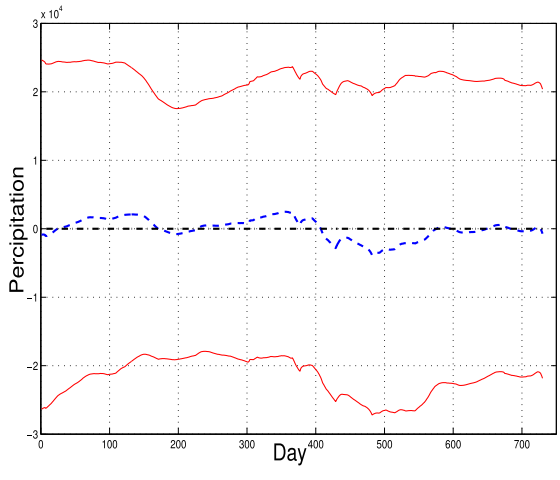

Fig. 7. 95% SCB for Precipitation.

intensity, which equals to test

$$H_0 : \beta_p(t) = 0 \quad \forall t \in [0,1]; \text{v.s. } H_1 : \beta_p(t) \neq 0, \exists t \in [0,1].$$

If the zero line is included in the SCB, we accept the hypothesis that the coefficient is not significant and could be omitted from the model; otherwise, we keep it in the model. We can also test whether the coefficients are constant, by attempting to include a constant horizontal line into the SCB. This is equal to testing

$$H_0 : \beta_p(t) = c_p \quad \forall t \in [0,1]; \text{v.s. } H_1 : \beta_p(t) \neq c_p, \exists t \in [0,1]$$

where c_p is a constant of each p. If the line is covered, we accept that the coefficient is constant; otherwise, it is not. As we center all the weather variables on their averages, the SCB of the $\beta_p(t)$ actually indicates the effect on the solar intensity. In Fig. 5, the zero line is not covered, while in Figs. 6 and 7, the zero line is covered by the 95% SCB. Therefore, we can conclude that for a level of 95%, temperature, humidity, and dew point have a strong effect on solar intensity, but the effect from wind and precipitation are weak. Also, we accept $\beta_1(t)$ to $\beta_4(t)$ as time-varying coefficients, because a constant horizontal line cannot be covered entirely in those SCBs. Therefore, we select three meteorological variables of temperature, humidity, and dew point, in the following models.

3) Performance evaluation of TMLM: The updated TMLM of three variables are used to predict the daily solar intensity from January 1 to February 28 in 2017. The predicted solar intensity data points are shown in Fig. 8, comparing to the actual observations. The predicting MAPE is 21.16% and RMSE is 28.65 watts/m^2. This result of TMLM is better than that of MLR with 24.72% and RMSE 33.49 watts/m^2. For the testing data of only 59 points, this overall performance is acceptable, however, some points have relatively large predicting errors, for example, the data from Day 6 to 8, Day 22 to 24, etc. Obviously, predictions from TMLM require further corrections.

C. Predicting Performance of HLM

The HLM model can be rewritten as

$$\tilde{y}_i = \hat{y}_i + E(\vec{x}_i, \hat{y}_i) + \hat{\beta}_0(i/n), \text{ for } i = 1, \ldots, n \tag{19}$$

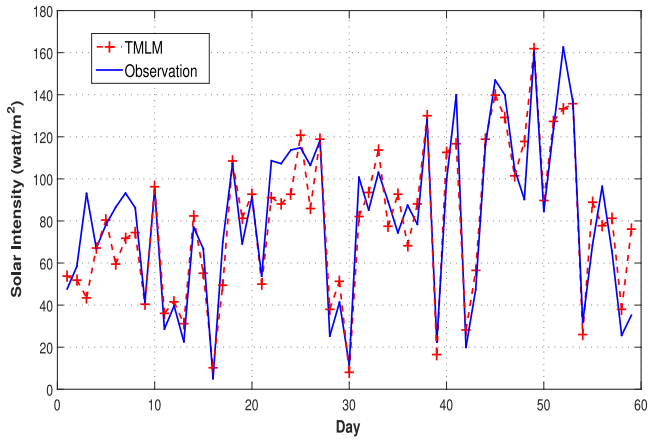

Fig. 8. TMLM-based solar intensity predictions versus observations.

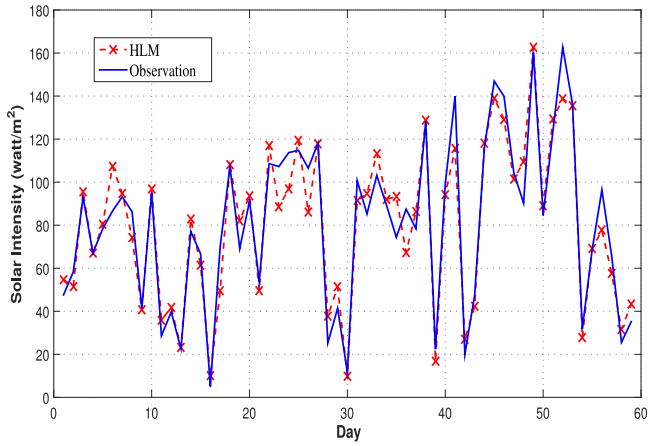

Fig. 9. HLM-based solar intensity predictions versus observations.

where $\tilde{y}_i$ is the corrected solar intensity predictions by HLM, $E(\cdot)$ is acquired from the trained GABP. The predicting results on the testing data are shown in Fig. 9. The MAPE and RMSE are calculated replacing $\hat{y}_i$ by $\tilde{y}_i$ in (17) and (18), respectively. From the figure, it can be seen that the predicting curve tracks the actual solar intensity much better than the previous TMLM model. The MAPE and RMSE of HLM are 14.18% and 20.93 watts/m^2 respectively, which are much smaller than those from TMLM. However, there are still some inaccurate predictions, i.e., Day 6, 23, 26, 35, 36, and 52. This implies GABP does not learn these points well and the model HLM needs further enhancement.

D. Predicting Performance of ALHM

According to Algorithm 1, we now apply the ALOHA to construct ALHM. Fig. 10 shows its predicting results, with the MAPE 13.68% and the RMSE 20.16 watts/m^2. The predicting MAPE and RMSE for the presented models are summarized and compared in Table I in the first four rows. Comparing the results from HLM, solar intensity predictions of four days are corrected, including Day 23, 26, 36, and 52. The predicting accuracy has been improved slightly, because the total testing data contain only 59 days, and 32 of them are already predicted very well.

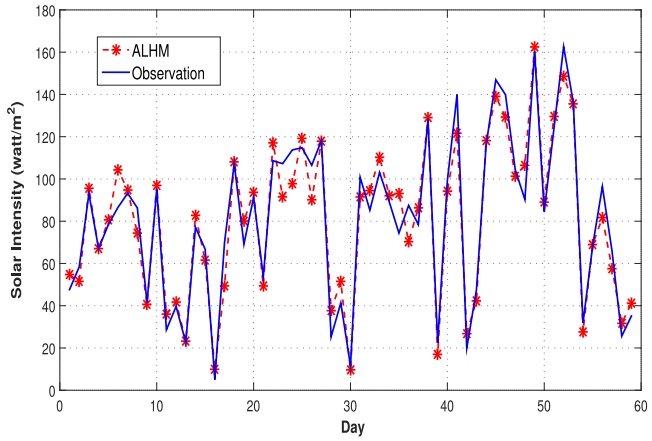

Fig. 10. ALHM-based solar intensity predictions versus observations.

TABLE I
PREDICTING PERFORMANCE ON MAPE (%) AND RMSE (WATTS/M^2) FOR DIFFERENT MODELS FROM JANUARY 1 TO FEBRUARY 28, 2017

Model	Algorithm	MAPE	RMSE
MLM	MLR	24.72	33.49
TMLM	LLE, SCB	21.16	28.65
HLM	GABP	14.18	20.93
ALHM	ALOHA	13.68	16.95

Further, ALOHA is designed to learn from the inaccurate data points which are also helpful in reducing the overall MSE. In this case, there are only three days added to the initial training data, although 24 days are considered as inaccurate points. This is because the ALOHA works very strictly in choosing the most valuable new data. So ALHM is able to improve gradually and adaptively. However, this does not mean the model improves very slowly, because the training process of GABP has some randomness. If the training path fits the new data very well, the model improves fast and greatly. To further evaluate the performance of ALOHA, we redistribute the original data to have fewer training data and more testing data.

We now select the data from January 1, 2015 to August 30, 2016 as the training data, and the left data from September 1, 2016 to February 28, 2017 are used to evaluate predicting performance of ALHM. Here, HLM decides the predicting function based on the training data, while the Updating HLM (UHLM) updates the training data for every new data, based on which the predicting function is renewed every day.

The predicting RMSE of HLM, UHLM, and ALHM are compared month to month in Table II, where the sizes of training data for different models are also listed. It can be seen that HLM has the largest predicting errors because of the fixed fewest training data. UHLM has the best predicting accuracy and the largest training data. ALHM performs very close to UHLM with much less training data. By comparing ALHM to HLM, it confirms the gradual improvement of ALHM in solar intensity predictions. Comparing to UHLM, ALHM successfully holds the most influential data and control the data size very well with comparable

TABLE II
MONTHLY AVERAGE RMSE IN WATTS/M^2 AND THE SIZE OF TRAINING DATA FOR HLM, UHLM, AND ALHM FROM SEPTEMBER 2016 TO FEBRUARY 2017

Month	HLM	Size	UHLM	Size	ALHM	Size
2016.09	28.63	609	27.54	639	28.38	611
2016.10	29.52	609	23.28	666	24.60	613
2016.11	27.86	609	20.46	692	21.33	616
2016.12	28.17	609	17.69	723	18.42	619
2017.01	26.35	609	16.05	754	16.88	621
2017.02	27.29	609	14.13	782	14.72	623

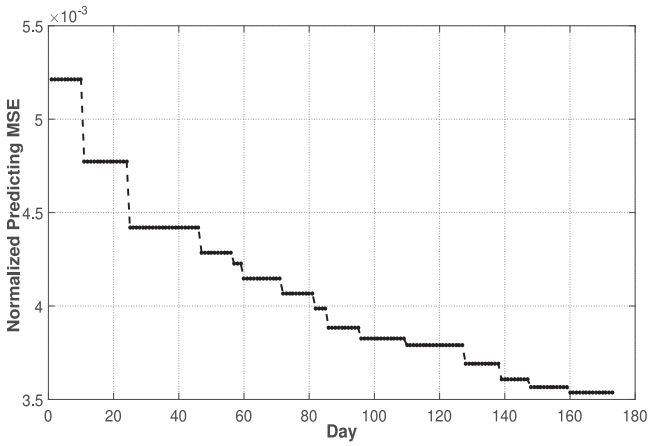

Fig. 11. Overall predicting MSE of ALHM for 173 days.

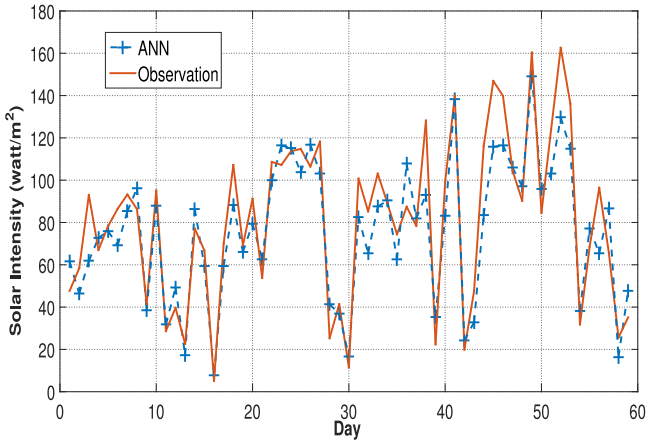

Fig. 12. ANN-based solar intensity predictions versus observations.

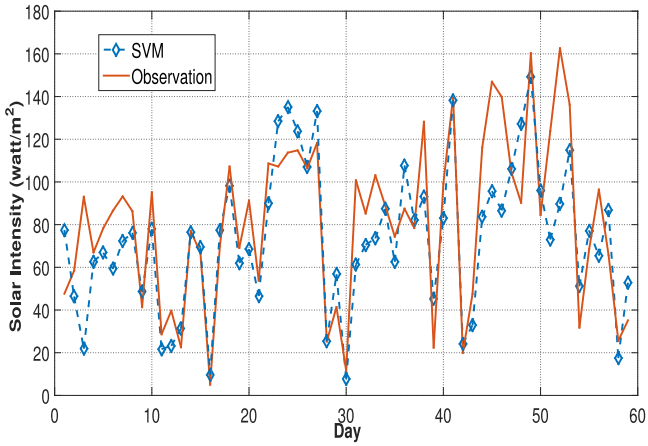

Fig. 13. SVM-based solar intensity predictions versus observations.

predicting accuracy. Comparing to ALHM under the previous training data, the average predicting RMSE from January 1 to February 28, 2017 reduces from 16.95 to 15.89 watts/m^2. The ALHM under new initial training data performs better, learning only 14 new data online, comparing to the previous one of four more months of data offline. This fully confirms the effectiveness of adaptive online learning ability of ALHM.

To further illustrate the learning process of ALHM, we show the decreasing of normalized predicting MSE through the testing data of six months in Fig. 11. In the figure, every step down of the curve indicates a learning happens. It is observed that among the total 173 days, only 14 days of data are allowed to be updated in the new training data, although 67 days of data are considered inaccurate. This shows again the strict filtering standard by ALOHA, and the learning ability of GABP.

E. Comparisons of ALHM to Two Benchmarks

We now compare ALHM to popular supervised learning models used in meteorological-data-based solar intensity forecasting problem, such as ANN [12], [14], and SVM [2]. Based on the same initial training data of 2015 and 2016, the predicting results using ANN and SVM are shown in Figs. 12 and 13, respectively. The MAPE of SVM and ANN are 20.39% and 18.41%, respectively, comparing to 13.68% for ALHM. Based on the same training data, ALHM achieves more precise predictions.

Then, we compare the performance of these models for short-time predictions. For ANN and SVM, we use the historical data from January 1 to December 30, 2016 as training data. For ALHM, we set the historical data from January 1 to June 30, 2016 as initial training data, and the data from July 1 to December 30, 2016 are used to improve the model. The models now predict every 5 min for two days of January 1 to January 2, 2017. The predicted daytime solar intensity of ALHM, ANN, and SVM for the two days are plotted in Figs. 14 and 15, respectively. The solar intensity in January 1 fluctuates more than January 2. Both figures show that ALHM predictions track the actual solar intensity data much better than the other two models. The MAPE for ALHM, ANN, and SVM are 8.66%, 15.24%, and 16.87%, respectively, for January 1, and 11.74%, 18.84%, and 21.05% for January 2.

Note that the predictions by SVM are not satisfactory, because it lacks a deep analysis of the solar power generation and weather data, by simply trying different SVM kernels. ANN tries to capture the relationship between solar intensity and meteorological variables directly. Although this is theoretically available as discussed in Section I, it is very difficult to realize in applications but in ALHM, TMLM focuses on linearity representation, and GABP focuses on learning the nonlinear relationship. They function together to achieve more accurate predic-

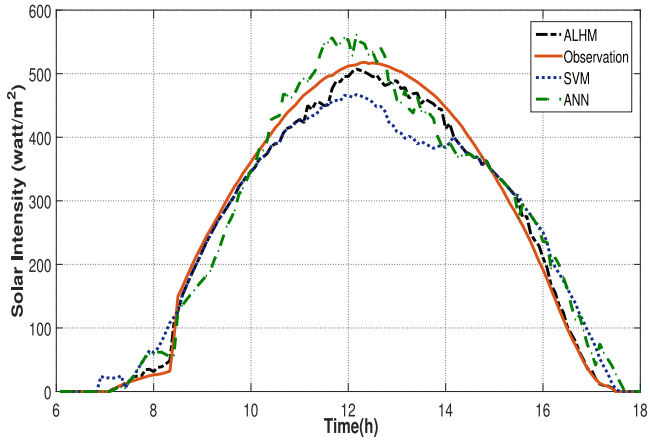

Fig. 14. Five-minute predictions from ALHM, ANN, and SVM versus observations on January 1, 2017.

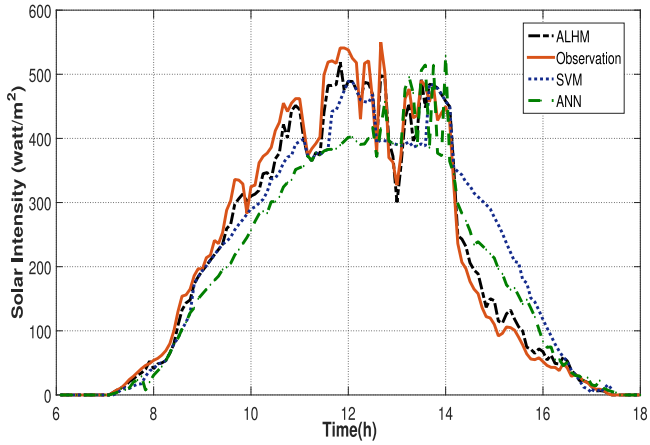

Fig. 15. Five-minute predictions from ALHM, ANN, and SVM versus observations on January 2, 2017.

tions. In conclusion, ALHM outperforms ANN and SVM for both daily and 5 min predictions.

V. Conclusion

In this paper, we developed ALHM for meteorological-data-based solar intensity predictions. We first presented the fundamental multiple linear model. Then, we propose the time-varying multiple linear model and hybrid learning model, which improve the model from the aspects of time series and intelligent learning. We then proposed the ALHM to learn the nonlinear relationship in the data to further improve the predicting accuracy. The proposed models were validated with trace-driven simulations.

Appendix

A. Construction of SCB

The approach of SCB analysis assumes locally stationary processes for both $\vec{X}(t)$ and $\epsilon(t)$ [23]. The locally stationary process guarantees the stationary property for local time series, and is useful for LLE. It actually belongs to a special class of nonstationary time series as

$$\vec{x}_i = \vec{G}(t_i, F_i), \quad \epsilon_i = H(t_i, F_i), \quad i = 1, 2, \ldots, n \tag{20}$$

where $\vec{G}(t_i, F_i)$ and $H(t_i, F_i)$ are measurable functions well defined on $t_i \in [0, 1]$, $F_i = (\ldots, \xi_{i-1}, \xi_i)$ with $\{\xi_i\}_{i\in\mathbb{Z}}$ are independent and identically distributed (i.i.d.) random variables, and $\mathbb{E}(\epsilon_i|\vec{x}_i) = 0$. Equation (20) can be interpreted as physical systems with F_i (and x_i and ϵ_i) as the inputs (and the outputs respectively), and G and H representing the underlying physical mechanism [24].

Based on the above assumptions, the central limit theorem for $\hat{\vec{\beta}}(t)$ states that: supposing $nh \to \infty$ and $nh^7 \to 0$ [24], then for any fixed $t \in (0, 1)$,

$$(nh)^{1/2}\left\{\hat{\vec{\beta}}(t) - \vec{\beta}(t) - h^2\vec{\beta}''(t)\mu/2\right\} \to N\left\{0, \Sigma^2(t)\right\} \tag{21}$$

where

$$\mu = \int_{\mathbb{R}} x^2 K(x)\, dx \tag{22}$$

$$\Sigma(t) = \left(M^{-1}(t)\Lambda(t)M^{-1}(t)\right)^{1/2} \tag{23}$$

$$M(t) = \mathbb{E}\left(\vec{G}(t, F_0)\vec{G}(t, F_0)^T\right). \tag{24}$$

The covariance matrix $\Lambda(t)$ can be further approximated using techniques proposed in (28).

Deriving from the central limit property and basic assumptions shown above, the $100(1-\alpha)\%$ asymptotic simultaneous confidence tube of $\vec{\beta}_C(t)$ can be constructed using

$$\tilde{\beta}_{C,\tilde{h}}(t) + \hat{q}_{1-\alpha}\hat{\Sigma}_C(t)\mathcal{B}_s \tag{25}$$

where $\tilde{\beta}_{C,\tilde{h}}(t)$ is the bias corrected estimator defined in (26), $\mathcal{B}_s = \{\vec{z} \in \mathbb{R}^s : |\vec{z}| \le 1\}$ is the unit ball, and s is the rank of a matrix $C_{p\times s}$, which we use for choosing different linear combinations of $\beta(t)$, and $\vec{\beta}_C(t) = C^T\vec{\beta}(t)$. We can construct the SCB for different linear combinations of $\beta_k(t)$ s by setting a different matrix $C_{p\times s}$. For example, if we set $C_{p\times 1} = [1, 1, 0, \ldots 0]$, we obtain the SCB of $\vec{\beta}_C(t) = C^T\vec{\beta}(t) = \beta_1(t) + \beta_2(t)$; if we set $C_{p\times 2} = [1, 0, 0, \ldots 0; 0, 1, 0, \ldots 0]$, the SCB of $\beta_1(t)$ and $\beta_2(t)$ becomes a tube at any time t. This is because when $s = 2$, the unit ball $\mathcal{B}_2$ becomes a unit circle. Thus, we simply take $s = 1$ in (25), and the SCB is constructed similarly to the confidence interval of the coefficients of the MLR: $\hat{\beta} \pm t_{\alpha/2,n-p}se(\hat{\beta})$, where $\text{se}(\hat{\beta})$ is the standard error of $\hat{\beta}$, and $t_{\alpha/2,n-p}$ is the upper $\alpha/2$ percentage point of the t_{n-2} distribution [20].

Similarly, the first term is the estimator of the time-varying coefficients corrected for bias by

$$\tilde{\beta}_{C,\tilde{h}}(t) = C^T\tilde{\beta}_{\tilde{h}}(t) = C^T\left(2\tilde{\beta}_{\tilde{h}/\sqrt{2}}(t) - \hat{\beta}_{\tilde{h}}(t)\right) \tag{26}$$

$\tilde{\beta}_{\tilde{h}}(t)$ can also be acquired by solving (6) using a corresponding kernel function $K^*(a) = 2\sqrt{2}K(\sqrt{2}a) - K(a)$ and an updated bandwidth $\tilde{h} = 2\hat{h}$ of the GCV selector $\hat{h}$.

The second term in (25) $\hat{q}_{1-\alpha}$ is actually the upper $\alpha/2$ percentage point of the normal distribution $N\{0, \Sigma^2(t)\}$ defined in (21), while the third term $\hat{\Sigma}_C(t)$ is the estimated standard error. The method of wild bootstrap is applied to obtain $\hat{q}_{1-\alpha}$.

Algorithm 2: Construction of SCB for Time-varying Coefficients.

1 Find a proper bandwidth $\hat{h}$ from GCV selector (10);
2 Let $\tilde{h} = 2\hat{h}$ and calculate $\tilde{\beta}_{C,\tilde{h}}(t)$ using (26) and (6);
3 Obtain the estimated $(1-\alpha)$th quantile $\hat{q}_{1-\alpha}$ via the bootstrap method;
4 Estimate $\hat{M}(t) = S_0(t^*)$ and $\hat{\Lambda}(t)$ by (28), and calculate $\hat{\Sigma}_C(t)$ according to (27);
5 Construct the $100(1-\alpha)\%$ SCB of $\vec{\beta}_C(t)$ using (25).

First, generate a large number i.i.d. vectors $\vec{v}_1, \vec{v}_2, \ldots, N(0, I_s)$, where $\vec{v}_i \in \mathbb{R}^p$ and I_s denotes the $s \times s$ identity matrix, and then calculate $q = sup_{0\le t\le 1} |\sum_{i=1}^{n} \vec{v}_i K^*((t_i - t)/\tilde{h})/(n\tilde{h})|$; repeat the previous step for a large number of times (say, 5000) to acquire the estimated $100(1-\alpha)\%$ quantile $\hat{q}_{1-\alpha}$ of q.

The estimate of the standard error $\hat{\Sigma}_C(t)$ is defined as

$$\hat{\Sigma}_C(t) = \left(C^T \hat{M}^{-1}(t)\hat{\Lambda}(t)\hat{M}^{-1}(t)C\right)^{1/2}. \tag{27}$$

We shall estimate $\hat{M}(t)$ and $\hat{\Lambda}(t)$, respectively. From the definition of $M(t)$ in (24), it can be estimated by $\hat{M}(t) = S_0(t^*)$, where $S_0(\cdot)$ is defined in (7), and $t^* = \max\{h, \min(t, 1-h)\}$. To obtain $\hat{\Lambda}(t)$, we first define two $p \times 1$ vectors $\vec{Z}_i = \vec{x}_i \hat{\epsilon}_i$ and $\vec{W}_i = \sum_{j=-m}^{m} \vec{Z}_{i+j}$, a matrix $\Omega_i = \vec{W}_i \vec{W}_i^T/(2m+1)$, and a function $g(t,i) = K((t_i - t)/\tau)/\sum_{k=1}^{n} K(t_k - t)$, where m and τ can be simply chosen as $m = \lfloor n^{2/7} \rfloor$ and $\tau = n^{-1/7}$. Then, $\hat{\Lambda}(t)$ can be calculated by

$$\hat{\Lambda}(t) = \sum_{i=1}^{n} g(t,i)\Omega_i. \tag{28}$$

All the above steps are summarized in Algorithm 2.

REFERENCES

[1] X. Fang, S. Misra, G. Xue, D. Buccella, and C. Yang, "Smart grid - The new and improved power grid: A survey," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 944–980, Dec. 2012.

[2] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *Proc. IEEE int. Conf. Smart Grid Commun.*, Brussels, Belgium, Oct. 2011, pp. 528–533.

[3] Y. Wang, S. Mao, and R. Nelms, "On hierarchical power scheduling for the macrogrid and cooperative microgrids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1574–1584, Dec. 2015.

[4] Y. Wang, S. Mao, and R. Nelms, "Distributed online algorithm for optimal real-time energy distribution in the smart grid," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 70–80, Jun. 2014.

[5] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay, "Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves," *IEEE Power Energy Mag.*, vol. 5, no. 1, pp. 411–419, Sep. 2014.

[6] R. Li, J. Li, and H. Li, "The short-term electric load forecasting grid model based on MDRBR algorithm," in *Proc. 2006 IEEE Power Eng. Soc. General Meeting*, Montreal, Canada, Jun. 2006, pp. 1–6.

[7] A. Tsikalakis and N. Hatziargyriou, "Artificial neural network for load forecasting in smart grid," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Qingdao, China, Jul. 2010, pp. 3200–3205.

[8] V. Coelho *et al.*, "A self-adaptive evolutionary fuzzy model for load forecasting problems on smart grid environment," *Elsevier Appl. Energy*, vol. 169, pp. 567–584, May 2016.

[9] R. Huang, T. Huang, R. Gadh, and N. Li, "Solar generation prediction using the ARMA model in a laboratory-level micro-grid," in *Proc. IEEE 3rd Int. Conf. Smart Grid Commun.*, Tainan, Taiwan, Nov. 2012, pp. 528–533.

[10] J. Shi *et al.*, "Genetic algorithm-piecewise support vector machine model for short term wind power prediction," in *Proc. World Congr. Intell. Control Autom.*, Jinan, China, Aug. 2010, pp. 2254–2258.

[11] F. Silva *et al.*, "Application of a hybrid neural fuzzy inference system to forecast solar intensity," in *Proc. Int. Workshop Database Expert Syst. Appl.*, Porto, Portugal, Sep. 2016, pp. 161–165.

[12] C. Dumitru, A. Gligor, and C. Enachescu, "Solar photovoltaic energy production forecast using neural networks" *Procedia Technol.*, vol. 22, pp. 808–815, Dec. 2016.

[13] R. Peña and A. Medina, "Using neural networks to forecast renewable energy resources," in *Proc. Int. Conf. Neural Comput. Theory Appl.*, Paris, France, Jan. 2011, pp. 401–404.

[14] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting - An approach using AutoEncoder and LSTM neural networks," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Budapest, Hungary, Feb. 2017, pp. 1–8.

[15] F.-C. Chen, "Back-propagation neural network for nonlinear self-tuning adaptive control," in *Proc. IEEE Int. Symp. Intell. Control*, Albany, NY, USA, Sep. 1989, pp. 274–279.

[16] W. Jin, Z. Li, L. Wei, and H. Zhen, "The improvements of BP neural network learning algorithm," in *Proc. Int. Conf. Signal Process.*, Beijing, China, Aug. 2002, pp. 1647–1649.

[17] J. Fan and R. Gijbels, *Local Polynomial Modelling and Its Applications*. London, U.K.: Chapman & Hall, 1996.

[18] R. L. Eubank, "Applied nonparametric regression," *Technometrics*, vol. 35, no. 2, pp. 225–226, Sep. 1999.

[19] P. Craven and G. Wahba, "Smoothing noisy data with spline functions: Estimating the correct degree of smoothing by the method of generalized crossvalidation," *Numerische Mathematik*, vol. 31, no. 4, pp. 377–403, Dec. 1978.

[20] D. Montgomery, "Introduction to linear regression analysis," *J. Roy. Statist. Soc.*, vol. 170, no. 3, pp. 856–857, Apr. 2007.

[21] Y. Shu and J. Zeng, "A back-propagation neural network model base on genetic algorithm and the application," *J. Ningbo Univ.*, vol. 13, no. 4, pp. 39–44, Dec. 2000.

[22] M. Liberatore and P. Shenoy, *The UMass Trace Repository*. [Online]. Available: http://traces.cs.umass.edu/. Accessed on: Jan. 8, 2018.

[23] D. Draghicescu, S. Guillas, and W. B. Wu, "Quantile curve estimation and visualization for nonstationary time series," *J. Comput. Graph. Statist.*, vol. 18, no. 1, pp. 1–20, Aug. 2009.

[24] Z. Zhou and W. B. Wu, "Simultaneous inference of linear models with time varying coefficients," *J. Roy. Statist. Soc., Series B (Statist. Method.)*, vol. 72, no. 4, pp. 513–531, Sep. 2010.