

LASSO & LSTM Integrated Temporal Model for Short-term Solar Intensity Forecasting

Yu Wang, *Member, IEEE*, Yinxing Shen, Shiwen Mao, *Senior Member, IEEE*, Xin Chen, *Member, IEEE*, and Hualei Zou, *Student Member, IEEE*

Abstract—As a special form of the Internet of Things, Smart Grid is an internet of both power and information, in which energy management is critical for making the best use of the power from renewable energy resources such as solar and wind, while efficient energy management is hinged upon precise forecasting of power generation from renewable energy resources. In this paper, we propose a novel least absolute shrinkage and selection operator (LASSO) and long short term memory (LSTM) integrated forecasting model for precise short-term prediction of solar intensity based on meteorological data. It is a fusion of a basic time series model, data clustering, a statistical model and machine learning. The proposed scheme first clusters data using k-means++. For each cluster, a distinctive forecasting model is then constructed by applying LSTM, which learns the non-linear relationships, and LASSO, which captures the linear relationship within the data. Simulation results with open-source datasets demonstrate the effectiveness and accuracy of the proposed model in short-term forecasting of solar intensity.

Index Terms—Internet of Things (IoT), Short-term solar power forecasting, Least absolute shrinkage and selection operator (LASSO), Long short term memory (LSTM), K-means++.

I. INTRODUCTION

Internet of things (IoT) is defined as uniquely identifiable objects that are organized in an Internet-like structure [1]. The internet of information, vehicles, power, and energy are all typical forms of IoT. With innovative structure and techniques, IoT promotes the new developments of many traditional industries, and the Smart Grid (SG) is one of them. As a special form of the IoT, SG is considered as an internet of both power and information [2]. In recent years, more and more IoT-based technologies have been developed to make SG a more sustainable, economic, safe and reliable power grid from many aspects [3]–[5]. Among them, energy management is one of the most innovative and important techniques, which can increase the efficiency, reliability, and economy of the SG [6]–[9].

Energy management in the SG is usually based on different timescales, depending on practical constraints and purposes [10]. At the daily scale, day-ahead strategies for power

devices and budget for utilities can be developed, according to the day-ahead predictions of daily average power generation and demand, which focus on state estimation and market guidance. At the hourly or minutely scale, the real-time power distribution can be optimized based on predicted power generation and demand updated every hour or several minutes, and the current power storage, which determines the power flows in SG for more reliable, efficient and economic operation. For both cases, energy management relies on predictions of power generation and demand in the SG [6]. And precise short-term predictions are more important to guarantee the power balance and system reliability, and thus require higher precision. Among the various renewable energy sources, solar power is highly fluctuating, and changes very fast in some cloudy days, making it difficult for accurate short-term forecasting [11]. Therefore, precise short-term prediction of solar power generation is highly demanded to achieve a highly efficient energy management scheme for the SG.

Solar power generation from solar panels are proportional to solar intensity, so it can be estimated by predicting short-term solar intensity [12]. Many recent works focus on applying the time series model to predict short-term solar intensity. The work in [13] conducts the short-term prediction of solar radiation using autoregressive integrated moving-average (ARIMA) and time delay neural network (TDNN). The authors of [14] propose a time-series regression model with ARIMA and artificial neural network (ANN) to predict solar intensity in the city of Al-Ain, United Arab Emirates. The temporal model works well when the weather is sunny and stable. However, in cloudy, heavily cloudy, rainy, and similar weathers, the time series based methods are not very accurate due to the high variations of solar intensity. Therefore, applying only temporal model is inadequate for further increasing the forecasting accuracy [11].

Machine learning techniques, which can capture the relationships between solar intensity and the meteorological variables, are also popular in forecasting renewable power generation. For example, support vector machine (SVM) is applied in [15], and a method based on artificial neural network (ANN) is proposed in [16]. Although many machine learning methods can achieve fairly good results [15]–[18], they do not perform a deep analysis of the data, and thus there is still room for more accurate predictions. We summarize the pros and cons of both time series and machine learning models in solar intensity predictions in Table I.

Recurrent neural network (RNN) acquires the output from not only the current input information but also the past

This work is supported by the National Natural Science Foundation of China under grant No. 51607087, the Fundamental Research Funds for the Central Universities of China under grant NO. XCA17003-06, the NSF under grant DMS-1736470, and by the Wireless Engineering Research and Education Center at Auburn University.

Y. Wang, Y. Shen, X. Chen, and H. Zou are with the Department of Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. S. Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA. Email: yuwang15@nuaa.edu.cn, xubanqiu@163.com, smao@ieee.org, chen.xin@nuaa.edu.cn, zouhualai@nuaa.edu.cn.

TABLE I
THE PROS AND CONS OF MAJOR EXISTING METHODS FOR
SHORT-TERM SOLAR INTENSITY FORECASTING

Model	Pros	Cons
Time Series Model	Accurate predictions in sunny and stable weathers	Not effective in fast-changing weathers
Machine Learning Model	Acceptable predictions in almost all weathers	Limitation on highly precise predicting

experience and input information, which is a good candidate to be integrated into the solar intensity forecasting model. However, when the distance between two cells in the RNN is too large, the vanishing gradient may cause losses of important information [19]. Fortunately, the long short-term memory (LSTM) method successfully solves this vanishing gradient problem by adding three gates: input gate, forget gate, and output gate into the RNN cell [20]. It is thus a suitable method to further increase the forecasting accuracy by capturing the non-linear relationships between solar intensity and the meteorological variables. On the other hand, there exist both linear and nonlinear relationships in the meteorological data. The linear representation can be captured by a regression method. To this end, the least absolute shrinkage and selection operator (LASSO) is a linear regression analysis method, which reduces over-fitting and enhances the prediction accuracy through variable selection and regularization [1].

Because of the complex and complicated relationships between the solar intensity and meteorological variables, a method integrating time series characteristics, statistical model and machine learning is highly desirable for precise short-term forecasting of solar intensity. By integrating these three methods, we classify the data into several clusters according to the idea of building different specific forecasting models for different weather types. K-means is the most popular clustering method for its computational efficiency [21]. Once the number of clusters is specified, the clustering result could be obtained. However, the randomness of initial sampling (e.g., bad data) affects the clustering results badly. The improved k-means++ solves this sampling problem so that satisfactory clustering results can be obtained [22]. Therefore, we apply k-means++ to cluster the weather data. For each cluster, we start from a basic temporal model because of its strong temporality in short-term prediction of solar intensity. We then apply an integrated forecasting model in each cluster, capturing both temporal and spatial characteristics. The spatial part is further divided into linear and nonlinear parts linked by optimal ratios. This way, the forecasting accuracy can be increased to a very high level.

The main contribution of this paper is to propose an integrated forecasting model for precise short-term predictions of solar intensity. The model architecture is shown in Fig. 1. It is a fusion of a basic time series model, data clustering, a statistical model, and machine learning, which is a comprehensive model considering every aspect inside the short-term solar intensity forecasting problem, including strong time

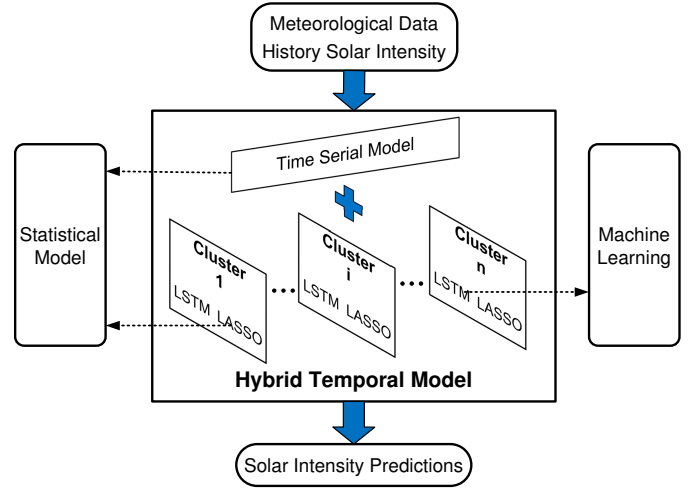


Fig. 1. Architecture of our proposed model.

correlation, weather complexity, linear and non-linear relationships. It generates different forecasting models for every different clusters from the k-means++ output, in which the time series model captures the strong time correlation, LSTM learns the non-linear relationship, and LASSO represents the linear relationship. Such a comprehensive model guarantees a very high forecasting accuracy of solar intensity, as will be shown in our performance evaluation and comparison study. Furthermore, our model is also applicable to other data-based short-term forecasting problems, such as predicting short-term wind power, loads of the power grid, and so on.

The remainder of this paper is organized as follows. We present the forecasting model construction in Section II. We propose our integrated short-term forecasting model in Section III. Simulation studies are presented in Section IV. Section V concludes this paper and discusses future work.

II. FORECASTING MODELS

In this section, we first introduce a general temporal model followed by model refinement from error correction. We then propose the integrated temporal model, which is based on the structure shown in Fig. 1.

A. Temporal Model

The data of solar intensity is usually recorded for every time interval, which can be represented using a time series model as follows [23]

$$Y(t) = f(Y(t - \delta_t), Y(t - 2\delta_t), \dots) + \epsilon(t), \quad (1)$$

where $Y(t)$ is the solar intensity at time t , $\epsilon(t)$ is the error at time t , and $f(\cdot)$ is a function which connects the recorded solar intensity data before t to the present $Y(t)$, δ_t denotes the time interval between two recordings, which is chosen from different timescales according to the practical application. For short-term predictions, δ_t is usually less than one hour. To simplify the expression without loss of generality, we use $t - k$

to represent $t - k\delta_t$ in our models, which do not specify δ_t , and thus, Eq. (1) is rewritten as:

$$Y(t) = f(Y(t-1), Y(t-2), \dots) + \epsilon(t). \quad (2)$$

In the temporal model, time correlation becomes stronger as δ_t gets smaller, i.e., data collected minutely correlate each other more tightly than data collected hourly, and $Y(t)$ and $Y(t-1)$ are more closely correlated than $Y(t)$ and $Y(t-2)$. Based on this, we can further simplify the model for short-term predictions as

$$Y(t) = Y(t-1) + \epsilon_t(t). \quad (3)$$

The shorter the δ_t , the smaller the $\epsilon_t(t)$. Also, this model works well for stable weather conditions such as sunny days. However, when it is partly cloudy, $\epsilon_t(t)$ could be large, causing more uncertainties in the model even when δ_t is chosen to be less than 5 minutes. Obviously, the model in (3) is not applicable in this case and a more comprehensive model is required.

B. Model Refinement

In (3), the error term $\epsilon_t(t)$ is actually the difference between $Y(t)$ and $Y(t-1)$. The major cause of this difference is from the weather condition and weather change, and thus, by relating them to the change of solar intensity, we can refine $\epsilon_t(t)$ as:

$$\epsilon_t(t) = g(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)) + \epsilon_r(t), \quad (4)$$

where $\vec{X}(t)$ is the vector of meteorological variables at time t , such as temperature, humidity, precipitation, etc.; $\vec{X}(t) - \vec{X}(t-1)$ is the change of meteorological variables in δ_t , and $\epsilon_r(t)$ is the remaining error, which is expected to be less than $\epsilon_t(t)$. Substituting (4) into (3), we have:

$$Y(t) = Y(t-1) + g(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)) + \epsilon_r(t), \quad (5)$$

where $g(\cdot)$ is a function connecting $(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1))$ to $Y(t) - Y(t-1)$. Here, we choose both $\vec{X}(t)$ and $\vec{X}(t) - \vec{X}(t-1)$, because the change of solar intensity in any δ_t may rely on the meteorological condition and change simultaneously. For example, for a sunny day in summer and winter, similar changes of meteorological data may cause very different amount of solar intensity changes. Also, the relationship between $\vec{X}(t) - \vec{X}(t-1)$ and $Y(t) - Y(t-1)$ is actually very complicated overall. However, there exist some useful relations, especially for some specific weather types and conditions, which help to further refine the model.

C. Integrated Temporal Model

Considering both linearity and non-linearity, the function $g(\cdot)$ is composed as

$$g(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)) = (\vec{X}(t), \vec{X}(t) - \vec{X}(t-1))^T \vec{\beta} + E(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)), \quad (6)$$

where the first term is a linear regression model representing the linear part in the data, and $\vec{\beta}$ is the vector of the regression coefficients; the second term is a function $E(\cdot)$ representing

the nonlinear part, which is usually revealed by machine learning methods. We then substitute (6) into (5) to obtain a comprehensive integrated temporal model:

$$Y(t) = Y(t-1) + (\vec{X}(t), \vec{X}(t) - \vec{X}(t-1))^T \vec{\beta} + E(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)) + \epsilon_r(t). \quad (7)$$

The corresponding forecasting model is thus written as

$$\tilde{Y}(t) = Y(t-1) + (\vec{X}(t), \vec{X}(t) - \vec{X}(t-1))^T \tilde{\vec{\beta}} + \tilde{E}(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)). \quad (8)$$

In order to apply this forecasting model, we need to identify $\tilde{\vec{\beta}}$ and $\tilde{E}(\cdot)$. To improve forecasting precision, solution methods are very decisive. As introduced in Section I, LASSO is a good option to get a better estimator $\tilde{\vec{\beta}}$. It is also capable of selecting meteorological variables, which is very useful to reduce the computational complexity. As for $\tilde{E}(\cdot)$, LSTM is a good choice for learning the nonlinear relationship. Moreover, linearity and non-linearity have different weights in different weather types. For example, in sunny days, linear relationship is the leading factor, while in cloudy days, non-linearity takes bigger role. So the weights allocated to them need to be carefully selected for different weather types. To solve the above problem, we propose the LASSO and LSTM integrated forecasting algorithm in Sec. III.

D. Model Characteristics

Before proceeding to the algorithm, it is necessary to examine the statistical characteristics of the forecasting model with respect to unbiasedness, consistency, efficiency, and sufficiency. However, since the proposed model (8) is an integration of statistical and machine learning methods, we cannot evaluate the model directly through statistical tests. Actually, the integrated model takes into account all the four characteristics. Here, we further analyze the model to show how these characteristics can be achieved.

In the meteorological data based short-term solar intensity forecasting problem, it is very difficult to find an unbiased estimator or predictor, if such estimator exists. This is because the relationships between solar intensity and the meteorological data are very complex and complicated, including linearity, non-linearity, time correlation and others. Because of this, the proposed model originates from a simple time series model (3), which is actually a first order auto-regressive model AR(1) of the parameter as 1. It is a non-stationary process and thus the error $\epsilon_t(t)$ is not white noise. In order to further reduce the bias of the model, we thus refine the error by (4).

In our forecasting model (7), data is partitioned into several clusters. Then for each cluster, we identify $\tilde{\vec{\beta}}$ and $\tilde{E}(\cdot)$ to apply the forecasting model. When the data size is approaching infinite, the error $\epsilon_r(t)$ is approaching zero; because of clustering, $\tilde{\vec{\beta}}$ and $\tilde{E}(\cdot)$ become more accurate. Theoretically, machine learning methods are able to capture any relationship, and the more data it learns, the better the model is. From this aspect, the proposed predictor is consisting.

Normally, solar intensity prediction performance in fast changing weathers are worse than that in sunny weathers. In

other words, the efficiency of a forecasting model depends mainly on the prediction performance in fast changing weathers. Because of this, we cluster the data into several groups so that a specific model is generated for every group to improve accuracy. This way, the overall performance is improved, and thus, the model efficiency is increased correspondingly.

To guarantee the sufficiency of the model, we make full use of the dataset and partition the dataset into training data, evaluation, data and testing data.

In summary, the integrated temporal model actually takes into account of all the four aspects, i.e., unbiasedness, consistency, efficiency, and sufficiency. Since theoretic analysis of the characteristics of an estimator is important but very difficult, it has to be replaced with quantitative simulations. We will evaluate the model characteristics through simulations in Sec. IV.

III. LASSO AND LSTM INTEGRATED ALGORITHM

In this section, we first present LASSO and LSTM and discuss their parameter selection. We then present the method to combine them together. Then we propose the integrated algorithm for short-time solar intensity forecasting.

A. Least Absolute Shrinkage and Selection Operator (LASSO)

Let $Y_1(t)$ denote $(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1))^T \vec{\beta}$, i.e., the linear part in (7), and the predicted $Y_1(t)$ is written as

$$\tilde{Y}_1(t) = (\vec{X}(t), \vec{X}(t) - \vec{X}(t-1))^T \tilde{\vec{\beta}}, \quad (9)$$

where $\tilde{Y}_1(t)$ is considered as the linear portion of the predicted changes of the solar intensity from $t-1$ to t . Because of its shrinkage function and parameters' selection, LASSO is used here to identify the estimated coefficients $\tilde{\vec{\beta}}$ as

$$\tilde{\vec{\beta}} = \underset{\vec{\beta}}{\operatorname{argmin}} \left\{ \sum_{i=1}^N \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j| \right\}, \quad (10)$$

where λ is a system parameter. LASSO is able to shrink large regression coefficients in order to reduce over-fitting by the constraint of the sum of the absolute value of the regression coefficients smaller than a fixed value, which filters out some coefficients, and thus effectively simplifies the model.

To acquire a fitted system parameter λ , two methods are commonly used: cross validation [24] and regularization path [25]. Cross validation is a very general and common technique finding an optimal parameter. Its weakness is the slow computation. Regularization path is able to achieve a good estimation accuracy with fairly quick computation. It also has the potential to achieve a high estimation accuracy even when information is lacking. Thus, we use the regularization path to obtain λ based on the following steps:

- 1) Choose a set of possible λ s and sort them in ascending order;
- 2) Execute the proposed algorithm for each λ and record their performance;
- 3) Plot the achieved precision performance versus λ ;

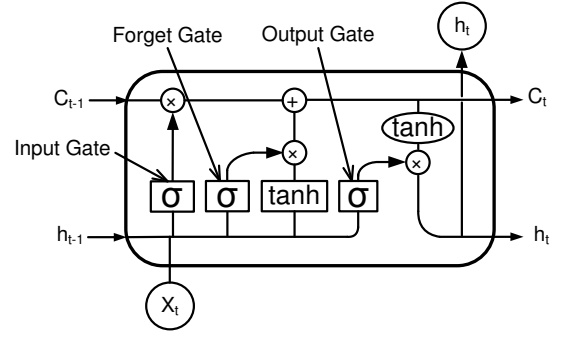


Fig. 2. The structure of LSTM.

- 4) Choose an acceptable point on the curve to guarantee the performance while achieving the maximized estimation speed due to sparsity.

B. Long Short Term Memory (LSTM)

For the non-linearity representing function $E(\cdot)$ in the model of (7), a specific machine learning method is required to learn it from the meteorological data and change of solar intensity. For simplification, we use $Y_2(t)$ to denote the non-linear part in (7), and thus we can write the predicted $Y_2(t)$ as

$$\tilde{Y}_2(t) = \tilde{E}(\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)), \quad (11)$$

where $\tilde{Y}_2(t)$ is the output of the function acquired by the learning method.

Neural network (NN) is a common method to capture the non-linear relationship in a complex dataset. RNN is an improved NN, which is able to combine both the current and previous information to find solutions. However, it is very difficult to obtain useful information if two RNN cells are far away from each other, due to the gradient vanishing problem. To address this issue, LSTM incorporates three gates into the RNN cell, as shown in Fig. 2. Each cell of LSTM sends two state variables to the next cell, which ensures the reliability of information transmission and thus avoids gradient vanishing. Moreover, the results from LSTM are highly repeatable [26]. Because of the above advantages, we apply LSTM to acquire a precise $\tilde{Y}_2(t)$.

Here, LSTM is used to generate the classification network and regression network, for classifying evaluation and prediction data, and learning non-linear characteristics respectively. Both the two networks have three major steps. The first two steps are feature extraction and the LSTM network, respectively. The third step is softmax classifier for the classification network, and a neural network for the regression network [27], [28].

1) *Feature Extraction:* For better feature extraction, we implement one fully connected layer with $\operatorname{ReLU}(\cdot)$ as the activation function, which is given as

$$z_t = \operatorname{ReLU}(Wx_t + b), \quad (12)$$

where x_t and z_t are the input and output of the fully connected layers, respectively, W and b are the weights and biases of the layers. The activation function is formulated as $\operatorname{ReLU}(\cdot) =$

$\max(x, 0)$. The $\text{ReLU}(x)$ function has several advantages such as sparse representation, efficient gradient propagation, and computation.

2) *LSTM Network*: After feature extraction, we use the LSTM for training optimal weights. For the classification LSTM network, we stack two layers of the LSTM network to obtain a stronger learning ability, which improves the performance of classification and regression.

3) *Softmax Classifier*: The output of the final cell's hidden state in the classification LSTM network is the input to a fully connected layer, which trains the output data using the softmax classifier. The softmax function maps the N dimensional vector to normalized data $p = [p_1, p_2, \dots, p_N]$, and

$$p_i = \frac{e^{h_f^T \omega_i}}{\sum_{n=1}^N e^{h_f^T \omega_n}}, \text{ for } i = 1, 2, \dots, N, \quad (13)$$

where N is the number of training data, h_f is the output vector of the final cell's hidden state in the LSTM network, and ω_i is the weight vector of the fully connected layer. Let $L(t)$ be the loss function, which is formulated as

$$L(t) = \frac{1}{2} \sum_{i=1}^N (p_i - y_i)^2, \quad (14)$$

where y_i is the true classification result for the i th training data. We then use Back Propagation Through Time (BPTT) and Real-Time Recurrent Learning (RTRL) to train the classification LSTM networks [29]. We also use the Adam Optimizer to improve the efficiency of optimization.

C. Data Clustering by K-means++

By LASSO and LSTM, we can use (8) to make predictions given any dataset. Based on the complete historical data, however, only one forecasting model will be constructed, which is too general to adapt to various weather conditions. Therefore, we need to cluster the data, and build a specific model for each cluster.

K-means++ clustering is a commonly used method, which selects k initial center points arbitrarily and clusters the data according to the distances from each data point to the centers. Its simple principle makes it suitable for fast computation for many applications, although it is sometimes difficult to get suitable number of k [30]. Here, we go through all possible k s in the specified range. To overcome the shortcoming of random selection of initial center points, which may lead to bad clustering performance, k-means++ applies the d^2 -sampling method [22] with the following procedure:

- 1) Select the first center point randomly;
- 2) Calculate the probability of selecting the next center by

$$p(x|C) = \frac{d(x, C)^2}{\sum_{x' \in \mathcal{X}} d(x', C)^2}, \quad (15)$$

where $\mathcal{X}$ represents a set of n points in $\mathbb{R}^D$, D is the dimension of $\mathcal{X}$, x denotes a point in $\mathcal{X}$, C is the set of center points and c is a selected center point, x' denotes the other points in set $\mathcal{X}$ except for x , and the function $d(x, C)^2$ is defined as $d(x, C)^2 = \sum_{c \in C} \|x - c\|_2^2$.

This way, the selected center points are far from each other with a high probability, and thus k-means++ is suitable for clustering the meteorological data. Therefore, we firstly cluster the data, and then find the forecasting model for each cluster.

D. LASSO and LSTM Integrated Forecasting Algorithm

We now propose the LASSO and LSTM integrated forecasting algorithm (LLIFA) for short-term solar intensity forecasting. Let $\mathbb{S} = \{\vec{X}(t), \vec{Y}(t)\}$ represent the dataset of meteorological and solar intensity. We first divide it into two subsets: the training set $\mathbb{T} = \{\vec{X}(t), \vec{Y}(t)\}_T$, and the evaluation set $\mathbb{E} = \{\vec{X}(t), \vec{Y}(t)\}_E$, where $\mathbb{T} \cup \mathbb{E} = \mathbb{S}$ and $\mathbb{T} \cap \mathbb{E} = \emptyset$. Then, we cluster the training set $\mathbb{T}$ into N groups of data, and each group is denoted as $\mathbb{T}_k = \{\vec{X}(t), \vec{Y}(t)\}_{T_k}$, where $k = 1, 2, \dots, N$. According to (6), the input variable for the k-means++ clustering is the vector $[\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)]$ in $\mathbb{T}$. $\mathbb{T}_k$ is then divided into two parts: the input vector $\vec{X}_{T_k} = [\vec{X}(t), \vec{X}(t) - \vec{X}(t-1)]_{T_k}$ and the output vector $\vec{Y}_{T_k} = \vec{Y}(t) - \vec{Y}(t-1)$ for training. We next divide each training output $\vec{Y}_{T_k}$ into the linear part $\vec{Y}_{1,T_k}$ and the non-linear part $\vec{Y}_{2,T_k}$:

$$\vec{Y}_{1,T_k} = \vec{Y}_{T_k} \cdot 1/(1 + \alpha_k), \quad (16)$$

$$\vec{Y}_{2,T_k} = \vec{Y}_{T_k} \cdot \alpha_k/(1 + \alpha_k), \quad (17)$$

where α_k is a positive ratio coefficient in the k th cluster. α_k is set to different values in different clusters, to adjust the weights of $\vec{Y}_{1,T_k}$ and $\vec{Y}_{2,T_k}$ in the forecasting model. Given α_k s, we can solve for the regression coefficients $\vec{\beta}$ and the function $\tilde{E}(\cdot)$ using LASSO and LSTM, respectively, to determine the forecasting model for each cluster. It is thus important to search for the optimal α_k s, as evaluated by the set $\mathbb{E}$.

First, we need to classify $\mathbb{E}$ into N clusters according to the classification standard in $\mathbb{T}$. However, k-means++ does not produce the same classification results, because $\mathbb{T}$ and $\mathbb{E}$ are different. So we use LSTM to learn the classification net *classnet* in $\mathbb{T}$. Through *classnet*, $\mathbb{E}$ can be clustered under the same rule into N clusters, denoted as $\mathbb{E}_k = \{\vec{X}(t), \vec{Y}(t)\}_{E_k}$, where $k = 1, 2, \dots, N$. Based on this, we can perform the traversing method to obtain the optimal α_k s. Considering both (16) and (17), α_k s are limited in $\alpha_k \in [\alpha_{min}, \alpha_{max}]$, which is determined according to the dataset. $\alpha_{k,j}$ gradually increases from $\alpha_{k,0} = \alpha_{min}$ to $\alpha_{k,j} = \alpha_{max}$ by an increment d_α , where $j = 0, 1, \dots, J$, and $J = \lfloor (\alpha_{max} - \alpha_{min})/d_\alpha \rfloor$. For every $\alpha_{k,j}$, the corresponding $\vec{\beta}_{k,j}$ and the function $\tilde{E}_{k,j}(\cdot)$ are generated. According to (8), (9) and (11), the predicted $\tilde{Y}_{E_k}(t)$ can be found for every solar intensity in every evaluation cluster $\mathbb{E}_k$. So under $\alpha_{k,j}$, we can calculate the sum of errors for each cluster by

$$e_{\alpha_{k,j}} = \sum_{t=1}^{T_k} |Y_{E_k}(t) - \tilde{Y}_{E_k}(t)|, \quad (18)$$

and then acquire the optimal $\hat{\alpha}_k$ by

$$\hat{\alpha}_k = \arg \min_{\alpha_{k,j}} \{e_{\alpha_{k,j}}\}. \quad (19)$$

Algorithm 1: Method of obtaining the optimal ratio coefficients $\hat{\alpha}_k$ in different clusters

- 1 Determine the range of $\alpha_k \in [\alpha_{min}, \alpha_{max}]$ according to the dataset;
 - 2 Start to test the ratio coefficients $\alpha_{k,j}$ from $\alpha_{k,0} = \alpha_{min}$;
 - 3 Divide each training output $\tilde{Y}_{T_k}$ into linear part $\tilde{Y}_{1,T_k}$ and non-linear part $\tilde{Y}_{2,T_k}$ based on (16) and (17);
 - 4 Generate the corresponding $\tilde{\beta}_{k,j}$ and function $\tilde{E}_{k,j}(\cdot)$ through LASSO and LSTM by training set $\mathbb{T}$;
 - 5 Find every predicted solar intensity $\tilde{Y}_{E_k}(t)$ in the evaluation cluster $\mathbb{E}_k$ according to (8), (9) and (11);
 - 6 Calculate the error sum $e_{\alpha_{k,j}}$ for each cluster by (18);
 - 7 Increase the ratio coefficients by increment d_α until $\alpha_{k,j} = \alpha_{max}$ and for each $\alpha_{k,j}$ repeat Steps 3 to 6;
 - 8 Acquire the optimal $\hat{\alpha}_k$ by (19).
-

Algorithm 2: The algorithm procedure of LLIFA

- 1 Divide the dataset $\mathbb{S} = \{\tilde{X}(t), \tilde{Y}(t)\}$ into the training set $\mathbb{T}$ and evaluation set $\mathbb{E}$;
 - 2 Cluster the training set $\mathbb{T}$ into N groups of data using k-means++;
 - 3 Train the classification net *classnet* using $\mathbb{T}$ and cluster $\mathbb{E}$ into N clusters using *classnet*;
 - 4 Acquire the ratio coefficients $\hat{\alpha}_k$ through Algorithm 1 and obtain the corresponding regression coefficients, the non-linear function at the same time, and the forecasting model for every cluster;
 - 5 Classify the new input data for a future time t into an appropriate cluster through *classnet*;
 - 6 Predict the solar intensity based on the forecasting model of the corresponding cluster.
-

This way, we are able to obtain the optimal ratio coefficients, the corresponding regression coefficients, the non-linear function, and thus the forecasting model of each cluster. We summarize the steps to acquire $\hat{\alpha}_k$ s in Algorithm 1.

To predict the solar intensity for a future time t , new input data is firstly classified into an appropriate cluster through the *classnet*. Then the solar intensity can be predicted based on the corresponding model of that cluster. We summarize the procedure to perform LLIFA in Algorithm 2, which is a fusion of LASSO and LSTM through k-means++ clustering. Note that to make sure the forecasting model for every cluster is precise, it needs to guarantee both the training dataset $\mathbb{T}$ and the evaluation dataset $\mathbb{E}$ are sufficiently large, so that the ratio coefficients in each forecasting model is significant. Also, we divide the training and evaluation data before clustering them separately, because the training data, and thus the choices of the ratio coefficients are more appropriate.

IV. SIMULATION STUDIES

In this section, we verify our proposed forecasting model and algorithms on two trace-driven datasets, and compare the performance with other benchmarks. The simulation results

show that the proposed forecasting model achieve outstanding performance for different datasets under different timescales, and outperforms other benchmarks.

A. Data Description

The first trace-driven dataset is acquired from the Davis weather station located in Amherst, MA, USA [31]. The meteorological data was collected every 5 minutes. The main recorded weather variables including temperature, wind chill, humidity, dew-point, wind speed, wind direction, and rainfall. The dataset is recorded from Feb., 2006 to Jan., 2013, which also contains some missing data and some errors, recorded as -100000 . In the simulation study, we excluded such errors and missing data. Moreover, we only consider the solar intensity on the day time when the solar intensity is nonzero, so the data of which the solar intensity is zero is also excluded. For short-term solar intensity forecasting, we mainly use the data from Jan. 1, 2011 to Feb. 28, 2013. We divide the data into three sub-sets: training, evaluation and testing data. The data from Jan. 1, 2011 to Dec. 31, 2012 is departed every other day into training data and evaluation data separately. This generates almost the same size of training and evaluation data, ensuring both datasets are large enough to get precise forecasting models for all clusters. The remaining data from Jan. 1 to Feb. 28, 2013 are taken as testing data.

The second dataset is recorded in Harnhill and Diddington in the U.K [32]. At each location, two weather stations are installed (four in total), which recorded every 30 minutes the data of rainfall, temperature, humidity, wind speed and so on. We use this the data recorded from Aug., 2011 to Dec., 2012. Missing data in datasets are represented by NaN. By excluding such invalid data, the remaining useful data are separated into the training, evaluation and testing data as same as the first dataset. The data from Aug. 1, 2011 to Sept. 30, 2012 is departed every other day into training data and evaluation data respectively. The remaining data are taken as testing data. Note that this data has a longer recording time interval than the first one, which helps us to evaluate the proposed forecasting model under different timescales.

B. Model Improvement with Data Analysis

We first use three fundamental models and their combinations to analyze the data and model characteristics in short-term solar intensity forecasting. According to Sec. II, we apply the following models to predict solar intensity every 5 minutes:

$$\tilde{Y}(t) = Y(t-1) \quad (20)$$

$$\tilde{Y}(t) = (\tilde{X}(t), \tilde{X}(t) - \tilde{X}(t-1))^T \tilde{\beta}_o \quad (21)$$

$$\tilde{Y}(t) = \tilde{E}_m(\tilde{X}(t), \tilde{X}(t) - \tilde{X}(t-1)) \quad (22)$$

$$\tilde{Y}(t) = Y(t-1) + (\tilde{X}(t), \tilde{X}(t) - \tilde{X}(t-1))^T \tilde{\beta}_m \quad (23)$$

$$\tilde{Y}(t) = Y(t-1) + \tilde{E}_m(\tilde{X}(t), \tilde{X}(t) - \tilde{X}(t-1)), \quad (24)$$

where (20) is a short-term time series model (TSM), (21) and (22) are based on LASSO and LSTM respectively, (23) is an integrated model combining TSM and LASSO, and (24) is another integrated model combining TSM and LSTM. Based

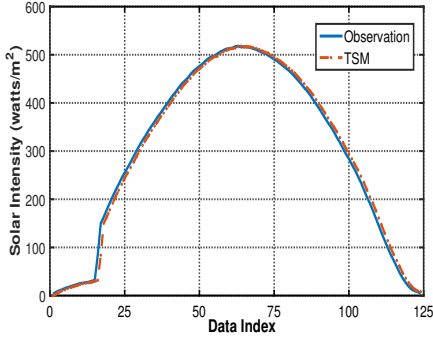


Fig. 3. 5-min solar intensity predictions by TSM on Feb. 8 vs. observations.

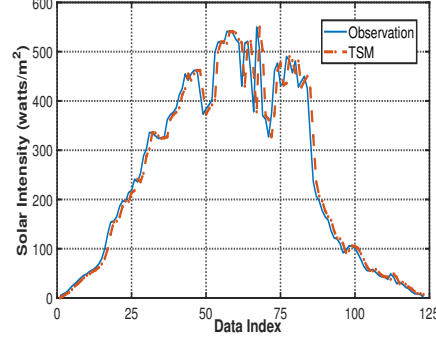


Fig. 4. 5-min solar intensity predictions by TSM on Feb. 19 vs. observations.

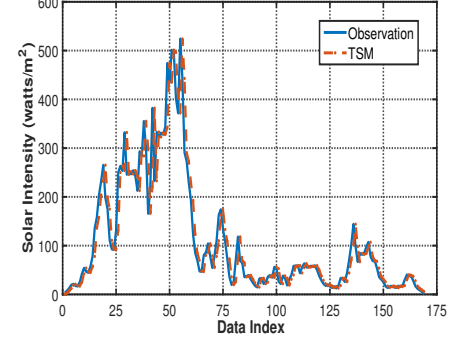


Fig. 5. 5-min solar intensity predictions by TSM on Jan. 25 vs. observations.

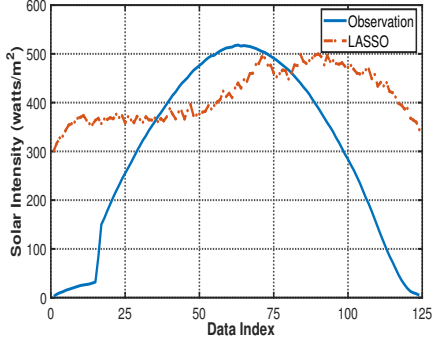


Fig. 6. 5-min solar intensity predictions by LASSO on Feb. 8 vs. observations.

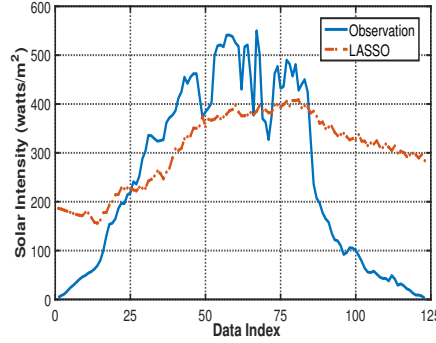


Fig. 7. 5-min solar intensity predictions by LASSO on Feb. 19 vs. observations.

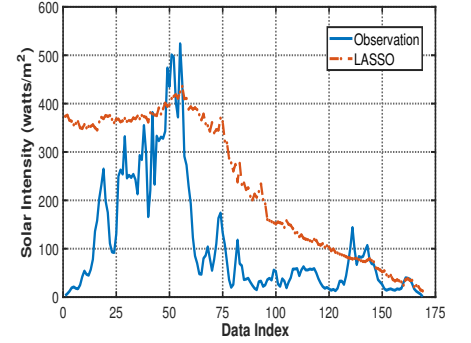


Fig. 8. 5-min solar intensity predictions by LASSO on Jan. 25 vs. observations.

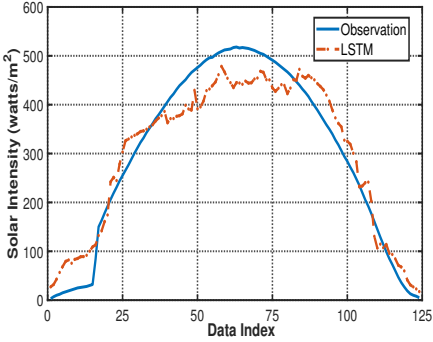


Fig. 9. 5-min solar intensity predictions by LSTM on Feb. 8 vs. observations.

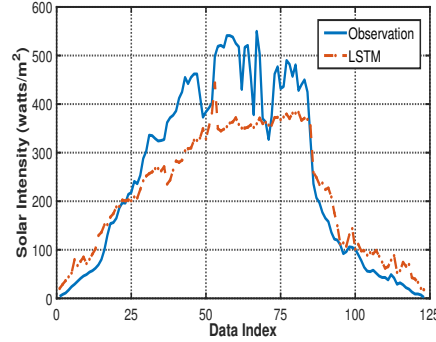


Fig. 10. 5-min solar intensity predictions by LSTM on Feb. 19 vs. observations.

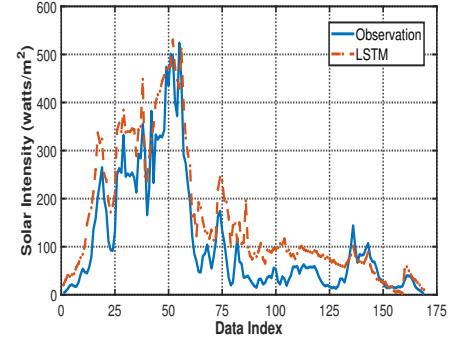


Fig. 11. 5-min solar intensity predictions by LSTM on Jan. 25 vs. observations.

on the UMASS data of 2011 and 2012, we find $\tilde{\beta}_o, \tilde{\beta}_m, \tilde{E}_m(\cdot)$ and $\tilde{E}_m(\cdot)$, and use the above five models to predict the solar intensity every 5 minutes for the first two months in 2013. We use both mean absolute percentage error (MAPE) and root mean square error (RMSE) to measure the accuracy of the forecasting models, which are summarized in Table II. We also plot the predicted solar intensity for Feb. 8, Feb. 19 and Jan. 25 based on three single models of TSM, LASSO and LSTM from Figs. 3 to 11. These three days represent three different weathers: stable, fluctuating and fast changing.

From Table II, we have several observations. Among the first three models, TSM achieves much better results than LASSO and LSTM, while LASSO has the worst performance whatever the weather is. This is because the temporal correlation

is very strong between the meteorological data and solar intensity. LASSO has poor performance shown also in Figs. 6 to 8, because the linear relationship in the data under short time period is weaker than that under longer timescale, such as daily or weekly. LSTM has better performance than LASSO because of its capability at learning and capturing strong time correlations in the data.

On the other hand, comparing the forecasting results between the integrated models and the first three ones, we find that both LSTM and LASSO can increase the predicting accuracy of TSM. For three days shown in Table II, comparing with TSM, the RMSEs decrease by 5.31%, 2.85%, and 2.58% respectively under TSM-LASSO, and decrease by 24.48%, 41.89%, and 68.47% respectively under TSM-LSTM. Note

TABLE II
FORECASTING ERRORS ON MAPE (%) AND RMSE ($watts/m^2$) OF FIVE
MODELS WITH UMASS DATA

Date		Feb. 8	Feb. 19	Jan. 25
TSM	RMSE	11.48	34.16	51.92
	MAPE	7.58	13.63	29.06
LASSO	RMSE	221.37	159.62	208.12
	MAPE	448.14	448.42	467.63
LSTM	RMSE	75.02	81.81	66.28
	MAPE	37.66	59.55	96.58
TSM-LASSO	RMSE	10.87	33.19	50.58
	MAPE	5.63	12.23	28.49
TSM-LSTM	RMSE	8.67	19.85	16.37
	MAPE	3.86	7.04	11.76

that the improvement of LSTM and LASSO on TSM differs for different weathers. LASSO can be a good assistant in stable days, while LSTM is good for cloudy days. They complement each other in assisting TSM. Based on these observations and analyses, we propose the LLIFA by deep fusion of LSTM and LASSO into TSM. And we notice that in different weathers, LSTM and LASSO have different performance, based on which, we cluster the data, find the regression coefficients for LASSO, train the function for LSTM and allocate their weights in the corresponding forecasting models.

C. Performance Evaluation of LLIFA

By applying LLIFA proposed in Sec. III-D, we need to cluster the data into an optimal number of sub-sets. We cluster the data into different numbers and calculate the forecasting errors for every cluster. We plot in Fig. 12 the forecasting errors for the three typical days Feb. 8, Feb. 19 and Jan. 25 under different number of clusters. Predictions on other days have the same trend. It shows the forecasting errors for all three days are minimized in 11 clusters. Fig. 12 indicates that when the clustering number is either too small or too large, the forecasting performance is poor. This is because given the same dataset, increasing the number of clusters reduces the amount of data in each cluster, which correspondingly reduces the accuracy of the models. On the other hand, a small number of clusters are not sufficient to capture the essential information in the complex weather data and thus cause large forecasting errors. We therefore set $N = 11$.

According to Algorithm 2, we use k-means++ to cluster the data into 11 clusters. For each cluster, we obtain a corresponding forecasting model which is used to predict the solar intensity. By comparisons, we build another model in the form of (8) without clustering. We compare the average forecasting errors in each cluster with that without clustering in Fig. 13 and Fig. 14. It can be seen that in every cluster, the forecasting error reduces much from no clustering, which verifies the effectiveness and importance of the clustering.

We plot the forecasting results of LLIFA for Feb. 8, Feb. 19 and Jan. 25 in Figs. 15, 16 and 17 respectively. The RMSEs and MAPEs for these three days are $7.83watt/m^2$, 2.81% ,

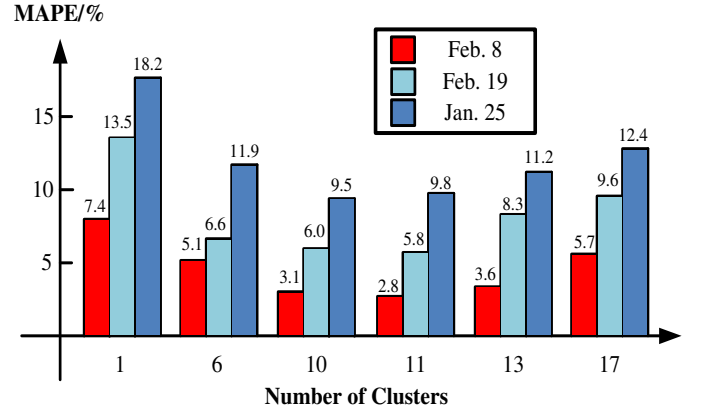


Fig. 12. Forecasting MAPEs of three days under different clusters.

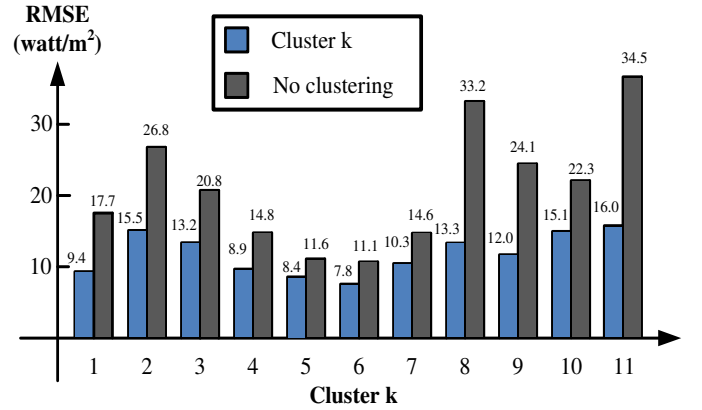


Fig. 13. Forecasting RMSEs under different clusters vs. no clustering.

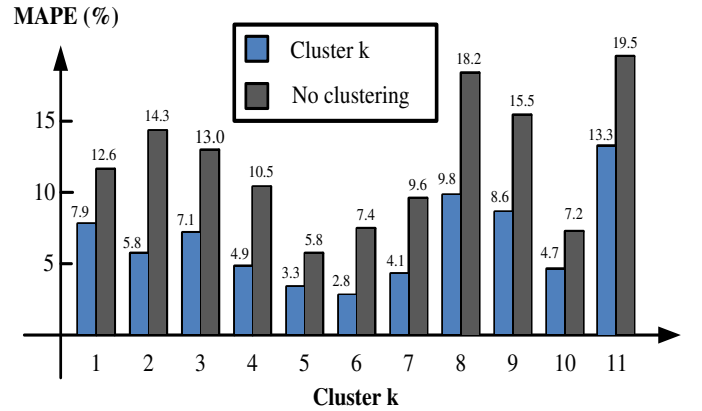


Fig. 14. Forecasting MAPEs under different clusters vs. no clustering.

$15.53watt/m^2$, 5.81% , $13.31watt/m^2$, 9.83% respectively, summarized in Table III. Note that in Fig. 17, the proposed LLIFA achieves the average MAPE of only 9.83% in forecasting the solar intensity every 5 minutes even in fluctuating or fast changing weathers. For stable sunny day, the predictions are very precise with MAPE as small as 3% in Fig. 15.

D. Comparisons of LLIFA to the Benchmarks

We now compare our proposed LLIFA scheme with a commonly used time series method ARIMA and a representative

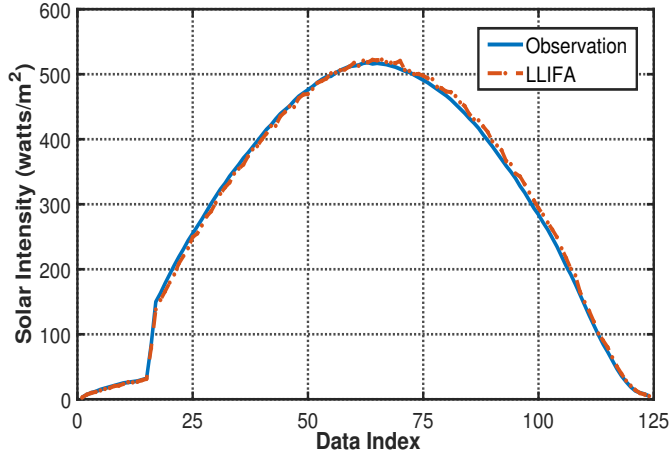


Fig. 15. 5-min solar intensity predictions by LLIFA on Feb. 8 vs. observations.

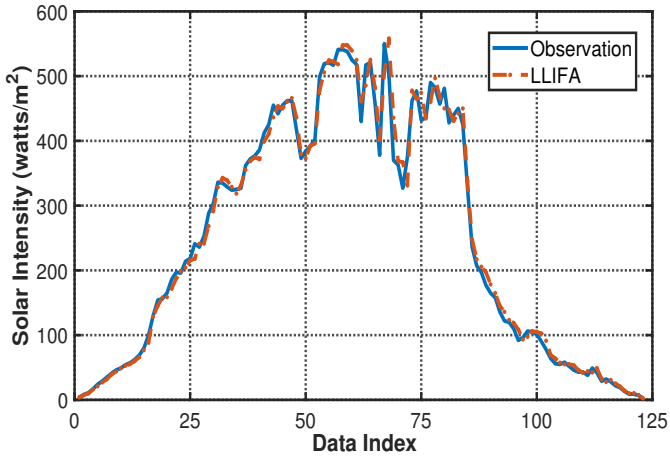


Fig. 16. 5-min solar intensity predictions by LLIFA on Feb. 19 vs. observations.

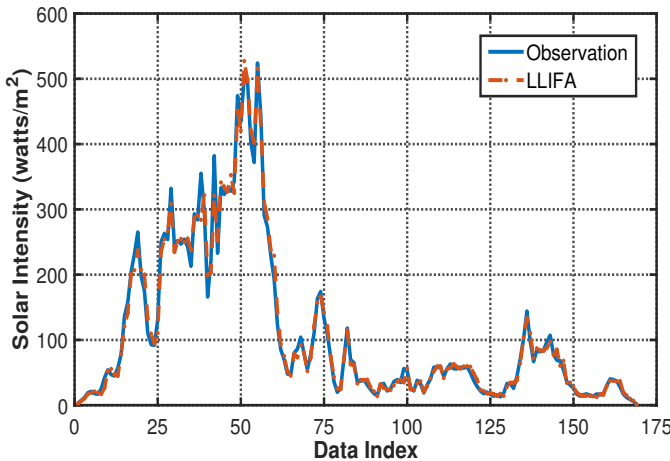


Fig. 17. 5-min solar intensity predictions by LLIFA on Jan. 25 vs. observations.

supervised learning method SVM. As a typical time series model, ARIMA can transform any non-stationary process into a stationary process through the differential time series. Solar intensity is actually a non-stationary time series, and thus

TABLE III
5-MIN FORECASTING ERRORS ON MAPE (%) AND RMSE ($watt/m^2$) BY LLIFA FOR THREE TYPICAL DAYS WITH UMASS DATA

Date	Feb. 8	Feb. 19	Jan. 25
RMSE	7.83	15.53	13.31
MAPE	2.81	5.81	9.83

TABLE IV
MEAN&MEDIAN VALUE OF MAPES (%) AND RMSES ($watts/m^2$) FOR 5-MIN PREDICTIONS WITH THREE MODELS FROM JAN. 1 TO FEB. 28

Model		Mean	Median
LLIFA	RMSE	10.67	8.34
	MAPE	4.73	3.16
ARIMA	RMSE	19.42	15.27
	MAPE	9.94	7.29
SVM	RMSE	79.04	76.89
	MAPE	67.52	58.65

ARIMA is applicable in solar intensity forecasting, especially in sunny and stable weather. On the other hand, SVM is a representative and widely adaptable machine learning method, which generates very good results in many application scenarios, when the optimal parameters are used. For example, SVM is applied in [15] for solar intensity forecasting. Therefore, we compare our model with ARIMA and SVM in our comparison studies.

Predictions are made every 5 minutes for the first two months in 2013 using UMASS weather data. The 5-min mean and median RMSEs and MAPEs for LLIFA, ARIMA and SVM are summarized in Table IV. The predicting results show that LLIFA outperforms ARIMA and SVM with averaged MAPE 4.73% and median MAPE 3.16%. ARIMA has fairly good performance which is much better than SVM, because ARIMA is good at modeling the short-term temporal relationship [13] while SVM lacks a deep analysis of the complex and complicated weather data, by simply trying different kernels [15]. LSTM works better than SVM with its stronger temporal learning capability. The results verify again the strong time correlation in the short-term forecasting problem. LLIFA fully considers the strong temporality and integrates both statistical regression and machine learning methods, so it has better forecasting performance.

E. Predicting Performance of LLIFA on a Longer Recording Time Scale

Next we test LLIFA on another trace-driven data in Didding-ton by forecasting the solar intensity every 30 minutes. This time, we cluster the data into $N = 10$ groups by checking the data with the same process as previous. We also select three typical days: stable, fluctuating and fast changing, and plot the corresponding forecasting results in Figs. 18, 19 and 20, respectively. The predictions using only LSTM are also plotted as a comparison. And the forecasting errors are summarized in

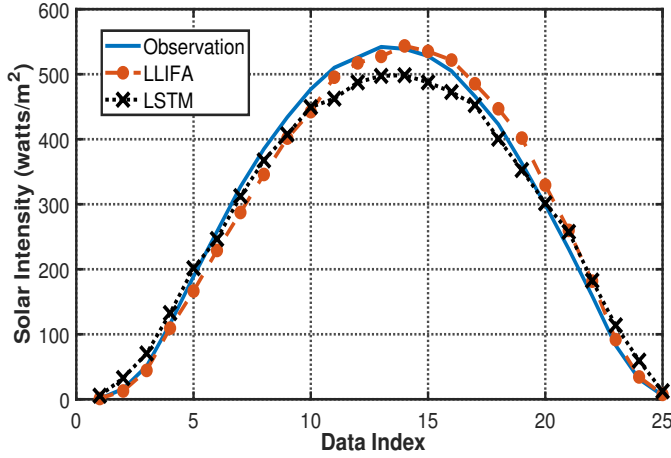


Fig. 18. 30-min predictions from LLIFA and LSTM vs. observations on a day in Cluster 6 of stable weather.

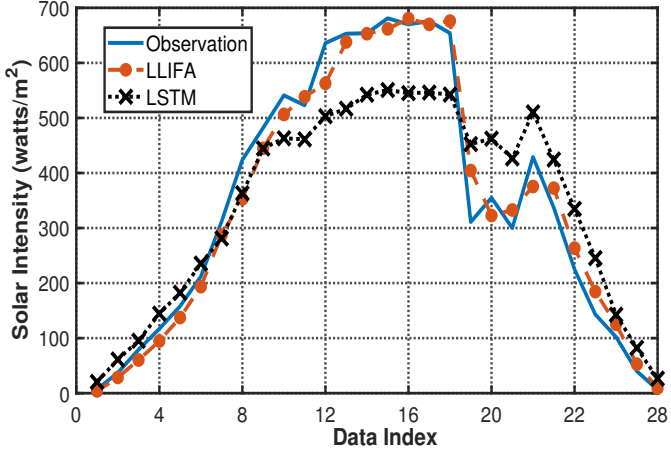


Fig. 19. 30-min predictions from LLIFA and LSTM vs. observations on a day in Cluster 2 of fluctuating weather.

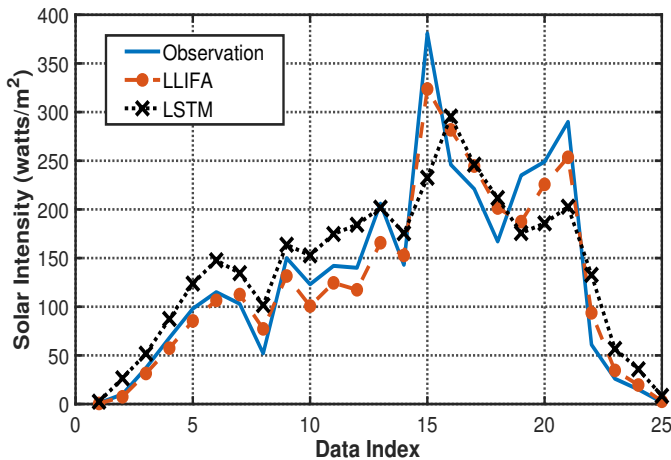


Fig. 20. 30-min predictions from LLIFA and LSTM vs. observations on a day in Cluster 8 of fast changing weather.

Table V. Note that these three days are clustered into Cluster 6, 2 and 8, respectively.

In a typical sunny day, LLIFA has very precise predic-

TABLE V
30-MIN FORECASTING ERRORS ON MAPE (%) AND RMSE ($watt/m^2$)
BY LLIFA AND LSTM FOR THREE TYPICAL DAYS WITH DIDDINGTON
DATA

Cluster #		6	2	8
LLIFA	RMSE	22.32	35.69	35.50
	MAPE	9.85	14.94	21.18
LSTM	RMSE	27.12	88.16	49.02
	MAPE	36.95	46.07	61.81

tions, tracking the actual observations closely. The RMSE and MAPE in Fig. 18 are $22.32 \text{ watts}/m^2$ and 9.85% respectively. It can be found from Figs. 19 and 20 that in fluctuating and fast changing weathers, the predicting performance of LLIFA is not ideal. The MAPEs are 14.94% and 21.18% respectively. This shows that when the weather changes fast, i.e., data index 19 to 23 in Fig. 19, and data index 15 to 23 in Fig. 20, LLIFA has bad performance, although it still works better than LSTM. This is same as the result from the previous UMASS data. In summary, it is most difficult to predict the solar intensity in fast changing weathers.

By comparing Table V with Table III, we could also find that LLIFA predicts better with the UMASS data than the Diddington data. This is because when the time interval of data recordings increases, the time correlation decreases and thus the forecasting precision of LLIFA decreases. In summary, LLIFA works great in short-term solar intensity forecasting especially in stable weathers. However, when the time interval between data recordings increase, the forecasting accuracy is getting down. However, LLIFA could still beat other methods because it integrates temporal model, clustering, statistical model and machine learning.

F. Evaluations on Model Characteristics

Finally, we sum up the errors between the actual solar intensity and the predictions using the proposed model for three different datasets of the two traces. The results are summarized in Table VI. For both UMASS and Diddington traces, the sum of predicting errors for three datasets are all very close to zero, indicating the proposed model is close to be unbiased. Unbiasedness appears more obviously in the UMASS trace, which is recorded every 5 minutes and thus has much more data than the Diddington trace. This implies the possible consisting characteristic of the proposed forecasting model. Also, as mentioned in Sec. II-D, the model applies efficient clustering (evaluated in Sec. IV-C), and makes full usage of the data, the proposed model is thus efficient and sufficient.

V. CONCLUSION AND FUTURE WORK

In this paper, we developed a LASSO and LSTM integrated temporal model for meteorological-data-based short-term solar intensity forecasting. We first presented the fundamental temporal model, followed by the model refinement and characteristics analysis. Then we developed the integrated temporal

TABLE VI
SUM OF THE PREDICTING ERRORS ($watt/m^2$) IN DIFFERENT DATA SETS
FOR UMASS AND DIDDINGTON DATA

	Training	Evaluation	Testing
# of UMASS data points	52500	52000	7412
$\Sigma[y(t) - \hat{y}(t)]$	-1	3	-6
# of Diddington data points	5200	5100	2076
$\Sigma[y(t) - \hat{y}(t)]$	2	-13	29

model which integrates regression model and machine learning method. We then proposed the LASSO and LSTM integrated forecasting algorithm to realize the integrated temporal model using k-means++ clustering, LASSO and LSTM. The proposed models were validated with trace-driven simulations under different timescales and compared to several benchmarks.

Although the proposed model can predict short-term solar intensity with high precision, the prediction accuracy decreases as timescale increases. So it may not be suitable for predicting solar intensity on longer timescales. Our future work is to propose an accurate general forecasting model which can predict the solar intensity precisely on different timescales.

REFERENCES

- [1] N. Tang, S. Mao, Y. Wang, and R. M. Nelms, "Solar power generation forecasting with a LASSO-based approach," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1090–1099, Apr. 2018.
- [2] Y. Wang, S. Mao, and R. M. Nelms, *Online Algorithms for Optimal Energy Distribution in Microgrids*, 1st ed. New York, NY, USA: Springer, 2015.
- [3] M. Li, P. He, and L. Zhao, "Dynamic load balancing applying water-filling approach in smart grid systems," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 247–257, Feb. 2017.
- [4] Z. Guan, J. Li, L. Wu, Y. Zhang, and J. Wu, "Achieving efficient and secure data acquisition for cloudsupported Internet of Things in smart grid," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1934–1944, Dec. 2017.
- [5] J. Pan, R. Jain, S. Pual, T. Vu, A. Saifullah, and M. Sha, "An Internet of Things framework for smart energy in buildings: Designs, prototype, and experiments," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 527–537, Dec. 2015.
- [6] Y. Wang, S. Mao, and R. M. Nelms, "Distributed online algorithm for optimal real-time energy distribution in the smart grid," *IEEE Internet of Things J.*, vol. 1, no. 1, pp. 70–80, Feb. 2014.
- [7] L. Yu, T. Jiang, and Y. Zou, "Distributed online energy management for data centers and electric vehicles in smart grid," *IEEE Internet of Things J.*, vol. 3, no. 6, pp. 1373–1384, Dec. 2016.
- [8] Y. Wang, S. Mao, and R. M. Nelms, "Online algorithm for optimal realtime energy distribution in the smart grid," *IEEE Trans. Emerg. Topics Comput.*, vol. 1, no. 1, pp. 10–21, Jun. 2013.
- [9] Y. Huang, S. Mao, and R. M. Nelms, "Adaptive electricity scheduling in microgrids," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 270–281, Jan. 2014.
- [10] H. Chen, P. Xuan, Y. Wang, K. Tan, and X. Jan, "Key technologies for integration of multitype renewable energy sources research on multi-timeframe robust scheduling/dispatch," *IEEE Trans. on Smart Grid*, vol. 7, no. 1, pp. 471–480, Jan. 2017.
- [11] 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.
- [12] Y. Wang, S. Mao, and R. M. Nelms, "On hierarchical power scheduling for the macrogrid and cooperative microgrids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1574–1584, Dec. 2015.
- [13] J. Wu and C. K. Chan, "The prediction of monthly average solar radiation with TDNN and ARIMA," in *Proc. Inter. Conference on Machine Learning and Applications*, Boca Raton, FL, USA, Jan. 2013, pp. 469–474.
- [14] H. A. N. Hejase and A. H. Assi, "Time-series regression model for prediction of mean daily global solar radiation in Al-Ain, UAE," *ISRN Renewable Energy*, vol. 2012, no. 1, pp. 1–11, Apr. 2012.
- [15] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *Proc. IEEE SmartGridComm'11*, Brussels, Belgium, Oct. 2011, pp. 528–533.
- [16] A. Mellit and A. M. Pavan, "A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy," *Solar Energy*, vol. 84, no. 5, pp. 807–821, May 2010.
- [17] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, vol. 148, no. 1, pp. 461–468, Apr. 2018.
- [18] Y. Sun, G. Szucs, and A. R. Brandt, "Solar PV output prediction from video streams using convolutional neural networks," *Energy & Environmental Science*, vol. 11, no. 7, pp. 1811–1818, Apr. 2018.
- [19] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Inter. Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, Canada, Oct. 2013, pp. 6645–6649.
- [20] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Dec. 1997.
- [21] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means clustering algorithm," *J. Roy. Stat. Soc.*, vol. 28, no. 1, pp. 100–108, Oct. 1979.
- [22] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proc. Eighteenth ACM/SIAM Symposium on Discrete Algorithms*, New Orleans, LA, Jan. 2007, pp. 1027–1035.
- [23] G. E. P. Box and G. M. Jenkins, "Time series analysis: Forecasting and control," *Journal of Time*, vol. 31, no. 4, pp. 303–303, Dec. 2010.
- [24] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B Methodol.*, vol. 58, no. 1, pp. 267–288, Jun. 1996.
- [25] P. Zeng, T. He, and Y. Zhu, "A Lasso-type approach for estimation and variable selection in single index models," *J. Comput. Graph. Stat.*, vol. 21, no. 1, pp. 92–109, Apr. 2012.
- [26] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. & Learning Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
- [27] X. Wang, Z. Yu, and S. Mao, "DeepML: Deep LSTM for indoor localization with smartphone magnetic and light sensors," in *Proc. IEEE ICC 2017*, Kansas City, MO, May 2018.
- [28] W. Yang, X. Wang, S. Cao, H. Wang, and S. Mao, "Multi-class wheat moisture detection with 5GHz Wi-Fi: A deep LSTM approach," in *Proc. ICCCN 2018*, Hangzhou, China, July/Aug. 2018.
- [29] Q. Wang, Y. Gu, L. Liu, and S. Kamijo, "DeepSpeedometer: Vehicle speed estimation from accelerometer and gyroscope using LSTM model," in *Proc. IEEE International Conference on Intelligent Transportation Systems*, Yokohama, Japan, Oct. 2017, pp. 1–6.
- [30] Y. Xu, G. Yue, and S. Mao, "User grouping for Massive MIMO in FDD systems: New design methods and analysis," *IEEE Access Journal*, vol. 2, no. 1, pp. 947–959, Sept. 2014.
- [31] University of Massachusetts. *The UMass Trace Repository*. Accessed: Mar. 2018. [online] Available: <http://traces.cs.umass.edu/index.php/Sensor/Sensor>.
- [32] The Detection of Archaeological Residues using Remote-sensing Techniques (DART) Project. *DART Weather Data*. Accessed: Mar. 2018. [online] Available: http://dartportal.leeds.ac.uk/dataset/dart_monitoring_weather_data/.



Yu Wang [S'13-M'17] received the B.S and M.S. degrees in instrument and meter engineering from Southeast University, Nanjing, China in 2008 and 2011, respectively. He received Ph.D. in electrical and computer engineering from Auburn University, Auburn, AL in 2015. Currently, he is an Assistant Professor in the Department of Electrical and Computer Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include smart grid, microgrid, renewable energy power management, and optimization.



Yinxing Shen received his BE degree in smart grid information engineering from Nanjing University of Science and Technology, Nanjing, China, in 2016. He is currently pursuing the M.S. degree in the Department of Electrical and Computer Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include smart grid, demand response, and microgrid optimization.



Shiwen Mao [S'99-M'04-SM'09] received his Ph.D. in electrical and computer engineering from Polytechnic University, Brooklyn, NY in 2004. He is the Samuel Ginn Distinguished Professor and Director of the Wireless Engineering Research and Education Center (WEREC) at Auburn University, Auburn, AL, USA. His research interests include wireless networks, multimedia communications, and smart grid. He is a Distinguished Speaker of the IEEE Vehicular Technology Society. He is an Area Editor of IEEE Internet of Things Journal, and on

the Editorial Board of IEEE Transactions on Mobile Computing, IEEE Transactions on Multimedia, IEEE Multimedia, ACM GetMobile, among others. He received the Auburn University Creative Research & Scholarship Award in 2018, the 2017 IEEE ComSoc ITC Outstanding Service Award, the 2015 IEEE ComSoc TC-CSR Distinguished Service Award, the 2013 IEEE ComSoc MMTC Outstanding Leadership Award, and the NSF CAREER Award in 2010. He is a co-recipient of the 2017 Best Conference Paper Award of IEEE ComSoc MMTC, the Best Demo Award from IEEE SECON 2017, the Best Paper Awards from IEEE GLOBECOM 2016 & 2015, IEEE WCNC 2015, and IEEE ICC 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems.



Xin Chen [S'99-M'04] received the B.S. and Ph.D. degrees in electrical engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1996 and 2001, respectively. From 2001 to 2003, he was a Chief Engineer with the Power Division of ZTE Corporation. From 2010 to 2011, he was an Invited Researcher with the Rensselaer Polytechnic Institute, Troy, USA. He is currently an Associate Professor at the Nanjing University of Aeronautics and Astronautics. His current research interests include power electronic

converters, distributed generation, and microgrids.



Hualei Zou [S'18] received the B.E. degree in Electrical Engineering and Automation from Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China in 2008 and M.E. degree in Power Electronics and Power Drives from Jiangsu University, Zhenjiang, China in 2011. Since 2014, she has been pursuing the Ph.D. degree in the Department of Electrical Engineering, NUAA, Nanjing, China. Her research interests include energy management in Smart Grid and Microgrid, Optimization.