# Feature Construction and Selection for PV Solar Power Modeling

Yu Yang[1], Jia Mao[1], Richard Nguyen[2], Annas Tohmeh[2], Hen-Geul Yeh[2]

***Abstract*— Using solar power in the process industry can reduce greenhouse gas emissions and make the production process more sustainable. However, the intermittent nature of solar power renders its usage challenging. Building a model to predict photovoltaic (PV) power generation allows decision-makers to hedge energy shortages and further design proper operations. The solar power output is time-series data dependent on many factors, such as irradiance and weather. A machine learning framework for 1-hour ahead solar power prediction is developed in this paper based on the historical data. Our method extends the input dataset into higher dimensional Chebyshev polynomial space. Then, a feature selection scheme is developed with constrained linear regression to construct the predictor for different weather types. Several tests show that the proposed approach yields lower mean squared error than classical machine learning methods, such as support vector machine (SVM), random forest (RF), and gradient boosting decision tree (GBDT).**

## I. Introduction

Solar power becomes increasingly important in the energy market as more PV panels are installed globally. However, the intermittent character of solar incurs a great safety issue because the high fluctuation of energy supply leads to the system instability and may damage connected appliances. Hence, an accurate prediction model of solar power generation is highly desired for the system integration and control [1], [2].

Solar power prediction approaches based on the physical, statistical, and machine learning models have been proposed. The physical models use either previous observations or numerical weather prediction (NWPs) as inputs. For example, the persistence (PSS) model simply assumes that the current power generation is equal to the previous one. The total sky imager (TSI) model relies on the image processing technique and cloud tracking for 15-30 minutes ahead prediction [3]. These two methods are limited to the short horizon prediction because the cloud cover may change rapidly. Many types of clear sky model can be used to estimate the solar irradiance [4], which then is inputted to a solar PV modeling algorithm for power prediction [5]. The statistical approach is extensively studied in [6], which uses autoregressive (AR) for the short-term prediction and AR with exogenous input (ARX) for the long-term forecast. In [7], a clear sky model is introduced to normalize the solar power data, and then an auto-regressive integrated moving average (ARIMA) model is built for the stochastic cloud cover. A probabilistic model is developed in [8] to determine the joint distribution of hourly-ahead horizontal irradiation and measured solar power supply. Many machine learning methods are applied for solar power prediction and show significant superiority over the traditional physical and statistical approaches. The neural network (NN) model has become very popular since the 1990s [9]. The forecast models based on linear, feed-forward, recurrent, and radial basis NN have been developed [10]–[12] for the global horizontal irradiance (GHI). Other machine learning methods, such as support vector machines (SVM) using multiple kernels [13], are also available in literature. A comprehensive comparison study on the day-ahead hourly forecast of solar power generation is presented in [14], where the second-order grey-box regression method, NN, quantile random forest (RF), k-Nearest Neighbors (kNN), and support vector regression (SVR) are investigated. Their results show that these approaches have similar overall accuracy. An ensemble average is proposed to synthesize these methods and achieve the best performance under all weather conditions.

The contribution of this paper is building a regression model based on the high-order basis functions for 1-hour ahead solar power prediction. The weather conditions, temperature, dew point, humidity, and wind speed are inputs to the predictor. The one-step (15-minute) past solar generation is introduced as an autoregressive term in the model. An essential innovation of this work is introducing Chebyshev polynomials and trigonometric functions into the regression model to form a higher dimensional feature space. Then, a wrapper method is employed to select suitable features for different weather conditions. Based on the selected features, a constrained least squares problem is solved to determine the model coefficients. In case studies, we show that the proposed approach is more accurate than SVR, RF, and gradient boosting decision tree (GBDT).

The remainder of this paper is organized as follows. The background knowledge and dataset are shown in Section 2. In Section 3, the regression model and feature selections are presented. In Section 4, several classical machine learning methods are implemented through scikit-learn package and compared with our method. The conclusion is drawn in the final Section 5.

## II. Background and Data

This work aims to predict 1-hour ahead solar power generation using weather data. Only short-horizon prediction is studied because long-range weather forecast may not be accurate, especially when the cloud cover plays a very important role in the output. The Long Beach, California weather record from January to June 2014 is gathered to extract the information of temperature, dew point, humidity, wind

[1]Yu Yang and Jia Mao are with Department of Chemical Engineering, California State University Long Beach yu.yang@csulb.edu

[2]Richard Nguyen, Annas Tohmeh, Hen-Geul Yeh are with the Department of Electrical Engineering, California State University Long Beach

speed, and weather type. Different from [13], precipitation is not considered as a useful feature since it does not vary sufficiently in California within the entire day. Regarding the weather type, we combine cloudy, mostly cloudy, and partly cloudy as one group. The haze, fog, and blowing dust are in the same group because they are all classified as horizontal obscuration. Therefore, three weather types are considered, including cloudy, fair, and haze. Here we do not study the rainy weather because no such data is available during the daytime at the studied location.

The California Solar Initiative (CSI) 15-minute interval data is used as the solar energy output. This dataset was built on the measured production data from 414 of the 504 solar systems and further improved by simulation for missing data and reflection of the true character of the system [15]. An important step is to match the location and timestamp of the solar power production and weather data. Here a PV system in Long Beach airport is studied, where the historical weather data is available with details. Because the sampling time of weather data is irregular, we align each solar power data item with its closest weather record. The solar power prediction models can be expressed as:

$$y(k+1) = F_{[i]}(y(k), \boldsymbol{u}(k)) \tag{1}$$

where $k$ represents the sampling time instant with 15 minutes interval; $y$ denotes the solar power output; $\boldsymbol{u} =$ [temperature, dew point, humidity, wind speed]; $F_{[i]}$ is the predictor function. Note that the weather type is not directly used as the input because associated cloud coverage is not quantified in the record. Instead, we use subscript $[i]$ to denote different weather type and develop their models separately. In prediction, the employed model can be switched based on the weather type at time instant $k$. Here $y(k)$ in the regression function is an autoregressive term to take the most recent measurement into account.

Considering that the sun elevation and azimuth vary month by month, the training, validation, and testing dataset may not cross a long period. Six datasets are designed and shown in Table I. The training set has 25 days, whereas validation and testing datasets all have 5 days.

TABLE I

DATE OF TRAINING, VALIDATION AND TESTING DATASET

| | Training | Validation | Testing |
|---|---|---|---|
| Dataset 1 | Jan 1-25 | Jan 26-30 | Jan 31-Feb 4 |
| Dataset 2 | Feb 1-25 | Feb 26-Mar 2 | Mar 3-7 |
| Dataset 3 | Mar 1-25 | Mar 26-30 | Mar 31-Apr 4 |
| Dataset 4 | Apr 1-25 | April 26-30 | May 1-5 |
| Dataset 5 | Apr 26-May 20 | May 21- 25 | May 26-30 |
| Dataset 6 | May 20-Jun 13 | Jun 14-18 | Jun 19-23 |

A simple model can be built upon the mean of power generation profiles in the training set, denoted as a positive variable $\bar{y}(k)$. However, such an average may not reflect the actual power dynamics under different weather conditions. In Fig. 1, the solar power profiles of every day in dataset 3 are plotted as an example. The thick dash line represents the mean power output. Large deviations occur when the weather conditions are significantly different from the average. The

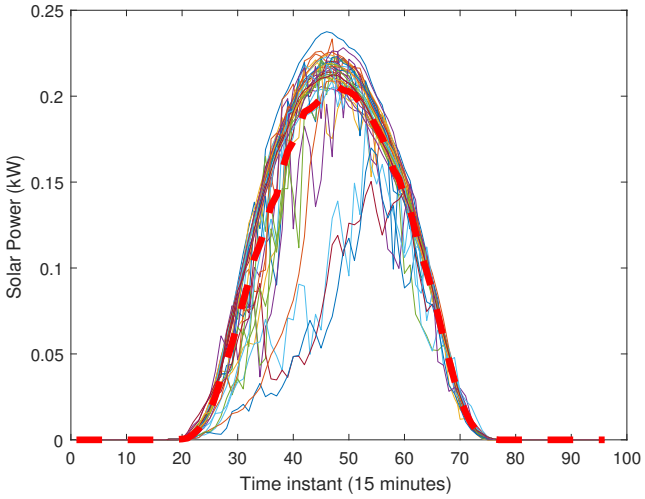


Fig. 1. Each curve represents a daily solar power profile.

mean value obtained from the training dataset is removed to reduce the time dependence. Then, models are developed to predict the deviation $y' = y - \bar{y}$ rather than the raw value $y$, such that the mean squared error (MSE) can be reduced significantly.

## III. PREDICTION MODEL

Starting from the easiest AR model, we denote the following basic regressor:

$$\phi(k+1) = [y'(k), \boldsymbol{u}_1(k), \boldsymbol{u}_2(k), \boldsymbol{u}_3(k), \boldsymbol{u}_4(k), \boldsymbol{u}_5(k)] \tag{2}$$

where $\boldsymbol{u}_1$ is the temperature; $\boldsymbol{u}_2$ is the dew point; $\boldsymbol{u}_3$ is the humidity; $\boldsymbol{u}_4$ is the wind speed; and $\boldsymbol{u}_5$ is the time. Here we omit the index of data items. Then, one may assume that output $y'(k+1)$ is linearly dependent on the input $\boldsymbol{u}(k)$ and autoregressive term $y'(k)$. However, such a simple model cannot achieve satisfactory performance. Instead, a high-order polynomial model is designed based on $\phi$. Clearly, there are so many options for polynomial terms. One cannot arbitrarily specify the degree and formulas of the polynomial used as features. Therefore, a feature construction and selection procedure is designed to determine which polynomial terms should be chosen.

### A. Feature Construction

Let us briefly describe candidate features. For the deviation outputs $y'(k)$, the bell shape data still can be found in some trials. Thus, the regressor $\phi(k)$ is extended by attaching two more variables: $\boldsymbol{u}_6(k) = \cos(\pi \boldsymbol{u}_5(k)/24)$ and $\boldsymbol{u}_7(k) = \sin(\pi \boldsymbol{u}_5(k)/24)$. This extended regressor becomes

$$\begin{aligned}\phi^*(k+1) =& [y'(k), \boldsymbol{u}_1(k), \boldsymbol{u}_2(k), \boldsymbol{u}_3(k), \boldsymbol{u}_4(k), \boldsymbol{u}_5(k), \\ & \boldsymbol{u}_6(k), \boldsymbol{u}_7(k)]\end{aligned}$$

Then, the normalized regressor is

$$\tilde{\phi}(k) = \frac{\phi^*(k)}{\bar{\phi}}$$

where denominator vector $\bar{\phi}$ can be chosen as the maximum absolute value of each variable in the data such that $\tilde{\phi}(k)$ is within the range $[-1, 1]$. The first kind Chebyshev polynomial can be constructed to form the following feature set:

$$\begin{aligned}
&\boldsymbol{C}_0 = 1,\ \boldsymbol{C}_1 = \tilde{\phi},\ \boldsymbol{C}_2 = 2\tilde{\phi}^2 - 1,\ \boldsymbol{C}_3 = 4\tilde{\phi}^3 - 3\tilde{\phi},\\
&\boldsymbol{C}_4 = 8\tilde{\phi}^4 - 8\tilde{\phi}^2 + 1,\ \boldsymbol{C}_5 = 16\tilde{\phi}^5 - 20\tilde{\phi}^3 + 5\tilde{\phi},\\
&\boldsymbol{C}_6 = 32\tilde{\phi}^6 - 48\tilde{\phi}^4 + 18\tilde{\phi}^2 - 1,\\
&\boldsymbol{C}_7 = 64\tilde{\phi}^7 - 112\tilde{\phi}^5 + 56\tilde{\phi} - 7\tilde{\phi},\\
&\boldsymbol{C}_8 = 128\tilde{\phi}^8 - 256\tilde{\phi}^6 + 160\tilde{\phi}^4 - 32\tilde{\phi}^2 + 1,\\
&\boldsymbol{C}_9 = 256\tilde{\phi}^9 - 576\tilde{\phi}^7 + 432\tilde{\phi}^5 - 120\tilde{\phi}^3 + 9\tilde{\phi},\\
&\boldsymbol{C}_{10} = 512\tilde{\phi}^{10} - 1280\tilde{\phi}^8 + 1120\tilde{\phi}^6 - 400\tilde{\phi}^4 + 50\tilde{\phi}^2 - 1
\end{aligned}$$

Only $\boldsymbol{C}_0 - \boldsymbol{C}_{10}$ are considered because more polynomials may render the feature selection step more computationally expensive. Except $\boldsymbol{C}_0$, all $\boldsymbol{C}_w\ \forall w \in \{1, 2, \ldots, 10\}$ are matrices with 8 columns. The proposed scheme increases the dimension of basis function and thus is able to represent complex dynamics of the system. Moreover, because the cloud cover impacts solar irradiance but is not reflected in existing features, another three inputs $\boldsymbol{u}_8, \boldsymbol{u}_9, \boldsymbol{u}_{10}$ are further introduced:

$$\boldsymbol{u}_8(k) = \begin{cases} 1 & \text{if cloudy at time instant } k \\ 0 & \text{else} \end{cases} \tag{3}$$

$$\boldsymbol{u}_9(k) = \begin{cases} 1 & \text{if mostly cloudy at time instant } k \\ 0 & \text{else} \end{cases} \tag{4}$$

$$\boldsymbol{u}_{10}(k) = \begin{cases} 1 & \text{if partly cloudy at time instant } k \\ 0 & \text{else} \end{cases} \tag{5}$$

Because the sky cover data is not quantified, one-hot encoding $\boldsymbol{u}_9, \boldsymbol{u}_{10}, \boldsymbol{u}_{11}$ can highlight the different types of clouds. The new features are the product of time and cloudy type:

$$\begin{aligned}
\boldsymbol{C}_{11} &= \boldsymbol{u}_8 \boldsymbol{u}_5/24\\
\boldsymbol{C}_{12} &= \boldsymbol{u}_9 \boldsymbol{u}_5/24\\
\boldsymbol{C}_{13} &= \boldsymbol{u}_{10} \boldsymbol{u}_5/24
\end{aligned}$$

The rational of introducing $\boldsymbol{C}_{11} - \boldsymbol{C}_{13}$ with time instant is that cloud incurs high deviation on the irradiance at noon, whereas the variation near sunset or sunrise is relatively small. $\boldsymbol{C}_{11} - \boldsymbol{C}_{13}$ are generated through feature interaction and not linearly dependent with any existing features.

### *B. Feature Selection*

In the previous section, several features are introduced to form a pool. Including all of them in the regression model can significantly reduce the training error. However, this is not true for the validation and testing datasets. Namely, overfitting may happen if unnecessary features contribute to the training process. To overcome this issue, we employ a sequential forward selection and backward elimination procedure. A subset of features will be chosen for each dataset such that the prediction model performs well on both training and validation set. In fact, a number of feature selection schemes based on the embedded and filter approaches have been proposed for process system analysis [16]–[19]. The wrapper method is employed in this paper because it evaluates the feature set directly based on the data fitting performance. Before discussing the details of feature selection, let us develop the training scheme.

For the weather type $i$, we build the predictor model $F_{[i]}$ with unknown coefficients $\boldsymbol{a}_{[i]}$:

$$F_{[i]} = \sum_{\boldsymbol{C}_{w,j} \in \Psi_{[i]}} \boldsymbol{a}_{[i],w,j} \boldsymbol{C}_{w,j} \tag{6}$$

where $\Psi_{[i]}$ represents the set of chosen features for weather type $i$; $\boldsymbol{C}_{w,j}$ represents $j^{th}$ column in the feature set $\boldsymbol{C}_w$. A constrained least squares is presented in ($\mathcal{LS}$) to obtain $\boldsymbol{a}_{[i]}$:

$$\min_{\boldsymbol{a}_{[i]}} \sum_{k \in \Gamma_{[i]}} (y'(k+1) - \sum_{\boldsymbol{C}_{w,j} \in \Psi_{[i]}} \boldsymbol{a}_{[i],w,j} \boldsymbol{C}_{w,j})^2 \qquad (\mathcal{LS})$$

$$\text{s.t.} \sum_{\boldsymbol{C}_{w,j} \in \Psi_{[i]}} \boldsymbol{a}_{[i],w,j} \boldsymbol{C}_{w,j} + \bar{y}(k+1) \geqslant 0 \tag{7}$$

where $\Gamma_{[i]}$ includes all data indexes for weather type $i$ within the training set. The objective function in ($\mathcal{LS}$) is to minimize the one-step prediction error. Eq. (7) requires the predicted solar power $\hat{y}(k+1) = \hat{y}'(k+1) + \bar{y}(k+1)$ greater or equal to zero.

Then, the flowchart of feature selection and elimination (Algorithm 1) for weather $i$ is shown in Fig. 2. The mean squared error (MSE) is the performance index, and the outcome is feature set $\Psi_{[i]}$ with model parameters $\boldsymbol{a}_{[i]}$. Here we identify $\boldsymbol{a}_{[i]}$ based on the training dataset and the feature evaluation is based on the validation dataset.

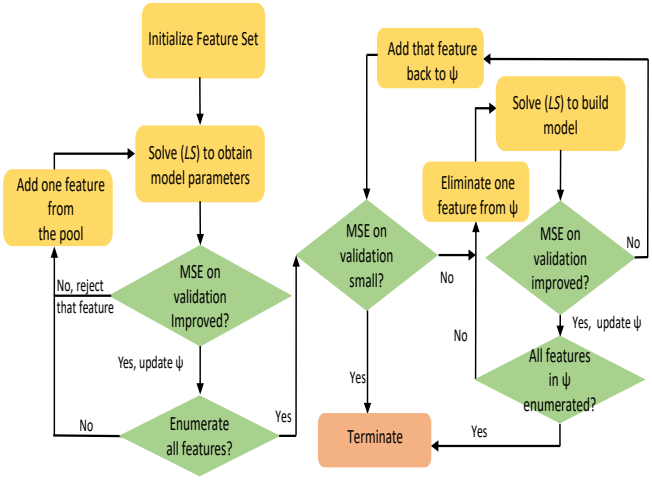


Fig. 2. Algorithm 1: Feature selection and elimination for weather $i$.

Several comments about Algorithm 1 are in order. **First**, ($\mathcal{LS}$) is a convex optimization problem with at most $10 \times 8 + 4 = 84$ decision variables, and thereby can be solved quickly. **Second**, Algorithm 1 is a wrapper method directly using MSE on validation set as a criterion to incorporate or eliminate features. Our method only finds a sub-optimal

solution, whereas the global optimality cannot be guaranteed. A possible improvement can be achieved by using Bayesian optimization to choose features. **Third**, Algorithm 1 only solves ($\mathcal{LS}$) to minimize one-step ahead prediction error. Directly minimizing the multi-step prediction error will lead to a high-order nonlinear optimization problem and all different weather types should be considered simultaneously. The resulting computational burden makes feature selection scheme extremely inefficient. **Fourth**, the selected feature set can be distinct for different datasets and weather types because the sun irradiation and cloud coverage may vary month by month.

Next, Algorithm 2 in Fig. 3 combines all weather types model together and evaluate the multi-step prediction error on the validation set. This algorithm continuously updates the feature subset for all types of weather until no improvement can be achieved.

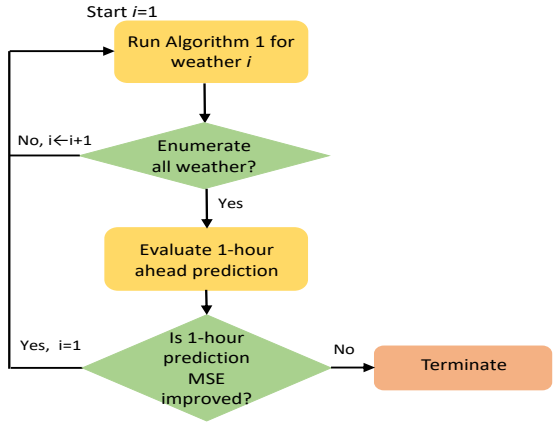


Fig. 3. Algorithm 2: Choose the model based on multistep prediction error.

The major difference between multi-step and one-step prediction is that we need to replace the autoregressive term by the prediction value. At time $k$, for one-step prediction $\hat{y}'(k+1|k)$, we use $y'(k|k)$, which is the measurement value. However, for $M$-step prediction $\hat{y}'(k+M|k)$, we need to use $\hat{y}'(k+M-1|k)$, which is based on the previous prediction. It implies that the prediction error at any step will be accumulated and impact the forecast in future steps.

A constraint can be embedded into ($\mathcal{LS}$) to ensure that the multi-step prediction is bounded. Let us define the normalized M-step regressor as

$$\tilde{\phi}(k+M+1|k) = \frac{\phi^*(k+M+1|k)}{\bar{\phi}}$$

where $\phi^*(k+M+1|k) = [\hat{y}'(k+M|k), \boldsymbol{u}_1(k+M), \ldots, \boldsymbol{u}_7(k+M)]$, $\forall M \geqslant 0$.

*Proposition 1:* If constraint (8) is integrated into ($\mathcal{LS}$) and $\bar{\phi} \geqslant 1$, the M-step ahead prediction is bounded.

$$\sum_{w,j} |a_{[i],w,j}| \leqslant 1 \tag{8}$$

**Proof:**

The first-kind Chebyshev polynomial $\boldsymbol{C}_0$ to $\boldsymbol{C}_{10}$ are within $[-1,1]$ because $\tilde{\phi}(k|k) \in [-1,1]$. Features $\boldsymbol{C}_{11}$, $\boldsymbol{C}_{12}$ and $\boldsymbol{C}_{13}$ are also within $[-1,1]$ based on the definition of $\boldsymbol{u}_5$, and $\boldsymbol{u}_8$ to $\boldsymbol{u}_{10}$. If (8) is embedded into ($\mathcal{LS}$), then the absolute value of the one-step ahead prediction is bounded by:

$$|\hat{y}'(k+1|k)| = |\sum_{w,j} a_{[i],w,j}\boldsymbol{C}_{w,j}| \leqslant \sum_{w,j} |a_{[i],w,j}||\boldsymbol{C}_{w,j}| \leqslant 1$$

Given $\bar{\phi} \geqslant 1$ and $|\phi_1^*(k+1|k)| = |\hat{y}'(k+1|k)| \leqslant 1$, there is $\tilde{\phi}(k+1|k) \in [-1,1]$. By repeating this process, we have $|\hat{y}'(k+M|k)| < 1$, $\forall M$-step prediction.

Proposition 1 shows that when $\bar{\phi} \geqslant 1$, the resulting absolute value of output prediction $\hat{y}'(k+M|k)$ is bounded by 1. Hence, the deviation output $y'(k)$ should be pre-scaled to the range $[-1,1]$ in advance. Then, the prediction error $|\hat{y}'(k+M|k) - y'(k+M)|$ is also bounded. Moreover, ($\mathcal{LS}$) is feasible even with (8) embedded because zero vector is always a feasible solution.

## IV. Results and Discussion

Six datasets in Table I are modeled using the proposed regression method with feature selection. For comparison, we also build SVR, RF, and GBDT models to predict $y'(k)$ for each weather type based on the basic regressor $\phi(k)$. These classical approaches are implemented using scikit-learn package 0.24. The GBDT is implemented through LightGBM [20].

Here the model performance is evaluated based on the MSE of 1-4 steps prediction, shown in (9),

$$\text{MSE} = \frac{\sum_{k=1}^{N-3}\sum_{h=1}^{4}(\hat{y}(k+h|k) - y(k+h))^2}{4(N-3)} \tag{9}$$

where $N$ is the number of data instances in a dataset. All model parameters are identified through minimizing the predication error on the training set. The hyperparameters of compared approaches (SVR, RF, LightGBM) are tuned by assessing the model performance on validation datasets. Here data shuffling and cross validation are not implemented because following chronological order is important to the application of this predictor. For SVR, its hyperparameters, including regularization, number of support vectors, type of kernel functions, are tuned through a library function GridSearchCV in scikit-learn. For RF, its hyperparameters are number of trees, maximum number of features for splitting, maximum number of tree levels, minimal number of data points before node splitting, and minimal number of data points in a leaf. For LightGBM, its hyperparameters are tuned using the Fast and Lightweight AutoML (FLAML) [21]. In addition, the max depth of lightGBM is also tuned to achieve better performance. Finally, the testing dataset is used to compare the true performance of all considered methods.

Algorithms 1 and 2 are implemented to select features for our model. Some of high-order polynomial features in $\boldsymbol{C}_0 - \boldsymbol{C}_{10}$ are selected for fair and haze weather. For cloudy weather, besides $\boldsymbol{C}_0 - \boldsymbol{C}_{10}$, $\boldsymbol{C}_{11} - \boldsymbol{C}_{13}$ are also chosen by algorithms to construct prediction models.

Dataset 4 is used as an example to illustrate the training results. The one-step prediction on Figs. 4-6 shows that the

proposed regression model achieves high accuracy for fair and haze weather but is less accurate for cloudy days. The same observation can be found for other datasets. The cloud coverage highly impacts the solar power generation and a quantitative description of cloud could be more helpful in the future research.

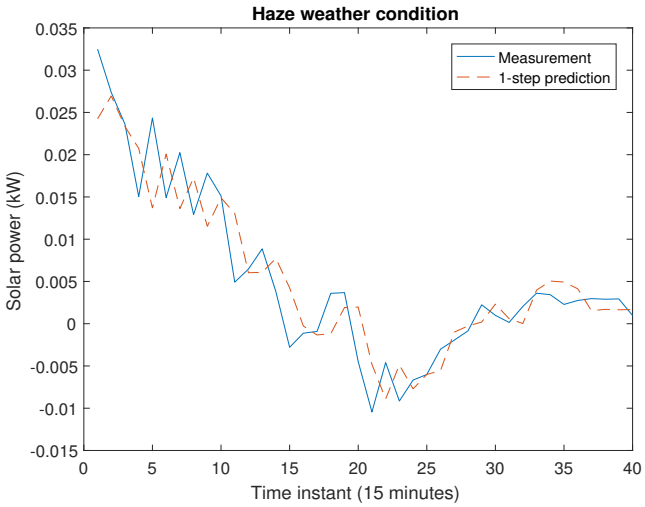


Fig. 4. 1-step ahead predictions on training set 4 haze weather.

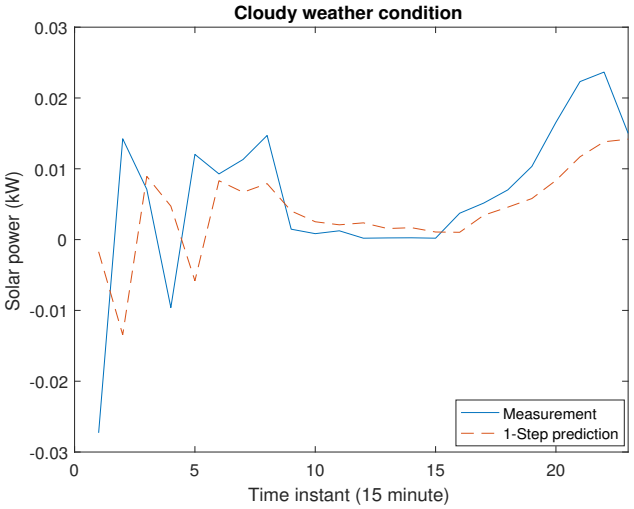


Fig. 5. 1-step ahead predictions on training set 4 cloudy weather.

Tables II-III present combined 1-4 step prediction errors, shown in (9), for each dataset. Table II shows the MSE on each validation dataset. Our method minimizes MSE on the validation set via feature selection, whereas other methods do this by tuning different hyperparameters. GridSearchCV enables SVR to achieve the best performance on validation datasets through exhaustive search. However, overly tuning hyperparameters may degrade the predictive generality.

The MSE on each testing dataset is a more important performance index to all data-driven models. Table III shows that the proposed model is only slightly worse than SVR on datasets 2, but is much better than all considered classical

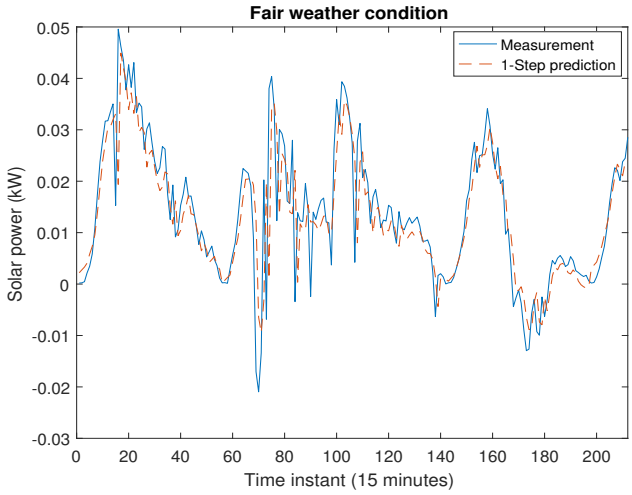


Fig. 6. 1-step ahead predictions on training set 4 fair weather.

machine learning methods on datasets 1, 3, 4, 5 and 6. This result implies that carefully designing and selection feature set for a simple regression model may lead to better results than advanced machine learning methods based on raw features. The constrained least squares ($\mathcal{LS}$) can be solved rapidly, and thus is suitable for the proposed feature selection procedure. A future investigation may combine feature selection with SVR to improve its prediction performance.

Figs. 7-8 show the measured and predicted values using the proposed approach on testing datasets 3 and 4. Dataset 3 is chosen because the measured power deviates from $\bar{y}$ with large fluctuations. Dataset 4 includes more fair weather data, and thus is less challenging than dataset 3. The prediction error in Fig. 7 is obvious, whereas in Fig. 8 is much smaller. It is not surprising that four-step prediction is less accurate than one-step prediction, but the difference is not too significant. Future work can be done to incorporate multi-step prediction into the model development on the training set.

TABLE II
COMBINE 1-4 STEPS MSE ON THE VALIDATION DATASET

| | Proposed Model | SVR | RF | LightGBM |
|---|---|---|---|---|
| Dataset 1 | 2.448e-4 | 2.302e-4 | 15.341e-4 | 13.600e-4 |
| Dataset 2 | 9.335e-4 | 3.810e-4 | 39.769e-4 | 35.705e-4 |
| Dataset 3 | 3.826e-4 | 4.920e-4 | 13.179e-4 | 9.087e-4 |
| Dataset 4 | 0.638e-4 | 0.627e-4 | 2.253e-4 | 1.543e-4 |
| Dataset 5 | 10.106e-4 | 5.242e-4 | 27.986e-4 | 32.700e-4 |
| Dataset 6 | 0.844e-4 | 0.825e-4 | 1.298e-4 | 1.179e-4 |

## V. CONCLUSION

A regression model is developed for one-hour ahead solar power prediction based on the weather data. The raw solar power generation data is detrended and combined with temperature, dew point, humidity, wind speed to form a basic feature vector. Next, this basic feature is augmented through Chebyshev polynomial and trigonometric functions.

TABLE III

COMBINED 1-4 STEPS MSE ON THE TESTING DATASET

| | Proposed Model | SVR | RF | LightGBM |
|---|---|---|---|---|
| Dataset 1 | 6.532e-4 | 7.026e-4 | 11.898e-4 | 18.233e-4 |
| Dataset 2 | 5.441e-4 | 5.423e-4 | 10.597e-4 | 7.700e-4 |
| Dataset 3 | 3.007e-4 | 5.722e-4 | 12.662e-4 | 14.026e-4 |
| Dataset 4 | 0.614e-4 | 2.525e-4 | 2.683e-4 | 2.679e-4 |
| Dataset 5 | 0.869e-4 | 6.452e-4 | 1.725e-4 | 1.302e-4 |
| Dataset 6 | 0.546e-4 | 0.987e-4 | 1.601e-4 | 1.314e-4 |

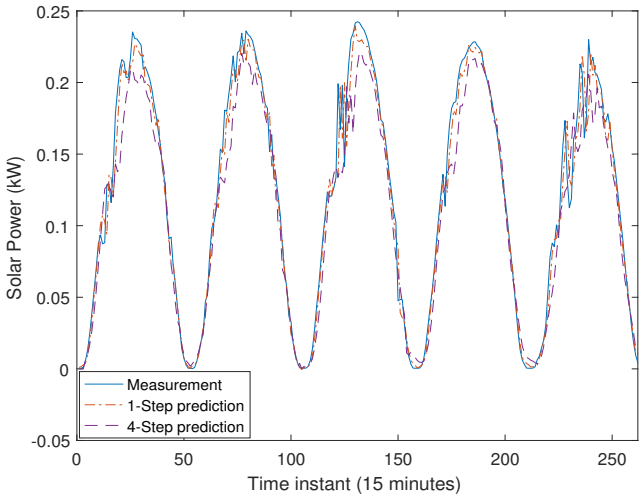


Fig. 7. Solar power 1-step and 4-step ahead predictions on testing set 3.

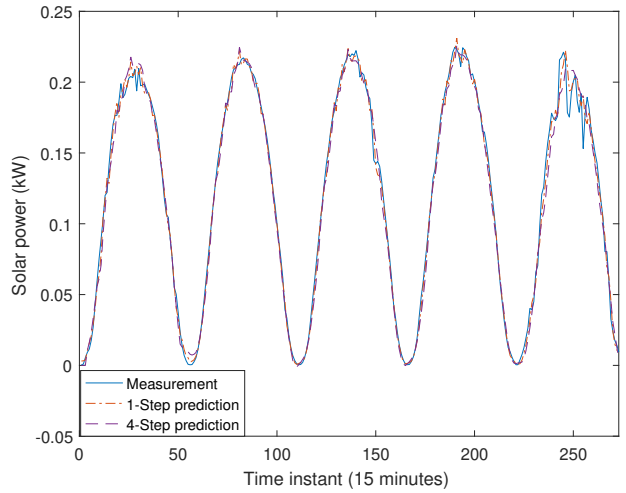


Fig. 8. Solar power 1-step and 4-step ahead predictions on testing set 4.

A linear combination model of resulting high-dimensional features is developed, whose coefficients are identified based on the training dataset. The feature space is further refined on the validation dataset, and the boundedness of multi-step prediction is shown. Finally, the proposed method is compared with classical machine learning methods, such as SVR, RF, and GBDT, on several testing datasets to demonstrate its superiority in prediction accuracy.

## REFERENCES


[1] P. Lauret, C. Voyant, T. Soubdhan, M. David, and P. Poggi, "A benchmarking of machine learning techniques for solar radiation forecasting in an insular context," *Solar Energy*, vol. 112, pp. 446-457, 2015.

[2] E. F. Camacho and M. Berenguel, "Control of solar energy systems," *IFAC Proceedings Volumes*, vol. 45, pp. 848-855, 2012.

[3] C. W. Chow, B. Urquhart, M. Lave, A. Dominguez, J. Kleissl, J. Shields, and B. Washom, "Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed," *Solar Energy*, vol. 85, no. 11, pp. 2881-2893, 2011.

[4] F. Antonanzas-Torres, R. Urraca, J. Polo, O. Perpñán-Lamigueiro, R. Escobar, "Clear sky solar irradiance models: A review of seventy models," *Renewable and Sustainable Energy Reviews*, vol. 107, pp. 374-387, 2019.

[5] C. T. M. Clack, "Modeling solar irradiance and solar PV power output to create a resource assessment using linear multiple multivariate regression," *Journal of Applied Meteorology and Climatology*, vol. 56, pp. 109-125, 2017.

[6] P. Bacher, H. Madsen, and H. A. Nielsen, "Online short-term solar power forecasting," *Solar Energy*, vol. 83, no. 10, pp. 1772-1783, 2009.

[7] B. Chowdhury and S. Rahman, "Forecasting sub-hourly solar irradiance for prediction of photovoltaic output," In: IEEE Photovoltaic Specialists Conference, 19th, New Orleans, LA, May 4-8, 1987.

[8] F. Loeper, P. Schaumann, M. Langlard, R. Hess, R. Bäsmann, and V. Schmidt, "Probabilistic prediction of solar power supply to distribution networks, using forecasts of global horizontal irradiation," *Solar Energy*, vol. 203, pp. 145-156, 2020.

[9] R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra, "Solar forecasting methods for renewable energy integration," *Prog. Energy Combust. Sci.* vol. 39, no. 6, pp. 535-576, 2013.

[10] A. Sfetsos and A. H. Coonick, "Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques," *Solar Energy*, vol. 68, no. 2, pp.169-178, 2000.

[11] L. Hontoria, J. Aguilera, J. Riesco, and P. Zufiria, "Recurrent neural supervised models for generating solar radiation synthetic series," *Journal of Intelligent and Robotic Systems*, vol. 31, pp. 201-221, 2001.

[12] J. Cao and X. Lin, "Study of hourly and daily solar irradiation forecast using diagonal recurrent wavelet neural networks," *Energy Conversion and Management*, vol. 49, no 6, pp. 1396-1406, 2008.

[13] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," IEEE International Conference on Smart Grid Communications (SmartGridComm), 2011.

[14] L. Gigoni, A. Betti, E. Crisostomi, A. Franco, M. Tucci, F. Bizzarri and D. Mucci, "Day-ahead hourly forecasting of power generation from Photovoltaic plants," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 2, pp. 831-842, 2018.

[15] https://www.californiadgstats.ca.gov/downloads

[16] M. Onel, C. A. Kieslich, and E. N. Pistikopoulos, "A nonlinear support vector machine-based feature selection approach for fault detection and diagnosis: Application to the Tennessee Eastman process," *AIChE Journal*, vol. 65, pp. 992-1005, 2019.

[17] M. Onel, C. A. Kieslich, Y. A. Guzman, C. A. Floudas, E. N. Pistikopoulos, "Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection," *Computers & Chemical Engineering*, vol. 116, pp. 503-520, 2018.

[18] R. Rendall, I. Castillo, A. Schmidt, S. Chin, L. H. Chiang, and M. Reis, "Wide spectrum feature selection (WiSe) for regression model building," *Computers & Chemical Engineering*, vol. 121, pp. 99-110, 2019.

[19] D. Shah, J. Wang, and Q. P. He, "Feature engineering in big data analytics for IoT-enabled smart manufacturing-Comparison between deep learning and statistical learning," *Computers & Chemical Engineering*, vol. 141, pp. 106970, 2020.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. "Lightgbm: A highly efficient gradient boosting decision tree," In Advances in Neural Information Processing Systems, pp. 3149-3157, 2017.

[21] C. Wang, Q. Wu, M Weimer, E. Zhu, "FLAML: A fast and lightweight AutoML library," Fourth Conference on Machine Learning and Systems (MLSys), 2021.