2021 IEEE Conference on Technologies for Sustainability (SusTech) | 978-1-6654-1927-7/20/$31.00 ©2021 IEEE | DOI: 10.1109/SusTech51236.2021.9467464

Solar Power Prediction in Different Forecasting Horizons Using Machine Learning and Time Series Techniques

Kesh Pun
Wichita State University
Wichita, KS USA
kbpun@shockers.wichita.edu

Saurav MS Basnet
Wentworth Institute of Technology
Boston, MA USA
basnets@wit.edu

Ward Jewell
Wichita State University
Wichita, KS USA
ward.jewell@wichita.edu

***Abstract*—Solar power generation is highly intermittent, nonlinear, and variable in nature. The increase in penetration level of solar energy resources poses technical challenges. An accurate forecasting model is crucial to minimizing these technical issues. Therefore, choosing the right forecasting technique for the right forecasting horizon is vital. In this study, the performance analysis of machine learning and time series forecasting techniques for various forecasting horizons has been investigated. Its accuracy, root mean square error (RMSE), and mean absolute error (MAE) have been compared to other techniques.**

***Index Terms*—Renewable Energy Resources, Forecasting, Machine Learning, Time Series.**

I. Introduction

Solar energy is one of the main sources of renewable energy. Extensive research and the increasing use of photovoltaic (PV) cells to generate utility-level electricity is driving the drop in cost, which can be easily seen by the 74% price difference of PV panels between 2010 and 2018 [1]. Solar power generation is highly nonlinear, variable, intermittent, and unpredictable in nature. Accurate solar power forecasting is crucial for unit commitment, grid stability, and cost effectiveness. Different forecasting techniques provide a variance in accuracy for different forecasting horizons. Therefore, to select the suitable forecasting technique with available data at a particular forecasting horizon is an essential task for power system workers to maintain the best operation.

The review of literature on solar power forecasting, which is primarily based on [2]-[4], involves mainly three types of methods: physical, statistical, and hybrid. Machine learning falls under statistical methods. Physical methods are numerical weather prediction (NWP), sky imagery, and satellite imaging. Statistical methods are more accurate for a short time horizon (one to six hours), while physical models are suitable for the long term [5].

Forecasting accuracy depends mainly on the feature vector that is input to the machine learning model. Buwei et al. [6] proposed an extension of the support vector machine (SVM) model based on data fusion, whereby they collected data from NWP, remote sensing, and station information. This extended model outperformed the SVM and artificial neural network (ANN) models in all time horizons, from 15 minutes to four hours. García-Hinde et al. [7] proposed a bootstrapped support vector regression (SVR) method where feature selection is performed using the bootstrapping method for solar irradiation prediction. This method showed the robustness over other feature selection processes. Masood et al. [8] examined machine learning techniques and key factors affecting PV generation forecasting, examining the correlation between PV power output and solar irradiance, air temperature, relative humidity, wind direction, and wind speed. Their study showed that irradiance, air temperature, and wind direction greatly impact PV power generation. The authors in [9] compared fuzzy logic and the ANN model. Here, fuzzy logic was developed using actual solar PV data, considering constant temperature and variable irradiance. Results showed that fuzzy logic outperformed the ANN model in terms of average error. The authors in [10] conducted a comparative study for very short-term solar power forecasting in ten-minute granularity for different forecasting techniques. Their results proved that the autoregressive integrated moving average (ARIMA) forecasting technique outperformed all other machine learning techniques. The authors in [11] compared the performance of two models—the least squares support vector regression (LS-SVR) and the feedforward neural network (FFNN)—using various performance metrics for intra-hour solar power forecasting. The input variables used in this study were solar irradiance and solar temperature. Results show that the LS-SVR gives better results in comparison to FFNN. The authors in [12] basically used the aerosol index (AI) and other physical parameters, such as temperature, wind speed, wind direction, relative humidity, pressure, and historical global horizontal irradiance (GHI), to train the ANN method to forecast solar power generation. Results show that the model gave a 4.67% mean squared error (MSE).

978-1-6654-1927-7/21/$31.00 ©2021 IEEE

Extensive research has been done to improve the machine learning model, the performance of which depends on the selection of feature vector and forecasting horizon. In this study, a comparative analysis of performance of SVR, kernel ridge regression (KRR), least absolute shrinkage and selection operator (LASSO), ridge regression (RR), and autoregressive moving-average (ARMA) for five different forecasting horizons is discussed using the same feature vector.

Figure 1 shows the variability of solar power generation over the year 2006 in Wichita, Kansas, while Figure 2 shows the variability of solar power generation over a partly cloudy day.

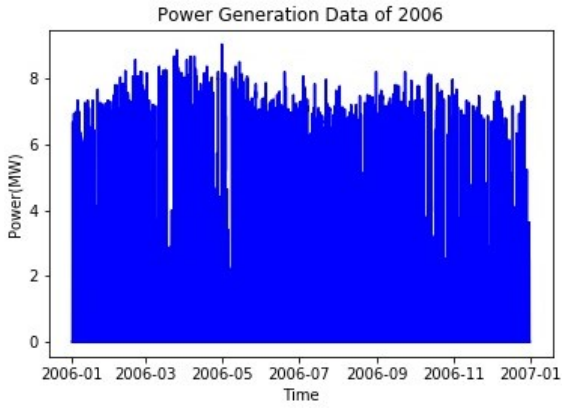

Fig. 1. Power output of 2006 in Wichita.

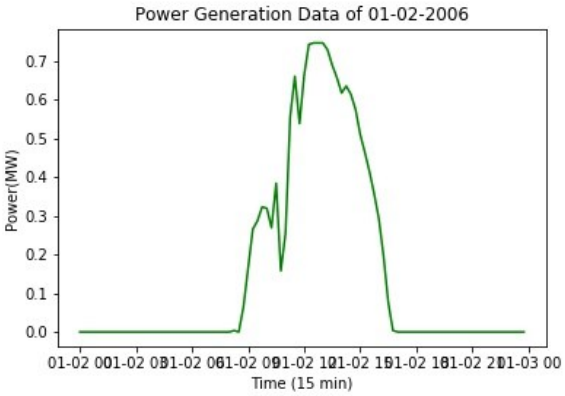

Fig. 2. Solar power data on partly cloudy day

II. Methodology

A photovoltaic cell is a device that converts sunlight directly into electricity. Basically, the PV cell is a semiconductor diode whose $p-n$ junction is exposed to sunlight. Sunlight is composed of photons of solar energy. When these photons strike the $p-n$ junction, it accelerates the movement of free electrons, which in turn produces electricity.

Mathematically, and referring to Figure (3), the PV cell can be modelled as follows:

$$I = I_{pv} - I_D - I_{Rp} \quad (1)$$

where I_{pv} is the current generated by the incident light, I_{Rp} is the current through parallel resistor R_P, and I_D is the Shockley diode equation, which can be represented as

$$I_D = I_0 \left[e^{\left(\frac{qV}{akT}\right)} - 1 \right] \quad (2)$$

Now, equation (1) can be rewritten, using equation (2) and considering Figure 3, as

$$I = I_{pv} - I_0 \left[e^{\left(\frac{V+R_S I}{V_t a}\right)} - 1 \right] - \frac{V+R_S I}{R_p} \quad (3)$$

where I_0 is the reverse saturation or leakage current of the diode, q is the electron charge $(1.60217646 \times 10^{-19} C)$, k is the Boltzmann constant $(1.3806503 \times 10^{-23} \frac{J}{K})$, T is the temperature of the $p-n$ junction (T), a is the diode ideal constant, and $V_t = \frac{N_S kT}{q}$ is the thermal voltage of the array with N_s cells connected in series. Moreover, R_s is the equivalent series resistance, and R_p is the equivalent parallel resistance of the array [13].

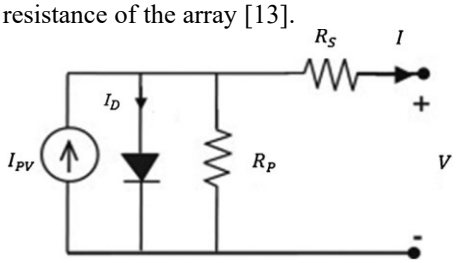

Fig. 3. Equivalent circuit of single-diode model of PV cell.

Solar power generation is dependent on many factors, and apart from weather data, important factors like lagged values, lagged day values, and solar irradiance values are correlated with solar power generation. Mathematically, these factors can be related to solar power generation as

$$P_g = y = w^T x + b \quad (4)$$

where P_g is the solar power generation, w is the coefficient vector of dependent variables, x is the input variables vector, and b is the bias.

By knowing the values of present and lagged values of power generation and solar irradiance, we can predict the solar power generation of a time step ahead using equation (4). There are different techniques to estimate the coefficient vector w and bias b in machine learning techniques. In this study, we have selected SVR, KRR, LASSO, and RR machine learning techniques.

A. Support Vector Regression

Support vector regression is a machine learning algorithm that uses past data to train it. It was proposed by Vapnik [14]. According to [15], suppose that training data are given as follows:

$T = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in \mathcal{X} \times \mathbb{R}$, training data

$\mathcal{X} = \mathbb{R}^d$, space of input patterns

Theoretically, the nonlinear relationship between x and y can be linearly described by function *f(x)* in a high-dimensional feature space:

$$f(x) = \omega.\varphi(x) + b \quad (5)$$
$$\omega \in \mathcal{X}, b \in \mathbb{R}$$

where ω is the weight vector, b is the bias, and $\varphi(x)$ is the high-dimensional feature space, which is linearly mapped from the input space x.

In ε-SV regression, there is a trade-off between the most ε-deviation and ω-flatness. Flatness of ω can be ensured by minimizing its norm. Mathematically,

$$\text{Minimize } \frac{1}{2}\|\omega\|^2 \quad (6)$$
$$\text{Subject to } y_i - \omega.\varphi(x) - b \le \varepsilon$$
$$\omega.\varphi(x) + b - y_i \le \varepsilon$$

Introducing slack variables to cope with infeasible constraints, if any of (5),

$$\text{Minimize } \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) \quad (7)$$
$$\text{Subject to } y_i - \omega.\varphi(x) - b \le \varepsilon + \xi_i$$
$$\omega.\varphi(x) + b - y_i \le \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \ge 0$$

where C is the user-defined constant and is greater than zero, which penalizes error greater than user-defined deviation, ε.

Finally, the SVR function is derived as

$$f(x) = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\,K(x, x_i) + b \quad (8)$$

which is the final equation of support vector regression, where $K(x, x_i)$ is the kernel function. For a radial basis function (RBF) kernel,

$$K(x, x_i) = \exp\left(-\gamma\left\|x_i - x_j\right\|^2\right) \quad (9)$$

where γ is the kernel parameter and is defined by the user.

B. Kernel Ridge Regression

Kernel ridge regression [16] combines ridge regression with the kernel trick. Thus, it learns a linear function in the data space transformed by the kernel.

Give the graining data as

$$S = ((x_1, y_1), (x_2, y_2), \ldots (x_i, y_i), \ldots (x_l, y_l)) \quad (10)$$

where $x_i \in R^n$, and $y_i \in R$.

Construct the linear function as

$$g(x) = \langle w, X\rangle = w^T x = \sum_{i=1}^{n} w_i x_i \quad (11)$$

for the least squares approximation, $g(x) \approx y$.

Define error as

$$f(x, y) = y - g(x) = \xi \quad (12)$$

The classical way to perform minimization in order to obtain optimal parameters of the regression is as follows:

$$C(w) = \frac{1}{2}\sum_i (y_i - w^T x_i)^2 \quad (13)$$

To work in feature space, regularize the above function in order to avoid overfit using the following cost function:

$$C = \frac{1}{2}\sum_i (y_i - w^T x_i)^2 + \frac{1}{2}\lambda\|w\|^2 \quad (14)$$

Use the optimality condition as

$$\frac{\partial C}{\partial w} = -x^T y + x^T x w + \lambda w = 0 \quad (15)$$

For $\lambda > 0$, the inverse exists as

$$w = (x^T x + \lambda I)^{-1} x^T y \quad (16)$$

Rearranging (16) yields

$$x x^T \alpha + \lambda\alpha = y$$
$$\alpha = (G + \lambda I)^{-1} y \quad (17)$$

where $G = xx^T$.

Use the gram matrix as

$$K = G = xx^T$$

which is composed of inner products of data:

$$K_{i,j} = \langle x_i, x_j\rangle$$

For the new observation (x', y'), the prediction is

$$\widehat{y'} = w^T x' \quad (18)$$

Recall (16):

$$w = (x^T x + \lambda I)^{-1} x^T y$$

Then,

$$\widehat{y'} = y^T (xx^T + \lambda I)^{-1} x x' \quad (19)$$
$$\widehat{y'} = y^T (K + \lambda I)^{-1} \kappa \quad (20)$$

where

$$\kappa = xx'$$

and

$$K_{ij} = x_i x_j$$
$$\kappa_i = x_i x'$$

Here, we can clearly see that $\widehat{y'}$ depends on K and κ, rather than x.

If we transform $x \to \phi(x)$, then we must calculate K and κ by the following function:

$$K_{ij} = f(x_i, x_j)$$
$$\kappa_i = f(x_i x')$$

One of the kernel functions we can use is the radial basis function, which is given by

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$
$$K(x, x') = \exp\left(-\gamma\|x - x'\|^2\right) \quad (21)$$

where $\gamma = \frac{1}{2\sigma^2}$.

C. LASSO Regression

Least absolute shrinkage and selection operator regression is a regressor method that performs variable selection and regularization in order to improve the prediction accuracy. It was proposed by Rob Tibshirani in 1996 [17] and then improved by others [18]. Mathematically, LASSO consists of a linear model with an added regularization term. Hence, the objective function is

$$\min \frac{1}{2n_{samples}}\|Xw - y\|_2^2 + \alpha\|w\|_1 \quad (22)$$

Thus, LASSO solves the minimization of the least squares penalty with additional $\alpha\|w\|_1$, where α is a constant, and $\|w\|_1$ is the l_1-norm of the coefficient vector. It uses the coordinate descent as the algorithm to fit the coefficients.

D. Ridge Regression

Ridge regression improves ordinary least squares problems by imposing a penalty on the size of the coefficients [19]. Therefore, the objective function on ridge regression becomes

$$min\|Xw - y\|_2^2 + \alpha\|w\|_2^2 \quad (23)$$

The parameter $\alpha \geq 0$ controls the amount of shrinkage, whereby the larger the value of α, the greater the amount of shrinkage.

E. ARMA Process

The ARMA process is a combination of the autoregressive and moving-average process. Mathematically, $ARMA_{(p,q)}$ can be defined as

$$\begin{aligned} X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} \\ = Z_t + \theta_1 Z_{t-1} \\ + \cdots + \theta_q Z_{t-q} \end{aligned} \quad (24)$$

where $\{Z_t\} \sim WN(0, \sigma^2)$ and the polynomials $\left(1 - \phi_1 z - \cdots - \phi_p z^p\right)$ and $\left(1 + \theta_1 z + \cdots + \theta_q z^q\right)$ have no common factors.

The process $\{X_t\}$ is said to be an ARMA process with mean μ, if $\{X_t - \mu\}$ is an $ARMA_{(p,q)}$ process [20].

F. Model Evaluation Criterion

For the model evaluation, R2_score, the MSE and mean absolute error (MAE) criterion are used, which can be formulated as follows:

$$R2_{Score} = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (25)$$

where

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$e_t = y_t - \widehat{y_t} \quad (26)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \quad (27)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |e_i| \quad (28)$$

where y_t is the real power at time t, y_t is the predicted power at time t, e_t is the error at time t, and *n* is the number of forecasting.

III. Experimental Results

Solar power generation data has been collected from national renewable energy laboratory (NERL) at location (37.65, –97.05) for 2006, and correspondingly, all three components of clear sky irradiances—Global Horizontal Irradiance (GHI), Direct Normal Irradiance (DNI), and Diffuse Horizontal Irradiance (DHI) for the same location—have been simulated using pvlib (Python programming module) [21], which is based on the Ineichen model. The sampling period of solar power generation is 5 minutes. Therefore, it was converted into 15 minutes, 30 minutes, 1 hour, 2 hours, 3 hours, and 4 hours, by averaging values over those periods. In this study, only historical data of solar power generation and clear sky irradiance has been used.

Throughout this work, all data processing, data manipulation, mathematical computation, and prediction were implemented in Python using the following libraries: Pandas [22]-[23], Matplotlib[24], Scikit-learn[25], NumPy, and Excel.

A. Feature Engineering

Feature engineering is a very crucial part of the machine learning technique. The performance of the model depends on the choice of the input feature vectors. Feature vectors were created from the historical solar power generation data and clear sky irradiance.

Solar power generation was correlated with lag values, and for a clear sky condition, its value is very close to day-ahead values. Therefore, three lag values and two lag day values were considered for the feature vectors, formulated as follows:

$$y_{lag1} = y_{t-1} \quad (29)$$

$$y_{lag2} = y_{t-2} \quad (30)$$

$$y_{lag3} = y_{t-3} \quad (31)$$

$$y_{lag1day} = y_{t-24} \quad (32)$$

$$y_{lag2day} = y_{t-48} \quad (33)$$

The main source of solar power generation is irradiance. Therefore, all three components of solar irradiance (GHI, DNI, and DHI) have been included in the feature vector.

To improve the performance of the machine learning model, the feature vector was scaled into a 0 – 1 scale using the following formula:

$$X_{std} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \quad (34)$$

$$X_{scaled} = X_{std} * (max - \min) + min \quad (35)$$

where the min, max = feature range, which is considered to be 0 – 1.

B. Selection of Parameters

Machine learning models are very flexible and sensitive to their hyperparameter selection values. Therefore, hyperparameters were optimally selected to avoid overfitting and underfitting of the machine learning model as well as to give the best performance on the basis of root mean square error (RMSE). For this, the GridSearch cross-

validation technique with a five-fold cross-validation was applied. Optimal hyperparameters of machine learning models are shown in Table I.

Table I
Parameters for Train Data Set

Time (hour)	SVR		KRR		LASSO	RR
	C	γ	α	γ	α	α
0.25	1000	0.01	0.001	0.1	1e-5	0.1
0.5	100	0.01	0.001	0.1	1e-5	0.1
1	1000	0.1	0.001	0.1	1e-5	0.01
2	1000	0.1	0.1	1.0	1e-5	0.1
3	1000	0.1	0.001	0.1	1e-5	0.01
4	100	0.1	0.01	0.1	1e-5	0.1

To implement the ARMA technique, time series data should be stationary. Stationarity of the solar power generation has been tested using the Augmented Dickey-Fuller (ADF) test [26]. This test is based on the presence of a unit root using the null hypothesis. If the p-value is less than 0.05, then it can be interpreted that the null hypothesis is rejected, meaning there is no presence of a unit root. After confirming the stationarity, optimal parameters of ARMA were selected using the Akaike information criterion (AIC) [27]-[28]. Table II shows the p-values and optimal parameters of the ARMA technique.

Table II
Parameters for Test Data Set

Time	P	q	p-value <0.05
15 min	2	1	0
30 min	2	1	2×10^{-17}
1 hour	5	3	1.45×10^{-19}
2 hour	5	2	2.18×10^{-15}
3 hour	4	5	3.84×10^{-12}
4 hour	5	4	7.48×10^{-10}

C. Results

The accuracy (R2_score) of different models in different forecasting horizons is shown in Figure 4 and Table III. From the data, it can be concluded that R2_scores of all the machine learning and time series techniques are similar for the 15-minute and 30-minute forecasting horizons, while they deviate from 1-hour onwards. SVR and KRR gave similar performances and outperformed all other techniques in all forecasting horizons. LASSO and RR gave similar performances in all forecasting horizons. The ARMA technique outperformed LASSO and RR for the 3- and 4-hour forecasting horizons.

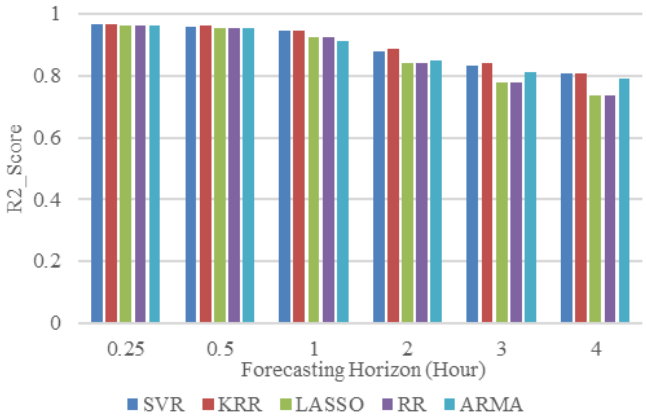

Fig. 4. R2_score of different techniques in different forecasting horizons.

Table III
R2_Score of Different Techniques in Different Forecasting Horizons

Time (hour)	SVR	KRR	LASSO	RR	ARMA
0.25	0.9643	0.9655	0.9621	0.962	0.9635
0.5	0.957	0.962	0.9548	0.9547	0.9536
1	0.9446	0.9445	0.9254	0.9255	0.9134
2	0.8801	0.8861	0.8423	0.8422	0.8481
3	0.8327	0.841	0.7793	0.7796	0.8117
4	0.8069	0.8074	0.734	0.7341	0.7896

Root mean square errors for all forecasting techniques in all forecasting horizons are presented in Figure 5 and Table IV. The RMSE is very similar for 15 and 30 minutes for all techniques. SVR and KRR gave similar results in all time horizons and also outperformed all other techniques. LASSO and RR gave very similar results in all time horizons. ARMA performed better than LASSO and RR techniques in the 2-, 3-, and 4-hour forecasting horizons in terms of RMSE.

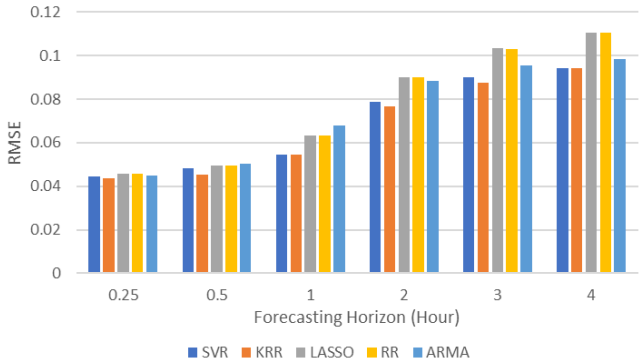

Fig. 5. RMSE of different techniques in different forecasting horizons.

Table IV
RMSE of Different Techniques in Different Forecasting Horizons

Time (hour)	SVR	KRR	LASSO	RR	ARMA
0.25	0.0444	0.0436	0.0458	0.0458	0.0449
0.5	0.0484	0.0455	0.0497	0.0497	0.0503
1	0.0545	0.0545	0.0633	0.0632	0.0681
2	0.0786	0.0766	0.0901	0.0901	0.0884
3	0.0899	0.0877	0.1033	0.1032	0.0954
4	0.0943	0.0942	0.1107	0.1106	0.0984

The mean absolute error for all techniques for all time horizons is shown in Figure 6 and Table V. LASSO and RR gave similar results in terms of MAE for all forecasting horizons. KRR techniques outperformed all other techniques. The ARMA technique gave a lower MAE than LASSO and RR in three- and four-hour forecasting horizons.

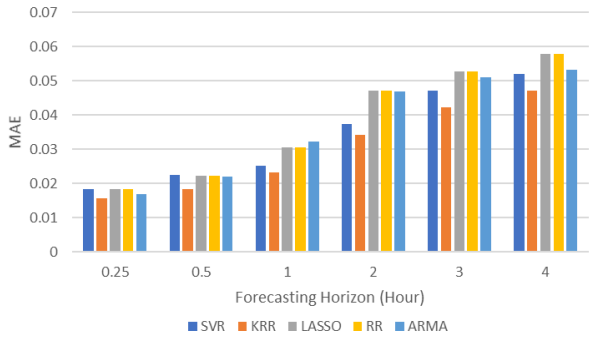

Fig. 6. MAE of different techniques on different forecasting horizons.

Table V
MAE of Different Techniques on Different Forecasting Horizons

Time (hour)	SVR	KRR	LASSO	RR	ARMA
0.25	0.0183	0.0157	0.0185	0.0185	0.017
0.5	0.0225	0.0183	0.0223	0.0223	0.022
1	0.0253	0.0233	0.0307	0.0307	0.0322
2	0.0375	0.0342	0.0471	0.0471	0.047
3	0.0471	0.0424	0.0528	0.0528	0.051
4	0.052	0.0472	0.0578	0.0578	0.0532

IV. Discussion and Conclusion

After examining the performance metrics of some selected machine learning and timeseries techniques, it can be concluded that kernel ridge regression outperformed all other techniques in terms of mean absolute error, while it gave similar results with support vector regression in terms of the R2_score and root mean square error. For the first 15 and 30 minutes of the forecasting horizon, all techniques gave similar results in terms of R2_score, RMSE, and MAE. SVR and KRR performed in a similar way and outperformed all other techniques, with the lowest R2_score being 0.80 for the 4-hour ahead forecasting, which is 0.73 for LASSO and RR, and 0.78 for the ARMA technique. LASSO and RR performed in a very similar manner in terms of all three-evaluation metrics but did not perform better than ARMA for the 3- and 4-hour ahead forecasting horizon. The ARMA technique performed similarly for the 15-minute, 30-minute, 1-hour, and 2-hour forecasting horizons, although better than the LASSO and RR techniques in the 3- and 4-hour forecasting horizons.

V. Future Work

Limitations of this study are that it does not include weather data, and it uses the same feature vector for all machine learning techniques in a particular forecasting horizon. An advanced comparison of the performance of machine learning techniques could be studied. This would include weather data and application of the feature selection technique so that suitable features will be applied to suitable machine learning techniques for a particular forecasting horizon.

References

[1] IRENA (2019), "Renewable Power Generation Costs in 2018," International Renewable Energy Agency, Abu Dhabi.

[2] M. N. Akhter, S. Mekhilef, H. Mokhlis, and N. Mohamed Shah, "Review on forecasting of photovoltaic power generation based on machine learning and metaheuristic techniques," IET Renewable Power Generation, vol. 13, no. 7, pp. 1009-1023, 2019, doi: 10.1049/iet-rpg.2018.5649.

[3] Rich H. Inman, Hugo T. C. Pedro, and Carlos F. M. Coimbra, "Solar forecasting methods for renewable energy integration," Progress in Energy and Combustion Science, vol. 39, no. 6, pp. 535-576, 2013.

[4] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de Pison, and F. Antonanzas-Torres, "Review of photovoltaic power forecasting," Solar Energy, vol. 136, pp. 78-111, October 2016.

[5] S. Jaidee and W. Pora, "Very short-term solar power forecast using data from NWP model," 2019 4th International Conference on Information Technology (InCIT), Bangkok, Thailand, 2019, pp. 44-49, doi: 10.1109/INCIT.2019.8912012.

[6] W. Buwei, C. Jianfeng, W. Bo, and F. Shuanglei, "A solar power prediction using support vector machines based on multi-source data fusion," 2018 International Conference on Power System Technology (POWERCON), Guangzhou, China, 2018, pp. 4573-4577, doi: 10.1109/POWERCON.2018.8601672.

[7] O. García-Hinde, and et al., "Feature selection in solar radiation prediction using bootstrapped SVRs," 2016 IEEE

Congress on Evolutionary Computation (CEC), Vancouver, BC, 2016, pp. 3638-3645, doi: 10.1109/CEC.2016.7744250.

[8] N. Masood, M. I. Asif, A. M. Alam, S. R. Deeba, and T. Aziz, "Forecasting of photovoltaic power generation: techniques and key factors," 2019 IEEE Region 10 Symposium (TENSYMP), Kolkata, India, 2019, pp. 457-461, doi: 10.1109/TENSYMP46218.2019.8971337.

[9] Z. P. Ncane and A. K. Saha, "Forecasting solar power generation using fuzzy logic and artificial neural network," 2019 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA), Bloemfontein, South Africa, 2019, pp. 518-523, doi: 10.1109/RoboMech.2019.8704737.

[10] W. Sharika, L. Fernando, A. Kanagasundaram, R. Valluvan, and A. Kaneswaran, "Long-term solar irradiance forecasting approaches—A comparative study," 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS), Colombo, Sri Lanka, 2018, pp. 1-6, doi: 10.1109/ICIAFS.2018.8913381.

[11] A. Fentis, L. Bahatti, M. Mestari, and B. Chouri, "Short-term solar power forecasting using support vector regression and feed-forward NN," 2017 15th IEEE International New Circuits and Systems Conference (NEWCAS), Strasbourg, France, 2017, pp. 405-408, doi: 10.1109/NEWCAS.2017.8010191.

[12] A. Kumar, M. Rizwan, and U. Nangia, "Artificial neural network-based model for short-term solar radiation forecasting considering aerosol index," 2nd IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, 2018, pp. 212-217, 2018, doi: 10.1109/ICPEICES.2018.8897290.

[13] M. G. Villalva, J. R. Gazoli, and E. R. Filho, "Comprehensive approach to modeling and simulation of photovoltaic arrays," IEEE Transactions on Power Electronics, vol. 24, no. 5, pp. 1198-1208, May 2009, doi: 10.1109/TPEL.2009.2013862.

[14] V. Vapnik, The Nature of Statistical Learning Theory. New York, USA: Springer, 1995.

[15] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," Statistics and Computing, vol. 14, pp. 199-222, 2004. [Online]. Available: https://alex.smola.org/papers/2004/SmoSch04.pdf [Accessed: Jan 18, 2020].

[16] Kevin P. Murphy, "Kernelized ridge regression," in Machine Learning: A Probabilistic Perspective, chapter 14.4.3, pp. 492-493. Cambridge, MA: The MIT Press, 2012.

[17] Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani, "Regularization paths for generalized linear models via coordinate descent," Journal of Statistical Software, vol. 33, iss. 1, January 2010.

[18] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale ℓ1-regularized least squares," IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 4, pp. 606-617, December 2007, doi: 10.1109/JSTSP.2007.910971.

[19] Marvin Gruber, Improving Efficiency by Shrinkage: The James–Stein and Ridge Regression Estimators, pp. 7-15. Boca Raton: CRC Press, 1998.

[20] Peter J Brockwell, "Introduction to time series and forecasting," in Springer Texts in Statistics, 3rd ed. Springer, 2016.

[21] William F. Holmgren, Clifford W. Hansen, and Mark A. Mikofski. "pvlib python: A python package for modeling solar energy systems." Journal of Open Source Software, vol. 3, no. 29, p. 884, 2018.

[22] Pandas development team, "Pandas-dev/pandas: Pandas 1.1.3," Feb 2020, Zenobo, version: 1.0.5, doi:10.5281/zenodo.3509134.

[23] Wes McKinney, "Data structures for statistical computing in Python," Proceedings of the 9th Python in Science Conference, Austin, Texas, 2010, pp. 56-61.

[24] J. D. Hunter, "Matplotlib: A 2D graphics environment," Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[25] Pedregosa, and et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[26] David A. Dickey and Wayne A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," Journal of the American Statistical Association, vol. 74, no. 366, pp. 427-431, 1979.

[27] Smith, Taylor G., and et al., pmdarima: ARIMA estimators for Python, 2017, http://www.alkaline-ml.com/pmdarima [Online; accessed 2020-06-27].

[28] H. Akaike, "A new look at the statistical model identification," IEEE Transactions on Automatic Control, vol. 19, no. 6, pp. 716-723, December 1974, doi: 10.1109/TAC.1974.1100705.