A Data-driven Approach for Forecasting State Level Aggregated Solar Photovoltaic Power Production

Mashud Rana
Data61
CSIRO
Sydney, NSW, Australia
mdmashud.rana@data61.csiro.au

Ashfaqur Rahman
Data61
CSIRO
Sandy Bay, TAS, Australia
ashfaqur.rahman@data61.csiro.au

Jiong Jin
School of Software and Electrical Eng.
Swinburne University of Technology
Melbourne, VIC, Australia
jiongjin@swin.edu.au

Abstract—**Reliable forecasting of power output from solar Photovoltaic (PV) systems is indispensable for successful penetration of increasingly high solar PV capacity into the structure of power systems. Forecasting is as equally important for efficient demand management as solar power output is variable in nature. Farm level forecasting is the primary focus of the current literature and various methods are available. However, the grid operators need to have the forecasts at aggregated regional level (e.g. state level) for decision making. Aggregated system level forecasting has received very limited attention in literature with only a very few methods so far. In this paper, we consider the task of forecasting PV power production at aggregated regional level (at state level in Australia) and present a data-driven approach. To develop the prediction models, we investigate three widely used machine learning algorithms for farm level solar power forecasting and two state-of-the-art deep learning algorithms. As the inputs to the prediction algorithms, we study two different feature sets based on the lagged power data and their descriptive statistics. An evaluation of the presented approach using PV power generation data from Australian Energy Market Operator (AEMO) demonstrates up to 6.30% improvement in forecast skill over the baseline models.**

Keywords— solar power forecasting, machine learning, deep learning, data re-sampling

I. INTRODUCTION

The power generation from solar PV systems has been growing with an unprecedented pace in recent years. Since 2007, the power generation from PV systems increased by more than 4300% reaching from 9.2 GW in 2007 to 404.5 GW in 2017 [1]. This trend is expected to continue as many countries have government legislations encouraging the use of renewable energy sources and set priorities to increase the capacity of renewable energy production over conventional fossil and nuclear sources. For example, Australian government aims to produce 30% of its total electricity supply from solar energy by 2050 [2].

Contrast to the power generation from traditional energy sources (e.g. fossil, nuclear), the power output from solar PV systems is highly variable due to its dependence on meteorological condition. The increasingly high penetration of such unstable power generation into the electricity grid originates imbalance between supply and demand which consequently introduces difficulties in real time power dispatching. Accurate forecasting of PV power generation is of utmost importance to overcome this challenge and ensure reliable and economic operations of the grid.

Many approaches have been proposed in the literature for forecasting solar PV power generation. They largely vary both in the application of prediction algorithms and feature sets used as inputs to those prediction algorithms. The most prominent prediction algorithms for solar PV forecasting include Neural Networks (NNs) [3, 4], Support Vector Regression (SVR) [5, 6], Random Forest (RF) and its variants [7], Autoregressive Integrated Moving Average (ARIMA) [8]. In terms of feature sets, the univariate approaches (e.g. [3, 9]) consider only lagged power data whereas multivariate approaches (e.g. [10, 11]) consider weather data in conjunction with lagged power data as inputs to the prediction algorithms.

Reviews (e.g. [12, 13]) of solar power forecasting literature suggest that most of the existing approaches focus on predicting the power output from a standalone PV plant or farm. The available approaches for farm level forecasting are not well suited for forecasting aggregated system level power generation aggregated from a large number of geographically dispersed PV plants with different physical parameters and diverse site-specific weather conditions [10]. Considering the rapidly growing installations of grid connected solar farms, aggregated system level forecasting is more appropriate for practical operations of the power systems [14]. It is particularly important for real time generation scheduling, maintaining reserve requirement to a minimum level, and analysis of contingencies. Hence, the design and operations of modern power systems are significantly dependent on the accurate information about the aggregated power production from distributed solar PV systems.

However, aggregated system level forecasting has been receiving attention only recently with only very few previous studies (e.g. [10, 15]). The main reason for lack of focus on system level forecasting is the unavailability of the data. Most recently AEMO has made power production data publicly available from all grid-connected solar power generation units across different states in Australia. The main objective of this study is to investigate state level aggregated PV power

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

forecasting for Australia using the AEMO data set. Specifically, we consider the task of day ahead forecasting of aggregated PV power output at half-hourly resolution for the state of New South Wales (NSW), Australia. Our main contributions in this paper can be summarized as follows:

1. We present a data-driven approach which considers only past solar PV power data to build and train the prediction model for state level aggregated PV power forecasting. The adopted approach applies three well-established machine learning algorithms (NNs, SVR and RF) and two state-of-the-art deep learning algorithms (Long Short-Term Memory (LSTM) recurrent networks and Convolutional Neural Networks (CNNs)) to develop prediction model. The two deep learning algorithms have not been investigated previously for aggregated system level PV power forecasting.
2. The proposed approach also investigates the predictive ability of two different feature sets as input to the prediction models. These feature sets have been constructed based on previous solar PV power data and their descriptive statistics. This simplifies the process of model development and ease its implementation for practical applications.
3. We also provide a comprehensive evaluation of the presented approach using eight months of data from AEMO. To the best of our knowledge, no previous studies used aggregated solar power data from Australian energy market.

This paper is organized as follows. Section II presents the problem statement. Section III describes the data set used. Section IV presents the forecasting approach and Section V outlines the experimental setup. Section VI presents and discusses the results. Finally, Section VII summarizes the key results and concludes the paper with future research directions.

II. Problem Statement

Given a time series of aggregated solar power production for a state in Australia up to day d: $P = [P^1, P^2, P^3, \ldots, P^{d-1}, P^d]$, where $P^i = [p_1^i, p_2^i, p_3^i, \ldots, p_n^i]$ represents the half-hourly power profile for day i, i.e. n observations of the power outputs measured at half-hourly intervals; the goal is to forecast the half-hourly power profile for next day *d+1* (i.e. next n future values $p_1^{d+1}, p_2^{d+1}, p_3^{d+1}, \ldots, p_n^{d+1}$) of the time series P.

In this study, we focus on developing prediction model utilizing previous solar PV power data as the inputs to the model. Solar power output from individual farms is influenced by local weather phenomena (e.g. solar irradiance, rain fall, temperature, etc). Therefore, multivariate model utilizing historical weather and power data, as well as predicted weather data is considered ideal for farm level forecasting. However, such multivariate model may not be suitable for aggregated system level forecasting since many geographically dispersed solar farms contribute to the aggregated system level power output. The local weather for each of these farms may vary significantly and inclusion of the huge volume of weather data for all farms into the input of the prediction model is very challenging and even could make the model inefficient [15]. Hence, this study aims to investigate the feasibility of univariate forecasting model for aggregated state level PV power production.

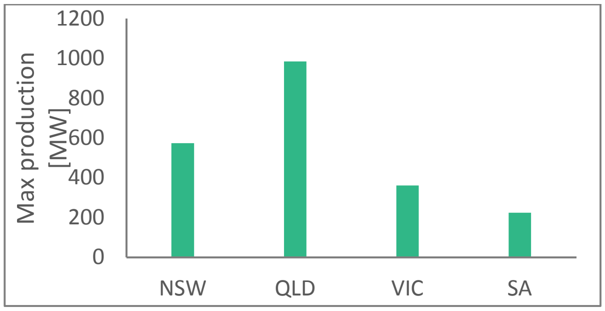

Fig. 1. State level aggregated maximum PV power production in Australia between Jan-Aug 2019

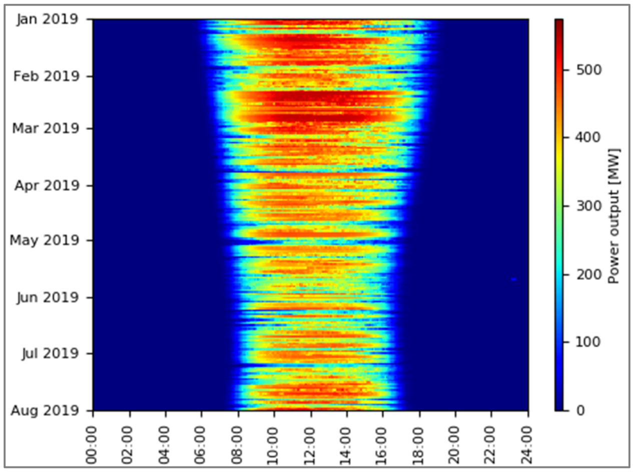

Fig. 2. Aggregated solar power generation in 5-min intervals for NSW (Jan-Aug 2019)

III. Data

A. Data set

We use solar power production data from AEMO. AEMO makes recent 13 months of solar power data publicly available and regularly updates the archive with the most recent data in [16]. In this study, we consider the half-hourly power generation data of all grid connected solar power generation farms registered in National Energy Market (NEM) participants list from 01-Jan-2019 to 28-Aug-2019 (total 240 days).

B. Data Aggregation and Pre-processing

For each state in Australia, the solar power production data from all the grid-connected and geographically dispersed PV plants across the state has been combined into one aggregated time series data by taking their sum. Fig. 1 shows the state-level aggregated maximum production recorded for four different states in Australia for the duration of the data set used in this study. QLD is the highest contributor with 45.96% of total 2144.05 MW solar PV power production in Australia followed by NSW with 26.79% of total production. VIC and SA have relatively lower production compared to other two states with 16.81% and 10.44% of total production, respectively.

AEMO uses one day ahead forecasting (i.e. pre-dispatch forecasting) for operational and planning activities for next day. The pre-dispatch forecasts for a day are done for 48 trading intervals – 1 for each 30-min period in a day. Hence, the time

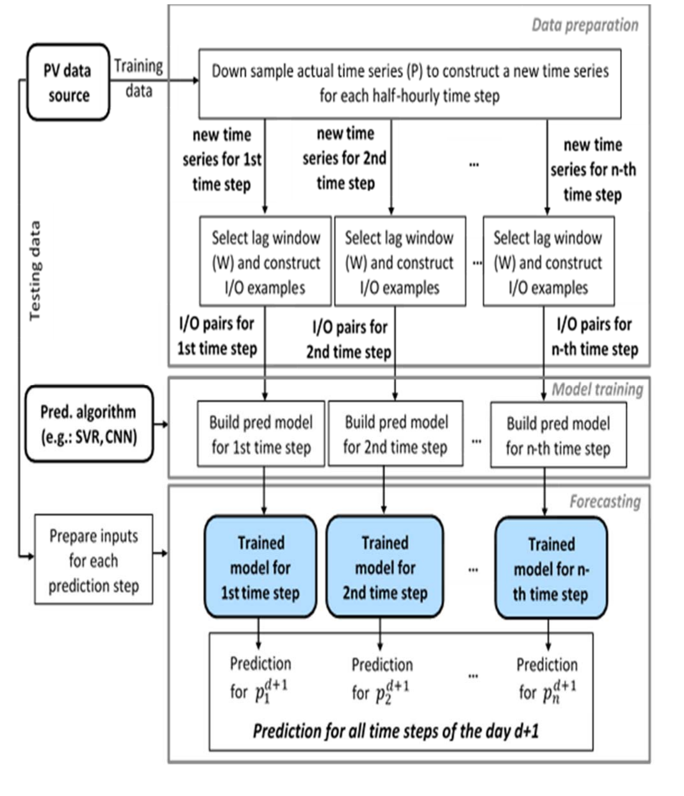
Data preparation
PV data source
Training data
Down sample actual time series (P) to construct a new time series for each half-hourly time step
new time series for 1st time step
new time series for 2nd time step
...
new time series for n-th time step
Testing data
Select lag window (W) and construct I/O examples
Select lag window (W) and construct I/O examples
...
Select lag window (W) and construct I/O examples
I/O pairs for 1st time step
I/O pairs for 2nd time step
I/O pairs for n-th time step
Model training
Pred. algorithm (e.g.: SVR, CNN)
Build pred model for 1st time step
Build pred model for 2nd time step
...
Build pred model for n-th time step
Forecasting
Prepare inputs for each prediction step
Trained model for 1st time step
Trained model for 2nd time step
...
Trained model for n-th time step
Prediction for p_1^{d+1}
Prediction for p_2^{d+1}
...
Prediction for p_n^{d+1}
Prediction for all time steps of the day d+1

Fig. 3. Schematic diagram of the adopted forecasting approach

resolution of the data has been adjusted from 5-min to 30-min by taking the average of solar power production data within each half-hourly window. This aligns the time resolution of the data with AEMO's trading intervals.

Fig. 2 shows the aggregated power output for Jan-Aug 2019 in the state NSW, Australia. The graphs showing aggregated solar power generation for other states are not presented as they share similar characteristics. From this graph, it is obvious that the power outputs before 6:00 in the morning and after 19:00 in the evening is either zero or very close to zero. This is because of the absence of solar irradiance outside this 13-h window. Hence, the final dataset includes the power generation between 6:00 to 19:00 inclusive and filter out the remaining samples.

IV. Forecasting approach

This Section presents a generic approach for day ahead forecasting of state level aggregated PV power production. The key idea of the adopted approach is to compute the forecasts for *n* half-hourly time-steps for next day (*d+1*) by combining the outputs from *n* prediction models – 1 model for each time-step. Fig. 3 presents a schematic diagram of the adopted forecasting approach. It consists of three main steps: (1) data preparation, (2) model development, and (3) forecasting, as described below.

A. Data Preperation

As shown in Fig. 3, the adopted approach divides the task of forecasting PV power output for *n* half-hourly time-steps for next day (*n=27* for our case since we consider data between 6:00 to 19:00 inclusive) into *n* different sub-tasks. For each half-hourly time-step, it first prepares a new time series by taking the samples that belong to that same time instance from all the days. For example, as shown in Fig. 4, for the first time step (*6:00*) the new down-sampled time series is constructed as: $p(6{:}00) = [p_{6:00}^1, p_{6:00}^2, p_{6:00}^3, \dots, p_{6:00}^d]$ and for the second time step (*6:30*) the new time series is constructed as: $p(6{:}30) = [p_{6:30}^1, p_{6:30}^2, p_{6:30}^3, \dots, p_{6:30}^d]$).

A forecasting model is then trained using features from the most recent W samples to make a 1-step ahead prediction on the each new down-sampled time series. The 1-step ahead prediction on each down-sampled time series corresponds to the prediction for a specific time of next day *d+1* (see Fig. 4; for ease of demonstration we use $W = 3$).

TABLE I. Description of input feature sets

Feature Set	Feature Description
FS$_1$ (17 features)	– the most recent observation – min, max, avg, std of the W lagged observations – slope of the regression line for W lagged observations – gradient at each point (except the first) of the W lagged observations where the gradient is computed using second order accurate central differences in the interior points and first order differences at the boundaries – min, max, avg, std of the first order difference of W lagged observations; – slope of the regression line for first order difference of W lagged observations;
FS$_2$ (7 features)	– W most recent observations

The selection of lag window length (W) is very important for building prediction models. A too large window requires many past observations. Whereas too small window may not contain the recent variability in the data that consequently results in poor prediction. In this study, we evaluated different values of W using 5-fold cross validation of the training data and found that W values higher than 7 do not significantly improve the prediction accuracy. Hence, we use $W=7$ for all prediction models.

Two different feature sets have been constructed from W lagged observations. The first feature set (FS$_1$) includes the descriptive statistics from both W lagged observations and the difference series of W lagged observations. The second set (FS$_2$) includes the W lagged observations themselves as the features. Table I summarizes the features in both FS$_1$ and FS$_2$.

B. Model Development

For every half-hourly time-step, a model is trained for single-step ahead prediction using input-output pairs prepared from the time series for that time-step. The adopted forecasting approach is generic and supports any machine learning algorithm for regression. In this study, we apply three well established machine learning algorithms and two state-of-the-art deep learning algorithms. Specifically, we apply NNs, SVR, RF, CNNs and LSTM recurrent networks. A brief overview of these prediction algorithms along with their parameters tuning process are provided below.

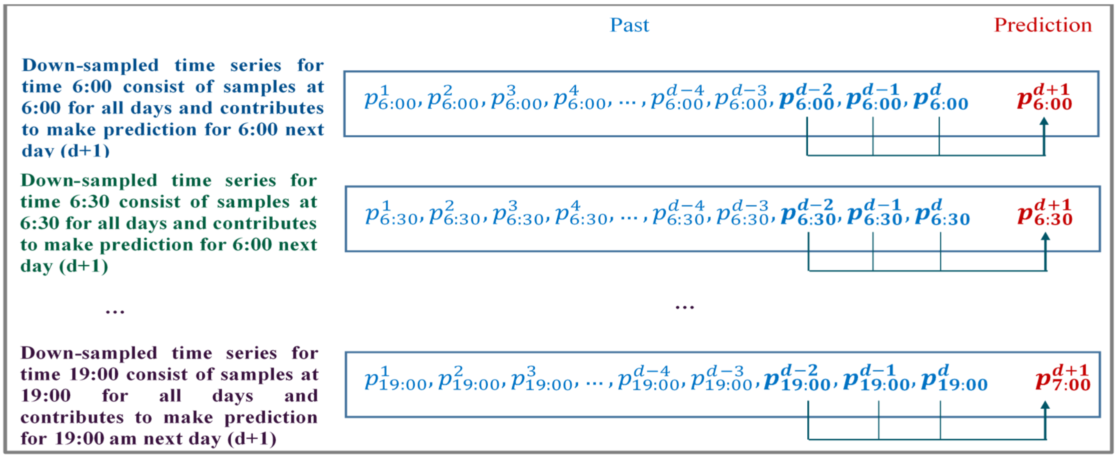
Past
Prediction
Down-sampled time series for time 6:00 consist of samples at 6:00 for all days and contributes to make prediction for 6:00 next day (d+1)
$p^1_{6:00}, p^2_{6:00}, p^3_{6:00}, p^4_{6:00}, \dots, p^{d-4}_{6:00}, p^{d-3}_{6:00}, p^{d-2}_{6:00}, p^{d-1}_{6:00}, p^{d}_{6:00}$
$p^{d+1}_{6:00}$
Down-sampled time series for time 6:30 consist of samples at 6:30 for all days and contributes to make prediction for 6:00 next day (d+1)
$p^1_{6:30}, p^2_{6:30}, p^3_{6:30}, p^4_{6:30}, \dots, p^{d-4}_{6:30}, p^{d-3}_{6:30}, p^{d-2}_{6:30}, p^{d-1}_{6:30}, p^{d}_{6:30}$
$p^{d+1}_{6:30}$
...
...
Down-sampled time series for time 19:00 consist of samples at 19:00 for all days and contributes to make prediction for 19:00 am next day (d+1)
$p^1_{19:00}, p^2_{19:00}, p^3_{19:00}, \dots, p^{d-4}_{19:00}, p^{d-3}_{19:00}, p^{d-2}_{19:00}, p^{d-1}_{19:00}, p^{d}_{19:00}$
$p^{d+1}_{7:00}$

Fig. 4. Data preparation and predictions in the adopted approach

NNs: NNs [17] are widely applied machine learning algorithms for forecasting solar power time series data. NNs can estimate any complex functional relationship (both linear and nonlinear) without any assumptions regarding the distribution of data. However, the performance of NNs depends on the network architecture and the random initialization of weights. To reduce this sensitivity, we apply an ensemble of NNs. To develop NNs based prediction models, we uses Multi-Layer Perceptron (MLP) NNs that are considered as universal function approximators [18].

For each of the *n* time-steps, a separate NNs ensembles has been developed using MLPRegressor from the scikit-learn library [19] for python. The parameters for each of these *n* NNs have been tuned separately using a grid search and applying 5-fold cross validation of the training data. For grid search, we consider following parameters: hidden_layer_sizes:[(i,) for i in range(10, 41, 5)], activation (activation function for the hidden layer): ['logistic', 'tanh', 'relu'], solver (weight optimization method): ['lbfgs'], early_stopping: [True], alpha (regularization term): [0.1, 0.01, 0.05, 0.005, 0.0001, 0], max_iter (maximum number of iterations): [2000], shuffle: [False].

To develop an ensemble, *m* replicates of the best estimator (i.e. NN that provides the highest score/lowest loss on 5-fold cross validation of training data) have been created – all the *m* replicates have same structures but assigned different random weights. These *m* replicates are then trained separately on the entire training data and their outputs are combined by taking average to compute the final prediction.

SVR: SVR [20] is another popular machine learning algorithm for energy time series prediction. Like NNs, SVR can learn complex functional relationship from examples but it has a smaller number of parameters to tune and is less likely to overfit the data. The key idea of SVR is to map the input data into a higher dimensional feature space using a non-linear transformation and then apply linear regression in the new space. The linear regression in the new space corresponds to non-linear regression in the original space. The task is formulated as an optimisation problem. The main goal is to minimize the error on the training data, but the flatness of the line and the trade-off between training error and model complexity are also considered to prevent overfitting.

Like NNs, a separate SVR model has been developed for each of the *n* time-steps using *epsilon-SVR* from the *scikit-learn* library. Each of these *n* SVR models has been optimized separately using a grid search of the following parameters: kernel = [‘linear’, ‘poly’, ‘rbf’], C (penalty parameter of the error term) = [0.1, 0.5, 1, 10], degree (degree of the polynomial kernel function , ignored by all other kernels)=[2,3], gamma (kernel coefficient for ‘rbf’, ‘poly’)=['scale', 0.1, 1], epsilon (defines a tube around the regression line that corresponds to the precision with which the input-output mapping will be approximated)=[0.1, 0.2, 0.5], tol (tolerance for stopping criterion)=[0.0001].

RF: RF [21] is a representative of the tree based learning algorithms. The basic idea of RF is to create a forest by fitting a number of Decision Tree [22] based learners that estimate the value of the target variable by generating a set of rules learned from the input features and then combine the prediction of all the individual decision trees by taking their average.

Each tree in the forest has been trained using a randomly drawn subset of input-output examples. All the subsets are equal in size and drawn using bootstrapping (i.e. sampling with replacement). The features for splitting the nodes of the tree are also always randomly permuted at each split. These increase the diversity in the learning process and lead to more robust overall predictions compared to the single Decision Tree based prediction algorithm.

Like NNs and SVR, RF has been implemented in python’s *scikit-learn* library and the parameters of each of the *n* RF models has been tuned based on exhaustive grid. The search space for grid search includes following: n_estimators (number of trees in forest)=[500, 1000, 2000], criteria (the function to measure the quality of a split)=[‘mse’], and bootstrap=[True].

CNNs: CNNs [23, 24] are state-of-the-art deep learning algorithms. One of the main advantages of CNNs is their ability to automatically learn useful features from data without prior knowledge and feature engineering. There are three main types of layers in a CNN: convolutional, pooling, and dense layers. The convolutional and pooling typically repeated several times (pooling layers follow a sequence of one or more convolutional layers and are intended to consolidate the features learned and expressed in the preceding layers feature map). The last layer is a fully connected layer which produces the output.

The convolutional layers apply filters to an input to create a feature map that summarizes the presence of detected features in the input. In particular, the filters (kernels) with pre-specified size in each convolutional layer are slided over the input to convolve the weight matrix of the filter with the input and compute the feature maps. The nodes in a convolutional layer use local connectivity and weight sharing to reduce the number of weights that are stored and learned, and lead to faster training. A pooling node computes the maximum/average (as configured) value of the set of adjacent convolutional nodes that are connected to it. The combination of a convolutional and pooling layer allows to extract suitable shift-invariant features from the input. CNNs models have been implemented in Python using Keras [31] and Tensorflow [32] libraries.

To select the appropriate structure and tune the parameters of the CNN model for each time-step, different architecture and parameter combination have been evaluated using 5-fold cross validation of training data. Specifically, the search space includes following: filters (the number of output filters in the convolution) =[5, 10, 20, 30, 40, (5,10), (10,10), (20,10), (30,10), (40,10)]; kernel_size (the length of the 1D convolution window)=[3, 5, 7]; padding='same', activation=['relu', 'tanh', 'sigmoid', 'linear], neurons (in dense layer)=[10, 20, 30, 40], activation (for dense layer)=['relu', 'tanh', 'sigmoid', 'linear'], optimizer='SGD', lr (learning rate)=[0.001, 0.005, 0.01, 0.05, 0.1], momentum=[0.99, 0.9, 0.8, 0.7, 0.6, 0.5], epochs=[200, 300, 500, 1000, 1500, 2000], batch_size=[1, 50, 100, 150, 200]. The batch normalization and dropout after each convolution layer have been applied to speed up the training and reduce overfitting. The CNN that provides the best accuracy on 5-fold cross validation of training data has been applied for predicting test data.

LSTM: LSTM [25, 26] are recurrent deep neural networks that have been specifically designed to overcome the vanishing and blowing up gradient problems of the standard recurrent NNs. The main advantage of LSTM is that they can learn the temporal context of input sequences in order to make better predictions for complex time series data. This means LSTM networks can model time series data without the need of specifying a fixed set of lagged observations. The neurons (also called nodes) of a LSTM network can manage their state using a set of gates that make them smarter than the neurons in conventional recurrent NNs.

Like CNNs, LSTM has been implemented in Python using Keras and Tensorflow. The suitable structure and the parameters of the LSTM models have been selected based on their different combination. The search space for LSTM structure and other parameters includes following: units (number of LSTM nodes): [5, 10, 20, 30, 40, (5,10), (10,10), (20,10), (30,10), (40,10)], activation=['relu', 'tanh', 'sigmoid', 'linear'], dropout (fraction of the units to drop for the linear transformation of the inputs) =[0, 0.1, 0.2, 0.3], activation (for dense layer)=['relu', 'tanh', 'sigmoid', 'linear'], batch_size=1, stateful=True, optimizer='SGD', lr (learning rate)=[0.001, 0.005, 0.01, 0.05, 0.1], momentum=[0.99, 0.9, 0.8, 0.7, 0.6, 0.5], epochs=[200, 300, 500, 1000, 1500, 2000].

C. *Forecasting*

To predict new data, the input features for each time-step have been prepared separately and feed to the corresponding trained prediction model. The outputs from all *n* prediction models combinedly construct half-hourly power forecasts for the target day *d+1*.

V. Simulation Settings

A. *Training and Testing Data*

We use the solar power production data for the state of NSW, Australia. The full data set for NSW has been divided into two non-overlapping subsets: *training* and *testing*. The *training* set contains all the data from 01-Jan-2019 to 31-Jul-2019 (212 days). The data in the *training* set is used to train the prediction models and tune their parameters through 5-fold cross validation. On the other hand, the *testing* set contains the data from 01-Aug-2019 to 28-Aug-2019 (28 days). The data in the *testing* set is used to evaluate the accuracy of prediction models.

B. *Evaluation Metrics*

To evaluate the accuracy of prediction models, two standard metrics are used: Mean Absolute Error (MAE) and Mean Relative Error (MRE).

MAE is the mostly cited metrics for forecasting energy time series data including solar PV power. It measures deviation of the predicted data from the actual data averaged over all the examples:

$$MAE = \frac{1}{D*N}\sum_{i=1}^{D}\sum_{j=1}^{N}\left|p_j^i - \hat{p}_j^i\right| \qquad (1)$$

where p_j^i and $\hat{p}_j^i$ are respectively the actual and predicted power output for the j-th time step of the day *i*; *N* is the number of time step in a day (*N=27* for our task); and *D* is the total number of days.

MRE is extension of MAE. It normalizes MAE by the maximum measured power output (C) of the time series and computes the prediction error as a percentage:

$$MRE = \frac{1}{D*N}\sum_{i=1}^{D}\sum_{j=1}^{N}\left|\frac{p_j^i - \hat{p}_j^i}{C}\right| \times 100\% \qquad (2)$$

VI. Results

This section presents and discusses the forecasting results. Firstly, it analyses the accuracy of the adopted approach using different prediction algorithms and evaluates the predictive ability of the two feature sets used as inputs to the prediction algorithms. It then evaluates the forecast skill of the adopted approach against two smart baseline models. It is important to note that all the results presented in this section are obtained using testing data set i.e. the out-of-sample data that is not utilized for tuning parameters of the models.

A. *Performance Using Different Prediction Algorithms*

Table II presents the forecasting accuracy of the adopted approach using five prediction algorithms (SVR, RF, NNs, CNNs, LSTM) that apply two different feature sets (FS_1 and FS_2) as inputs.

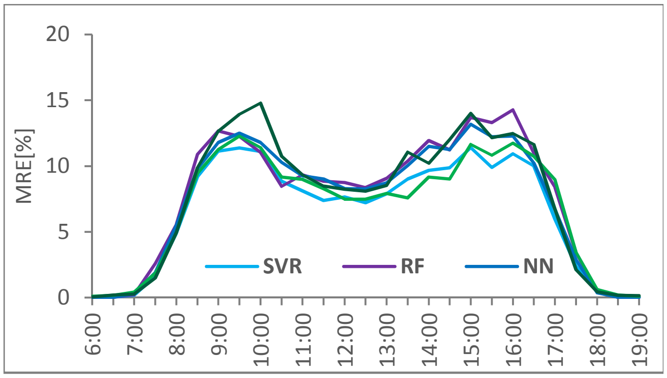

Fig. 5. Prediction accuracy (MRE) of the adopted approach utilizing different prediction algorithms for all half-hourly intervals

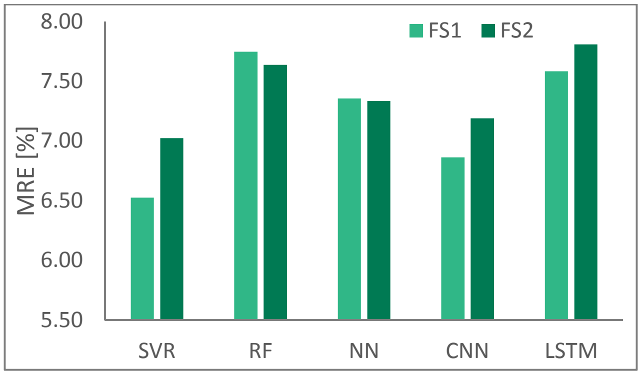

Fig. 6. Prediction accuracy (MRE) of the adopted approach with five prediction algorithms using two input feature sets

Overall, the adopted approach utilizing SVR as the prediction algorithm and FS_1 as the inputs achieves the highest prediction accuracy: MAE= 35.54 MW and MRE=6.53%. CNNs using the same feature set (FS_1) as inputs come next in the ranking followed by NNs using FS_1. The prediction accuracy of the LSTM and RF are the lowest regardless of feature set used. The top ranked SVR (using FS_1 as inputs) shows at least 4.90%, 11.03%, 13.96%, and 14.57% improvements of accuracy (either MAE or MRE) over the CNNs, NNs, LSTM, and RF, respectively. These improvements have been found statistically significant at $p<=0.05$ (Wilcoxon rank-sum test). A plot showing the actual vs predicted data using the most accurate model (i.e. SVR with FS_1 feature set) for entire duration (4 weeks) of the test data set is provided in Appendix.

TABLE II. ACCURACY OF THE ADOPTED APPROACH WITH DIFFERENT PREDICTION ALGORITHMS USING TWO INPUT FEATURE SETS

Pred. Algorithm	Input Set	MAE [MW]	MRE [%]
SVR	FS_1	35.54	6.53
	FS_2	38.25	7.02
RF	FS_1	42.18	7.75
	FS_2	41.60	7.64
NNs	FS_1	40.06	7.36
	FS_2	39.94	7.34
CNNs	FS_1	37.37	6.86
	FS_2	39.15	7.19
LSTM	FS_1	41.30	7.59
	FS_2	42.53	7.81

Fig. 5 presents the breakdown of the prediction accuracy (MRE) for all half-hourly time steps between 6:00 to 19:00. It shows that there is not a substantial difference in accuracy of the prediction algorithms in the early morning. However, as the time increases from 8:00, the two top ranked algorithms (SVR and CNNs) show superior accuracy over other three algorithms (RF, NNs, and LSTM) until 17:00 in the evening. This indicates that both SVR and CNNs can model the solar power data well and provide better prediction compared to other three algorithms during the time window when fluctuations in the data are higher. Hence, it can be concluded that both SVR and CNNs are more viable for the aggregated state level power production data we consider in this study.

Moreover, to better understand what makes SVR to perform better than any other models, the optimized parameters of the best performing SVR models (i.e. SVR using FS_1 as inputs) for all time steps have been thoroughly investigated. Analysis reveals that SVR chooses linear kernel and assigns higher value (10) for penalty parameter (C) of the error term for the first few hours (6:00 to 8:00) at the beginning of the day and last few hours (16:30 to 19:00) at the end of the day. During these two window of time, the power production is relatively low and the fluctuation is less prominent, hence SVR with linear kernel can model the data well. For the remaining time of the day (8:30 to 16:00), SVR selects polynomial kernel with degree either 2 or 3 and assigns lower value (0.5 to 1) for penalty parameter of the error term depending on the time. During this period, the solar power output shows more variability and the selection of polynomial kernel to model the highly fluctuating data is well justified. The ability of SVR to select the suitable kernel functions and other associated parameters based on the complexity of the data for different time steps makes it well suited to model the solar power data set, decrease the chance of overfitting (if complex kernel were used during no or less fluctuation in data) and underfitting (if linear kernels were used during the time of high fluctuation in data), and provide better prediction overall.

B. *Comparison of Input Feature Sets*

Fig. 6 plots the accuracy (in terms of MRE) of the adopted approach with five prediction algorithms (SVR, RF, NNs, CNNs, and LSTM) using two different input feature sets (FS_1 and FS_2) to facilitates visual comparison of the predictive power of the feature sets. The graph with MAE is not shown as it is very similar to Fig. 6.

It is noticed that the two top ranked algorithms (SVR and CNNs) and LSTM obtain the better accuracy if they use first feature set (FS_1) as inputs. SVR, CNNs, and LSTM respectively achieves 7.03%, 4.55%, and 2.88% statistically significant improvement in accuracy (either MAE or MRE) if they use FS_1 as inputs instead of FS_2. On the other hand, only RF shows slightly better accuracy if it uses second feature set (FS_2) as inputs. For NNs, the accuracy is very similar with FS_1 and FS_2, and the difference is not statistically significant. These observations suggest that feature extraction from the raw solar power data is beneficial for most prediction algorithms and using them as inputs helps to capture patterns and recent trend in the solar power production data.

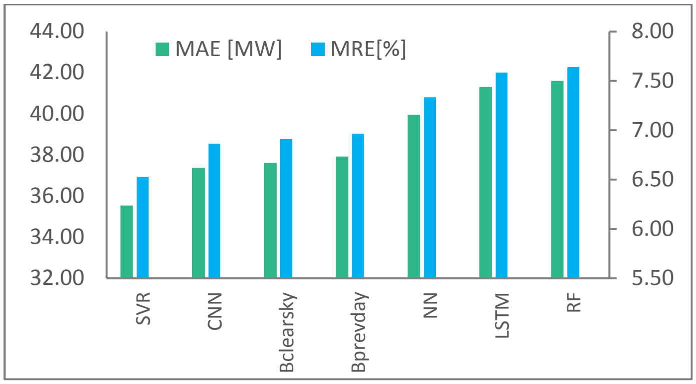

Fig. 7. Best accuracy of different prediction algorithms and baselines

C. *Forecast Skill with Respect to Baselines*

To assess the forecast skill of the adopted approach using different prediction algorithms, the best performance for each algorithm has been compared with that of two persistence models considered as baselines. The first baseline ($B_{prevday}$) assumes the similar weather conditions for consecutive days and considers the solar power outputs from previous day (*d*) as the predictions for next day (*d+1*). Thus, predictions for $p_1^{d+1}, p_2^{d+1}, p_3^{d+1}, \dots, p_n^{d+1}$ are given by $p_1^{d}, p_2^{d}, p_3^{d}, \dots, p_n^{d}$. On the other hand, second baseline ($B_{clearsky}$) assumes clear-sky weather condition and considers the 90th percentile of the power outputs at same time-step over last 3 months as the prediction for a time-step next day *d+1*.

Fig. 7 visually presents the best accuracy results of the adopted approach achieved by different prediction algorithms (using either FS_1 or FS_2) with that for two baseline models. It shows that only SVR and CNNs (using FS_1 as inputs) provide better accuracy compared to both baseline models. The forecast skills of the SVR with respective to $B_{prevday}$ (MAE= 37.93 MW, MRE=6.97%) and $B_{clearsky}$ (MAE= 37.61 MW, MRE=6.91%) are 6.30% and 5.52%, respectively. CNNs also shows improvements over two baselines: 1.47% and 0.65%, respectively.

On the other hand, NNs, LSTM, and RF show poor performance and their accuracy (MAE or MRE) are well below compared to two baselines. There are two main possible reasons that could cause relatively poor performance of NNs, LSTM, and RF. Firstly, lack of enough training data - there are only 212 input-output examples have been used to train the prediction models. Increasing the size of training data spanning over larger time period may lead to inclusion of input-output examples with diverse weather conditions and consequently improve the performance of the prediction models. The second reason could be the input features used. The features considered in FS_1 and FS_2 may not be sufficient for generalizing the input-output relationships in the training data. Using additional features that better describe power output can help to improve the accuracy of the prediction models. The future work will focus on these two areas to investigate possible improvement in accuracy of all prediction algorithms.

VII. Conclusions and Future Work

We presented a generic data-driven approach for day ahead forecasting of aggregated PV power production in half-hourly intervals for the state of NSW, Australia. The presented approach investigated five widely used machine learning and deep learning algorithms to develop forecasting models; and two different feature sets as inputs to the prediction algorithms. The parameters of the prediction models were tuned using a 5-fold cross validation of training data spanning seven months (Jan-Jul 2019).

The performance of the adopted approach evaluated using out-of-sample data spanning 1 month (1-28 Aug 2019) showed that SVR and CNNs provided the most accurate prediction. The prediction accuracy in terms of MAE and MRE are 35.54 MW and 6.53% for SVR; and 37.37 MW and 6.86% for CNNs. Both SVR and CNNs not only showed superior performance over NNs, LSTM and RF but also outperformed two smart persistence models considered as baselines. SVR achieved 4.90-14.57% improvement of accuracy over all other models (including CNNs and baselines). The improvement of CNNs was in the range of 0.65-10.17% over NNs, RF, LSTM, and baselines. Most importantly, SVR and CNNs showed superior performance over the other models during the time between 8:00-17:00 when the solar power output fluctuates more. The better forecast skill of the SVR and CNNs compared to the baselines, and their ability to provide more accurate predictions during the time of high variability in the data suggest that both algorithms are very promising and can be applied for practical applications for forecasting aggregated system level solar power prediction. The good performance of the deep learning algorithm CNNs compared to traditional feed forward NNs also indicates that although NNs are the most popular in the literature for farm level prediction, CNNs could be a better choice for aggregated system level forecasting of solar PV power production.

The noteworthy advantage of the presented forecasting approach is that it only uses the lagged power data (and their descriptive statistics) as inputs to the prediction algorithms. This makes the forecasting of aggregated state level solar PV production relatively easy since identifying the meteorological factors for many geographically distributed solar farms across a state and using those factors for forecasting aggregated state level production could make the prediction model inefficient and complicated.

In future, we plan to investigate following: 1) advanced deep learning algorithms such as LSTM Encoder-Decoder and CNN-LSTM Encoder-Decoder for developing prediction model; 2) unsupervised machine learning techniques (such as clustering) to group the data and then developing separate prediction model for each group; 3) extracting new features and using them as the inputs to the prediction model; and 4) increasing the size of data training data to include most recent data from AEMO.

References

[1] "Global market outlook for solar power (2018-2022)," Solar Power Europe, 2018.

[2] C. Commission:, "The Critical Decade: Australia's Future - Solar Energy," 2013.

[3] M. Rana, I. Koprinska, and V. G. Agelidis, "Forecasting solar power generated by grid connected PV systems using ensembles of neural networks.," in *International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 2015.

[4] M. Rana and I. Koprinska, "Neural network ensemble based approach for 2D-interval prediction of solar photovoltaic power," *Energies,* vol. 9, pp. 829-845, 2016.

[5] J. Shi, W.-J. Lee, Y. Liu, Y. Yang, and P. Wang, "Forecasting power output of photovoltaic systems based on weather classification and support vector machines," *IEEE Transactions on Industry Applications,* vol. 48, pp. 1064-1069, 2012.

[6] M. Rana, I. Koprinska, and V. G. Agelidis, "2D-Interval Forecasts for Solar Power Production," *Solar Energy,* vol. 122, pp. 191-203, 2015

[7] C. Persson, P. Bacher, T. Shiga, and H. Madsen, "Multi-site solar power forecasting using gradient boosted regression trees," *Solar Energy,* vol. 150, pp. 423-436, 2017.

[8] H. T. Pedro and C. F. Coimbra, "Assessment of forecasting techniques for solar power production with no exogenous inputs," *Solar Energy,* vol. 86, pp. 2017-2028, 2012.

[9] M. Rana and A. Rahman, "Multiple Steps ahead Solar Photovoltaic Power Forecasting Based on Univariate Machine Learning Models and Data Re-sampling " *Sustainable Energy, Grids and Networks,* vol. 21, pp. 1-12, 2020.

[10] X. Zhang, Y. Li, S. Lu, H. F. Hamann, B.-M. Hodge, and B. Lehman, "A Solar Time Based Analog Ensemble Method for Regional Solar Power Forecasting," *IEEE Transactions Sustainable Energy,* vol. 10, pp. 268-279, 2019.

[11] M. Rana, I. Koprinska, and V. G. Agelidis, "Solar power forecasting using weather type clustering and ensembles of neural networks," in *International Joint Conference on Neural Networks (IJCNN)*, Canada, 2016.

[12] S. Sobri, S. Koohi-Kamali, and N. A. Rahim, "Solar photovoltaic generation forecasting methods: A review," *Energy Conversion and Management,* vol. 156, pp. 459–497, 2018.

[13] R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra, "Solar forecasting methods for renewable energy integration," *Progress in Energy and Combustion Science,* vol. 39, pp. 535-576, Jul 2013.

[14] E. Lorenz, T. Scheidsteger, J. Hurka, D. Heinemann, and C. Kurz, "Regional PV power prediction for improved grid integration," *Progress in Photovoltaics: Research and Applications,* vol. 19, pp. 757-771, 2011.

[15] Y. Zhang, M. Beaudin, H. Zareipour, and D. Wood, "Forecasting Solar Photovoltaic power production at the aggregated system level," in *North American Power Symposium (NAPS)*, 2014, pp. 1-6.

[16] *Australian Energy Market Operator (AEMO) - Market Data NEMweb.* Available: http://www.nemweb.com.au/

[17] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature,* vol. 323, pp. 533-536, 1886.

[18] [18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks Are Universal Approximators," *Neural Networks,* vol. 2, pp. 359–366, 1989.

[19] *Scikit-learn : Machine Learning in Python*. Available: https://scikit-learn.org/stable/index.html

[20] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in *Advances in Neural Information Processing Systems*. vol. 9, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds., ed: MIT Press, Cambridge, 1997, pp. 281-287.

[21] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, pp. 5–32, 2001.

[22] Y. Wang and I. H. Witten, "Inducing Model Trees for Continuous Classes," in *European Conference on Machine Learning (ECML)*, Czech Republic, 1997, pp. 1-10.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, pp. 2278-2324, 1998.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *International Conference on Neural Information Processing Systems (NIPS)*, 2012.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation,* vol. 9, pp. 1735-1780, 1997.

[26] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation,* vol. 12, pp. 2451-2471, 2000.

APPENDIX

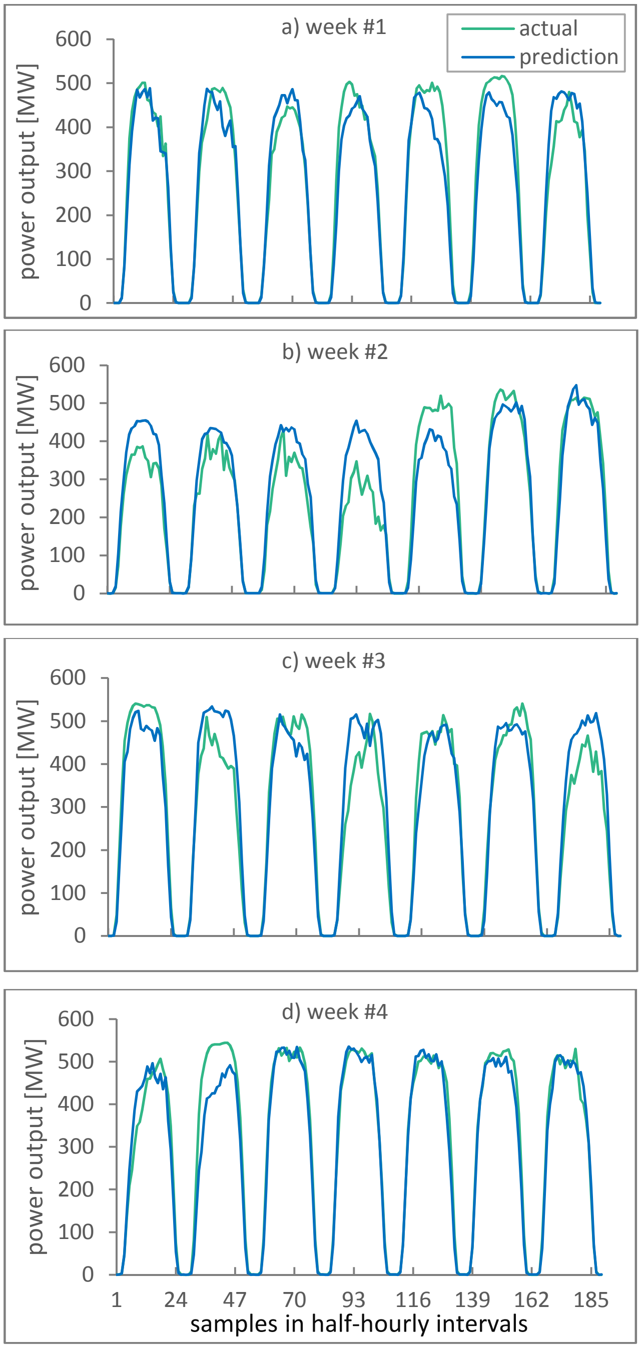

Fig. 6. Actual vs predicted data using the best performing model (SVR with FS1 input feature set)