Journal of the Society of Automotive Engineers Malaysia
Volume 4, Issue 3, pp 307-322, September 2020
e-ISSN 2550-2239 & ISSN 2600-8092

Machine Learning-Based Solar Power Energy Forecasting

N. C. Nath[*1], W. Sae-Tang[1] and C. Pirak[2]

[1]Department of Software System Engineering,
[2]Department of Communication and Smart System Engineering,
The Sirindhorn International Thai-German Graduate School of Engineering, King Mongkut's University of Technology North Bangkok, Bangsue, Bangkok 10800, Thailand

[*]*Corresponding author: nayan.n-sse2017@tggs.kmutnb.ac.th*

ORIGINAL ARTICLE **Open Access**

Article History:

Received
21 May 2020

Received in revised form
15 Jul 2020

Accepted
17 Jul 2020

Available online
1 Sep 2020

Abstract – *The expanding interest in energy is one of the main motivations behind the integration of solar energy into electric grids or networks. The exact expectation of solar oriented irradiance variety can improve the nature of administration. This coordination of solar-based vitality and exact expectations can help in better arranging and distributing of energy. Discovering vitality sources to fulfil the world's developing interest is one of the general public's major difficulties for the coming fifty years. In this research, two machine learning techniques utilized for hourly solar power forecasting are presented. The solar power prediction model becomes robust and efficient for solar power energy forecasting once the redundant information is removed from raw data, experimental data is transformed into a settled range, the best features selection method is chosen, four different weather profiles are made based on different weather conditions and the right time series machine learning algorithm is chosen*

Keywords: Solar power energy forecasting, feature selection, weather profiles, neural network, long short-term memory

Journal homepage: www.jsaem.saemalaysia.org.my

1.0 INTRODUCTION

Global energy demand is increasing, and finding adequate supplies of clean energy for the future is one of mankind's most overwhelming difficulties. Having a precise expectation can help in better arranging and circulating of energy. The world's interest for energy is anticipated to dramatically increase by 2050 and shall significantly multiply before the century is over. Gradual enhancements to existing energy systems will be insufficient to supply this interest economically. Solar power forecasting depends on various factors including knowledge of the sun's path, the atmosphere's condition, the scattering process, and the characteristics of a solar energy plant utilizing the sun's energy to create solar power. Thus, the main motivation of this paper is the precise expectation of solar-powered irradiance variation, which can upgrade the quality of services. This combination of solar energy and precise forecast can help in better planning and distributing of energy. The sustainable power source is becoming progressively

essential in fulfilling global energy demands. One specific element of the energy power source is that the yield of a power plant depends, to a great extent, on climate conditions, and precise gauges of intensity is an important prerequisite to be incorporated into the power grid.

The Global Energy Forecasting Competition 2014 featured a unique approach to the forecasting of hourly power outputs. In the approach, the forecast is made as regards the probability distribution of the target variable, rather than forecasting one single value. This methodological difference from a single value forecast is huge and will provide stakeholders in the industry with more information to incorporate into their daily work. As a side effect, the new method must be used efficiently (Nagy et al., 2016).

In the present aggressive and dynamic condition, the energy supply, request, and costs are becoming unstable and unpredictable. Increasing more basic leadership forms in the energy industry requires a complete standpoint of the questionable future. Numerous leaders are depending on probabilistic forecasts to measure these vulnerabilities, as opposed to point estimates. In this paper, the term "solar power forecasting" is utilized and alludes to the forecast energy industry. Forecasting the solar power energy industry incorporates, but is not limited, to the anticipating of the supply, request, and cost of power, gas, water, and sustainable power source assets (Tao et al., 2016). Various types of methods have been introduced such as gradient boosting machines (GBM), k-nearest neighbour (k-NN), etc. Intra-hour solar oriented irradiance estimating is performed by anticipating cloud movements and areas in (Chi et al., 2011).

Cloud areas are anticipated for five minutes ahead and re-enactments have demonstrated that for beachfront regions, cloud figure blunder increases as the forecast horizon increases. A neural network-based forecasting strategy is produced in (Chu et al., 2015) to enhance forecast precision of (1) a physical model dependent on cloud following, (2) auto backward moving normal and (3) k-th closest neighbour strategies for forecast horizons of 15 minutes and less. An ongoing audit of the best in class can be found in (Inman et al., 2013). Many papers on energy estimating have been published over the past 50 years. An outline of energy anticipating, following the determining points back to the origin of the electric power industry.

In this research, two different techniques are used for solar power energy forecasting, namely time series forecasting. Two novel machine learning algorithms, neural network (NN) and long short-term memory (LSTM) which is a very special kind of recurrent neural network, are the focus of this study. This paper is organized as follows. The introduction part, background, and recent work shall be introduced. Continuing with this, the machine learning-based method comprises the theoretical foundation of feature selection, neural net, recurrent neural net, and long short-term memory. Experiment design shall consist of data processing, feature selection, and different weather profiles. Experimental results are then presented with performance evaluation, graphical representation and finally, the conclusion shall summarize the research. The significant conclusion to this study shall be the prediction of solar power energy forecasting on an hourly basis and the analysis of two different machine learning algorithms.

2.0 MACHINE LEARNING BASED METHOD

There are two main parts for the assessment which calculates the mutual information score (part of data processing) and the algorithmic method that has been used during the experiment.

2.1 Mutual Information

Regression is intimately related to predictably, in the sense of $\hat{X} = (X|Y)$ being a prediction (a guess) of the random variable X in terms of the variable Y. So, "regression gets us how Y explains X" but only in that weak sense. It's true that if X, Y are independent then $\hat{X} = (X|Y)$ =E(X) (no regression, X is" not predictable" from Y). But the reverse is not true. It can happen that but even then, X, Y are not independent.

$$\mathrm{E}\,(\mathrm{X}|\mathrm{Y}) = \mathrm{E}\,(\mathrm{X}) \tag{1}$$

The concept of lack of regression/predictability is weaker than the concept of independence (and the concept of uncorrelatedness is still weaker) also regression is not symmetric. It can happen that

$$\mathrm{E}(\mathrm{X}|\mathrm{Y}) = \mathrm{E}(\mathrm{X}) \text{ but } \mathrm{E}(\mathrm{Y}|\mathrm{X}) \neq \mathrm{E}(\mathrm{X}) \tag{2}$$

- so one would say that X helps us to guess Y, but Y does not help us to guess X. Formally, the mutual information of two discrete random variables X and Y can be defined as:

$$\mathrm{I}(\mathrm{X};\mathrm{Y}) = \sum_{\mathrm{y}\in\mathrm{Y}} \sum_{\mathrm{x}\in\mathrm{X}} \mathrm{p}\,(\mathrm{x},\mathrm{y}) \log\left(\frac{\mathrm{p}(\mathrm{x},\mathrm{y})}{\mathrm{p}(\mathrm{x})\mathrm{p}(\mathrm{y})}\right) \tag{3}$$

where $p(x, y)$ is the joint probability of X and Y, and $p\ (x)$ and $p(y)$are the merginal probability distribution functions of X and Y respectively. In the case of continuous random variables, the summation is replaced by a definite double integral:

$$\mathrm{I}(\mathrm{X};\mathrm{Y}) = \int_{\mathrm{y}}\int_{\mathrm{x}} \mathrm{p}(\mathrm{x},\mathrm{y}) \log\left(\frac{\mathrm{p}(\mathrm{x},\mathrm{y})}{\mathrm{p}(\mathrm{x})\mathrm{p}(\mathrm{y})}\right) \mathrm{dxdy} \tag{4}$$

where $p(x, y)$ is now the joint probability density function of X and Y, and $p(x)$ are the marginal probability density functions of X and Y respectively.

2.2 Neural Network

Rosenblatt structured the principal learning neural computer, the perceptron, in 1958 of a push to imitate human learning (Rosenblatt, 1958).

The neural network has its beginning stages in undertakings to find numerical representations of data processing in biological systems. Beyond question, it has been used widely to cover a broad assortment of different models, a ton of them have been the subject of distorted cases as for their biological credibility. From the perspective of uses of example pattern recognition, be that as it may, biological authenticity would drive unnecessary confinements. The best model of example acknowledgment is the feed-forward neural network, otherwise called the multi-layer perceptron (Bishop, 2006).

2.3 Feed Forward Neural Net

When we use a feedforward neural network to accept an input x and produce an output $\hat{y}$, information flows forward through the network. The inputs x provide the initial information that then propagates up to the hidden units at each layer and finally produces $\hat{y}$. This is called forward propagation. During training, forward propagation can continue onward until it produces a scalar cost $J(\Theta)$ (Goodfellow et al., 2016). The linear regression models are based on linear combinations of fixed nonlinear basis functions $\emptyset j(x)$ and the form generated from this is:

$$y(x, \omega) = \int \left(\sum_{j=1}^{M} \omega_j \emptyset_j (x)\right) \quad (5)$$

Here, ƒ (*) is the activation function for a regression problem. If the input variables are x_1,...,x_D, the M linear combination of the input is:

$$aj = \sum_{i=1}^{D} \omega_{ji}^{(1)} x_{i+} \omega_{j0}^{(1)} \quad (6)$$

Here,$j = 1$,..., M, and the superscript (1) signifies that the respective parameters are from the first layer of the network. $\omega_{ji}^{(1)}$ are the weights and $\omega_{j0}^{(1)}$are the biases of the network. The quantity a_j are known as activation. Then they get transformed and give:

$$z_j = h(a_j) \quad (7)$$

These quantities are the output of basis function and called hidden units. When we linearly combine them for output unit activations, it takes the form:

$$a_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} z_j + \omega_{k0}^{(2)} \quad (8)$$

Here k = 1,...,K, and K is the total number of outputs and the superscript (2) signifies that the respective parameters are from the first layer of the network and $\omega_{k0}^{(2)}$ are biases. At this time the output unit activations are transformed using an inappropriate activation function to give a set of network outputs. In a regression problem, the activation function is the identity, so that $yk = ak$. The combination of different steps gives the whole network function for sigmoidal output unit activation functions,

$$yk = (x, \omega) = \sigma\left(\sum_{j=1}^{M} \omega_{kj}^{(2)} h\left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_{i+} \omega_{j0}^{(1)}\right) + \omega_{k0}^{(2)}\right) \quad (9)$$

where, k = 1,..., K is the total number of outputs. This transformation corresponds to the second layer of the network, and again the $W_{k0}^{(2)}$ are bias parameters. Finally, the output unit activations are transformed using an appropriate activation function to give a set of network outputs yk. The choice of the activation function is determined by the nature of the data and the assumed distribution of target variables.

2.4 Recurrent Neural Net

A recurrent neural network (RNN) is a class of artificial neural networks where associations between nodes form a directed graph along a temporal sequence. This enables it to show temporal dynamic behaviour. Compare with neural network input is associated with the hidden layer and the output layer but in the recurrent neural network there is a feedback connection that allows us to pass the input in current timestamp and immediate past timestamp and the weight is shared with the subsequent layers. There is some limitation for the recurrent neural net which called handling long and short-term dependencies problem.

As we know, we need to calculate the scaler weight during the back-propagation of time algorithm and it could turn to infinity (when the Eigen vector is greater than one) which is called gradient exploding problem though we can handle this issue by introducing gradient clipping while gradient is lower than zero it could vanish (when the Eigen vector is lower than zero). That means, we can lose our information or the network could stop learning at this stage. For handling this issue, LSTM (long short-term memory) was introduced which combined with a special gated mechanism (forget gate, input gate, and output gate).

Unlike feedforward neural networks, RNNs can use its internal state (memory) to process sequences of inputs. Long short-term memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. LSTM was explicitly designed to avoid the long-term dependency problem**,** which helps the layers to keep the gradient stable.

2.5 Long Short-Term Memory

Long short-term memory (LSTM) units are units of a recurrent neural network (RNN). An RNN made of LSTM units is frequently called an LSTM network. A typical LSTM unit is made of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. To construct an architecture for memory cells and gate units that allows for constant error flow through special, self-connected units without the disadvantages of the naive approach, extend the constant error carrousel CEC embodied by the self-connected, linear unit j from constant error flow: naive approach by introducing additional features (Hochreiter & Schmidhuber, 1997).

A multiplicative input gate unit is introduced to protect the memory contents stored in j from perturbation by irrelevant inputs. Likewise, a multiplicative output gate unit is introduced which protects other units from perturbation by currently irrelevant memory contents stored in j. The resulting, more complex unit is called a memory cell. The j-th memory cell is denoted c_j. Each memory cell is built around a central linear unit with a xed self-connection (the CEC). In addition to net_{c_j}, c_j gets input from a multiplicative unit out_j (the "output gate"), and from another multiplicative unit in_j(the" input gate"). $in'_j s$ activation at time t is denoted by $y^{in_j}(t)$, $out'_j s$ by $y^{out_j}(t)$, We have:

$$y^{out_j}(t) = f_{out}\left(net_{out_j}(t)\right) ; y^{in_j}(t) = f_{in_j}\left(net_{in_j}(t)\right) \quad (10)$$

where

$$net_{out_j}(t) = \sum_u \omega_{out_j} uy^u(t-1) \tag{11}$$

and

$$net_{in_j}(t) = \sum_u \omega_{in_j} uy^u(t-1) \tag{12}$$

We also have,

$$net_{c_j}(t) = \sum_u \omega_{c_j} uy^u(t-1) \tag{13}$$

The summation indices u may stand for input units, gate units, memory cells, or even conventional hidden units if there are any. All these different types of units may convey useful information about the current state of the net. For instance, an input gate (output gate) may use inputs from other memory cells to decide whether to store (access) certain information in its memory cell. There even may be recurrent self-connections like wc_jc_j it's up to the user to determine the network topology. At time, c_j's output $y^{cj}(t)$is computed as

$$y^{c_j}(t) = y^{out_j}(t)h\left(s_{c_j}(t)\right) \tag{14}$$

where the "internal state" $s_{c_j}(t)$ for $t > 0$.

$$s_{c_j}(0) = 0, s_{c_j}(t) = s_{c_j}(t-1) + y^{in_j}(t)\mathrm{g}\left(net_{c_j}(t)\right) \tag{15}$$

The differentiable function g squashes net c_j; the differentiable function h scales memory cell outputs computed from the internal state s_{c_j}. The LSTM contains special units called memory blocks in the recurrent hidden layer. The memory blocks contain memory cells with self-connections storing the temporal state of the network in addition to special multiplicative units called gates to control the flow of information. Each memory block in the original architecture contained an input gate and an output gate. The input gate controls the flow of input activations into the memory cell. The output gate controls the output flow of cell activations into the rest of the network.

Later, the forget gate was added to the memory block. This addressed a weakness of LSTM models preventing them from processing continuous input streams that are not segmented into subsequence. The forget gate scales the internal state of the cell before adding it as input to the cell through the self-recurrent connection of the cell, therefore adaptively forgetting or resetting the cell's memory. In addition, the modern LSTM architecture contains peephole connections from its internal cells to the gates in the same cell to learn the precise timing of the outputs (Sak et al., 2014). An LSTM network computes a mapping from an input sequence $x = (x_1,...,x_T)$ to an output sequence y = $(y_1,...,y_T)$ by calculating the network unit activations using the following equations iteratively from t = 1 to T:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \tag{16}$$

$$\mathrm{f}_t = \sigma\left(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f\right) \tag{17}$$

$$c_t = \mathrm{f}_t \odot c_{t-1} + i_t \odot \mathrm{g}(W_{cx}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o) \tag{18}$$

$$o_t = \sigma\left(W_{ox}\, x_t + W_{om} m_{t-1} + W_{oc} c_t + b_o\right) \quad (19)$$

$$m_t = o_t \odot h(c_t) \quad (20)$$

$$y_t = \sigma\left(W_{ym} m_t + b_y\right) \quad (21)$$

where the W terms denote weight matrices (e.g. W ix is the matrix of weights from the input gate to the input), W_{ic}, W_{fc} , W_{oc} are diagonal weight matrices for peephole connections, the b terms denote bias vectors (b_i is the input gate bias vector), σ is the logistic sigmoid function, and i, f , o and c are respectively the input gate, forget gate, output gate and cell activation vectors, all of which are the same size as the cell output activation vector m, is the element-wise product of the vectors, g and h are the cell input and cell output activation functions, generally and in our model $relu$ is the network output activation function.

3.0 EXPERIMENT DESIGN

3.1 Data Description & Matrix Representation

This research aims to predict solar power generation with hourly time series data (on a rolling basis) for solar power plants located in a certain region of Australia. The dataset included 12 weather variables with the timestamp, as obtained from the European Centre for Medium-range Weather Forecasts (ECMWF). Approximately 14 months' hourly historical power simulation data (01st of April 2012 to 30th of May 2014) from three solar power plants located in a certain region of Australia have been collected. Raw solar data comprise 12 independent weather variables, namely Total column liquid water (var1), Total column ice water (var2), Surface pressure (var3), Relative humidity at 1000 mbar (var4), Total cloud cover (var5), 10-meter U wind component (var6), 10-metre V wind component (var7), 2-meter temperature (var8), Surface solar rad down (var9), Surface thermal rad down (var10), Top net solar rad (var11), and Total precipitation (var12) with three different zone data. During the experiment, only one zone data was used.

In the raw data, total column liquid water is indicated by var1 while total column ice water is represented by var2, and so on. For the machine learning algorithm, all data features are prepared as a matrix. The matrix representation includes a total of 14 independent variables (12 weather variable, date, and hour) as the input matrix while the target matrix is solar power. The input and target matrix below illustrates how the matrix appears in the machine learning algorithm.

$$\begin{bmatrix} Date & Hour & var1 & \cdots & va12 \\ x1,1 & x1,2 & x1,3 & \cdots & x1,14 \\ x2,1 & x2,2 & x2,3 & \cdots & x2,14 \\ x3,1 & x3,2 & x3,3 & \cdots & x3,14 \\ x4,1 & x4,2 & x4,3 & \cdots & x4,14 \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ xN,1 & xN,2 & xN,3 & \cdots & xN,14 \end{bmatrix} \begin{bmatrix} SolarPower \\ y1 \\ y2 \\ . \\ . \\ . \\ . \\ . \\ yN \end{bmatrix}$$

$$[\ \underline{\boldsymbol{X_1}} \quad \underline{\boldsymbol{X_2}} \quad \underline{\boldsymbol{X_3}} \quad \cdots \quad \underline{\boldsymbol{X_{N,}}},14\ ] \qquad \begin{bmatrix} \underline{\boldsymbol{y}} \end{bmatrix}$$

Input matrix **Target matrix**

3.2 Feature Selection

Feature selection attempts to diminish the dimensionality of data and also find the best parameter with an impact on the target variable. Figure 1 shows the importance of each feature, where Hour illustrates the most important and variable 3 (var3: Surface pressure) indicates the least important feature. Feature selection involves picking the most pertinent features (attributes) among the course of action of original ones. This progression is basic for the structure of regression and classification.

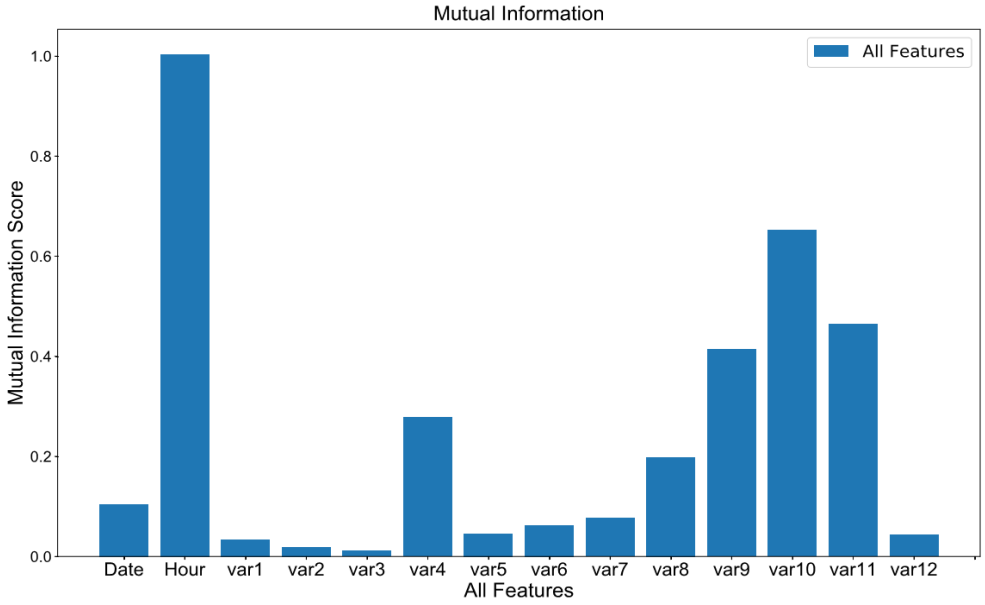

Figure 1: Mutual information

The term "relevant" is related to the impact of the variables on the prediction error of the variable to be regressed (target variable). The relevant criterion can be based on the performance of a specific predictor (wrapper method), or some general relevance measure of the features for the prediction (filter method). The selection method uses a dissimilarity matrix originally developed for classification problems, whose applicability is extended to regression and built using the conditional mutual information between features concerning a continuous relevant variable representing the regression function. Applying an agglomerative hierarchical clustering technique, the algorithm selects a subset of the original set of features (Carmona et al., 2011).

3.3 Selected Features & Weather Profiles

In this part, four weather profiles were made for different weather situations. It was possible to train the model with the selected six best features but it must not be fully dependent on one solar power prediction model for all the months in a year as weather changes continuously. For generating weather profiles, two best features (var10 & var11) among six selected features were selected. Among the six selected features, four different datasets were generated for the experiment based on the condition of relative humidity (var10) and 2-meter temperature (var11) average value (var10 & var11: the best two features among six selected features), the rest of the columns were constant for four weather profiles. The most important features selected: hour, surface thermal rad down (var10), top net solar rad (var11), surface solar rad down (var9), relative humidity at 1000 mbar (var4), 2-meter temperature (var8). Based on four

conditions, four weather profiles were created. The conditions for producing the weather profiles are given below:

Create weather profiles:

$$\text{var10} \geq \overline{X_{10}} \ \& \ \text{var11} \geq \overline{X_{11}} \quad \text{[Profile 1]}$$

$$\text{var10} \geq \overline{X_{10}} \ \& \ \text{var11} < \overline{X_{11}} \quad \text{[Profile 2]}$$

$$\text{var10} < \overline{X_{10}} \ \& \ \text{var11} \geq \overline{X_{11}} \quad \text{[Profile 3]}$$

$$\text{var10} < \overline{X_{10}} \ \& \ \text{var11} < \overline{X_{11}} \quad \text{[Profile 4]}$$

where $\overline{X_{10}}$, $\overline{X_{11}}$ is the average of Surface thermal rad down (var10) and Top net solar rad (var11), initially the average value was calculated from features variable var10 and var11. All data observations greater and lower than the average value of variables var10 and var11 were separated by different files, named Profile 1 and Profile 2, and an average value lower than or equal to var10 and greater than or equal to var11 was named Profile 3. Finally, the average value greater than or equal to var10 and lower than or equal to var11 was separated into another file named Profile 4. During the experiment for every weather profile, 60% of data were chosen for training, 20% for validation, and the rest were for testing.

4.0 RESULTS

4.1 Performance Evaluation

Calculation of the final prediction score for the hourly solar power energy forecasting model includes mean absolute error (MAE) and root mean square error (RMSE) used during the experiment. MAE is a typical measure of figure mistake in time series data. RMSE is a commonly used measure of the contrasts between qualities anticipated by a model or an estimator and the qualities observed. It shows the example standard deviation of the contrasts between anticipated values and observed values.

Prediction using NN and LSTM produced an almost similar result, but LSTM performance was slightly better than NN because of its special gated mechanism (discussed in the LSTM part). LSTM was designed for handling long-term dependency, and for time series data, the network could find the pattern which occurred a long time ago. For that reason, LSTM performed well for all the weather profiles. Table 1 and Table 2 below illustrate the hourly predicted result of NN and LSTM, respectively.

4.2 Neural Net Predicted Results

The consequences of the neural network of the examples recovered toward the end of the activity are summed up in Table 1. Based on MAE (mean absolute error), four profiles have led to an accurate result.

Table 1: Neural net predicted results

Dataset	Predicted Power	
	MAE	RMSE
Profile 1	0.0490	0.1053
Profile 2	0.0640	0.1127
Profile 3	0.0381	0.0724
Profile 4	0.0598	0.1254

4.3 LSTM Predicted Results

The long short-term memory, or LSTM, is perhaps the best RNN as it defeats the issues of preparing an intermittent system and thus has been widely utilized. Compared to the neural network, it produced more accurate results where each profile performed precisely as shown in Table 2.

Table 2: Long short-term memory predicted results

Dataset	Predicted Power	
	MAE	RMSE
Profile 1	0.0381	0.0910
Profile 2	0.0678	0.1167
Profile 3	0.0234	0.0478
Profile 4	0.0261	0.0727

The model error rate for both cases was between 1% to 6% according to the MAE score. For four weather profiles, LSTM error rate was 2% to 3% except for weather profile 2. In all those cases, network trained with little quantity of data because the data processing part only cleaned data selected for the machine learning model and had filtered out all the redundant information from the raw datasets. Although the quantity of data observations was not large, LSTM performed well in all experiments leading to 97% accuracy on average. In each case, LSTM performance was better than NN. By calculating the average value, LSTM gave 97% and NN gave 96% final accuracy based on the MAE score.

5.0 GRAPHICAL REPRESENTATION

This section discloses all the experimental graph results (LSTM and NN). Figure 2 to Figure 5 show the hourly predicted results on a day with a neural network while Figure 6 to Figure 9 demonstrate the same profile results with LSTM. Each profile reveals the 24-hours observation results (blue and yellow lines in the figures indicate the actual and observations). The original data comprised different seasons combinations of data, and each profile had different characteristics. Both cases profile 1 to profile 4 demonstrate the final predicted test results with actual data.

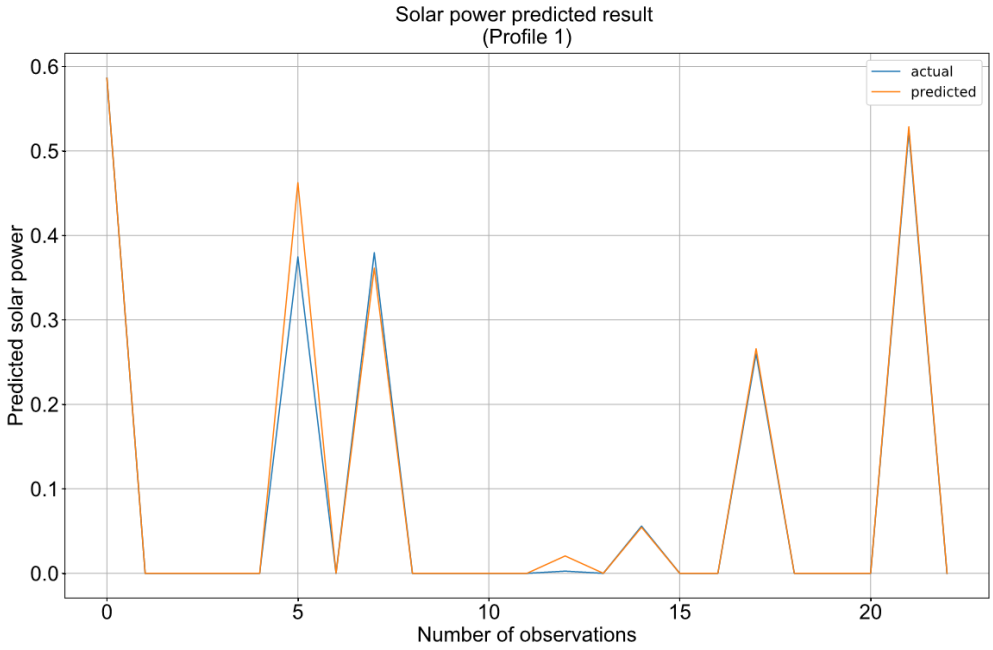

Figure 2: NN predicted result (profile 1)

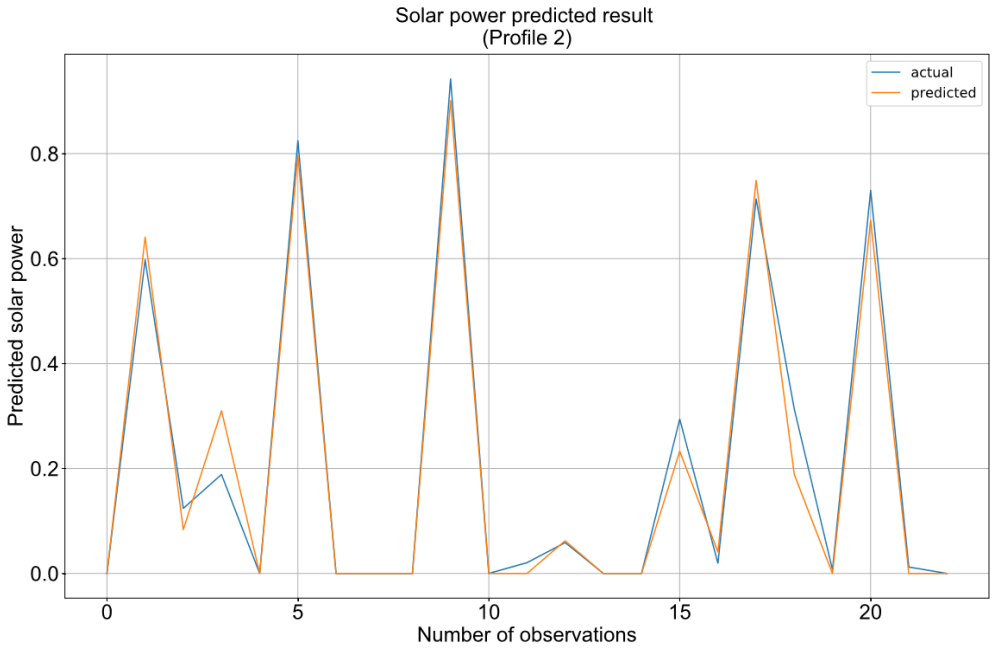

Figure 3: NN predicted result (profile 2)

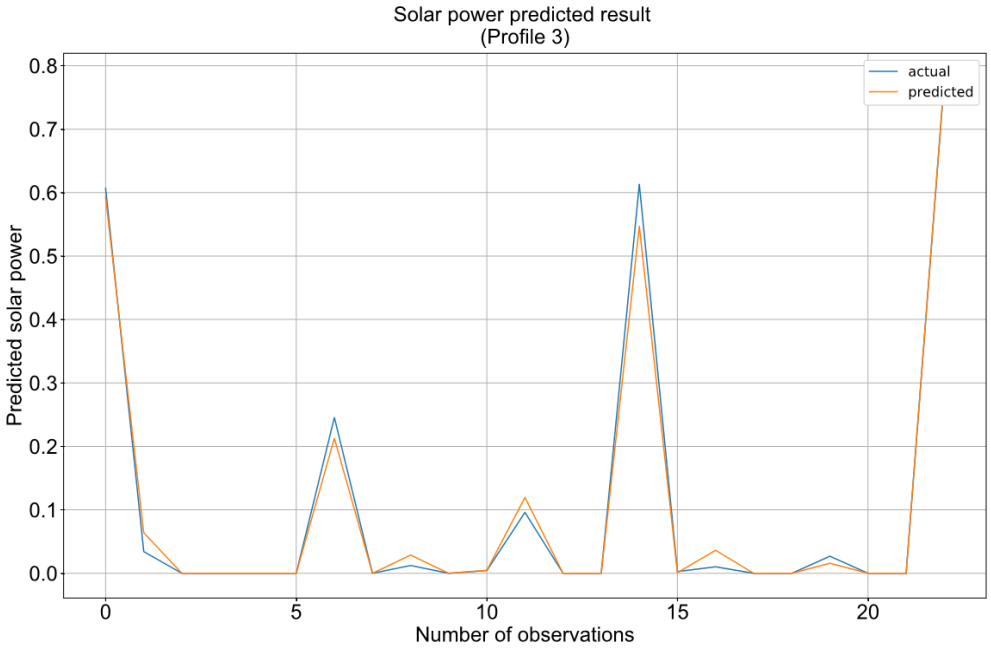

Figure 4: NN predicted result (profile 3)

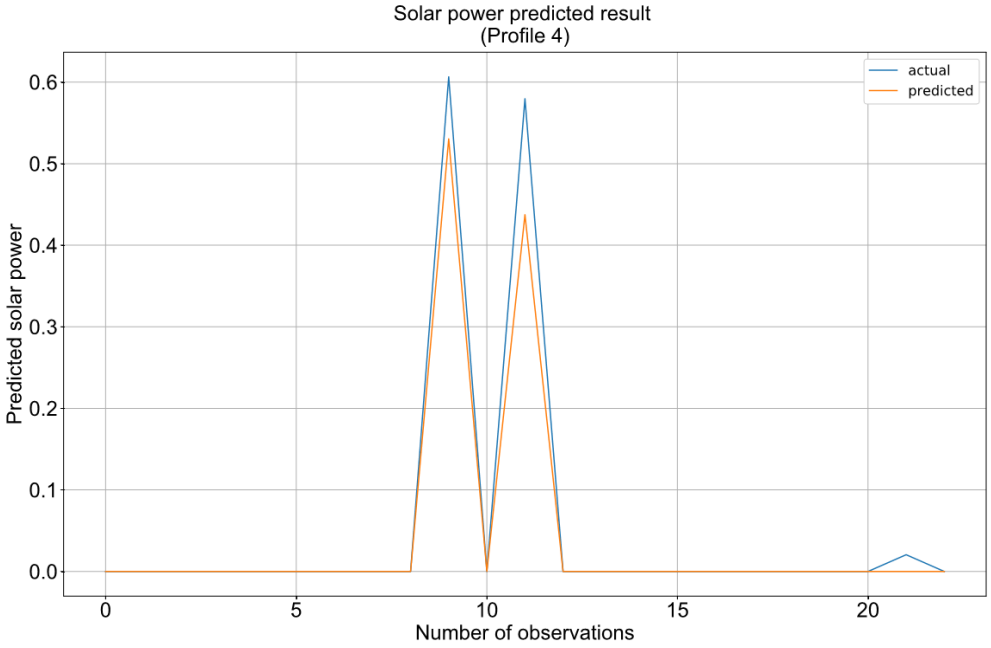

Figure 5: NN predicted result (profile 4)

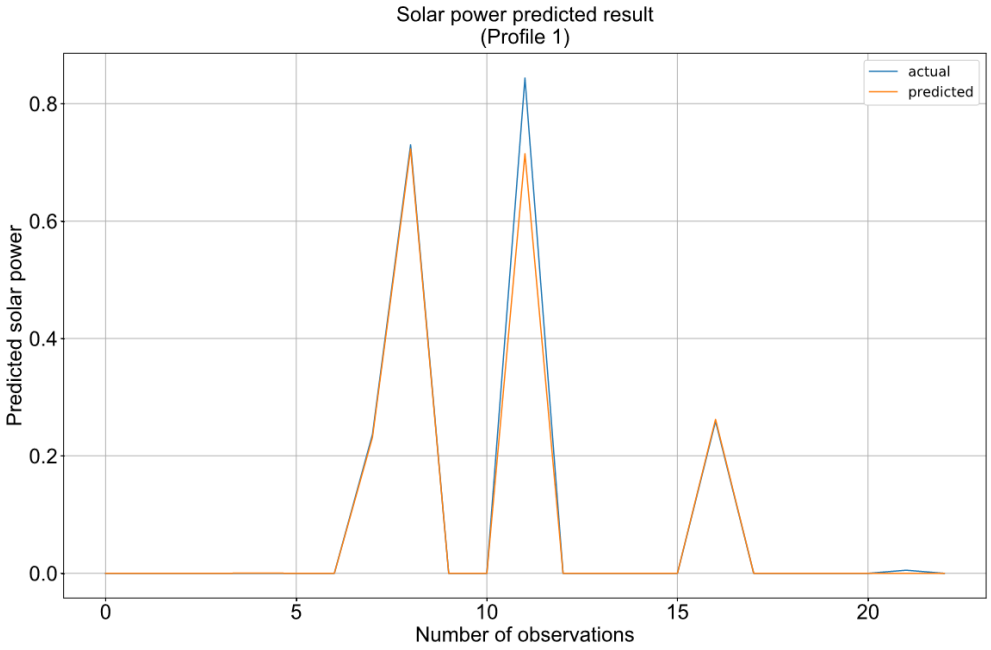

Figure 6: LSTM predicted result (profile 1)

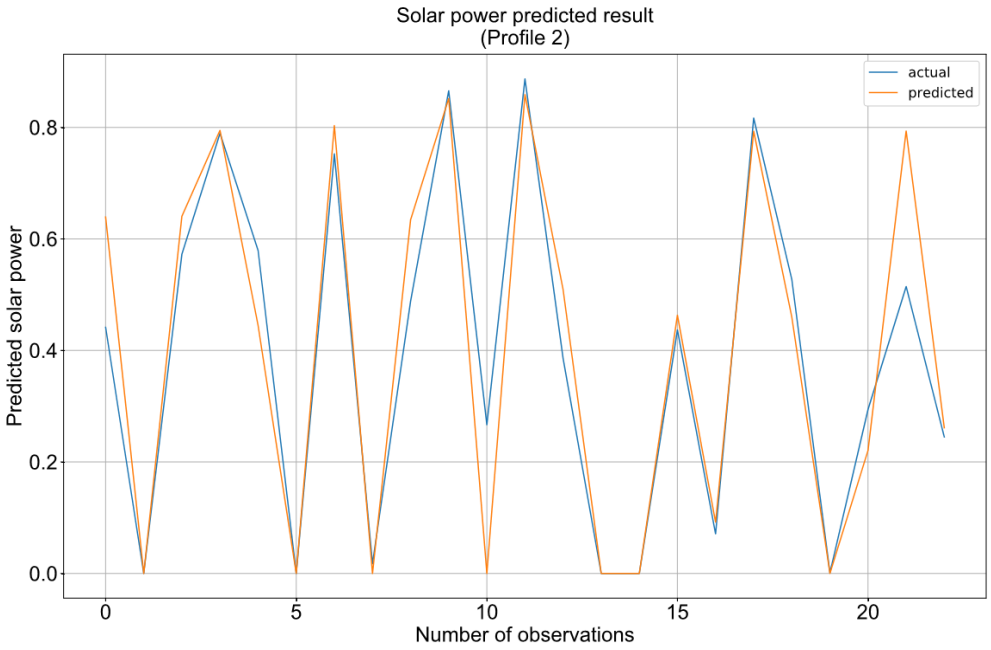

Figure 7: LSTM predicted result (profile 2)

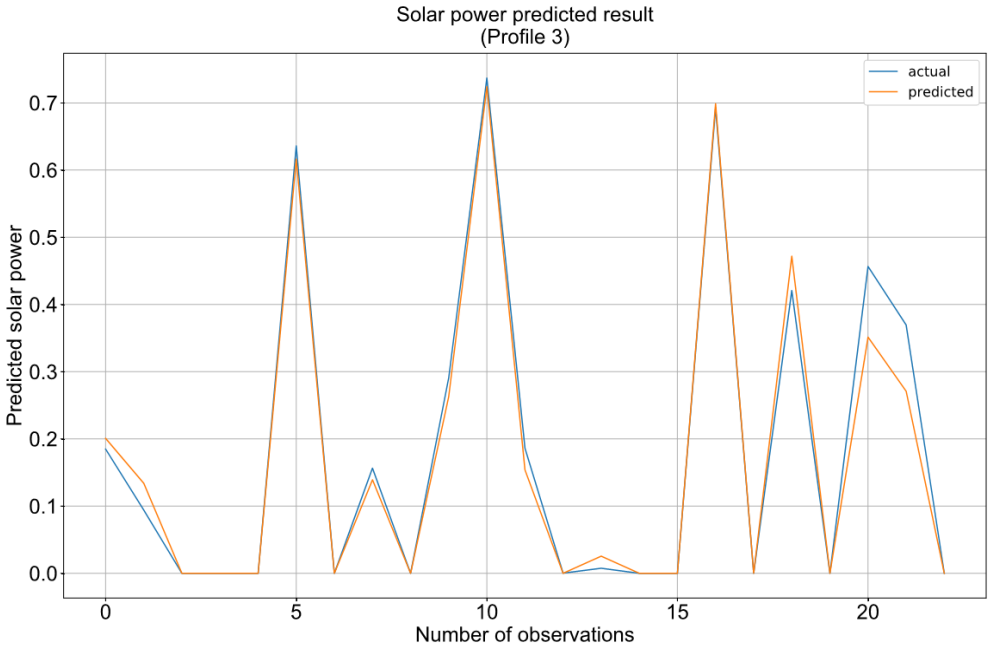

Figure 8: LSTM predicted result (profile 3)

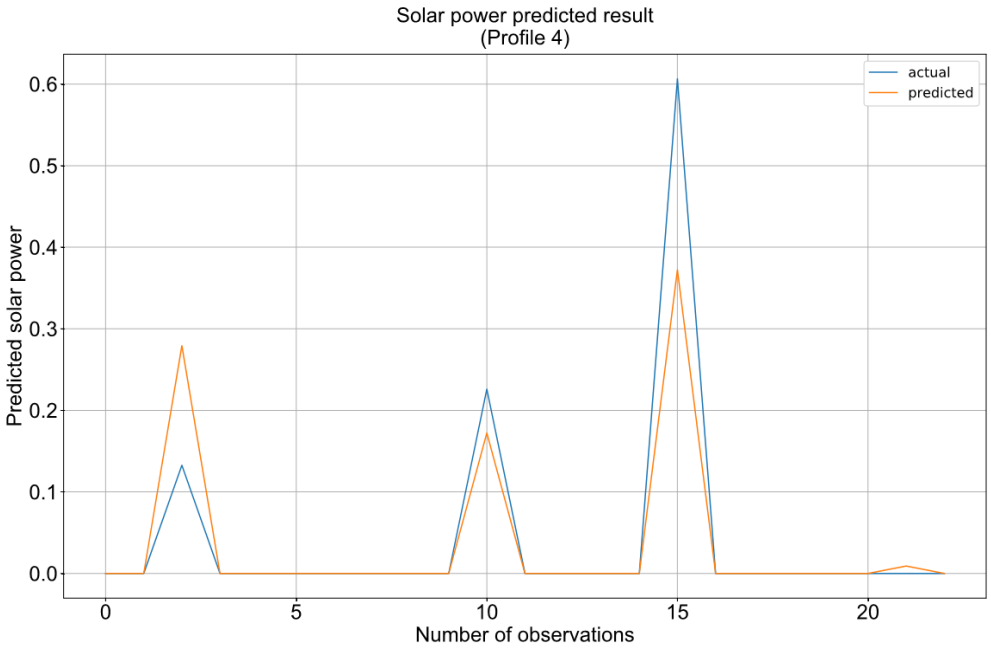

Figure 9: LSTM predicted result (profile 4)

At this point, when data were prepared just once through LSTM, they carried on equivalent to NN as both had a similar system structure. The thought behind LSTM was to store the last yield in memory and utilized that as a contribution for the following progressive advance. LSTM was sort of a repetitive system and worked well with sequence data. The above result illustrates that NN and LSTM performed well. LSTM performed slightly better where 24-hour observations had been plotted each day from profile 1 to profile 4 on a completely independent test dataset.

5.0 CONCLUSION

Solar power energy forecasting with two novel machine learning techniques was presented in this study. Solar forecasting presents more data to partners than point determining, and is a method that requires more research for sustainable power source anticipating. The final outcomes demonstrated that outfit techniques could be utilized to accomplish remarkable outcomes for a solar-based power system. Two different novel data mining and machine learning techniques, namely neural network (NN), and long short-term memory (LSTM) were used in the study. NN performed well but for time series forecasting, LSTM was more reliable because of its special gated mechanism. Both machine learning models were created based on one-year data with four different weather profiles (Profile 1 to Profile 4). Data cleaning (removing redundant information from data, duplicate values, etc.) and applying the data normalization method made the weather profiles data as a clean dataset which forced the network to train well. It also predicted well in the final test phase.

It was also possible to train the network with one dataset (without weather profiles) but in a real-world situation, it was not possible to simply rely on one model because the weather is constantly changing. In addition, making one model could be very complex for the network to accurately predict hourly solar power energy. Splitting the dataset into four different weather profiles made the learning method robust and efficient for good accuracy. The data were fed into the algorithms to train and test the model. According to the theoretical foundation, LSTM should perform better than NN due to its special gated mechanism as we discussed in the machine learning-based method. NN gave us 4% final error where LSTM gave 3%. The one-year time series data were used to train the machine learning model although getting more data could make the model even more efficient. Using the one-year data with different weather profiles, LSTM predicted all the data points accurately. By using more data with the same machine learning method, it could make the model even more robust and efficient in a real-world situation.

REFERENCES

Bishop, C. (2006). *Pattern recognition and machine learning*. NY, USA: Springer -Verlag.

Carmona, P.L., Sotoca, J.M., Pla, F., Hing Phoa, F.K., & Bioucas-Dias, J.M. (2011). *Feature selection in regression tasks using conditional mutual information*. Paper presented at 5th Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA 2011, Las Palmas de Gran Canaria, Spain.

Chi, W.C., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., & Washom, B. (2011). Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. *Solar Energy*, *85*(11), 2881-2893. doi: 10.1016/j.solener.2011.08.025

Chu, Y., Urquhart, B., Gohari, S.M.I., Pedro, H., Kleissl, J., & Coimbra, C.F.M. (2015). Short-term reforecasting of power output from a 48 MWe solar PV plant. *Solar Energy*, *112*, 68-77.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, MIT, USA: The MIT Press.

Hochreiter, S., & Schmidhuber, J.A. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Inman, R.H., Pedro, H.T.C., & Coimbra, C.F.M. (2013). Solar forecasting methods for renewable energy integration. *Progress in Energy and Combustion Science*, *39*(6), 535-576. https://doi.org/10.1016/j.pecs.2013.06.002

Nagy, G.I., Barta, G., Kazi, S., Borbély, G., & Simon, G. (2016). GEFCom2014: Probabilistic solar and wind power forecasting using a generalized additive tree ensemble approach. *International Journal of Forecasting*, *32*(3), 1087-1093. doi: 10.1016/j.ijforecast.2015.11.013

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386-408. doi: 10.1037/h0042519

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modelling. Retrieved from https://arxiv.org/pdf/1402.1128.pdf

Tao, H., Pinson, P., Shu, F., Zareipour, H., Troccoli, A., & Hyndman, R.J. (2016). Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. *International Journal of Forecasting*, *32*(3), 896-913. https://doi.org/10.1016/j.ijforecast.2016.02.001