Weather Based Photovoltaic Energy Generation Prediction Using LSTM Networks

Sahar Arshi
Faculty of Engineering and Environment
University of Northumbria,
Newcastle, UK
sahar.arshi@northumbria.ac.uk

Li Zhang
Faculty of Engineering and Environment
University of Northumbria
Newcastle, UK
li.zhang@northumbria.ac.uk

Rebecca Strachan
Faculty of Engineering and Environment
University of Northumbria
Newcastle, UK
rebecca.strachan@northumbria.ac.uk

***Abstract*— Photovoltaic (PV) systems use the sunlight and convert it to electrical power. It is predicted that by 2023, 371,000 PV installations will be embedded in power networks in the UK. This may increase the risk of voltage rise which has adverse impacts on the power network. The balance maintenance is important for high security of the physical electrical systems and the operation economy. Therefore, the prediction of the output of PV systems is of great importance. The output of a PV system highly depends on local environmental conditions. These include sun radiation, temperature, and humidity. In this research, the importance of various weather factors are studied. The weather attributes are subsequently employed for the prediction of the solar panel power generation from a time-series database. Long-Short Term Memory networks are employed for obtaining the dependencies between various elements of the weather conditions and the PV energy metrics. Evaluation results indicate the efficiency of the deep networks for energy generation prediction.**

Keywords—Photovoltaic systems, Solar panels, Long Short Term Memory, Energy Forecasting.

I. INTRODUCTION

The usage of solar power generators has been encouraged in recent years owing to various environmental benefits. The forecasting of Photovoltaic (PV) power is thus an uprising research branch. However, the power generated by the PV solar devices are highly affected by weather conditions. The energy generated by photovoltaic systems is variable in terms of seasonality and weather factors. The solar panel orientation (East facing or West facing) affects PV output. The imported, exported, and generated energy are important terminologies used in the field. The imported energy refers to the electricity taken from the network by the household, whereas the exported energy refers to the electricity generated, but not consumed by the household [1]. The energy exported is the excess electricity which is injected to the network. The prediction of the generated energy is of great importance to grid operators whose responsibility is to keep the imported and exported energy balanced in the distribution system [2]–[5].

A feed-in-tariff scheme was proposed by UK in 2010 with the main purpose of reducing UK's carbon emission [1]. In accordance to the feed-in-tariff scheme, a low carbon network project was defined in 2015, entitled "Validation of Photovoltaic Connection Assessment Tool". The project involved collecting real-life data for trial purposes, as well as ensuring that the connection assessment tools were fit for purpose [1]. There were some other objectives and benefits related to the project at the time. These include understanding the underlying associated PV generator behaviours by data analysis. The other target was to identify the underlying issues in the connection procedures. The project was successful in achieving a validated and practical connection assessment approach. As a cross product, a rich dataset was produced which is useful for distribution network operators and academic institutions. The produced database is deposited in London datastore and provides the basis for this research study.

In this research, the PV energy generation database will be studied from a different viewpoint rather that the initial targets of the UK Networks project (as mentioned above). The research is in accordance with the related native weather conditions. The output of a PV system highly depends on local environmental circumstances as well as the sun radiation. The sunlight can be occluded by the clouds. The wind blowing direction can bring clouds and effect the way the panel receives the sunlight. There is clearly a correlation between solar radiation and PV outputs. It is often assumed that a PV system is likely to have its maximum output on a clear sunny day. Nevertheless, the temperature is also an important factor on the efficiency of PV generators. The work in [1] demonstrated that PV generators could potentially produce higher levels of electricity on cloudy days. These kinds of analysis results may often be in contradictory with the common beliefs. Therefore, different environmental factors impact the overall energy generation divergently.

In this paper, various weather related data are analysed. An attribute selection procedure is employed for identifying the most important weather factors which affect the generated power. Later, a Long-Short Term Memory (LSTM) network

model is employed for predicting the PV systems power output. The LSTM architecture is equipped with memory units that can be useful in forecasting the temporal variations of PV generated power metrics.

II. Literature review

A. Related Research for Predicting Solar Power Generation

Prediction of PV generated power can be traced back in various studies. The previous approaches in the area of power generation forecasting can be categorized into four major groups including AI, physical, statistical, and hybrid methods.

AI models are one of the popular methods for forecasting the outputs of PV plants. Various machine learning models have been used ever since. Some of the previous research in the area applied artificial neural networks, while recent studies target tools such as deep learning models. Some examples of AI applications in solar radiation forecasting for PV output prediction can be found in [4], [6], [7]. We discuss some of the studies in detail below.

At the University of Illinois, the researchers applied some linear and non-linear machine learning algorithms for forecasting the solar energy generation [8]. They recorded the solar panels outputs at the campus of Illinois University. Their work used weather information and selected day light observations. The employed methods included weighted linear regression trees and LSTM. The later achieved the best performance in their application. The study also found the time-series correlations between all the weather attributes and actual energy output. They found that the cloud coverage, humidity, visibility and dewpoint were the most important features for solar panels output forecast. A similar research in [8] also stated that they could have taken advantage of weather dependency in a way that previous weather conditions would affect the current weather conditions in their model. Instead, they randomized the training dataset.

In [6] , Meyers applied the 3 hourly and 5 minutes time-step intervals to identify the ramp events or deviation from a long-term trend within a short-term period. Their work used auto-regressive model, K-nearest neighbour and artificial neural networks.

In [9], the researchers suggested the usage of LSTM Recurrent Neural Network (LSTM-RNN) to predict the generated power of PV systems using a time-series database with one varying feature. They employed the hourly interval dataset over a period of a year's time. The LSTM model outperformed other methods using the one-featured time-series. The work used various LSTM architectures as well as basic LSTM for regression by using window techniques (LSTM with memory between batches). Their database comes from two sites in Egypt. Other machine learning tools were also applied in their study. However, the LSTM method achieved the best prediction in terms of lower mean absolute error and mean square error compared to those of other methods they applied.

Besides machine learning methods, physical models were also applied for predicting PV system output. The usage of satellites and numerical weather prediction models are two of the physical models which were used in the past for forecasting the solar radiation and indirectly predicting the output of PV systems. The research in the solar power prediction is intertwined with weather forecasting approaches, as the later affects the outcome of the energy generated by PV systems. In [10], the numerical weather prediction models were employed for photovoltaic and solar power generation forecasting. Inman and colleagues in [11] provided a comprehensive literature review on solar forecasting methods for renewable energy integration. They discussed the prediction method for solar resources (weather conditions forecasting) and the power output of the solar plants. Some other recent review works on solar power forecasting can be found in [12].

Moreover, statistical methods have also been used for predicting PV outputs. Statistical models are usually based on the short-term historical data. Auto-Regressive-Integrated-Moving-Average models (ARIMA) are examples of statistical models [13]–[15]. Forecasting the PV output time-series data using ARIMA models was among the first collections of research in this area. In [16], ARMA model with different parametrization are employed to analyse and forecast the residuals in daily solar radiation time-series in Malaysia sites.

Finally, hybrid methods combine various methods together for prediction. For instance, AI models were integrated with physical methods for PV system output prediction. The study in [17] discussed on how the sky images were analysed for forecasting the output of Photovoltaic facility plant in Nevada using image processing techniques.

In this research, an LSTM model as a deep learning model was employed for modelling the time-series dependencies between weather condition changes and solar power generation metrics. The predictions are performed up to 48 time-steps (number of unrollings). Moreover, more than one solar power factors were predicted in a multistep-ahead prediction approach. The model also embeded the dependency pertaining to weather conditions changes.

B. Long Short Term Memory Networks

The LSTM architecture was originally proposed by Sepp Hochreiter and Jurgen Schmidhuber in 1997 [18]. An LSTM network is a Recurrent Neural Network (RNN) [19] which consists of memory units. An LSTM network refines the vanishing gradient problem associated with RNNs [20]. An LSTM cell consists of a memory cell and multiplicative gates which work as regulators. LSTM networks learn short-term or long-term dependencies between elements of time-series data [5]. This characteristic makes LSTM networks suitable for making predictions.

An LSTM cell is similar to RNN in its recurrent architecture. However, LSTM cells are equipped with a memory cell which makes them more desirable for broader ranges of applications. The LSTM architecture includes some gates as well as the input gate (I_t), output gate (O_t), and forget gate (F_t). The variations of LSTM networks may not necessarily include all the mentioned gates. Instead, they may have other gates suited for the specific related applications.

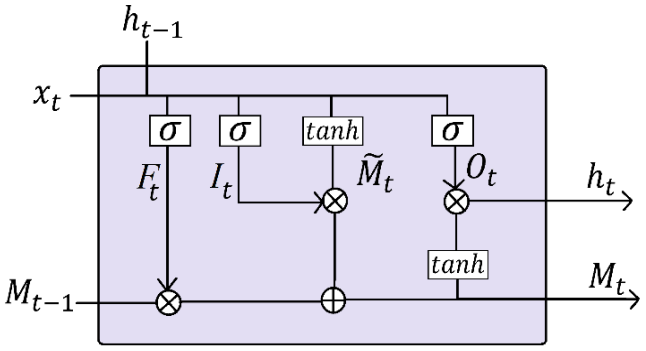


Fig 1. LSTM network cell architecture. There are three major gates involved in this version of LSTM cell. These gates consist of a forget gate, input gate and output gate. The gates have sigmoid and/or tanh layer embedded in their design. The input value (x), previous hidden (h_{t-1}), and previous memory cell (M_{t-1}) information, would structure the current hidden (h_t), and memory cell (M_t). The sign, i.e. the cross in the circle, stands for pointwise multiplication operation, and the operator, i.e. plus in the circle, shows the adding operation.

In this section a brief overview of the functionality of the main LSTM gates are discussed. The input gate determines how the flow of the input data updates the memory state. The forget gate controls whether information should remain in the memory or be forgotten (and to what extent). The output gate sets the extent of which the output is affected by input value and memory unit information[21], [22].

The time-series data are presented as a sequence to the input layer of an LSTM network. The LSTM blocks may also be presented with the previous hidden timestep data of the same LSTM layer (h_{t-1}). The (x_t and h_{t-1}) are passed through different gates as a concatenated vector. The forget gate has a sigmoid layer, which receives the current input and the hidden state of the previous step. The forget gate determines to what extent to maintain or abolish the memory state (M_{t-1}) information. In the next step, a decision is made, i.e. a pointwise multiplication operation is embedded within the gate to obtain $F_t * M_{t-1}$.

The input value and previous hidden state values are also guided through other gates. The next important gate is the input gate layer. In this phase the extent of new information to be stored is determined. The same vector is passed through a $tanh$ layer to obtain the $\widetilde{M}_t$ value. In this stage, the $\widetilde{M}_t$ and I_t values are multiplied together to obtain ($I_t * \widetilde{M}_t$). This result is added up to the result of the previous stage from the forget gate. On this account, the new state cell is achieved. Finally, the output of the LSTM cell is influenced by the memory cell state, the input value and previous hidden state. The Equations 1-6 clarify the above learning process from computational point of view.

$$F_t = \sigma[W_{hf}h_{t-1} + W_{xf}x_t + b_f] \quad (1)$$

$$I_t = \sigma[W_{hi}h_{t-1} + W_{xi}x_t + b_i] \quad (2)$$

$$\widetilde{M}_t = \sigma[W_{hm}h_{t-1} + W_{xm}x_t + b_m] \quad (3)$$

$$M_t = F_t * M_{t-1} + I_t * \widetilde{M}_t \quad (4)$$

$$O_t = \sigma[W_{ho}h_{t-1} + W_{xo}x_t + b_o] \quad (5)$$

$$h_t = O_t * \tanh(M_t) \quad (6)$$

III. Methodology

A. Dataset

The dataset comes from the London datastore [23]. London datastore provides a repository of free databases for public which can be applied for research purposes too. The database employed for this research consists of data related to voltage, current, power, energy and weather from low voltage substations and domestic sites with solar panels. The measurements are collected over 480 days from 27 July 2013 to 19 Nov 2014. There are 20 substations and 10 domestic premises. The database consists of 171 million individual measurements. The measurements take place every one minute during summer 2014 while they are collected at 10 minutes interval throughout the days on a 6 months' time-span. There are also hourly measurements available which come with provided minimum and maximum measurement values. The hourly measurements cover more than one year time. The measurements are collected from customer endpoints, feeders, and networks endpoints and substations.

B. Attribute Selection

In this particular research, the customer endpoints database together with weather database are studied. The power database and weather database were merged together for the related hourly measurements, which are appropriately arranged in records. There was a phase of data preparation involved before the training took place. The wind direction feature is a nominal attribute in the original database and provides the wind orientation. This data feature was converted to numerical data for this application. There were also a data cleaning phase applied which took place to compensate for missing measurements in the database. Records with missing data were whether removed or replaced with data from adjacent time-series records.

The weather data was not available for all the sites. Therefore, a limited selection of the databases was possible at the time. The customer end-point for YMCA, and Maple Drive East were taken into account and their related weather measurements were selected from a separate weather database (also provided in the same data repository in London datastore). There were also a selection to be made for the timespan and the sampling intervals provided for each site. The databases with hourly interval measurement within a yearly span were chosen. This would provide a larger cover over weather changes varieties. (The one minute interval database only covers summer 2014, and the 10-minute database includes 6 months data only. This ended up in the decision of taking the longer time span database to study more varieties of the weather changes). Combining the customer end-point hourly data and weather database leaves 7001 records for YMCA, and 6611 records for Maple Drive East, responsively.

A visual inspection on the mutual correlation of different variables in the database reveals some patterns within the data. There is negative correlation between *in_air_density* and

temperature, and positive correlation between *wind_chill* and dewpoint. It was commonly observed that some of the power related features like *S_Gen_Min* and *I_Gen_Min_filtered* have perfect one to one correlation with each other. Moreover, the mutual correlation of different weather related attributes reveal similar correlational patterns with power features. For example, the correlation of *Hi-Speed* with *I_Gen_Min_Filtered, I_Gen_Max_Filtered, P_Gen_Min, P_Gen_Max, S_Gen_Min, and S_Gen_Max* have similar patterns.

In order to perform the attribute selection process, Weka software [24] was used. The attribute evaluator applied was correlation attribute evaluation using ranker search method. For Weka to work properly we removed all the power related features except *Q_Gen_Min*, which was left as the target (as if we were about to use it in a regression problem).

Table 1 shows the resulted attribute evaluation scores. The features that were discarded from YMCA dataset include: *Rain, ArcInt, THSWIndex, RainRate, WindTx, WindSamp, ISSRecept, InHum, InEMC, HeatD_D, InAirDensity, OutHum* (also shown as grey colour coded cells in the table). These features have the least scoring values in the attribute evaluation process, therefore they were considered as the least desirable features. The attribute evaluation process also took place on the Maple Drive East database. The removed weather features include: *THSWIndex, Rain, RainRate, ISSRecept, WindTx, WindDir, WindSamp, HiDir, InHum, InEMC, InAirDensity, OutHum, HeatD_D.*

The energy related features included in the training were also filtered: The features related to imported energy were ignored in this research. The reason is that it was presumed that the imported energy shows the household energy consumption. The exported energy has also been deprecated, since the focus is on the generated energy by the PV system itself. The relevant power generated features which were used in this research are: *I_Gen_Min_Filtered, I_Gen_Max_Filtered, P_Gen_Min, P_Gen_Max, Q_Gen_Min, Q_Gen_Max, S_Gen_MIN, S_Gen_Max, thdI_Gen_Min, thdI_Gen_Max*

TABLE I. WEATHER ATTRIBUTES EVALUATION USING WEKA CORRELATION ATTRIBUTE EVALUATION USING RANKER SEARCH METHOD FOR YMCA SITE

Rank	Attribute	Score	Rank	Attribute	Score
1	HiSolarRad	0.649559	18	WindRun	0.120447
2	SolarRad	0.6171	19	WindSpeed	0.120447
3	SolarEnergy	0.617099	20	HiDir	0.092432
4	InTemp	0.513786	21	Bar	0.089296
5	InHeat	0.50967	22	Rain	0.007006
6	ET	0.494994	23	ArcInt	0
7	InDew	0.450145	24	THSWIndex	0
8	TempOut	0.435542	25	RainRate	0
9	HiTemp	0.433011	26	WindTx	-0.00065
10	WindChill	0.426131	27	WindSamp	-0.00757
11	HeatIndex	0.417224	28	ISSRecept	-0.0108
12	LowTemp	0.409244	29	InHum	-0.37357
13	THWIndex	0.408528	30	InEMC	-0.40546
14	CoolD_D	0.288888	31	HeatD_D	-0.41545
15	HiSpeed	0.224074	32	InAirDensity	-0.49965
16	DewPt	0.172591	33	OutHum	-0.51351
17	WindDir	0.151451			

a) Preparing Data for Training: Batch Generation

LSTM networks were used for predicting the future power related factors regarding the weather features. This model is implemented in python using TensorFlow [26]. With respect to performing the training task, the data needs to be presented in a specific format. The data was separated to training and testing chunks and a separate MinMaxScaler from Scikit-learn library [27] was used for normalizing the training and testing sets respectively. Around 85% of the time-series data records were chosen for training and the last 15% remaining consecutive time-steps were chosen for testing.

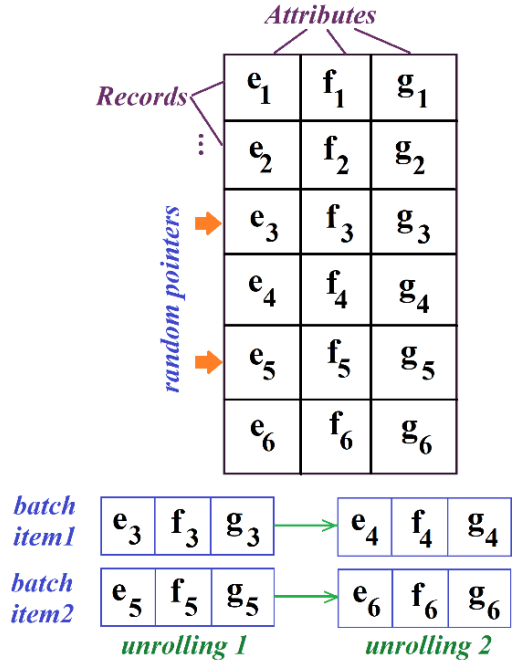


Fig 2. A schematic diagram of the first batch making process. Six records and their three related attributes are shown in this figure. A collection of random pointers are generated which determine the starting locations for producing the batch items. The records from the next time-step are selected to provide the unrollings. Two steps of unrolling are provided for the two batch items in this example.

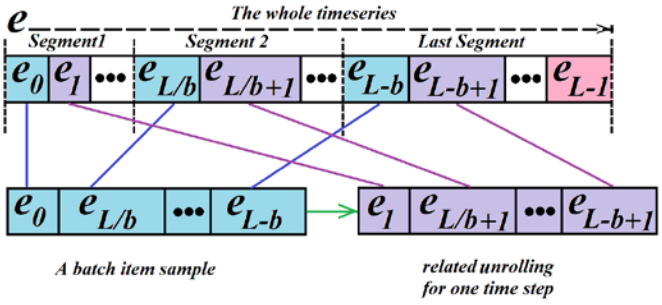


Fig 3. A schematic view of the second batch making process. For simplicity, only one feature is considered. The time-series is divided into segments with equal sizes. A collection of pointers are set to point to the starting location of the segments. The elements of a batch item and their related unrollings are selected on this account.

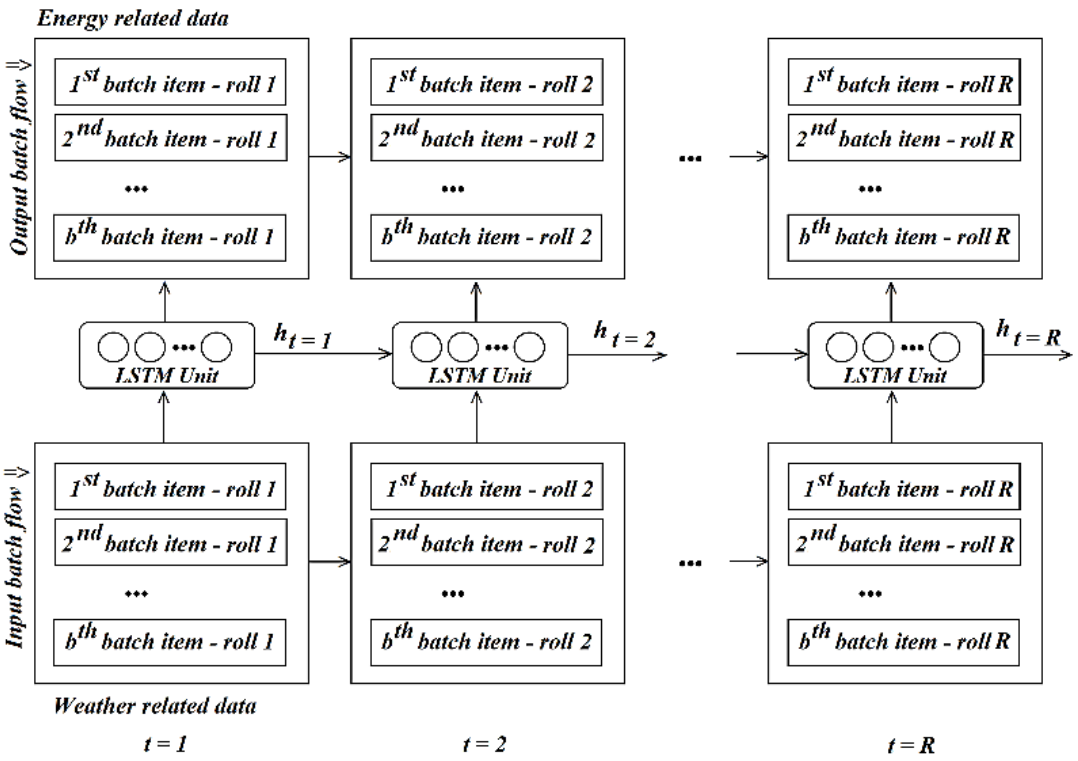


Fig 4. The unrolling process for training LSTM network. The LSTM unit may consist of multiple cells arranged in hidden layers. An LSTM cell architecture was presented in figure 1.

There were no overlapping between the training and test sets. The numbers of the features in the training sets were 21 for YMCA, and 20 for Maple Drive East. All the training features selected were weather data. The number of output factors was selected to be 10. All the output factors were related to be solar energy generation metrics at customer's endpoint. The shape of the data for both the training and testing chunks was the same. Another important step in preparing the data for training LSTM models, like any other neural network model, was to provide the data in the batch format. Two experiments took place in terms of designing the batches. The format of the data batches presented to LSTM network is worth while discussing here. The batch size was selected to be 100 and the number of unrollings was set to be 48. The interpretation of the number of unrolling is in the following: At each of the time-steps, the network takes the output of the antecedent time-steps and feeds them in to the network. The batches were generated in an unrolled format consisting of consequent time-steps of data. The number of unrolling determines the amount of time-steps which are used for training the network in each iteration.

Two methods for generating the batches were used in this research. In the first approach, a number of 100 indexes (equalling the batch size number) were selected randomly from the time-series. A number of 48 (number of unrollings) consecutive records were derived from the database; starting from the preselected indexes. This leaves us with the input batch of three dimensional shape of: (batch size, number of weather features, number of unrollings). The counterpart output batch takes the three dimensional form of: (batch size, number of power features, number of unrollings) [28]. This process for batch generating by shuffling the starting time of a (48-long) time-steps takes place in each training epoch. Figure 2 shows a schematic view of the first undertaken procedure for generating the batches. An alternative approach for generating the batches refers to the way the items in the batch are sampled from the time-series. Like-wise the previous approach, the starting indexes can be selected randomly. However, the unrollings do not take place by selecting the very next adjacent record in the time-series [29]. Instead, the whole time-series is divided to *S* number of segments with equal sizes. The number of segments is obtained by dividing the timeseries length by the batch size. A collection of indices is set to pointing to the starting locations of the segments. The batch generation process takes place in every iteration. Random numbers (smaller than segment size) are added to the pointer locations. This is to make sure that in every iteration we should have different batch. A pointer governs on the element of the time-series to be chosen. Those elements would eventually form an item of the batch. Figure 3 depicts the element selection for an item of the batch and its related one-time step unrolling. The figure shows one dimension of the data for simplicity (for example only the *solarRad* feature). Generating the items in a batch is similar to each other. For avoiding complications, the figure only suggests the generation of one item in the batch including its unrolling in time.

Finally, the training can take place after generating the batches. The LSTM network was chosen to have 2 hidden layers each consisting of 200 cells. The number of training iterations was set to be 40,000. The learning rate was set to be 0.01. Figure 4 illustrates a conceptual design which shows the training process of an LSTM model. The batches can be generated in either of the approaches demonstrated previously.

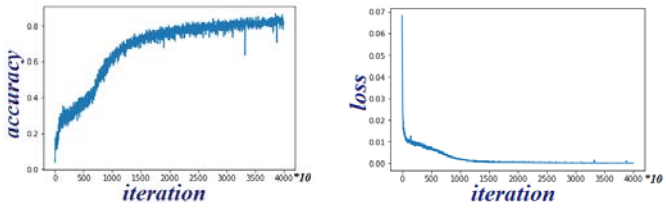
From YMCA database, 6000 samples were chosen for training, and 1001 samples were used for testing. A total of the train samples of 5619 and the test samples of 992 were used for Maple Drive road dataset were also employed. We discuss the evaluation results in the next section.

IV. Results

In this section the results for training the model with YMCA and Maple Drive East sites databases are provided. Figure 5, and 6 show the accuracy and loss values for LSTM trained with the first batch making and second batch making approaches consecutively. Figure 6 has more dramatic jumps. Both of the training curves have similar values at the end of 40000 iterations limit.

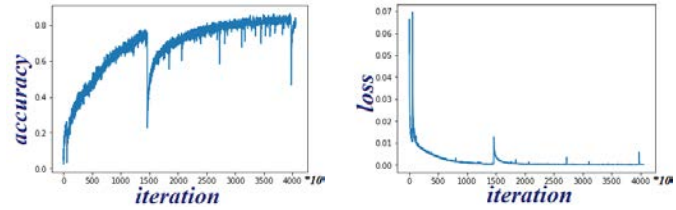
Figure 7, and 8 present the accuracies and losses for Maple Drive East database trained with first and second batch generating algorithms. Figure 8 shows a major drop in the accuracy while training. The algorithm could not recover from it within the remaining training iterations limit. The nature of the first batch generating algorithm was more suitable for Maple Drive East database. This might be because the data cleaning on Maple Drive database was more severe, because there were a larger proportion of the database which had empty records.

Figure 9 illustrates the output of the model versus the original data. They are respectively shown with red and black lines in the figure. The data is from YMCA testing data. In this figure only three output features are shown. The black line is the original data and the red lines show the predictions. The predictions follow the black line throughout the timeseries almost perfectly.



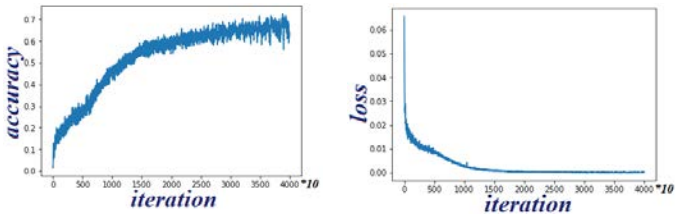
(a) (b)

Fig 5. (a) Training the LSTM network with YMCA time-series data using the first batch making approach. The accuracies are improved over iterations. 40000 accuracy values are depicted in this image. (b) The loss values associated with training the LSTM network using YMCA training set.



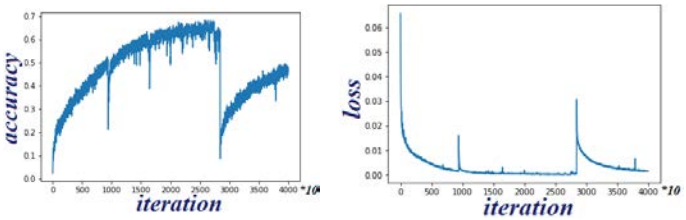
(a) (b)

Fig 6. (a) Training the LSTM network with YMCA time-series data using the second batch making approach. 40000 accuracy values are depicted in this image. (b) The loss values.



(a) (b)

Fig 7. (a) Training the LSTM network with Maple Drive site time-series data using the first batch making approach. (b) The associated loss values.



(a) (b)

Fig 8. (a) Training the LSTM network with Maple Drive site time-series data using the second batch making approach. (b) The associated loss values.

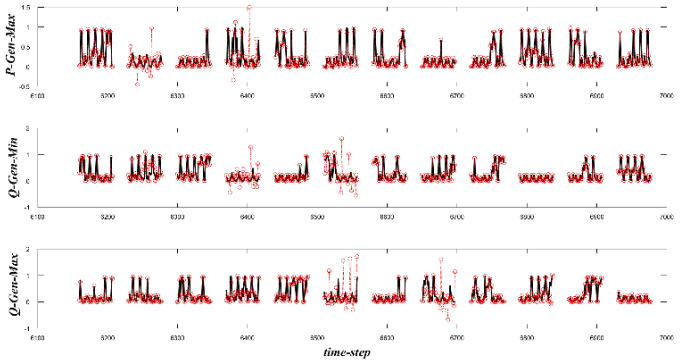


Fig 9. Example of the original data and LSTM output on the testing data from YMCA. There are 3 output features shown in this figure. The black lines show the original data stream. The red dotted lines show the output predictions by the LSTM network. This figure also depicts the batch items with 48 number of unrollings. The starting location for windowing the batches was chosen on every 70 time-series intervals for this example.

TABLE II. The average accuracies and losses in the training and testing phases, for different sites

Training using…	Sites	Training Accuracy	Training Loss	Testing Average Accuracy	Testing Average Error
First batch making approach	YMCA	0.828	0.00015	0.8125	0.00017
	Maple Drive Road	0.643	0.00022	0.6339	0.00025
Second batch making approach	YMCA	0.839	0.00016	0.8024	0.00018
	Maple Drive Road	0.448	0.00145	0.1779	0.0204

Table 2 shows the average accuracies and losses obtained by training and testing the model using YMCA and Maple

Drive East customer endpoint database. Training the model using YMCA database shows more promising results.

V. Conclusion

Photovoltaic systems are a source of clean energy, since they do not contribute to carbon gas emission and are accessible by many households. The environmental fluctuations affect the output of the PV system which can cause problems in electricity distribution network. In order to avoid the incurring issues in the power networks, predicting future solar panel outputs is essential. The model discussed in this work predicts the solar panels energy output regarding weather conditions.

The proposed model is able to predict the solar output under different weather circumstances (which are normal to the location of the solar power plant). The algorithm needs fine-tuning to work perfectly for all the sites. The first batch generating approach seems more promising for a time-series data that does not have many missing values. The database can be expanded to include other varieties of weather conditions to make the related training model more general.

In the future directions, we aim to study the effects of the hyper-parameters on the performance of the algorithm [31-33]. The number of hidden layers and their consisting neurons are another important aspect that is going to be explored. The learning rate, the number of iterations and the TensorFlow optimizer (In this implementation Adam optimizer was used [30]) can be reviewed in order to find the best practical parametrization of the model.

Moreover, although weather prediction is not the ultimate goal of this research, the model is able to contribute to real-time weather forecast since the training data are presented in time series sequences. On this account, a hybrid model could be developed in future research where numerical weather forecast can be embedded. It can be fed to the model in real time to more accurately predict the energy output at the solar power station. We also aim to employ the resulting model combined with Convolutional Neural Networks to tackle other challenging computer vision and image classification problems [34-38].

References

[1] M. Wilcox, “Validation of Photovoltaic (PV) Connection Assessment Tool, Networks, UK Power,” 2015.

[2] A. K. Yadav and S. . Chandel, “Artificial Neural Network based Prediction of Solar Radiation for Indian Stations,” Int. J. Comput. Appl., vol. 50, no. 9, pp. 1–4, 2012.

[3] Y. Li, Y. Su, and L. Shu, “An ARMAX model for forecasting the power output of a grid connected photovoltaic system,” Renew. Energy, vol. 66, pp. 78–89, 2014.

[4] A. Yacef, R. Benghanem, M. Mellit, “Prediction of daily global solar irradiation data using Bayesian neural network: A comparative study,” Renew. Energy, vol. 48, pp. 155–160, 2012.

[5] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, “Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks,” no. July, 2017.

[6] B. Meyers, “Short time horizon solar power forecasting,” pp. 1–6, 2017.

[7] A. Mellit, “Artificial Intelligence technique for modelling and forecasting of solar radiation data: a review,” Int. J. Artif. Intell. Soft Comput., vol. 1, no. 1, pp. 52–76, 2008.

[8] A. Kuzmiakova, G. Colas, and A. Mckeehan, “Short-term Memory Solar Energy Forecasting at University of Illinois,” no. December, pp. 1–6, 2017.

[9] N. Computing, M. Abdel-nasser, and K. Mahmoud, “Accurate photovoltaic power forecasting models using deep LSTM-RNN Accurate photovoltaic power forecasting models using deep LSTM-RNN” , October, 2017.

[10] R. Perez et al., “Comparison of numerical weather prediction solar irradiance forecasts in the US, Canada and Europe,” Sol. Energy, vol. 94, pp. 305–326, 2013.

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

[12] D. K. and I. I, “Solar Power Forecasting: A Review,” Int. J. Comput. Appl., vol. 145, no. 6, pp. 28–50, 2016.

[13] E. J. Hannan and M. Deistler, “Statistical theory of linear systems. Wiley series in probability and mathematical statistics. New York: John Wiley and Sons.Hannan & Deistler (1988, p. 227): Hannan, E. J.; Deistler, Manfred (1988). Statistical theory of linear systems. Wiley series in p,” 1988.

[14] G. Box and G. C. Jenkins, Gwilym M. Reinsel, “Time Series Analysis: Forecasting and Control (Third ed.). Prentice-Hall. ISBN 0130607746.,” 1994.

[15] E. J. Hannan, “Multiple time series. Wiley series in probability and mathematical statistics. New York: John Wiley and Sons.,” 1970.

[16] Y. Sulaiman, M. H. Oo, M. Abd. Wahab, and A. Sulaiman, “Analysis of residuals in daily solar radiation time series. Renewable Energy.,” vol. 11, p. 97–105., 1997.

[17] B. Urquhart, C. Chow, A. Nguyen, J. Kleissl, J. Sengupta, Manajit Blatchford, and D. Jeon, “Towards intra-hour Solar Forcasting Using Two-Sky imagers at a Large Solar Power Plant,” 2012.

[18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1–32, 1997.

[19] A. Karpathy, J. Johnson, and L. Fei-fei, “Visualizing and Understanding Recurrent Networks,” pp. 1–11, 2016.

[20] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, A Field Guide to Dynamical Recurrent Neural Networks,” in IEEE Press, 2001.

[21] C. Olah, “Understanding LSTM Networks,” 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[22] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to Forget: Continual Prediction with LSTM,” Neural Comput., vol. 12, no. 10, pp. 2451–2471, 2000.

[23] UK_Power_Networks, “Photovoltaic (PV) Solar Panel Energy Generation data,” 2017. [Online]. Available: https://data.london.gov.uk/dataset/photovoltaic--pv--solar-panel-energy-generation-data.

[24] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update; SIGKDD Explorations,” vol. 11, no. 1, 2009.

[25] I. Kononenko, E. Šimec, and M. Robnik-Šikonja, “Overcoming the myopia of inductive learning algorithms with RELIEFF,” Appl. Intell., vol. 7, no. 1, pp. 39–55, 1997.

[26] M. Abadi et al., “Large-scale machine learning on heterogeneous systems, Software available from tensorflow.org.” 2015.

[27] A. Pedregosa, F. Varoquaux, G. Gramfort et al., “Scikit-learn: Machine Learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[28] S. Chhabra, “Understanding LSTM in Tensorflow(MNIST dataset),” 2017. [Online]. Available: https://jasdeep06.github.io/posts/Understanding-LSTM-in-Tensorflow-MNIST/.

[29] T. Ganegedara, “Stock Market Predictions with LSTM in Python,” 2018. [Online]. Available: https://www.datacamp.com/community/tutorials/lstm-python-stock-market.

[30] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” in ICLR, 2015, pp. 1–15.

[31] T. Tan, L. Zhang, C.P. Lim, B. Fielding, Y. Yu, and E. Anderson. “Evolving Ensemble Models for Image Segmentation Using Enhanced Particle Swarm Optimization”, IEEE Access, March 2019.

[32] D. Pandit, L. Zhang, S. Chattopadhyay, C.P. Lim, and C. Liu. (2018). “A Scattering and Repulsive Swarm Intelligence Algorithm for Solving Global Optimization Problems”. Knowledge-Based Systems.

[33] T. Tan, L. Zhang, S.C. Neoh, and C.P. Lim, C.P. (2018). “Intelligent Skin Cancer Detection Using Enhanced Particle Swarm Optimization”. Knowledge-Based Systems.

[34] K. Mistry, L. Zhang, S.C. Neoh, C.P. Lim, and B. Fielding. (2017). A micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition. IEEE Transactions on Cybernetics. 47 (6) 1496–1509.

[35] L. Zhang, K. Mistry, S.C. Neoh, and C.P. Lim. (2016). Intelligent facial emotion recognition using moth-firefly optimization. Knowledge-Based Systems. Volume 111, Nov. 2016, 248–267.

[36] W. Srisukkham, L. Zhang, S.C. Neoh, S. Todryk and C.P. Lim. (2017) Intelligent Leukaemia Diagnosis with Bare-Bones PSO based Feature Optimization. Applied Soft Computing, 56. pp. 405-419. ISSN 1568-4946.

[37] P. Kinghorn, L. Zhang and L. Shao. (2019). A Hierarchical and Regional Deep Learning Architecture for Image Description Generation. Pattern Recognition Letters.

[38] P. Kinghorn, L. Zhang and L. Shao. (2018). A region-based image caption generator with refined descriptions. Neurocomputing. 272 (2018) 416-424.