Scientific Research Publishing

Journal of Computer and Communications, 2018, 6, 91-97
http://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Prediction of Solar Radiation Using Data Clustering and Time-Delay Neural Network

Chee Keong Chan, Yi Hong Ler

School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Email: eckchan@ntu.edu.sg

How to cite this paper: Chan, C.K. and Ler, Y.H. (2018) Prediction of Solar Radiation Using Data Clustering and Time-Delay Neural Network. *Journal of Computer and Communications*, **6**, 91-97.
https://doi.org/10.4236/jcc.2018.612009

Received: August 24, 2018
Accepted: December 23, 2018
Published: December 26, 2018

Abstract

In this paper, a combination of data clustering and artificial intelligence techniques are used to predict incoming solar radiation on a daily basis. The data clustering technique known as Perceptually Important Points is proposed, where time-series data is grouped into clusters separated by key characteristic points, which are later used as training data for an artificial neural network. The type of network used is known as a Focused Time-Delay Neural Network, and an analysis of the data is performed using the Mean Absolute Percentage Error scheme.

Keywords

Prediction, Clustering, Neural Networks, Artificial Intelligence

1. Introduction

Currently, fossil fuels such as crude oil, coal and gas are the main resources of energy that are used in today's world. Fossil fuel reserves are expected to deplete in the near future, with studies showing the depletion of oil and gas as fuels expected to occur as soon as 35 to 37 years from the time of writing [1].

As a means to slow down the depletion of fossil fuels, renewable sources of energy are being utilized as alternatives to the aforementioned fossil fuels. Solar energy is one of the alternative sources of energy being researched, as the influx of solar radiation on the Earth's surface is several magnitudes larger than the global power consumption of humanity as a whole [2].

The ability to accurately predict the incoming solar radiation is an important factor to improve the efficiency of a solar energy conversion system. One of the methods utilized to predict incoming solar radiation is the use of an empirical model. The empirical model is a technique which uses meteorological parame-

ters as inputs to predict future values of solar radiation.

The main shortcomings of the empirical model are its focus on long-term prediction, its reliance on existing meteorological data, as well as its inability to identify abnormalities and account for sudden changes in data.

This paper proposes an alternative to the use of empirical models, which is the utilization of a combination of pattern recognition through data clustering techniques [3] as well as data modelling through artificial neural networks [4]. Through the clustering of data, an organized set of inputs can be used to more accurately train the neural network for use in predicting data. A focused time-delay neural network [5] is then used on each cluster.

2. Perceptually Important Point (PIP)

Perceptually Important Point (PIP) is a concept introduced by Chung FL, Fu TC, Luk R, and Ng V [6]. The purpose of PIP is to define the shape of a graph using points on the graph that are shown to be "critical points": points which signify a point of change in the trend of the graph. The graph can be divided into clusters separated by the PIPs, which allows for data to be grouped by trend similarity.

The process behind identifying PIPs is as follows:

1) The start and end of the graphs are set to be the starting and end points.

2) The point on the graph with the greatest vertical distance is set as the first PIP, and is also used as the end point of the first cluster, and the start point of the second cluster.

3) Subsequent PIPs are obtained by finding a point with the greatest vertical distance from the start and end points of each cluster, forming new clusters.

Algorithmically, the process behind segmenting the graph through PIPs can be explained as follows (**Figure 1**).

Points P1 and P2 on the time-series are established and a gradient is obtained between the two points.

A point on the gradient Pc and a point on the time-series Pn with the same x-axis value are obtained. Initially, $(x_1 + 1)$ is selected as the initial x-axis value.

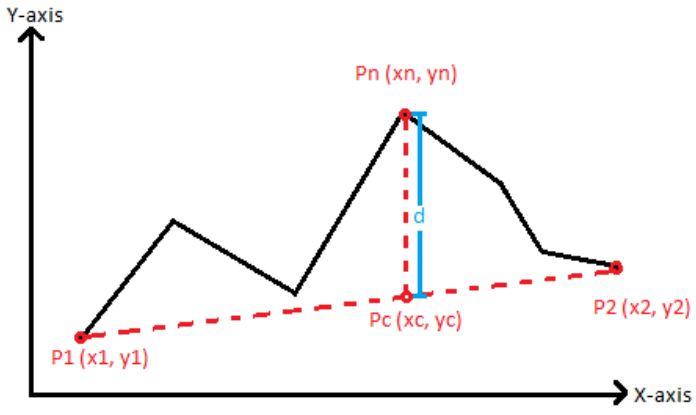

Figure 1. Finding first point.

The difference in values of the y-axis of points Pn and Pc is obtained, and set as distance d.

The x-value of Pc and Pn are incremented by one step and the distance d is calculated again. If the value of d obtained is greater than the previous value of d, the new value is stored. Otherwise, the new value is discarded and the old value of d is kept as the greatest vertical distance.

The x-value of Pc and Pn is repeatedly incremented until P2 is reached.

The time-series can subsequently be segmented into cluster 1 (data from points P1 to Pn) and cluster 2 (data from points Pn to P2) (**Figure 2**).

The process is repeated recursively for cluster 1 and cluster 2 to obtain points Pm and Pl, which segments the time-series into 4 clusters: P1 to Pm, Pm to Pn, Pn to Pl and Pl to P2.

3. FTDNN Using MATLAB

The algorithm for training and utilizing a Focused Time-Delay Neural Network (FTDNN) can be obtained from MathWorks' Neural Network Toolbox [7] [8]. FTDNN was selected as the neural network of choice due to its speed relative to other forms of neural networks.

All values or names within <> braces are subject to change as per user specifications.

Firstly, the cluster dataset has to be loaded and converted into a time sequence using the following commands:

load <data>

y = y(1:<size_integer>);

y = con2seq(y);

The FTDNN is then created using the following commands, with the tapped delay lines, hidden layer neurons and number of epochs being variable depending on optimal parameters:

<network_name> = newfftd(y,y,[1:<delays_integer>],<number_of_neurons_integer>);

<network_name>.trainParam.show = <show(integer)>;

<network_name>.trainParam.epochs = <number_of_epochs_integer>;

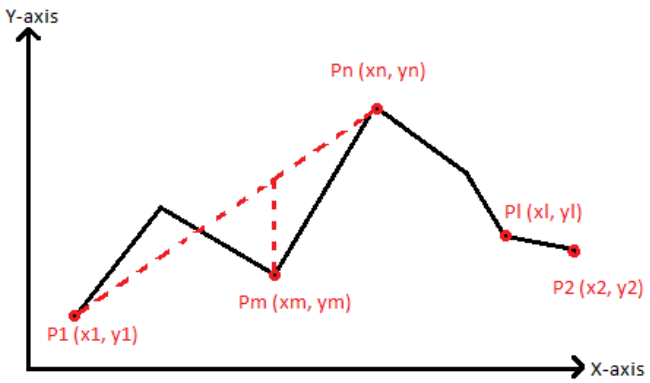

Figure 2. Finding subsequent points.

The prediction begins on the value in the series after the delay, and the initial values in the delay are also required to be loaded:

p = y(<delays+1_integer>:end);

t = y(<delays+1_integer>:end);

i = y(1:<delays_integer>);

The network is then trained to perform one-step-ahead prediction:

<network_name> = train(<network_name>,p,t,i);

The network is then ready for use through the calling of the network as a function:

<result> = <network_name>(<target>);

The resulting prediction can subsequently be converted for plotting:

<result> = cell2mat(<result>);

plot(<result>);

4. Results

Using the PIP method described earlier, 3 points are obtained. Due to the recursive nature of the PIP algorithm, an odd number of points will always be obtained.

For this paper, all solar radiation data is obtained from the Geography Weather Station of the National University of Singapore.

As shown in **Figure 3**, the point (705, 784) is seen to be a PIP, indicating a turning point in the graph. (705, 784) in the context of **Figure 3** indicates that on the 705th minute of the day, there is an incoming solar energy of 784 WH/m^2.

The first cluster is selected to be data from minute 405 to minute 705. The point at minute 480 is ignored to reduce the number of clusters due to the similarity in trend from the data prior to and after minute 480.

The next 17 readings are used to form cluster 2, and the subsequent 20 readings are used to form cluster 3. Readings before cluster 1 and after cluster 3 are

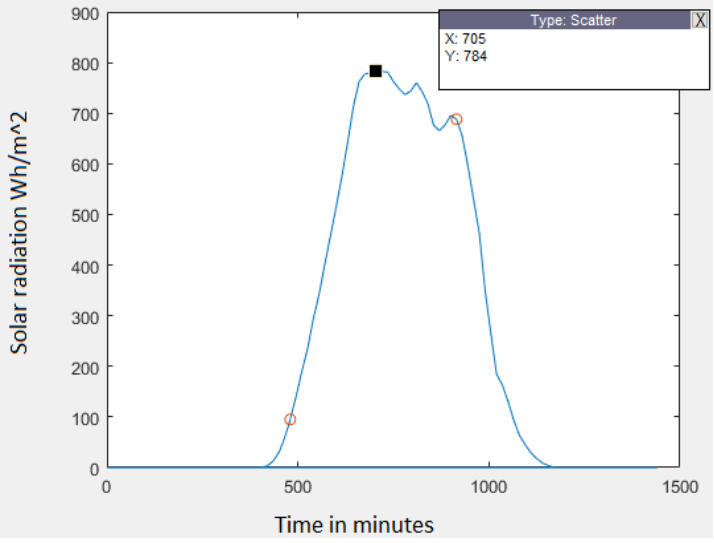

Figure 3. Executing PIP on 1st August 2017 solar radiation data.

ignored due to them being approximately zero, which are not required for use as training data as it indicates that there is no solar radiation incident during that period of time.

Figure 4 shows a visual representation of the actual data (blue) plot against the predicted data (orange). The data is obtained using 1 day of training data as inputs for the network, to predict a day's worth of solar radiation. Visually, the two sets of data are shown to be fairly close.

Using MAPE [9], it can be observed that for most of the readings, the percentage difference between the forecast and actual value are sub-10%, ignoring the values that are close to 0 due to the large deviation it causes in percentage errors when the values are small (**Figure 5**).

Figure 6 shows the result of utilizing the FTDNN on all 3 clusters individually, with 15 days of training data used as the input for each cluster.

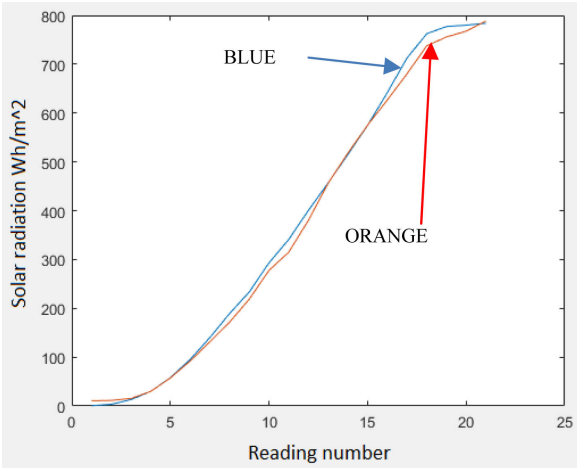

Figure 4. FTDNN on cluster 1.

Reading #	Forecast	Actual	|A-F|/|A|
1	10.68424	0.555333	18.23932
2	11.3054	3.412	2.313423
3	15.43352	13.032	0.184279
4	29.77609	29.83667	0.00203
5	57.42128	58.08	0.011342
6	91.98741	95.19333	0.033678
7	131.3044	140	0.062112
8	171.5724	189.2	0.093169
9	218.7286	233.5	0.063261
10	277.6207	293.1	0.052812
11	314.947	341	0.076402
12	380.1995	400.5667	0.050846
13	456.1203	456.7667	0.001415
14	517.877	515.0333	0.005521
15	575.2269	574.7333	0.000859
16	626.6113	641.7667	0.023615
17	680.3223	712.6333	0.04534
18	737.709	762.3333	0.032301
19	755.9874	777.3333	0.02746
20	767.5858	780	0.015916
21	788.55	784	0.005804

Figure 5. Percentage differences in readings of cluster 1.

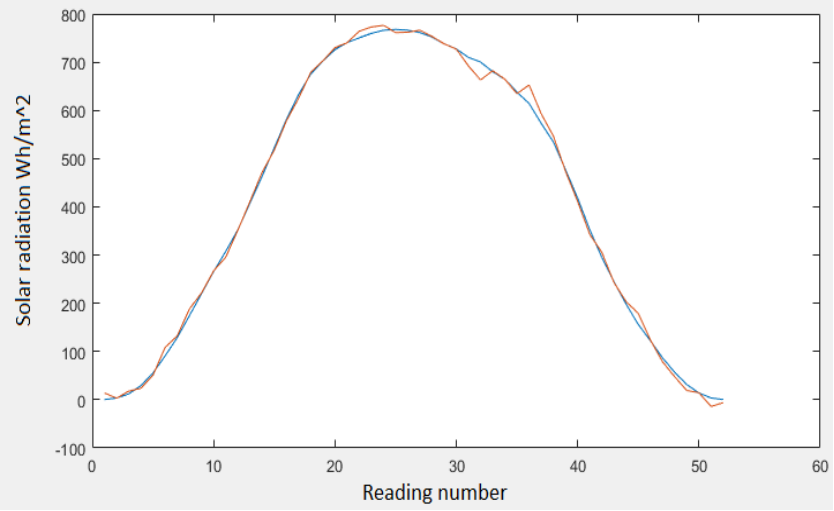

Figure 6. FTDNN on 3 clusters combined.

The resulting adjusted MAPE values of cluster 1, 2 and 3 are 3.890%, 4.129% and 1.180% respectively.

While MAPE is shown to not be entirely exact in portraying the accuracy of the prediction, it is sufficient to show a basic level of competency of the network in performing predictions.

5. Conclusions

In this paper, a system combining the use of data clustering and neural networks is proposed to optimize the prediction of solar radiation.

The paper provides a fundamental level of knowledge on Perceptually Important Points and the focused time-delay neural network that was used in the project.

The use of the methodologies discussed are not limited to the prediction of solar radiation, but can also be used in a more general case for other fields such as the prediction of stock market trends or water current strength for turbines.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Shafiee, S. and Topal, E. (2009) When Will Fossil Fuel Reserves Be Diminished? *Energy Policy*, **37**, 181-189. https://doi.org/10.1016/j.enpol.2008.08.016

[2] Lewis, N.S. (2007) Toward Cost-Effective Solar Energy Use. *Science*, **315**, 798-801. https://doi.org/10.1126/science.1137014

[3] De Soete G. and Carroll, J. (1994) K-Means Clustering in a Low-Dimensional Euclidean Space. In: Diday, E., Lechevallier, Y., Schader, M., Bertrand, P. and Burtschy, B., Eds., *New Approaches in Classification and Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization*. Springer, Berlin, Hei-

delberg. https://doi.org/10.1007/978-3-642-51175-2_24

[4] Haykin, S. (2009) Neural Networks and Learning Machines. Prentice Hall/Pearson, New York.

[5] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K. (1989) Phoneme Recognition Using Time Delay Neural Networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, **37**, 328-339. https://doi.org/10.1109/29.21701

[6] Fu, T.C., Chung, F.L., Ng, V. and Luk, R. (2001) Evolutionary Segmentation of Financial Time Series into Subsequences. Proceedings of 2001 *Congress on Evolutionary Computation*, Seoul, 27-30 May 2001, 426-430.

[7] Miller, S. (2015) How to Build a Neural Network. https://stevenmiller888.github.io/mind-how-to-build-a-neural-network/

[8] Mathworks (2017) Neural Network Toolbox (r2017b): Design Time Series Time-Delay Neural Networks. https://www.mathworks.com/help/nnet/ug/design-time-series-time-delay-neural-networks.html

[9] Swamidass, P. (2000) MEAN ABSOLUTE PERCENTAGE ERROR (MAPE). Encyclopedia of Production and Manufacturing Management. Springer, Boston, MA. https://doi.org/10.1007/1-4020-0612-8_580