Improving the Quality of Load Forecasts Using Smart Meter Data

Abbas Shahzadeh, Abbas Khosravi, and Saeid Nahavandi

Centre for Intelligent Systems Research,
Deakin University
{ashahzad,abbas.khosravi,saeid.nahavandi}@deakin.edu.au
http://www.deakin.edu.au/research/cisr

Abstract. *For the operator of a power system, having an accurate forecast of the day-ahead load is imperative in order to guaranty the reliability of supply and also to minimize generation costs and pollution. Furthermore, in a restructured power system, other parties, like utility companies, large consumers and in some cases even ordinary consumers, can benefit from a higher quality demand forecast. In this paper, the application of smart meter data for producing more accurate load forecasts has been discussed. First an ordinary neural network model is used to generate a forecast for the total load of a number of consumers. The results of this step are used as a benchmark for comparison with the forecast results of a more sophisticated method. In this new method, using wavelet decomposition and a clustering technique called interactive k-means, the consumers are divided into a number of clusters. Then for each cluster an individual neural network is trained. Consequently, by adding the outputs of all of the neural networks, a forecast for the total load is generated. A comparison between the forecast using a single model and the forecast generated by the proposed method, proves that smart meter data can be used to significantly improve the quality of load forecast.*

Keywords: Smart Meters, Clustering, Neural Networks, Load Forecast, Wavelet Transformation

1 Introduction

A study by Hobbs et al. in 1999, reports that in a 10000 MW system, when the forecast error is in the range of 3% to 5%, every 1% reduction in forecast error can save between 0.1% to 0.3% in generation costs [1]. According to this reference, this saving would translate to $US1.6 million per year in 1999 for a 10 GW electrical system. Considering the fact that the electricity market has been restructured in many countries and that the price of electricity is directly affected by the size of demand, it is safe to assume that the savings due to an improvement in the accuracy of forecast, can be much bigger now than it used to be back in 1999. For instance, in many electricity markets, the trading is carried out in two phases namely day-ahead phase and real-time phase. Naturally the price per

MWh in the real-time phase is higher. Electricity market in California can be taken as an example. In California, the average day-ahead price is $52/MWh while the average real-time price is $70/Mwh[2]. Obviously in such a system, a utility company can make huge savings by optimizing its purchases in the day-ahead market which in turn requires an accurate demand forecast.

There has been a huge interest in load forecast in academia. Many forecast methods have been applied to this problem. Artificial Neural Networks [3], Regression Analysis [4], Fuzzy Modelling [5], Interval Type-2 Fuzzy Logic Systems [6] and Support Vector Machines [7] are some of the methods that have been used to provide solutions for the load forecast problem. However, despite years of study, there is still room for improvement in the performance of forecast.

At the same time, smart meters are about to completely replace traditional meters all over the world. The main difference between smart meters and traditional meters is the frequency of recording the electricity consumption of consumers. In the case of older meters, the consumed energy was usually recorded once per month. However smart meters can record energy consumption in every 15 minutes or at higher frequencies. Consequently a large volume of data is available now which was not accessible in the past.

In this paper it will be demonstrated how by using smart meter data, the forecast error can be significantly decreased. The rest of this paper is organized as follows. In section 2, the dataset used for testing the proposed method in this paper is introduced. In section 3, the problem is formulated and the concept of clustering is explained. In section 4, a single model for forecasting the aggregated load is introduced and the results are reported to be used as a benchmark for comparison. In section 5, wavelet transformation and a clustering method called I-k-means [8] are explained. In Section 6, the proposed method is applied to the dataset and the forecast results are compared with the single model method. Finally, section 7 concludes the paper.

2 Smart Meter Dataset

A very important prerequisite for a research about smart meter data is having access to a dataset of smart meter records. Unfortunately, most utility companies are not willing to share their smart meter data with researchers mainly due to privacy concerns. However, there is a dataset provided by Irish Social Science Data Archive (ISSDA)[1], which has been made available for academic and research purposes. The data is gathered by Commission for Energy Regulation (CER) in Ireland and contains smart meter data of more than 6000 Irish consumers from 14 July 2009 to 31 December 2010 [9]. In this research only the smart meter records of 3176 residential consumers has been used.

[1] www.ucd.ie/issda

3 Problem Statement

As explained in section 2, the dataset used in this paper contains the information of M=3176 consumers. The data is provided as M time series vectors. The vector corresponding to consumer number i is given in equation (1) in which, n shows the length of the time series.

$$\mathbf{l}_i = \{l_i^1, l_i^2, ..., l_i^n\} \tag{1}$$

The main goal of this paper is to generate two forecasts using two different methods and then compare the results. The first method uses only total load of the system consisted of all M consumers. The second method is explained by detail in section 3.2.

3.1 Single Model Method Formulation

The total load of the system can be expressed as a time-series like equation (2). In this formula, n shows the number of time intervals in the period of interest.

$$\mathbf{L}_{total} = \{L_{total}^1, L_{total}^2, ..., L_{total}^n\} \tag{2}$$

Each element of $\mathbf{L}_{total}$ in (2) is calculated using (3). In other words, in each time interval like t the total load of the system can be easily calculated by adding the loads of all the consumers as expressed by equation (3). In (3), M shows the number of consumers in the network.

$$L_{total}^t = \sum_{i=1}^{M} l_i^t \tag{3}$$

A model like the one depicted in figure 1, can be used to generate a forecast for the total load. The model itself will be explained in detail in the next section. However the output of the model will be a time series given in equation (4).

$$\hat{\mathbf{L}}_{total} = \{\hat{L}_{total}^1, \hat{L}_{total}^2, ..., \hat{L}_{total}^n\} \tag{4}$$

To measure the difference between the actual load and the foretasted load, Mean Absolute Percentage Error (MAPE) calculated using equation (5) is used in this work.

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \frac{\hat{L}_{total}^t - L_{total}^t}{L_{total}^t} \tag{5}$$

3.2 Clustering-Based Method Formulation

The main concept of the clustering method is portrayed in figure 2. In this method, the consumers are divided into C clusters denoted as S_1, S_2, ..., S_C. Number of clusters (C) can be any number between 1 and M (number of consumers). When C=1, the clustering method basically reduces to the single model

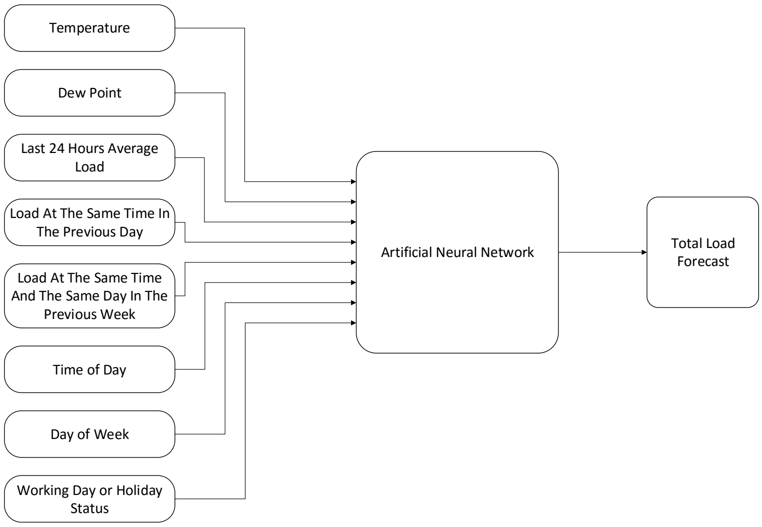

Fig. 1. The ANN model used for forecasting the total load of the system in the single model method and the total load of each cluster in the clustering-based method

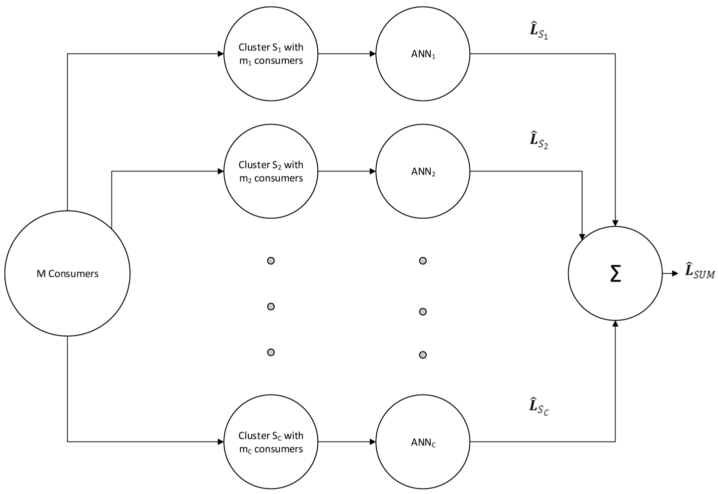

Fig. 2. In the clustering-based method, consumers are grouped into C clusters. For each cluster an individual ANN model generates the forecast . Adding all the forecasts, gives a forecast for the total load of the system.

method explained in the previous subsection. C=M means that for each consumer, an individual model will be trained. The generated forecast for cluster S_k is represented by $\hat{\mathbf{L}}_{S_k}$ as given in equation (6).

$$\hat{\mathbf{L}}_{S_k} = \{\hat{L}^1_{S_k}, \hat{L}^2_{S_k}, ..., \hat{L}^n_{S_k}\} \tag{6}$$

The total load forecast using clustering method is calculated by adding all $\hat{\mathbf{L}}_{S_k}$ values together as given in equation (7) below.

$$\hat{\mathbf{L}}_{Sum} = \sum_{k=1}^{C} \hat{\mathbf{L}}_{S_k} \tag{7}$$

In (7), $\hat{\mathbf{L}}_{Sum}$ shows the total load forecast calculated using the clustering method. By applying formula (5) and using the elements of $\hat{\mathbf{L}}_{Sum}$ instead of $\hat{\mathbf{L}}_{total}$, the forecast error for this latter method can be calculated. A comparison between the two calculated MAPEs can reveal the improvement in the accuracy of forecast using the clustering method.

4 Single Model Forecast

The model portrayed in figure 1, was used to predict the total load of the network. This model has eight inputs with descriptive titles in the figure. The artificial neural network used here is a two-layer network, with 20 sigmoid neurons in the hidden layer and a linear neuron in the output layer. It was trained using the back-propagation algorithm called Levenberg-Marquardt in MATLAB.

Data from 1 February 2010 to 7 March 2010 was used for training the network. The period from 9 March 2010 to 16 March 2010 was used for testing the performance of this model. The experiment was repeated 150 times. The average MAPE using this method was 6.15%.

5 Wavelet Transformation and Interactive k-means

In this section the Wavelet Transformation (WT) formulas will be presented and the concept of Interactive k-means method, which is used to group the consumers into different clusters will be explained. The formulas and notation used for explaining WT are mostly taken from [10].

Before the application of WT to a time series, the time axis of the time-series should be scaled to cover the range from 0 to 1. The signal should also be resampled and expressed as the summation of 2^n samples similar to equation (8) in which $u_i^{(n)}$ is defined by (9).

$$f(x) = \sum_{i=1}^{2^n} a_i \times u_i^{(n)}(x) \tag{8}$$

$$u_i^{(j)}(x) = \begin{cases} 1 & \text{if } \frac{i-1}{2^j} \leq x < \frac{i}{2^j} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The result of applying WT to the function described by (8) for l successive iterations, is given in (10). In this equation, l shows the level of WT and can be any number between 0 and n. In the case of $l = 0$, equation (10) will be the same as (8) and in the case of $l = n$, which corresponds to the deepest level of transformation, only one coefficient (the average of all samples) will remain. The step function term $u^{(n-l)}$ in (10) is again defined by (9).

$$f^{(l)}(x) = \sum_{i=1}^{2^{n-l}} a_i^{(l)} \times u_i^{(n-l)}(x) \tag{10}$$

It should be noted that WT converts a function to a sum of two sets of coefficients namely "detail coefficients" and "approximation coefficients". In this analysis only approximation coefficients are of interest. Equations (9) and (10) basically show how applying WT can give coarser approximations of the signal by increasing the number of iterations. The initial function has 2^n samples with each sample covering a width of $\frac{1}{2^n}$ of the time axis. After applying one iteration of WT, the converted function will have 2^{n-1} samples which is half as many as the initial function. The width of each sample will be doubled so that the new representation of the signal, covers the same area of the time axis as the original function. After the first iteration of WT, each sample will have a width of $\frac{1}{2^{n-1}}$.

Selecting the best number of WT iterations for optimum feature extraction often results in a dilemma. On the one hand, by opting for deeper levels of WT, the number of features in each time series will decrease which in turn can significantly lower the computational cost of data mining. On the other hand, selecting shallower levels of WT, i.e. a smaller number of WT iterations, preserves higher levels of detail in the signal which can improve the quality of the analysis.

Vlachos et al. in [8], have proposed a method to reconcile these two apparently opposing objectives. In their method, called "Interactive k-means", the clustering algorithm is first applied to a coarse representation of the data which is a result of a series of WT iterations. Then, the results of this initial clustering are used to calculate the centroids of the clusters for the next sweep of the algorithm which is applied to a finer representation of data. The same procedure is repeated until the finest level is reached (the original time series before the application of any transformation) or when using shallower levels of WT does not result in any changes in the clustering outcome. The algorithm is summarized in table 1.

6 Case Study

For each consumer in the database, a mean daily load profile is calculated by averaging all 24-hour load profiles during the training period. Since the data is measured in half-hourly periods, each average load profile, initially has 48 elements. In order to use WT for feature extraction, first the time series signals are re-sampled. In this case, a re-sampling phase to convert 48 elements to 64 elements is used. The time axis is also scaled to cover the range between 0 and 1, so that each sample covers $\frac{1}{64}$ of the time axis. After this step, the average

Table 1. Interactive k-means algorithm for clustering consumers according to their daily average load profile

1	Calculate, re-sample and present each consumer's average daily load profile by a function similar to $f(x)$ in equation (8)
2	Set $l := n - 1$
3	Generate C centroids for clusters S_1 to S_C. If $l = n - 1$, generate centroids randomly, otherwise use the results of level $l + 1$.
4	Apply the k-means algorithm. Use 2^{n-l} elements of $f^{(l)}$ from equation (10) as the feature set for each consumer.
5	If $l = 0$ or no change in the clusters in the last run of k-means, stop. Otherwise set $l := l - 1$ and go to 3.

load profile of each consumer is represented by equation (8). Since there are $64 = 2^6$ elements in each time series, n is equal to 6 in this case. Consequently l can range between 0 and 6. The I-k-means algorithm as presented in table 1, will start from $l = 5$ which corresponds to the approximation coefficients after the application of 5 iterations of wavelet transformations. At this level, the load profile of each consumer is condensed in only two coefficients. In the next step, the centroids of the calculated clusters are used to initialize the next k-means algorithm which works on 4 coefficients. The procedure continues according to the algorithm presented in table 1.

After the clustering phase, a neural network is trained for each cluster. The predictions of these ANNs are added together to give a forecast for the total load. The procedure is repeated for different number of clusters ranging from 1 to 100 clusters. The MAPE for each number of clusters in this range is displayed in figure 3.

The results in figure 3, show that up to a certain point, as the number of clusters increases, the quality of forecast improves. By using 24 clusters, a MAPE as low as 3.78% is achieved. However increasing the number of clusters beyond this point, does not result in any improvements in the accuracy of forecast. Therefore, in this case, using 24 clusters, gives the optimum results. By comparing the results attained by clustering (MAPE=3.78%) with the results from section 4 (MAPE=6.15%), a decrease of almost 40% in forecast error can be observed.

7 Conclusion

A method for using smart meter data to improve the accuracy of load forecast was presented. Wavelet Transformation was used in the feature extraction phase and then a clustering method called I-k-means [8] was utilized to group consumers according to their wavelet coefficients. Then an artificial neural network was used to generate a forecast for each group and eventually a forecast for the total load was generated. It was demonstrated that by using smart meter data and clustering, the forecast error can be decreased by almost 40%.

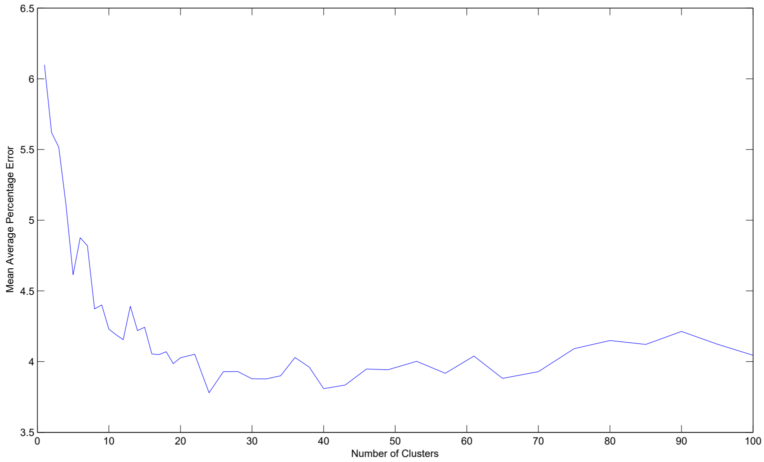

Fig. 3. The graph shows how MAPE changes when the number of clusters increases

References

1. B. Hobbs, S. Jitprapaikulsarn, S. Konda, V. Chankong, K. Loparo, and D. Maratukulam, "Analysis of the value for unit commitment of improved load forecasts," *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1342–1348, Nov. 1999.
2. A. Albert and R. Rajagopal, "Cost-of-service segmentation of energy consumers," *Power Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–9, 2014.
3. D. Park, M. El-Sharkawi, I. Marks, R.J., L. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 442–449, May 1991.
4. A. Papalexopoulos and T. Hesterberg, "A regression-based approach to short-term system load forecasting," *IEEE Transactions on Power Systems*, vol. 5, no. 4, pp. 1535–1547, Nov. 1990.
5. P. Mastorocostas, J. Theocharis, and A. Bakirtzis, "Fuzzy modeling for short term load forecasting using the orthogonal least squares method," *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 29–36, Feb. 1999.
6. A. Khosravi, S. Nahavandi, D. Creighton, and D. Srinivasan, "Interval type-2 fuzzy logic systems for load forecasting: A comparative study," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1274–1282, Aug. 2012.
7. P.-F. Pai and W.-C. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Conversion and Management*, vol. 46, no. 17, pp. 2669–2688, 2005.
8. M. Vlachos, J. Lin, E. Keogh, and D. Gunopulos, "A Wavelet-Based Anytime Algorithm for K-Means Clustering of Time Series," in *In Proc. Workshop on Clustering High Dimensionality Data and Its Applications*, 2003, pp. 23–30.
9. C. Alzate and M. Sinn, "Improved electricity load forecasting via kernel spectral clustering of smart meters," in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2013, pp. 943–948.
10. Y. Nievergelt and Y. Nievergelt, *Wavelets made easy.* Springer, 1999, vol. 174.