International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 4, No.4 (Oct-2015)

http://dx.doi.org/10.12785/ijcds/040403

Load Prediction in Smart Grid Networks

Karthik Sivakoti and Mohammad Mozumdar

Electrical Engineering, California State University Long Beach, USA

Received 6 June 2015, Revised 13 July 2015, Accepted 31 August 2015, Published 1 October 2015

Abstract: Efficient forecasting and load prediction for maintaining the accurate DR (Demand Response) ratio is a key factor in implementing and deploying the Smart-Grid networks [1]. There are a plethora of techniques and models suggested by forecasters over the decades, the most accurate and feasible being – artificial neural networks, linear regression technique and the curve fitting algorithm. Researchers have demonstrated extreme zeal and effort in devloping algorithms which could derive the best effeciency, thus saving excess production than demand. For example, the work descrbied in the paper [2] puts forward the prediction values to be at an accuracy of around 95%. A hybrid algorithm has been presented in this paper, which has been practically proved to have a forecasting efficiency much higher than the conventional methods. Using the artificial neural networks for training the model with historical data and fluctuations in demand, the linear regression method has been used for implementing the temperature sensitivity, namely – dew point, humidity, wind speed, seasonal variations and location of the smart-meter. Together along with the curve fitting algorithm, the proposed hybrid algorithm has been practically implemented by taking data from smart-meters across the United States to determine their efficiency of implementation. The proposed algorithm described in this paper encountered a marvelous prediction accuracy of 99.2% - 99.45%, which promises vast reduction in the power wasted by power utility companies owing to the mismatch within the DR rates from the consumer end and is far accurate than the predictions made by [2].

Keywords: Load Forecast, Smart-Grid Networks, Artificial Neural Networks, Linear Regression, Curve Fitting algorithm, Temperature Sensitivities.

1. Introduction

Power utility facilities across the world are currently faced with the prominent challenges of reducing their greenhouse gas emissions, limiting the peaks of energy cost, optimizing their grids, increasing the supply of eco-friendly renewable source for meeting the energy demand along with efficient power storage and advanced metering infrastructure to monitor efficiently the DR (Demand Response) integrity [3][4][5]. Smart Grid networks have started to replace the conventional electric utilities in recent times, owing to it being an optimized self-healing, intelligent monitoring, communication and control technology. Apart from being more eco-friendly and sustainable, it could provide a higher resilience, reliability and support for the ever-growing economy.

Accurate forecasting of the demand in power load is one of the pivotal aspects, which needs to be considered while focusing on the framework of the Smart-Grid network. Predicting the consumed load accurately limits the wastage of energy and saves astronomical figures of dollars invested by the power utilities on a global scale. To add to the plethora of benefits derived, it would even greatly reduce the greenhouse emissions owing to the matching production and consumption ratio. Forecasters have come up with various algorithms aiming at curbing the above mentioned challenges of which – artificial neural networks, linear regression algorithm and the curve fitting, have proved out to be the most accurate. In this paper, a hybrid algorithm is proposed which uses artificial neural networks for training and testing the data to be forecasted. Moreover, linear regression technique for adding on the temperature sensitivities of humidity, wind speed and location is also proposed. There could be a scenario where some power utilities may not have access to meteorological facilities, which would render the extrinsic variable factors of temperature fluctuations useless and eventually would get down the predicted efficiency. Owing to this factor, historical data has been used in this paper to commensurate with the seasonal variations. Our proposed hybrid forecasting algorithm has been proved practically to have benefited commercially with a forecasting efficiency of 99.2% – 99.45%.

The remaining portions of this paper have been organized as the follows. Section 2 provides an insight over the current forecasting technologies being

E-mail: Karthik.Sivakoti@student.csulb.edu, Mohammad.Mozumdar@csulb.edu

implemented globally and Section 3 puts forward the methodology and the architecture of the proposed hybrid algorithm. Section 4 would detail out the practical implementation of this algorithm by simulating the model and testing upon training the model with smart-meter data from a Commercial Shopping Complex in Los Angeles, United States. On a final note, Section 5 would provide the conclusion of this paper along with a summarization of the findings.

2. RELATED WORK

Predictions and forecasting has always intrigued the human mind. To determine accurately the amount of power to be generated by a power utility company, thus resulting in ecological, monetary and resource conservation, a major emphasis is laid upon forecasting mechanisms.

Under this section, different methodologies have been presented which are currently under use by power utility plants on a global scale.

A. Artificial Neural Networks

The advanced functionalities of the artificial neural network (ANN) are possible due to the interconnected neurons within the network which constantly evolve based upon the synaptic weights. They then back-calculate the weighted average by comparing the threshold value fed with this average and fire the corresponding output signal.

Once appropriate values have been fed to the input layer neurons, the evolution per layer begins wherein the output neurons are triggered because of which the ANN is synonymously even known as the feed-forward algorithm. Synaptic weights and thresholds together contribute towards the training phase of the neurons. Training a neural network involves the optimization of the nonlinear functions dealt within the model. Various methods for solving the nonlinear functions exist – Gauss-Newton algorithm, Levenberg-Marquardt algorithm and genetic algorithm which are all aimed towards finding the local optima and finally getting the forecasted values.

In [6], authors investigated the ability of neural network to predict the hourly load using Particle Swarm Optimization (PSO) algorithm to adjust weights and biases in order to minimize the error objective function of a neural network to obtain a short term load forecasting. PSO shows its ability to minimize the error function with high efficiency. In this study, it was found that selecting proper values of network structure, input data, and PSO parameters are major factors that affect the performance accuracy of the network. It shows also that using weather condition data can positively influence the training and testing results.

In [7], authors presented an algorithm which combined both time series and regression approaches. Their proposed algorithm utilizes a layered perceptron ANN which uses weather information for modeling. The ANN is able to perform non-linear modeling and adaptation. It does not require assumption of any functional relationship between load and weather variables in advance.

In [8], authors used eleven different weather parameters to make load forecasting more accurate. The parameters used for comparing the performance of learning techniques are: mean absolute percentage error, million floating-point operations performed per learning and time taken per learning. The first parameter gives the accuracy of recall: the second, the complexity of the learning technique; the third, a measure of the speed of the learning technique.

B. Linear Regression Technique

In the linear regression algorithm, the scenario is expressed in terms of a linear equation involving several independent variables and a single dependent variable, which would itself be contained within a set of independent attributes. If β is supposed to be the regression parameters which is dependent on the variables x, the load Y can be expressed in terms of the equation (1):

$$Y(t) = \beta_0 x_0(t) + \beta_1 x_1(t) + \beta_2 x_2(t) + \ldots \beta_n x_n(t) \qquad (1)$$

The least square estimate technique is used within the regression model, for minimizing the squared residual sum [9]. The accuracy of a forecasted reading upon passing Phase II, i.e. regression algorithm relies within the extent of the regression function fitting the dataset [10]. Checks are thus imposed into this model by using the Curve fitting algorithm, to make sure that the fitting curve does not exceed to the limit of hampering the whole system [11].

To summarize the inspiration of this model, the individual methodologies of ANN, Linear Regression were used and curve-fitting, comprising of factors – dew point, humidity, temperature, seasonal variations, was utilized. Previous works in this area of load forecasting have only focused on improvising either of the aforementioned methods. Uniquely, our proposed hybrid forecasting algorithm, brings in the propagation training-set ideas of the ANN, along with the variables used to filter out irregularities by using the regression algorithm and curve-fitting to perform multiple iterations mapping the predicted curve based on historical data.

3. ARCHITECTURE OF THE HYBRID ALGORITHM

The proposed algorithm begins with the insignificant attributes being discarded from the data set, followed by the data undergoing the first phase of training and testing through the neural networks before being forwarded to the second phase of linear regression implementation. Our proposed algorithm has been described in Fig. 1.

```
%Hybrid algorithm
extract readings from database;
select the factorization sequence;
outline the filters;
   for Validation Set = Accuracy > 99.5%
      train the sets with user inputted nodes;
      test the sequence;
      search upon the grid for accruacy ratios;
      exogenous factors – linear regression;
      temperature/ Dew Point/ Humditiy
      seasonal variations/ holiday or not;
   end;
pick up a sub-sample test;
check the validation of the tests followed;
```

Figure 1. Algorithm explaining the hybrid proposed algorithm

The initial training set is matched for 70% and 30% of the data being reserved for testing the model, followed by a 50:50 proportion after an ideal number of epochs. The Neural Networks have been implemented using the Levenberg-Marquardt (LM) algorithm.

LM algorithm is an iterative approach for solving the least square equations, involving the forecasting problems. Let us assume J to be the Jacobian matrix of the system under consideration, λ being the damping factor for the Levenberg system, Ϭ being the weight update vector and E being the error vector at the output nodes of the framework deployed. The weight update vector reminds the forecaster of the adjustments he needs to make while the training process for the nodes involved along with the damping factor. The LM algorithm, can be further described to process and solve the equation (2):

$$(J^t J + \lambda I)\, Ϭ = J^t E \qquad (2)$$

Each epoch involved during the phase 1 would involve computing the Jacobian followed by the error gradient, followed by updating the weights using the Ϭ. Further, a squared error sum is calculated followed by decreasing the value of the damping factor once convergence has been reached.

Based on constant simulations, the optimal value for the ANN training and testing sets have been registered to be kept as shown in Table.1.

Table 1. Optimal metric values for ANN Simulation

Learning Rate	0.5
Training period	15000 ms
Hidden Layers	1
Momentum	0
Nodes	25

A. Error determination

The Mean Absolute Percentage Error (MAPE) value serves as the quantitative measure for how efficient a trend estimating method is. If A_t resembles the actual value and F_t is the predicted forecasted value, then MAPE of the particular model is given by the equation (3):

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \qquad (3)$$

B. Data Analysis

Readings of the smart-meter from a commercial shopping complex in Los Angeles (USA) and a power plant situated in New York (USA) were used to test the efficiency and compare the MAPE values of the proposed algorithm with the forecasting models being used currently within power utility organizations. The readings were recorded with an interval of every 5 minutes and have been time stamped individually. To further provide the reader a better insight into the recorded readings, a sample of the readings has been shown in Table 2.

Table 2. Readings received from the Smart-meter

Timestamp	Timeline	Value (Watts)
1325376300	1/1/2012 0:05	136.7605
1325376600	1/1/2012 0:10	136.7605
1325376900	1/1/2012 0:15	137.5465
1325377200	1/1/2012 0:20	147.7642
1325377500	1/1/2012 0:25	144.6203
1325377800	1/1/2012 0:30	138.3325
1325378100	1/1/2012 0:35	133.6166
1325378400	1/1/2012 0:40	137.5465
1325378700	1/1/2012 0:45	133.6166

An initial analysis of the data sets was extrapolated over Matlab to provide the user with a higher insight into the data and variations over a regular interval of 5 minutes, daily, weekly, and monthly basis.

The recorded data gets divided initially into 70% training set followed by 30% of the data being reserved for testing the efficiency. Matching the patterns and back-calculating the weights of the nodes, as specified in the Section 3, an initial output vector weights are recalibrated. Setting the training period and number of epochs as mentioned in Fig. 3, provides the optimal functionality for the training set [12]. Gradually, the forecaster shifts the training and testing node ratio to 1:1.

The next step involves the data to transverse through series of linear regression states. The prediction variables are highly skewed owing to which transformation would be carried out followed by the temperature variables of temperature changes, seasonal variations, dew point and humidity being added from the historical data to the system. The algorithm followed for implementing the hybrid algorithm has been summarized in Fig. 2 and 3.

```
For LM pattern used in Neuron Networks
%Forward computation
    for all layers
        for all neurons in the layer
            calculate net;
            divide into training set;
            training ratio = 70/100;
            training function = train_lm;
            validation ratio = 15/100;
            divide into computational set;
            calculate output;
            test ratio = 15/100;
            calculate slope;
            end;
     end;
```

Figure 2. Algorithm summarizing the forward propagation

```
For LM pattern used in Neuron Networks
%backward computation
Initial delta as slope;
     for all outputs
           calculate error;
           time delay = (input delay, hidden
           layer);
     end;
     for all neurons in the previous layer
         multiply delta through weights;
         sum the back-propagated delta;
     end;
multiply delta by slope;
```

Figure 3. Algorithm summarizing the backward propagation

4. Practical Application of the Algorithm

Once the simulation begins, the ANN network begins forming the network and training it accordingly. Framing the epochs till the appropriate convergence is formed, it forwards it through series of Regression platforms for improving the temperature sensitivity along with the curve fitting algorithms [13].

The MAPE value registered by the hybrid algorithm is 0.86%, which is of a much higher optimal solution when compared with the ANN or Regression algorithm in itself. Plots are created by breaking down the predicted and forecasted data together for a period of whole year, followed by monthly, daily, weekly and hourly basis which are displayed in Fig. 4- 7.

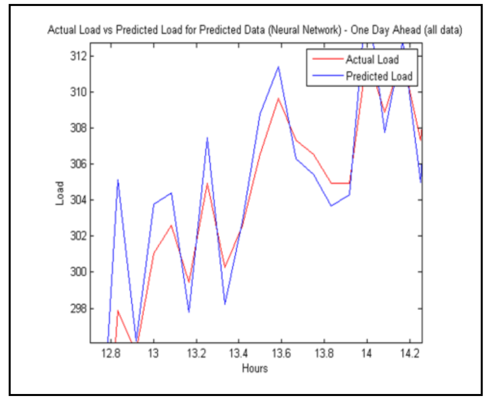

Figure 4. Prediction analysis for Neural Networks

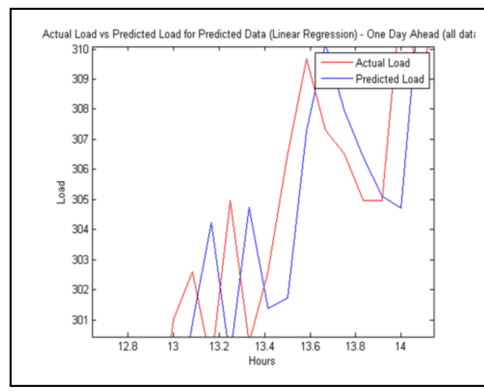

Figure 5. Prediction analysis for Linear Regression

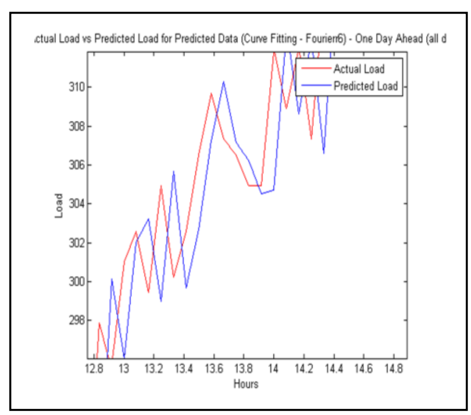

Figure 6. Prediction analysis for Curve fitting model

Figures 4-7 depicts the original load consumed (watts) being plotted upon with respect to time (hours) as the red line and the blue line as the predicted load by the respective algorithm used.

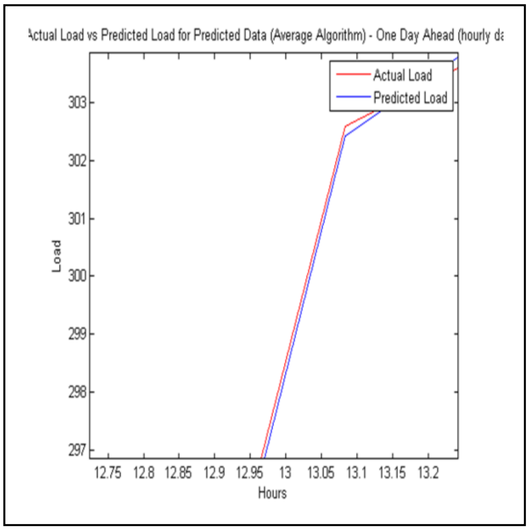

Figure 7. Prediction analysis upon using the proposed hybrid algorithm

5. Conclusion

The proposed algorithm has provided a MAPE of 0.86% upon using the best virtues of both the ANN and linear regression and utilizing curve fitting. The similar problem was solved using ANN, Linear Regression and Curve-Fitting algorithm individually and a MAPE of 1.0636%, 1.5651% and 1.614% respectively when compared with the proposed algorithm which gave a MAPE of 0.866% [14] [15]. These findings have been displayed in Fig. 8 and Table 3. In Fig.8, the red line is the actual load plotted upon and the blue line corresponds to the predicted values upon using the hybrid algorithm. Compared with the other methods being used by the power utility companies and suggested by researchers in the current field, the closeness and accuracy of the prediction provided by the hybrid algorithm is of a higher degree. The summarization of the simulations has been provided under table. 3.

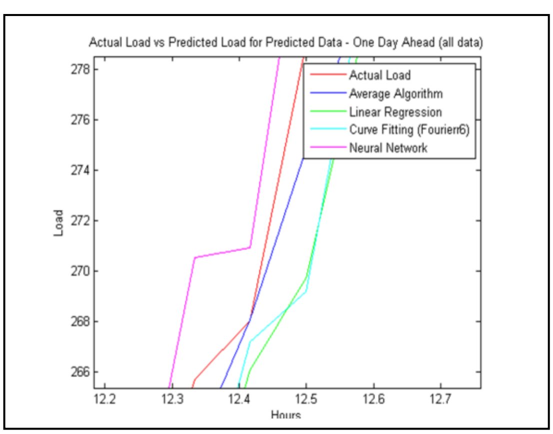

Figure 8. Comparing the other algorithms with the proposed algorithm

Table 3. Comparison of the MAPE values for different algorithms

Proposed algorithm Predicted loads	0.866
Proposed algorithm Historic loads	1.107
Neural Networks Predicted loads	1.0636
Neural Networks Historic loads	1.2778
Linear Regression Predicted loads	1.565
Linear Regression Historic loads	1.985
Curve Fitting Predicted loads	1.6142
Curve Fitting Historic loads	2.06

References

[1] Chen, C.S.; Leu, J.T., "Interruptible load control for Taiwan Power Company," Power Systems, IEEE Transactions on , vol.5, no.2, pp.460,465, May 1990

[2] Hao-Tian Zhang; Fang-Yuan Xu; Long Zhou, "Artificial neural network for load forecasting in smart grid," Machine Learning and Cybernetics (ICMLC), 2010 International Conference on , vol.6, no., pp.3200,3205, 11-14 July 2010

[3] Farhangi, H., "The path of the smart grid," Power and Energy Magazine, IEEE , vol.8, no.1, pp.18,28, January-February 2010

[4] Metke, A.R.; Ekl, R.L., "Security Technology for Smart Grid Networks," Smart Grid, IEEE Transactions on , vol.1, no.1, pp.99,107, June 2010

[5] Lotufo, A.D.P.; Minussi, C.R., "Electric power systems load forecasting: a survey," Electric Power Engineering, 1999. PowerTech Budapest 99. International Conference on , vol., no., pp.36,, Aug. 29 1999-Sept. 2 1999

[6] Bashir, Z.A., El-Hawary, M.E., 2007. Short-term load forecasting using artificial neural networks based on particle swarm optimization algorithm. In: Canadian Conference on Electrical and Computer Engineering, CCECE 2007, pp. 272–275

[7] Park, D.C.; El-Sharkawi, M.A.; Marks, R.J., II; Atlas, L.E.; Damborg, M.J., "Electric load forecasting using an artificial neural network," Power Systems, IEEE Transactions on , vol.6, no.2, pp.442,449, May 1991

[8] Saini, L.M.; Soni, M.K., "Artificial neural network based peak load forecasting using Levenberg-Marquardt and quasi-Newton methods," Generation, Transmission and Distribution, IEEE Proceedings- , vol.149, no.5, pp.578,584, Sep 2002

[9] Nose-Filho, K.; Lotufo, A.D.P.; Minussi, C.R., "Short-Term Multinodal Load Forecasting Using a Modified General Regression Neural Network," Power Delivery, IEEE Transactions on , vol.26, no.4, pp.2862,2869, Oct. 2011

[10] Ceperic, E.; Ceperic, V.; Baric, A., "A Strategy for Short-Term Load Forecasting by Support Vector Regression Machines," Power Systems, IEEE Transactions on , vol.28, no.4, pp.4356,4364, Nov. 2013

[11] Ahlert, K.-H.; Block, C., "Assessing the Impact of Price Forecast Errors on the Economics of Distributed Storage Systems," System Sciences (HICSS), 2010 43rd Hawaii International Conference on , vol., no., pp.1,10, 5-8 Jan. 2010

[12] Mirsu, R.; Tiponut, V., "Parallel model for Spiking Neural Networks using MATLAB," Electronics and Telecommunications (ISETC), 2010 9th International Symposium on , vol., no., pp.369,372, 11-12 Nov. 2010

[13] Margrave, F.; Babu, N.R.; Bradshaw, A.; Collins, I., "MATLAB-neural networks toolbox hardware post-processor," Applied Control Techniques Using MATLAB, IEE Colloquium on , vol., no., pp.6/1,6/3, 26 Jan 1995

[14] Moghram, I.S.; Rahman, S., "Analysis and evaluation of five short-term load forecasting techniques," Power Systems, IEEE Transactions on , vol.4, no.4, pp.1484,1491, Nov 1989

[15] Drezga, I.; Rahman, S., "Short-term load forecasting with local ANN predictors," Power Systems, IEEE Transactions on , vol.14, no.3, pp.844,850, Aug 1999

Mohammad M. R. Mozumdar is a tenure track faculty in the Electrical Engineering department of the California State University at Long Beach and an ex-postdoc from the University of California, Berkeley. He received a Ph.D. in electronics and communication engineering from Politecnico di Torino, Italy. His novel ideas of model based design for sensor networks made profound impact on engineering and industrial communities and have been published in book chapters, renowned journals, conference proceedings, major scientific magazines and also translated in several different languages. Dr. Mozumdar's research interests include methodologies and tools for embedded system design, especially in the domain of sensor networks; energy efficient building management and control system design; cloud computing; cyber physical system; methodology for the design of distributed embedded systems subject to high real time, safety and reliability constraints.

Karthik Sivakoti is a graduate student in the Electrical Engineering Department of the California State University at Long Beach and has finished his bachelor’s degree in electronics and telecom from West Bengal University of Technology, India. He has a professional experience in the field of Networking and Telecom for 3 years and is currently working for Extron Electronics, Anaheim, California, United States in the field of Network Research and Development. He has been working with Professor Mozumdar in the NES-Lab at CSULB for over a year working with Smart-Grid Networks.