Long Term Electrical Load Forecasting considering temperature effect using Multi-Layer Perceptron Neural Network and k-Nearest Neighbor algorithms

Mihir R.Patel[1]
CSPIT-CHARUSAT UNIVERSITY
Electrical Engineering,
Changa, Gujarat, India
e-mail: mihirrpatel.ee@charusat.ac.in

Ravi V.Patel[2]
CSPIT-CHARUSAT UNIVERSITY
Information Technology,
Changa, Gujarat, India
e-mail: ravipatel.it@charusat.ac.in

Dharmesh Dabhi[3]
CSPIT-CHARUSAT UNIVERSITY
Electrical Engineering,
Changa, Gujarat, India
e-mail: dharmeshdabhi.ee@charusat.ac.in

Jignesh S.Patel[4]
CSPIT-CHARUSAT UNIVERSITY
Electrical Engineering,
Changa, Gujarat, India
e-mail: jigneshpatel.ee@charusat.ac.in

Abstract— **Electrical power planning plays an essential role in energy sector development. Unpredictability of electrical power demand creates difficulties in grid management as load demand continues to rise. Thus, the task and accuracy of load forecasting become crucial to facilitate in the processes of optimal unit commitment (UC), economic dispatch, and power system networks stability. There are number of methods for electrical load forecasting, starting from the basic numerical methods to machine learning methods have been applied in the past for forecasting. In this paper, considering the correlation between time, temperature and electricity load demand for the electrical load forecasting. Here, by using machine learning methods for the electrical load forecasting the one is Multilayer Perceptron Neural Network and the other one k-Nearest Neighbor algorithms. The results of these methods are compared with mean absolute percentage error (MAPE).**

Keywords— Long Term Load Forecasting, Machine learning, Multi-layer Perceptron Neural Networks, Neurons, k-Nearest Neighbor, Neighbor Validation

I. INTRODUCTION

Load Forecasting is one of the most developing field and challenging field in past years. Load forecasting may be defined as the measure of precision of the change between the actual and forecast value of future electrical load demand. The main advantages of an accurate load forecast are: 1. The system operator gets better observability and controllability of the system, 2. It reduces uncertainties and costs, and 3. Possible penalties can be avoided for the owners and operators of the plant due to the inherent error between the actual production and forecasted energy of the plant [1]. The importance of the issue has motivated the development of many studies worldwide with the goal of obtaining accurate forecasts. The accurate or nearly-accurate forecasts of electrical power demand allow the grid operators to adjust the load in order to optimize the energy balance from other sources. It also plays a significant role in energy management system.

Neural Networks (NNs) have been applied effectively to solve some more difficult real-world problems such as pattern classification, electrical load forecasting, function approximation, and nonlinear mapping for which orthodox methodologies have proven ineffective and time consuming [2]. Availability of the fast computers, the responsiveness in NNs has been grew in the present-day digital world. NNs is trained using actual electrical load and temperature data from the past. NNs can learn and establish associated factors between input data sets and equivalent objectives. After completion training, the NNs can be used to forecast the output of new self-determining input data [3]. The k-NN algorithm is used for cataloging objects based on neighboring training examples in the feature space. k-NN is a type of instance-based learning, where the function is only approximated locally and all calculation is deferred until classification [4]. The KNN is the fundamental and simplest classification technique when there is little or no prior knowledge about the distribution of the data [4].

This paper use the data set given from the Global Energy Forecasting Competition 2018. The first homework part which is provided by Dr. Tao Hong and their team. In this data set has hourly temperature data of year 2005 to 2008 of total 28 weather station and load data of year 2005 to 2007. These datasets are going use to forecast load of year 2008 [5]. Here in this experiment we are using two algorithms for the forecasting the one is Multilayer Perceptron Neural Network and the other one is k-Nearest Neighbor algorithms. Data from year 2005 to 2007 is used to provide training and testing on the data set that predict load of year 2008 [5]. By experimenting it is proved that it is possible to accurately load forecast to save cost and efficient management of energy.

This article is structured in five different sections, first one is introduction and the second one is about long term load forecasting. The third section introduce the forecasting methods using NN and k-NN algorithms. The fourth section explain discussion of training results. The final fifth section gives conclusion, future scope and references.

II. LONG TERM LOAD FORECASTING

Long term load forecasting forecasts within the interval range of one year or more than that. The factors that affect long term load forecasting are previous years' load. The forecasting procedure depends on the manner in which historical time series data is analyzed and on the type of information available at the time the forecast is prepared. Various techniques have been applied to the problem of Long term load prediction. Long term load forecasting includes a broad range of parameters like: 1. Energy supply and price, 2. economic and demographic data and their forecast, 3. historical load and weather data, 4. population or number of customers, 5. Regional development, 6. time factors, 7. Facilities investment or sales, 8. Random disturbances [6]. Here in this paper, considering the historical load and temperature data for the forecast next year hourly load.

III. Forecasting using Neural Networks and k-Nearest Neighbor

A. *Forecasting using Neural Networks*

Application of NNs are: prediction, filtering, signal recognition, identifying processes, improving the performance of classical mathematical programing. A detailed illustration of NNs is presented in Fig. 1. The system of interconnected neurons, that have connection between them which can be either strong or weak, are called weights. The weights can be adjusted based on understanding the dataset and an algorithm such as backpropagation. After completion of training, it can be used as a task to get whatever output is needed [7].

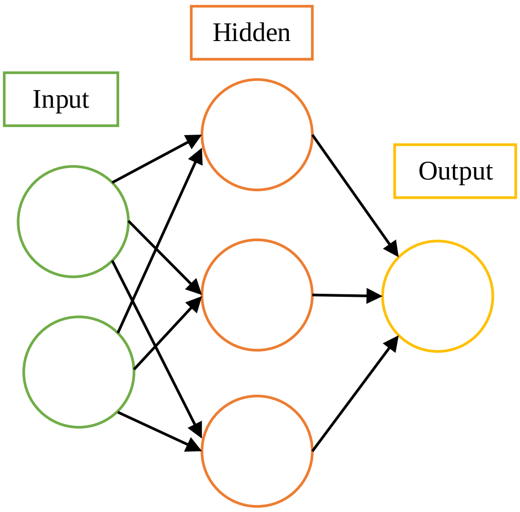

Fig.1. NNs diagram [7]

The neurons are usually prepared in layers where every neuron of a layer connects with every neuron form the layer above and the one below it. The first layer is called the input layer, which takes the input from the data set and transmits it through the network which consists of number of hidden layers, where the computation is performed and then the data set is pressed to the topmost layer, called the output layer [7].

Multi-layer Perceptron (MLP) is a supervised learning algorithm that learns a function $f(.):R^m \rightarrow R^o$ by training on a dataset, where is the number of dimensions for input and is the number of dimensions for output. Given a set of features $X= x_1, x_2, \ldots, x_m$ and a target y, it can learn a non-linear function approximate for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. Figure 2 shows a one hidden layer MLP with scalar output [8]. The leftmost layer, known as the input layer, consists of a set of neurons $\{x_i|x_1,x_2,\ldots,x_m\}$representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation $w_1m_1+w_2x_2+\ldots+w_mx_m$, followed by a non-linear activation function $g(.):R \rightarrow R$- like the hyperbolic tan function. The output layer receives the values from the last hidden layer and transforms them into output values. The advantages of Multi-Layer Perceptron are: Capability to learn non-linear models and Capability to learn models in real-time [8]. Here the tuning parameter for this method are given in table 1.

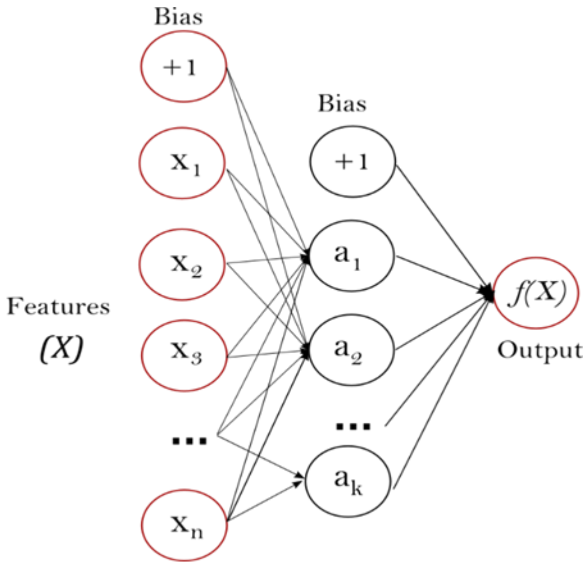

Fig. 2. One hidden layer MLP [8].

B. *k-Nearest Neighbor*

The principle behind nearest neighbor techniques is to discover a predefined number of training samples, neighboring in distance to the new point, and expect the label from these. The k-NN is the essential and naivest classification method when there is little or no previous information about the supply of the data. The NN rule is the simplest form of k-NN when k = 1, this technique classified each sample equally to its neighboring samples. Hence, if the classification of a sample is unidentified, then it might be expected by considering the classification of its nearest neighbor samples. Specified an unknown sample to the training set, altogether the distances between the unidentified sample and all the samples in the training set can be calculated then the distance with the smallest value corresponds to the sample in the training set closest to the unknown sample. Therefore, the unknown sample may be classified based on the classification of this nearest neighbor [9,10].

The selection of suitable neighbor k affects the classification performance of k-NN, for the small training data set sample size can affect the optimal neighborhood k degradation of the classification performance of k-NN. Commonly, the classification results are highly responsive to two facts, 1. k is too small- the data sparseness and the noisy, mislabeled points, and 2. k is too large- many outliers within the neighborhood from other classes. So, the selection of value k's is very sensitive to classification performance. Also, the simplest majority elective of merging the class labels for k-NN can be a problematic if the nearest neighbors differ extensively over their distances and the closer ones more reliably indicate the class of the query object. With the objective of speaking the sensitivity issue of different choices of the neighborhood size k, some weighted voting approaches have been established for k-NN. Commonly, the larger values of k are more excepted to the noise presence, and make boundaries smoother between classes [9,10]. Here the tuning parameter for this method are given in table 1.

IV. Training, Testing and Results

By considering hourly 28 weather station temperature data set for four years from 2005 to 2008. Also, Three years of the load data set from years 2005 to 2007 [5]. For first three years i.e., from 2005 to 2007 by providing the training to the algorithms with both load and temperature data set, for year 2008 and by applying testing to the same algorithms. After the testing it is compared with the actual data set with forecasted one, and calculate the Mean Absolute Percentage Error(MAPE). MAPE is defined as follows:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{A_i}\right|$$

Where A_i is the actual load on day i and F_i the forecast value of the load on that day, and N represents the total number of data (hours).

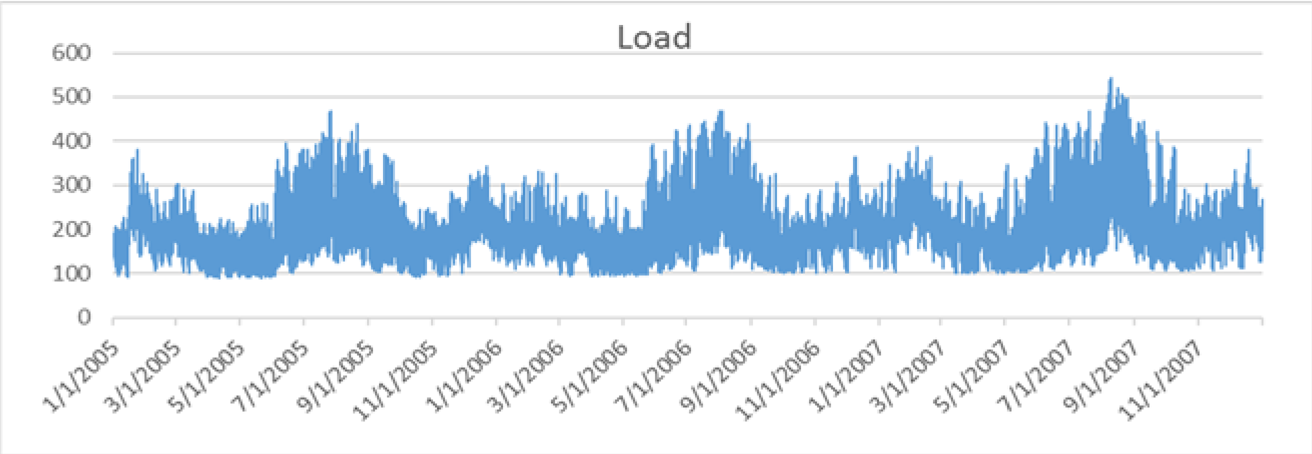

Fig. 3. Actual hourly load (kW)

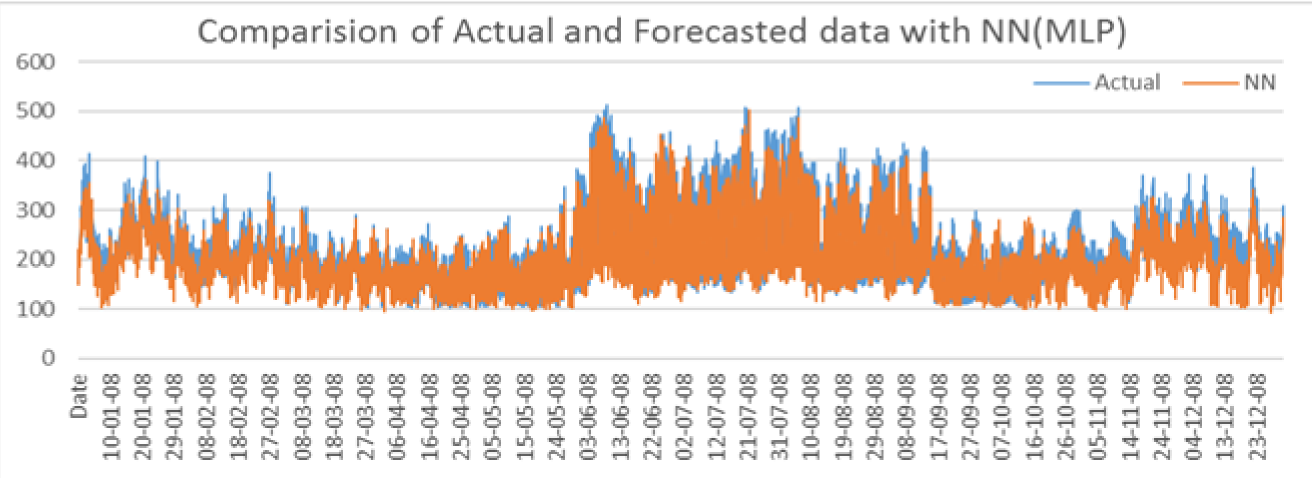

Fig.4. Actual and Forecasted load using NN(MLP)

Algorithm Summary with Model parameter:

Table-1: Model Parameter

Model Parameter for k-NN	
Number of neighbors	36
Metric	Euclidean
Weight	Uniform
Model Parameter for NN (Multi-Layer Perceptron)	
Neurons per hidden layers	100
Activation	ReLu
Solver	Adam
Max iterations	500

From table-1 for k-NN the parameter indicates as: 1. number of nearest neighbors shows the value of k, 2. Metric can be Euclidean which indicates "straight line", distance between two points, and 3. The Weights can be used are Uniform that shows all points in each neighborhood are weighted equally. Similarly, for the Multi-Layer Perception the parameter indicates as: 1. Neurons per hidden layer defined as the ith element represents the number of neurons in the ith hidden layer, 2. Activation function for the hidden layer is ReLu which indicates the rectified linear unit function, 3. Solver for weight optimization is Adam which indicates stochastic gradient-based optimizer, and 4. Total numbers of iteration.

Figure 3 shows graphical hourly actual load which is used for the testing the model. Considering all four years of temperature data set for forecasting the load of year 2008. Figure 4 and 5, shows the compression between actual and forecasted load using NN(MLP) and k-NN respectively for year 2008. Figure 6 shows the total four years of load profile where, four years of actual load and one year of forecasted data set. Table 2 shows the comparative MAPE of forecasted data using NN(MLP) and k-NN. It can be reduce the MAPE by tuning the algorithm with more accuracy. Moreover, by considering weather station selection method the result may improve [11]. Also, it can be considered holiday, seasonality, and recency effect for more accuracy in forecasted results [12].

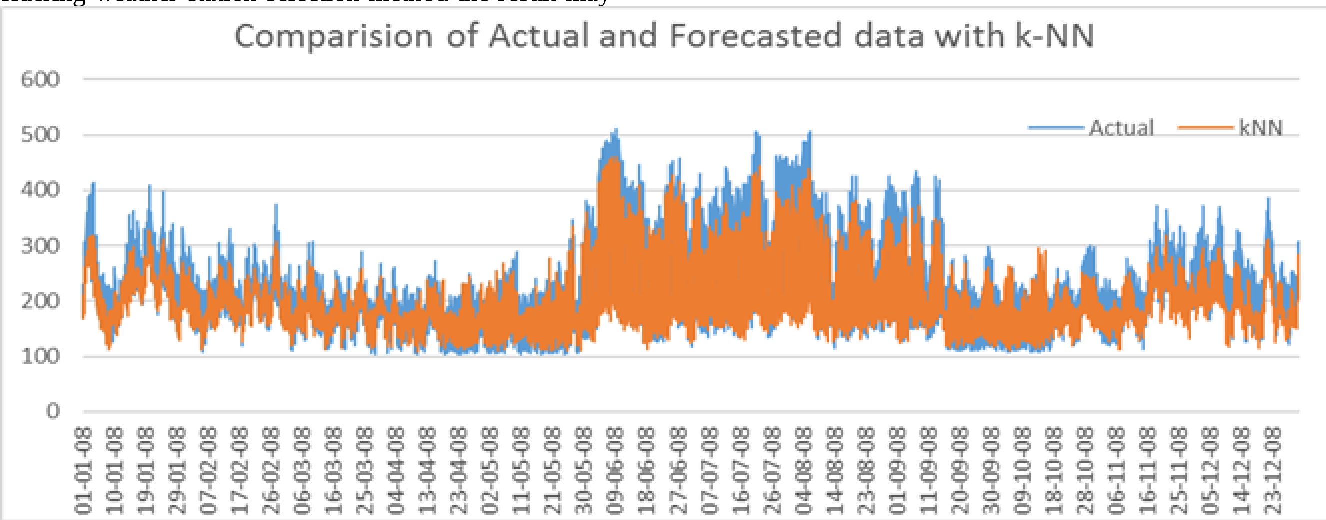

Fig. 5. Actual and Forecasted load using k-NN

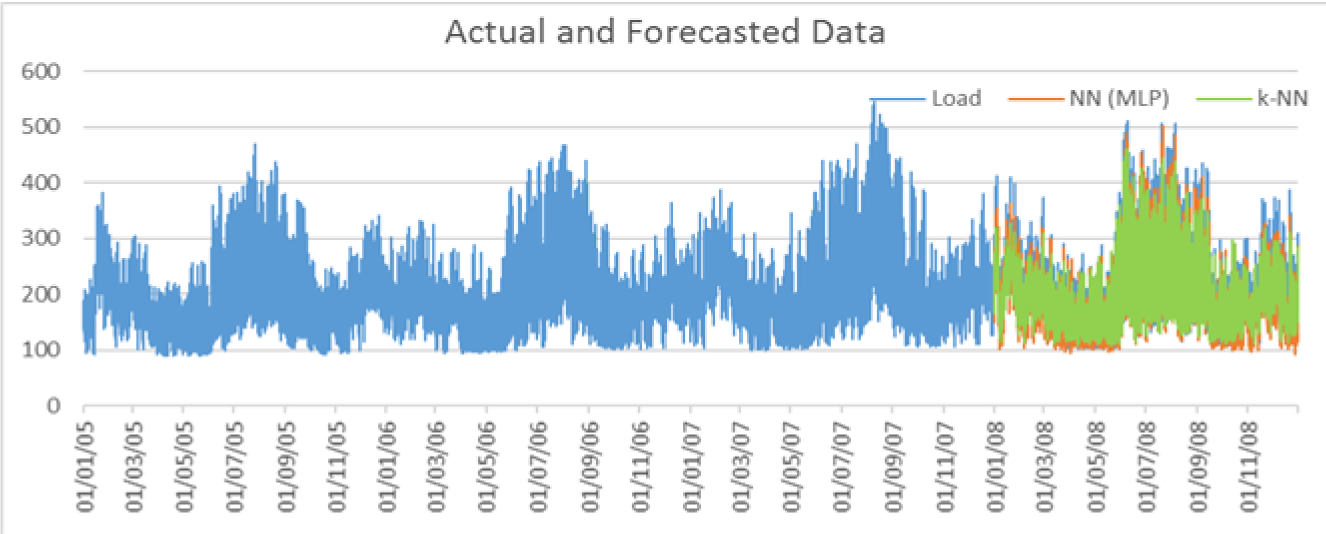

Fig. 6. Load Profile along with Forecasted data

Table-2: comparative MAPE of forecasted data

Algorithm	MAPE
NN(MLP)	7.8
k-NN	10.8

IV. CONCLUSION:

This work presented the results of long term load forecasting, with considering the temperature effect using the NN(MLP) and k-NN method. By comparing both the methods with the actual and forecasted data set, to calculate the MAPE. For particular model parameter, from table 2, it is observe that MAPE of NN(MLP) comparatively less than the k-NN method. It can be reduce the MAPE by tuning the algorithm with more accuracy. Also by considering holiday, seasonality, and recency effect it is possible to have more accuracy in forecasted results.

REFERENCES

[1]. Soliman SA. “Electrical load forecasting; modeling and model construction”. Butterworth-Heinemann. 2010.

[2]. Haykin S. “Neural Networks a Comprehensive Foundation”. 2nd ed, NY, USA: Macmillan College Publishing Company; 1999.

[3]. EA. Feilat, D. Talal Al-Sha’abi and MA. Momani “Long-term load forecasting using neural network approach for Jordan's power system”. Eng Press. 2017; 1(2): 43-50.

[4]. Devroye, L. (1981) "On the equality of Cover and Hart in nearest neighbor discrimination", IEEE Trans. Pattern Anal. Mach. lntell. 3: 75-78.

[5]. T. Hong, P. Wang, and H. L. Willis, “A naÃ¯ve multiple linear regression benchmark for short term load forecasting,” in 2011 IEEE Power and Energy Society General Meeting, pp. 1–6, 2011.

[6]. H. Daneshi, M. Shahidehpour, A.L. Choobbari, “Long-Term Load Forecasting in Electricity Market,” IEEE Int. Conf. on Electro/Inf. Tech., 2008, pp. 395-400, May 18-20, 2008.

[7]. Catalin Mihai, Elena Helerea, “Load forecasting using artificial neural networks for industrial consumer,” IEEE Int. Conf. on MPS, 2017.

[8]. https://scikit-learn.org

[9]. Devroye, L. & Wagner, T.J. (1982) "Nearest neighbor methods in discrimination, In Classification, Pattern Recognition and Reduction of Dimensionality", Handbook of Statistics, 2: 193–197. North-Holland, Amsterdam.

[10]. S B Imandoust et al., " Application of K-Nearest Neighbor (k-NN) Approach for Predicting Economic Events: Theoretical

Background ", Int. Journal of Engineering Research and Applications, Vol. 3, Issue 5, Sep-Oct 2013, pp.605-610.

[11]. Tao Hong, Pu Wang, Laura White, "Weather station selection for electric load forecasting", International Journal of Forecasting 31 (2015), pp.286-295.

[12]. Pu Wang, Bidong Liu, Tao Hong, "Electric load forecasting with recency effect: A big data approach", International Journal of orecasting 32 (2016), pp.585–597.