



A Novel Approach for Short Term Load Forecasting Using Load Profile Index Vectors by Development of BPNN and GRNN Algorithms

K. N. Bhanuprakash¹, Supreetha Patel T P², Nikshepa T³, Santosh C⁴

Principal, Dept. of EEE, Kalpataru Institute of Technology, Tiptur, Karnataka, India¹

Assistant Professor, Dept. of CSE, Kalpataru Institute of Technology, Tiptur, Karnataka, India²

Assistant Professor, Dept. of CSE, Dayanand Sagar College of Engineering, Bangalore, Karnataka, India³

Student, Dept. of EEE, KSSEM, Bangalore, Karnataka, India⁴

ABSTRACT: The historical hourly load data for each day is represented as index vectors which in turn are used for training and running the ANN to generate the forecast result. A new technique of eliminating highly discrepant historical data during the training of ANN is presented. BPNN as well as GRNN neural networks are used to get the forecast result.

KEYWORDS: Load Profile, Profile Index Vector, ANN, BPNN, GRNN, STLF

I. INTRODUCTION

Modern Power System has grown in size and complexity. The constraints imposed on the utilities operating environments in recent years for security, economic, environmental and regulatory reasons have made the task of the system operator increasingly more difficult. Moreover, power systems are operating closer to their design capabilities and security margins. This coupled with continuously escalating margins and depleting natural resources has made it mandatory for utilities to operate the system as optimally as possible. In this context more complex and fast decision making process are required. Operating personnel need more timely and meaningful information on which to base their numerous important decisions. These problems associated with the power system necessitate the need for modern energy control centre.

Load forecasting plays an important role in power system operation and control. Basically, the active power demands at the various load buses need to be estimated ahead of time in order to plan the generation and distribution schedules, contingency analysis and for checking the system security. These operational applications require load forecasts to be available for lead times (forecasting intervals) of a few seconds to a several minutes duration. Such forecasts are usually referred to as 'very short term' load forecasts. Load forecasts with larger values of the lead time are also needed for other applications. For example, for the allocation of the spinning reserve, it would be necessary to predict the load demands at least half an hour to a few hours ahead. Load forecasts with such lead times are usually referred to as 'short term' forecasts. On the other hand, preparing to meet the load requirements at the height of the winter or the summer season may require a load forecast to be made a few days to a few weeks in advance. Forecasts with such lead times constitute 'medium term' load forecasts. Finally, to plan the growth of the generation capacity, it would be necessary to make a 'long term' load prediction which may involve a lead time of a few months to a few years.

A majority of recently reported approaches are based on neural network and many researchers have presented good results. The attraction of the method lies in the assumption that neural networks are able to learn the properties of the load which would otherwise require careful analysis to discover. However the development of the methods is not exhausted and the lack of comparative results on different model variables is a problem. Therefore to make use of the



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

techniques in a real application, a comparative analysis of the properties of different model types used in the neural network seems necessary.

In this paper short term load forecasting using ANN techniques are developed based on the selection of input variables, and determination of optimum number of neurons. Further historical hourly data is represented as index vectors which in turn are used for training and running the ANN to generate the forecast result, and a method of STLF based on the similarity of the load shapes is presented. It is observed all developed algorithms is robust, faster training, prevents overtraining and is suitable for online implementation in an energy control centre.

One of the primary objectives of cognitive radio (CR) ad-hoc networks is to facilitate an efficient utilization of The Load Profile of a Power System gives the variation of the load with respect to time. Basically, the load profile represents the shape of the load curve [5]. The hourly load profile for each day is an important data set in power system analysis, classification, protection, and forecasting, scheduling and overall management [1]. Profile indexing of a numerical Data Vector, is a method of generating a corresponding integer vector having the same sort order as that of the data vector. The resulting integer vector is called the Profile Index Vector (PIV) of the Data Vector. The PIV represents the profile of the load curve when the corresponding numerical data vector represents the hourly load [4]. The size of the Data Vector and its PIV are same. No earlier work is found on PIV's.

III. LITERATURE SURVEY

H.Yu, and Ki.Lee et. al. [1] has proposed in his study on load profile data, which can be collected through the automatic meter reading system, are analyzed for getting demand patterns of customer. The load profile data include electricity demand in 15 minutes interval. An algorithm for clustering similar pattern is developed using load profile data. As results of classification, representative curves for the same groups are generated. The demand characteristics of the groups are discussed. Also, the compositions of demand contract and industrial classification in each group is presented.

Wikipedia –the encyclopedia [2] –Linear regression refers to any approach to modelling the relationship between one or more variables denoted by y and one or more variables denoted by x , such that the model depends linearly on the unknown parameters to be estimated from the data.

Simon Haykin [3] has proposed the application of multilayer perception successfully to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as the error propagation algorithm.

A wide variety of procedures for short term forecasts have been reported in the literature. The earliest technique for load forecasting was published in 1918 and this was reported in the paper by M.S Sachdev R. Billinton [1] which also includes representative cross section of publications.

The earlier techniques used in the short term load forecasting (STLF) were mainly on state estimation methods, spectral expansion methods and regression models and the recent techniques are based on ANN method, Fuzzy Inference method, support vector machines, etc.. The report prepared by the load forecasting working group summarizes the papers that were published before 1981. Some recent reviews have been reported in CIGRE study committee [5].

Ibrahim Moghram and Saiffur Tahman [6] analyzed five different short term load forecasting techniques namely Multiple linear regression, Stochastic Time series, General Exponential Smoothing, State space method and Knowledge based approach and they have compared the results.

D. C. Park et. al. for the first time proposed a neural network based STLF model with three layers and a delta rule for training the network. The network provided accurate forecast for hourly, peak and daily total loads. Tests were conducted over one winter season and average errors were 1.4%, 2.04% and 1.68% respectively for the forecasts. K. Y. Lee and J. H. Park [7] suggested a non linear model with back propagation algorithm for estimating the parameters of the model. They compared two methods, static approach which forecasts 24 hours load simultaneously with dynamic method approach that forecast 24 hours load sequentially and concluded that dynamic approach performs better than the static method.

Kun-Long Ho et. al. [8] suggested a multi layered ANN with an adaptive learning algorithms for STLF. The effects of learning rates, momentum the number of presentations in an iterations, etc., were extensively studied and found that once trained by the algorithm, the ANN can yield the desired hourly forecast very efficiently and accurately.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Kwang-Ho Kim et.al.[9] proposed a new short term load forecasting methods for special days in anomalous load conditions. These days include public holidays, and consecutive holidays, and days preceding and following holidays. The proposed method uses a hybrid approach of ANN based technique and fuzzy inference method to forecast the hourly loads of special days.

III. PROBLEM FORMULATIONS

A. Definition of PIV

Let D be a Data Vector of size N . Then its PIV is designated as $\text{piv}(D)$. $\text{piv}(D)$ is also of size N . It is made up of integers 1 to N without any repetitions. The sort index of $\text{piv}(D)$ is made equal to that of D .

Examples:-

1. Let $D = [30\ 40\ 50\ 60\ 70]$ then, its PIV would be, $\text{piv}(D)=[1\ 2\ 3\ 4\ 5]$. The sort index vector of D designated by $\text{siv}(D)$ and that of $\text{piv}(D)$ designated by $\text{siv}(\text{piv}(D))$ both are same as given by, $\text{siv}(D)=\text{siv}(\text{piv}(D))=[1\ 2\ 3\ 4\ 5]$.
2. Let $D=[60\ 30\ 40\ 70\ 50]$ then, $\text{piv}(D)=[4\ 1\ 2\ 5\ 3]$. The sort index vector is, $\text{siv}(D)=\text{siv}(\text{piv}(D))= [2\ 3\ 5\ 1\ 4]$. This means, the position of the element in both D and $\text{piv}(D)$ is 2, the position of the next smallest element is 3 and so on.

B. Determination of $\text{piv}(D)$

1. For a given D , find its sort index vector $\text{siv}(D)$, say using the matlab built in sort function as,
 $[E\ Q] = \text{sort}(D)$ (1)
Then, $\text{siv}(D)$ is given by the vector Q .
2. Find the $\text{siv}(Q)$ by sorting Q itself as,
 $[F\ P] = \text{sort}(Q)$ (2)
then $\text{siv}(Q)$ is given by P . Then P gives the $\text{piv}(D)$, thus $\text{piv}(D)$ is obtained as, $P = \text{piv}(D) = \text{siv}(\text{siv}(D))$, then $\text{siv}(D)$ and $\text{siv}(P)$ would be same.

C. Properties of $\text{piv}(D)$

Let the size of D be N . Let vector $P=\text{piv}(D)$, then

1. Size of P is also N .
2. P is a permutation vector containing integers from 1 to N , without any repetition.
3. $\text{siv}(P)=\text{siv}(D)$.
4. The Euclidean distance between D and P is minimum compared to any other permutation vector of size N . Thus P is a closest permutation vector representation of D .
5. The covariance between D and P is maximum compared to that between D and any other permutation vector of size N .
6. The dot product between D and P is maximum compared to that between D and any other permutation vector of size N .
7. The norm of P for different Data vectors of size N , is constant given by $\sqrt{N(N+1)(2N+1)/6}$. Thus when several data vectors, all of size N , are stored in the form of their PIV's, the norm of these PIV's are all same. Therefore, when used as input training vectors for ANN's, the scaling and normalization are already completed.

D. PIV as the Encoded version of the Data Vector

Basically, the PIV of a Data Vector is an encoded version of the Data Vector. Thus PIV representation is a form of encoding the given Data Vector. Eqs.(1) and (2) are used for encoding. In the case of Load Flow data vector, the range of the Load is large and wide, while its PIV occupies a small integer range 1 to N .

E. Decoding (Recovery) of the Data Vector from its PIV

Let D be a Data Vector of size N . Let vector $P=\text{piv}(D)$. Then the values of D can be recovered from P by simple Linear Regression with minimum MSE as follows [2]. The recovered value \hat{D} is given by,

$$\hat{D} = \alpha + \beta * P \quad (3)$$



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Where α and β are the regression parameters given by,

$$\beta = \frac{\text{cov}(P, D)}{\text{var}(P)} \quad (4)$$

$$\alpha = \text{mean}(D) - \beta * \text{mean}(P) \quad (5)$$

From the known values of α , β and P , the approximate \hat{D} can be recovered. From Eq. (3), the j^{th} elements of \hat{D} and P are related as,

$$\hat{D}(j) = \alpha + \beta * P(j) \quad \text{For } j=1,2,\dots,N \quad (6)$$

From Eq. (6),

$$P(j) = \frac{(\hat{D}(j) - \alpha)}{\beta} \quad (7)$$

IV. NEXT HOUR LOAD FORECASTING USING BPNN WITH PIV'S AS INPUTS

The ANN technique indirectly stores the historical data in the form of PIV's as their weights during training and then the present day forenoon data matching pattern is recalled to find the closest similar day's next hour load as the output of the ANN [9]. This output would be the forecast value at the next hour on the present day. We have chosen two types of ANN in this paper [6]. The more popular BPNN (Back Propagation Neural Network) is a scalar real number.

A. BPNN (Back Propagation Neural Network)

We have chosen BPNN to get the forecast results as it is found to be robust and efficient for short term forecasting [3]. The PIV's of Load Flow data are the inputs to the BPNN. The elements of PIV's are integers in the range 1 to N and their norms are all same. Hence they are well suited for training and running of BPNN.

B. Description of the ANN used

A three layer Feed Forward ANN is used [7]. The input layer has 5 neurons because the size of the input vector is 5x1. The 5 inputs are the 5 preceding hourly PIV data on the forenoon of the day from hours 8 to 12. The number of neurons in the hidden layer is selected to be 9. These are found to give better results. The output layer has 1 neuron corresponding to the load at the 13th hour. This load is converted into an equivalent profile element using Eq.(7) which is based on linear regression[11]. The reason for conversion is, to scale the BPNN output so that it belongs almost to the same range as its input elements. Weights are updated by the back propagation Algorithm which uses the gradient descent method using the function 'traingd'.

C. Preparation of the input and target output during training

For each historical day, the load data from hours 8 to 12 is converted into its equivalent PIV using Eqs.(1) and (2). The size of each input PIV is 5x1. Five immediate previous hours 8 to 12 are chosen, because these values are found to have maximum influence on the 13th hour load. The corresponding target output value is obtained for that historical day by converting the load data at 13th hour into its Equivalent PIV element using Eq.(7). For the historical day i , let this target output value during training be $T(i)$, given by, (Eq.(7) is rewritten)

$$T(i) = \frac{(D(i,13) - \alpha(i))}{\beta(i)} \quad (8)$$

Where $\alpha(i)$ and $\beta(i)$ are the regression coefficients already calculated using the previous 5 hourly data $D(8:12)$ and the corresponding PIV vector P for that day.

D. Exclusion of discrepant Historical days during training

For a given historical day, the 5 hourly data from hours 8 to 12, converted into its equivalent PIV is the input. The output is also an element belonging to PIV corresponding to the 6th element of PIV (corresponding to the 13th hour). Under normal conditions the range of PIV elements for all the 6 hours would be from 1 to 6. If $T(i) < 1$, it means the load at 13th hour is less than the lowest among the first five values. Hence it is an abnormal situation. Therefore if $T(i)$ calculated from Eq.(8) lies within 1 to 6, that day is accepted. Otherwise, that day is rejected. Thus for a given



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

historical day i , find $T(i)$ using Eq.(8). If $1 \leq T(i) \leq 6$, accept that day for training. Else that day is excluded during training. This process eliminates discrepant/outlying days while training. With this elimination, the training speed is increased and the predicted output is more accurate.

E. Training to get the actual output

Training is by back propagation using the matlab Neural Network Tool box. Different set of training parameters are used to find the best combination to get good results. Once the training is over, the matlab ANN simulation function is used to get the output for the applied present input. The present input is the PIV corresponding to the 5 hourly load data of the present test (predict) day, from hours 8:12.

The corresponding regression parameters α and β , for the 5×1 data, is calculated using Eqs.(4) and (5) for future use while forecasting the 13th hour load. The output will be the closest PIV element from the historical data corresponding to the present day's previous five hour data. This PIV element value is converted into its Equivalent Load Data using Eq. (7). These operations can be listed as follows.

1. Take the present test day data from hours 8:12 Call this as vector D. The size of D is 5×1 .
2. Get P, the PIV for this D using Eqs (1) and (2).
3. Calculate α and β for this P and D using Eqs.(4) and (5).
4. Apply the 5×1 P vector as the input to the trained ANN.
5. Get the output and call it as P(6).
6. From P(6), get D(6), the predicted load using Eq.(6).

F. Extension of the method to cover the entire day

This forecast method can be applied to the succeeding hours by taking the previous hourly data as the inputs.

For the hour 14, use the data from hours 9:13,

For the hour 15, use the data from hours 10:14, and so on.....

V. NEXT HOUR LOAD FORECASTING USING GRNN WITH PIV'S AS INPUTS AND OUTPUTS

Generalized Regression Neural Network (GRNN) has a radial basis layer and a special linear layer [3]. It is very well suited for vector inputs and target vector outputs which are nearer and similar to each other. In GRNN, the number of neurons required is large compared to BPNN, but the training period is very short.

A. Design details

The input layer as well as the output layer has as many neurons as the number of training/target vectors [8]. The input training vectors are the PIV's of the previous K hours, where K can take values generally from 6 to 9. Thus if the prediction hour is h, then the input vector P is the PIV's corresponds to the hourly data from hour (h-K) to (h-1) and the target output vector T is the PIV's of the previous K hourly data counting from h, that is from (h-k+1) to h. Therefore if K=6 and h=13, then the input vectors are from the hours (7:12) and the target output vectors are from the hours (8:13). The GRNN results are obtained for different values of K and h. The number of training vectors used are Q=100,150 and 200. Other suitable values can also be used.

B. Parameters and variables used

Let d be the prediction day and h be the forenoon prediction hour. h can take values in the range 12...24. The loads on that day, at hours previous to h should be known. This could be by measurement data or by any reliable forecast method. K is the size of the input vector. The range for K can be generally 5:8. Matrix L is used to represent the hourly load for the historical days as follows.

$L(r,c)$, the element at row r and column c, represents the load at hour r and day c. The rows of L matrix represent the hours in the range 1:24 and the columns of L represent the corresponding days. For the prediction at hour h, we use previous K hour data. Therefore a sub matrix of L is used in training and prediction. The sub matrix L_{sub} required for training is formed as,

$$L_{sub}=L(h-K:h-1,d-Q:d-1) \quad (9)$$



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Where Q is the number of historical days used for training. Here the range $d-Q:d-1$ represents the previous Q days, excluding abnormal days if any. The range $h-K:h-1$ represents the previous K hours. Each column of L_{sub} represents the hourly load sequence on that day. These column vectors are converted into equivalent PIV's using Equations similar to Eqs. (1) and (2). This results in the training input matrix P. The columns of P are the PIV's of the corresponding columns of L_{sub} . Thus,

$$P(:,j)=PIV(L_{sub}(:,j)), \text{for } j=1,2,..Q \quad (10)$$

$$\text{Eq.(10) can be expressed as, } P=PIV(L_{sub}) \quad (11)$$

In Eq.(11), the PIV operation is applied column wise. Thus the input training matrix is obtained. The size of P is $K \times Q$. The target output is similar, it includes the hour h data as,

$$T=PIV(L(h-K+1:h,d-Q:d-1)) \quad (12)$$

Thus the column vectors of T have K rows starting from hours $h-(K-1)$ to the forecast hour h. Therefore during training, the output vector always contains the hour h data as the last element. The size of T is also $K \times Q$.

C. Creation and training of GRNN for forecasting

The GRNN is created using P and T using the matlab function *newgrnn(P,T)*. Once created, it is trained automatically without calling any explicit training function. After creation and training, the GRNN is ready to generate the forecast value. The prediction day data from hours $(h-K:h-1)$ is converted into its equivalent PIV. That is, the present input is given by,

$$P_{in}=PIV(L(h-K:h-1,d)) \quad (13)$$

The size of P_{in} is $K \times 1$. The output of GRNN is designated as T_{out} which is the PIV version of the Load vector for that day for hours $h-(K-1) : h$. The elements of T_{out} are not exact PIV elements, because the exact PIV elements are integers while T_{out} elements are general non integers but very close to the exact PIV's. T_{out} is the GRNN encoded version that represents the corresponding Load vector. Therefore, T_{out} can be decoded to get the Load Vector $L(h-K+1:h,d)$. T_{out} and $L(h-K+1:h,d)$ are highly correlated and we can use the Linear Regression relation as given by Eq.(3) to determine the Load Vector $L(h-K+1:h,d)$ from T_{out} . But the hourly load data from $h-K+1:h-1$ (previous K-1 hours) is already known. This is encoded as the first K-1 elements of T_{out} . We have to calculate the hour h data which is encoded as the last (K^{th}) element of T_{out} .

D. Calculation of the forecast load at hour h

The hourly load data from $h-K+1:h-1$ is already known. The corresponding encoded values are available from the first K-1 elements of T_{out} . Therefore, they are related by the decoding relation as given by Eq.(3). That is, $L(h-K+1:h-1,d)$ and $T_{out}(1:K-1)$ are related as,

$$L(h-K+1:h-1,d) = \alpha + \beta * T_{out}(1:K-1) \quad (14)$$

Let L_{pr} represent $L(h-K+1:h-1,d)$ and T_{pr} represent $T_{out}(1:K-1)$. That is,

$$L_{pr}=L(h-K+1:h-1,d) \quad (15)$$

$$T_{pr}=T_{out}(1:K-1) \quad (16)$$

From, Eqs.(15) and (16), Eq.(14) can be written as,

$$L_{pr} = \alpha + \beta * T_{pr} \quad (17)$$

Subscript pr is used to indicate the prediction day data of previous K-1 hours. α and β are obtained similar to Eqs.(4) and (5) as,

$$\beta = \frac{\text{cov}(T_{pr}, L_{pr})}{\text{var}(T_{pr})} \quad (18)$$

$$\alpha = \text{mean}(L_{pr}) - \beta * \text{mean}(T_{pr}) \quad (19)$$

α and β are the regression coefficients for the data range from $h-(K-1)$ to $h-1$ having K-1 sequential data elements which are already known. Now, we assert that the same coefficients α and β can be used to predict the next sequential value as,



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

$$L(h : d) = \alpha + \beta * T_{out} (K) \tag{20}$$

Where $T_{out} (K)$ is the last (K^{th}) element of T_{out} which is given out by the GRNN. In this method the assertion is that, if a linear regression holds good for previous ($K-1$) pairs, then it holds good for the next K th pair also. This is based on the fact that when K is reasonably large, and the two sequences are closely correlated, the regression coefficients α and β would be almost same with ($K-1$) pairs or K pairs.

E. Exclusion of discrepant Historical days during training

For a given historical day, the hour h forecast load is calculated using Eq.(20), where d is the day under consideration [10]. The Actual load at this hour on that day is also known. Then the absolute percentage forecast error is calculated as $APE=100*Abs((Actual Value - Forecast value))/Actual value$.

If $APE>5\%$ on any day, that day is excluded from training.

➤ Design and Training

1. Chose a suitable value for K and Q , the size of the input vector and the number of training vectors respectively.
2. From the historical data, get the input training matrix whose columns are the input training vectors.

➤ Forecasting

Apply the test day input to GRNN and get the corresponding output. Calculate the prediction value.

VI. CASE STUDIES FOR BPNN-KARNATAKA SYSTEM INDIA

A new method of STLF based on the similarity of the load shape is presented [14]. The historical hourly load data for each day is represented as index vectors which in turn used for training and running the ANN to generate the forecast result.

Historical Load Data from the Karnataka Southern Grid is used here. Out of 330 days, last 30 days are used as test (predict) days and the earlier 300 days are used as historical days [13]. Variations in environmental parameters and contextual parameters are assumed to be reflected in the load pattern itself. For the purpose of illustration, sample historical data for 5 days from day 36 to 40 are shown in Table-1. Table-1 also shows the corresponding PIV's.

In this case, day 36 and 38 are discarded because their PIV elements, in the last row, which is the target output, are <1 . These days are marked with # sign to indicate that they are to be discarded during training. The input and output of the ANN for test days are shown in Table-2. The inputs are the PIVs of size 5×1 , corresponding to the hours 8 to 12. The forecast load is shown in row forecast. Percentage error is shown in the last row of Table-2. %error is calculated as, $\%error=100*abs(actual value - forecast value) / (actual value)$

It can be seen from the result, that except for test day 4th, the percentage error is relatively small.

	Historical Day 36 #		Historical Day 37		Historical Day 38 #		Historical Day 39		Historical Day 40	
Hours	Load	PIV Input	Load	PIV Input	Load	PIV Input	Load	PIV Input	Load	PIV Input
8	2667	1	2601	1	2847	1	2943	1	2581	1
9	2739	3	2789	3	2896	3	3112	4	2645	2
10	2804	4	2790	4	2963	5	3083	3	2788	3
11	2912	5	2889	5	2942	4	3197	5	2965	4
12	2709	2	2769	2	2853	2	3081	2	3067	5
α	2590.7		2588.5		2803.9		2921.5		2421.6	
β	58.5		59.7		32.1		53.9		129.2	
13	2622	0.5 # Target Output	2852	4.4 Target Output	2758	-1.4 # Target Output	3098	3.3 Target Output	2901	3.7 Target Output

Table-1 Training Data for BPNN

Historical Load for 5 days from day 36 to 40. The target output is shown next to the load in the last row in bold

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Hours	Test Day 1		Test Day 2		Test Day 3		Test Day 4		Test Day 5	
	Load	PIV	Load	PIV	Load	PIV	Load	PIV	Load	PIV
8	4409	4	4252	3	4331	2	4409	3	4202	4
9	4062	1	4087	1	4250	1	4160	2	3787	1
10	4129	2	4191	2	4347	3	4101	1	3951	2
11	4564	5	4667	5	4760	5	4518	5	4424	5
12	4379	3	4487	4	4357	4	4420	4	4184	3
α	3923.4		3900.0		4095.2		3993.4		3652.1	
β	128.4		145.6		104.6		109.4		152.5	
13 (forecast)	4341	3.25	4419	3.56	4482	3.7	4377	3.5	4148	3.25
13 (Actual)	4265		4379		4451		4162		4104	
Abs(error)	76		40		31		215		44	
%error	1.7835		0.9061		0.6921		5.1574		1.0761	

Table-2 Results obtained on running the BPNN.

Test day Load for hours 8 to 12 and their PIV's for 5 days. ANN output and forecast load for hour 13 is also shown.

The plots of the actual and forecasted load and their %Abs Error for daily and hourly load are shown in below figures – 1,2,3,4.

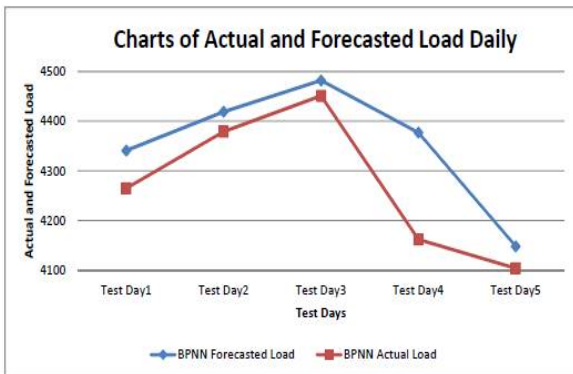


Fig. 1: Actual and Forecasted Load (Daily) for BPNN

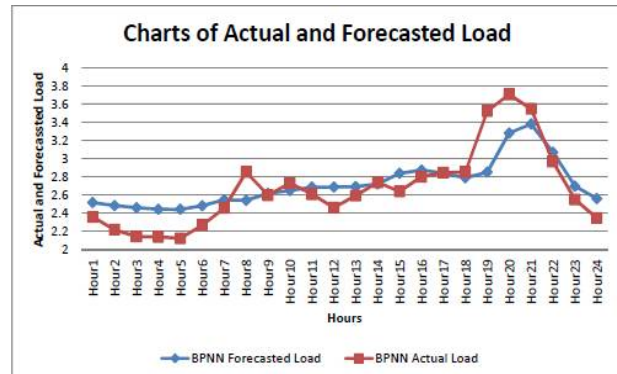


Fig. 2: Actual and Forecasted Load (Hourly) for BPNN

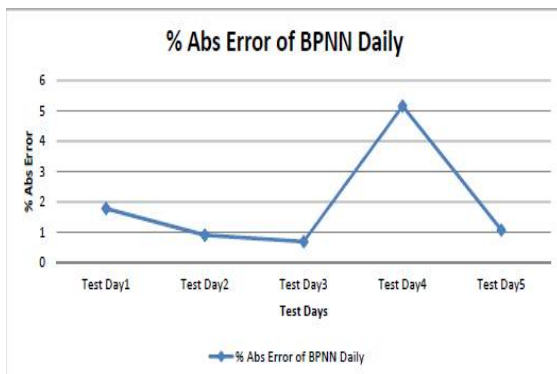


Fig. 3: Absolute Error (Daily) for BPNN

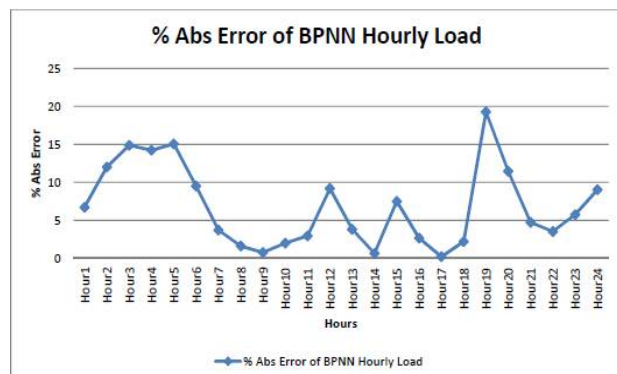


Fig. 4: Absolute Error (Hourly) for BPNN



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

VII. CASE STUDY FOR GRNN

Data set is same as used earlier for BPNN. Training data for 5 historical sample days are shown in Table-3. The 300 Historical days are designated as Historical Day 1 to Historical Day 300. The sample days are from 36 to 40.

		Historical Day 36				Historical Day 37				Historical Day 38			
		Hours (8:12)		Hours (9:13)		Hours (8:12)		Hours (9:13)		Hours (8:12)		Hours (9:13)	
Hours	Load	PIV Input	Load	PIV Target Output	Load	PIV Input	Load	PIV Target Output	Load	PIV Input	Load	PIV Target Output	
8	2667	1			2601	1			2847	1			
9	2739	3	2739	3	2789	3	2789	2	2896	3	2896	3	
10	2804	4	2804	4	2790	4	2790	3	2963	5	2963	5	
11	2912	5	2912	5	2889	5	2889	5	2942	4	2942	4	
12	2709	2	2709	2	2769	2	2769	1	2853	2	2853	2	
13			2622	1			2852	4			2758	1	
		Historical Day 39				Historical Day 40				PIV = Profile Index Vector			
		Hours (8:12)		Hours (9:13)		Hours (8:12)		Hours (9:13)					
Hours	Load	PIV Input	Load	PIV Target Output	Load	PIV Input	Load	PIV Target Output					
8	2943	1			2581	1							
9	3112	4	3112	4	2645	2	2645	1					
10	3083	3	3083	2	2788	3	2788	2					
11	3197	5	3197	5	2965	4	2965	4					
12	3081	2	3081	1	3067	5	3067	5					
13			3098	3			2901	3					

Table-3. Training Data for GRNN

Historical Load for 5 days from day 36 to 40. Load values from hours 8:12 are common to both input and output side

The load data on test days are used to test the trained GRNN to get the forecast values. The difference between the forecast value and the actual value gives the error. The resulting values of forecasting for 5 test days are shown in Table-4.

		Test Day 1				Test Day 2				Test Day 3			
Hours	Actual Load	PIV Input P_{in}	GRNN Output T_{out}	PIV of Actual Data	Actual Load	PIV Input P_{in}	GRNN Output T_{out}	PIV of Actual Data	Actual Load	PIV Input P_{in}	GRNN Output T_{out}	PIV of Actual Data	
8	4409	4	3.7870		4252	3	3.4958		4331	2	2.7089		
9	4062	1	1.2535	1	4087	1	1.2342	1	4250	1	1.0070	1	
10	4129	2	1.7675	2	4191	2	1.7963	2	4347	3	2.6991	2	
11	4564	5	5.9990	5	4667	5	5.9986	5	4760	5	5.9937	5	
12	4379	3	4.6836	4	4487	4	4.7535	4	4357	4	4.9679	3	
13	4265		3.5093	3	4379		3.7216	3	4451		3.6236	4	
α	3923.4				3900.0				4095.2				
β	128.4				145.6				104.6				

Table-4. Results obtained on running the GRNN

The GRNN output T_{out} and the corresponding Actual PIV's are shown side by side for comparison

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Hours	Test Day 4				Test Day 5			
	Actual Load	PIV Input P_{in}	GRNN Output T_{out}	PIV of Actual Data	Actual Load	PIV Input P_{in}	GRNN Output T_{out}	PIV of Actual Data
8	4409	3	3.7870		4202	4	3.4958	
9	4160	2	1.2535	2	3787	1	1.2342	1
10	4101	1	1.7675	1	3951	2	1.7963	2
11	4518	5	5.9990	5	4424	5	5.9986	5
12	4420	4	4.6836	4	4184	3	4.7535	4
13	4162		3.5093	3	4104		3.7216	3
α	3993.4				3652.1			
β	109.4				152.5			

Table-4(contd). Results obtained on running the GRNN
The GRNN output T_{out} and the corresponding Actual PIV's are shown side by side for comparison

Predicted and Actual values on the test days are shown in Table-5. Error= abs(Actual Load – Forecast Load)

	Test Day 1	Test Day 2	Test Day 3	Test Day 4	Test Day 5
Actual Load	4265	4379	4451	4162	4104
Forecast Load	4382.1	4357.4	4419.7	4375.2	4196.9
Abs(Error)	117.1	21.6	31.3	213.2	92.9
% Abs(Error)	2.75	0.49	0.70	5.12	2.26

Table-5. Actual Load, Forecast Load, Absolute value of the error and percentage Absolute error are shown

The plots of the actual load and the forecasted load and their %Abs error are shown in figures-5,6,7,8.

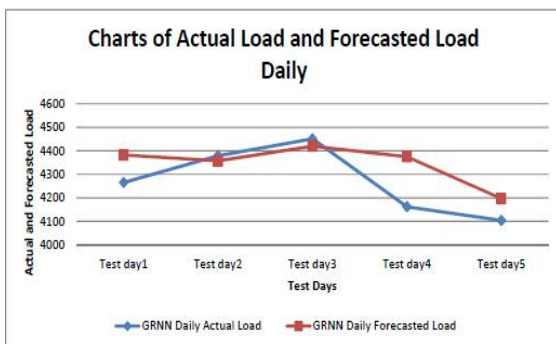


Fig. 5: Actual and Forecast Load (Daily) for GRNN

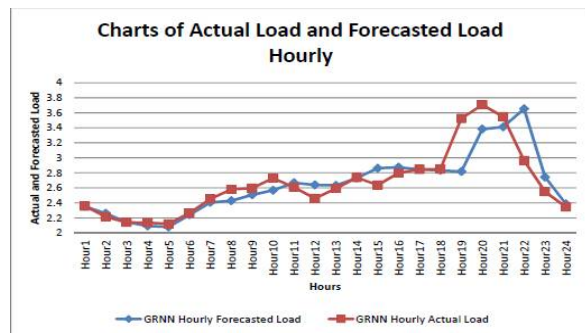


Fig. 6: Actual and Forecast Load (Hourly) for GRNN

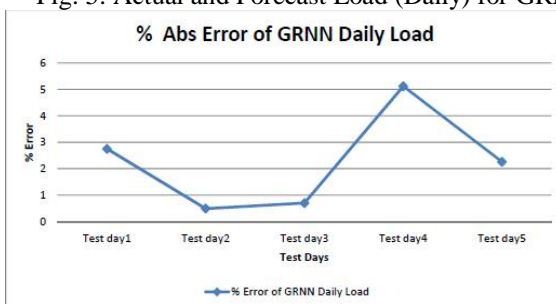


Fig. 7: Absolute Error (Daily) of GRNN

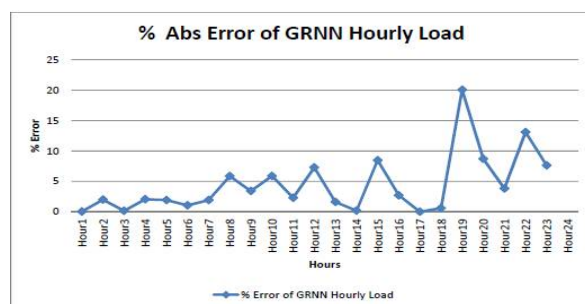


Fig. 8: Absolute Error (Hourly) of GRNN



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

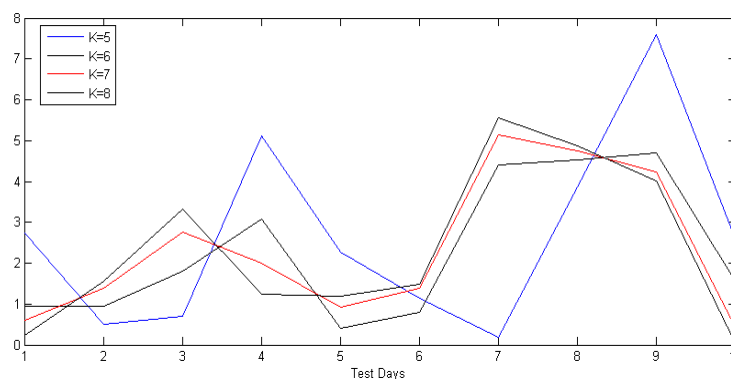
Vol. 5, Issue 3, March 2016

Thirty test days APE (Absolute Percentage Error), and the MAPE (Mean APE) are shown in Table-6 for different values of K, where K is the size of the input Vector.

Test Day	K=5	K=6	K=7	K=8
1	2.75	0.94	0.61	0.22
2	0.49	0.95	1.38	1.55
3	0.70	1.80	2.77	3.32
4	5.12	3.07	2.00	1.23
5	2.26	0.41	0.92	1.19
6	1.15	0.80	1.40	1.48
7	0.18	4.41	5.14	5.56
8	3.85	4.54	4.76	4.87
9	7.60	4.71	4.22	4.01
10	2.60	1.57	0.42	0.05
11	1.59	0.74	1.19	1.27
12	6.35	4.51	4.56	4.34
13	3.19	0.45	0.18	0.46
14	2.69	0.92	0.50	0.31
15	0.94	2.77	3.04	3.31
16	6.19	3.53	3.28	3.23
17	8.61	6.54	7.35	7.05
18	5.55	3.10	3.15	3.00
19	4.90	1.88	2.51	2.47
20	0.92	3.86	3.77	3.94
21	4.95	1.65	1.30	1.13
22	0.20	1.78	1.89	2.01
23	4.70	3.78	2.89	2.56
24	3.33	7.48	2.97	2.97
25	3.85	3.48	3.84	3.80
26	0.27	5.56	0.37	0.28
27	1.59	1.77	1.72	1.92
28	0.05	1.98	1.83	1.94
29	0.32	0.31	1.11	1.22
30	4.22	1.62	1.49	1.49
MAPE	3.0371	2.6974	2.4183	2.4053

Table-6 APE values for 30-days

From this result, we see that, as the size of the input vector increases, the MAPE value decreases. Variation of APE over first 10 test days (out of 30) with respect to the size of the input vector is shown in the 2D plot of Fig 9.





International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

Fig.9. Variation of APE with respect to the test days

Size of Input Vector	Q=150	Q=200	Q=250	Q=300
K=5	3.0705	3.0577	3.0557	3.0371
K=6	2.3797	2.3770	2.3753	2.6974
K=7	2.4560	2.4154	2.4145	2.4183
K=8	2.4934	2.4245	2.4120	2.4053

Table-7. Variation of MAPE for 30 test days, with respect to the number of training vectors and the size of the input vector, Q=Number of Training Vectors (Number of Training Days)

From the result it can be seen that MAPE does not change much, as the number of training vectors vary over the range 150 to 300 days.

VIII. DISCUSSION

Forecasting using PIV's requires more numerical computation, because of encoding and decoding numerical data sets. The regression coefficients α and β have to be calculated for every data vector. The α and β values also vary as the size of the vector varies. This involves additional calculation when the size of the input vector is increased to take care of more and more previous hourly loads. But, because of the availability of modern high speed computers, the calculation part is not a bottle neck in the forecasting process.

IX. CONCLUSION

The use of Profile Index Vectors (PIV's) for Load Profiling and forecasting is entirely a new technique. PIV's provide a compact and convenient representation of load profiles. The data sets are integers in a specific range, normalized and occupy the minimum space for storage. The integer form helps in ANN training. The training of ANN becomes fast and consistent because of the limited range of the values of the input training vector. The PIV encoded form is also useful for data validation, anomaly detection and easy and efficient load pattern analysis. In this method, abnormal days can be easily detected during training, and are discarded while training. PIV's are basically permutation vectors within certain range. Therefore, rough set techniques also can be applied for data set reduction. From our, results we conclude that, PIV's are very well suited for STLF. This technique can be extended to estimate peak loads and day-ahead forecasting.

Here, we are using two types of ANN, namely BPNN and GRNN. Compared to BPNN's, GRNN's have quicker training times and easy to design. GRNN's are well suited when the inputs and outputs of the ANN's are spread over a small numerical range. But, to handle nonlinearities, BPNN is well suited. Best results are obtained by forecasting using BPNN's and then validating the forecast by GRNN.

REFERENCES

- [1] Yu, and Ki. Lee, "A Method for classification of electricity demands using load profile data", International Conference On Computer And Information Science, pages 164-168, Nov 2005
- [2] WWW <http://en.wikipedia/wiki/the> free Enclyopedia-Linear regression
- [3] Simon Haykin Mc, "Neural Networks" A Comprehensive Foundation Second edition Prentice Hall International Inc Chapter 4 Page no 178-184.
- [4] M. S. Sachdev R. Billinton C. A. Peterson, "Representative Bibliography on Load Forecasting" IEEE Transactions on Power Apparatus and systems vol. PAS-96, no. 2, March/April 1977.
- [5] CIGRE Study Committee Present, "A Survey on load forecasting and load management", ELECTRA(145) Page No:69-89,1992.
- [6] Ibrahim Moghran and Saiffur Tahman, "Analysis and evaluation of five short term load forecasting techniques", IEEE tr. on Power systems, vol.4, No. 4, pp.1484-1494,October, 1989.
- [7] K.Y. Lee, J.H. Park, "Short-term load forecasting using an Artificial Neural Network", IEEE Tr. on Power Systems., vol. 7, Page No 124-130, February, 1992.
- [8] Kun-Long Ho, Yuan-Yin Hsu, C. c. Yang, "Short term load forecasting using a multilayer neural network with an adaptive learning algorithm", IEEE Tr. on Power Systems, vol. 7, pp. 141-149, February 1992.
- [9] C.N. Lu, H.T. Wu and S. Vemuri, "Neural network based short term load forecasting", IEEE Tr. on Power Systems, vol. 8, Page No 336-342, Feb 1993.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 3, March 2016

- [10] Kwang-Ho Kim, Hyoung-Sun Youn, and Yong-Cheol Kang. “Short-Term Load Forecasting for Special Days in Anomalous Load Conditions Using Neural Networks and Fuzzy Inference Method”, IEEE Transactions on Power Systems, VOL. 15, NO. 2, MAY 2000.
- [11] Kyung-Bin Song, Young-Sik Baek, Dug Hun Hong, and Gilsoo Jang, “ Short-Term Load Forecasting for the Holidays Using Fuzzy Linear Regression Method”, IEEE Transactions on Power Systems. VOL, 20, NO.1, FEBRUARY 2005.
- [12] Srinivas and Amit Jain, “A Methodology for Short Term Load Forecasting Using Fuzzy Logic and Similarity”.
- [13] M.M. Othman, Member, IEEE, 1M.H.H. Harun, and 1IMusirin, Member, IEEE “Short Term Load Forecasting Using Artificial Neural Network with Feature Extraction Method and Stationary Output” 2012 IEEE International Power Engineering and Optimization Conference (PEOCO2012), Melaka, Malaysia: 6-7 June 2012.
- [14] A Comparative Study of BPNN, RBFNN and ELMAN Neural Network for Short-Term Electric Load Forecasting: A Case Study of Delhi Region publication at: <http://www.researchgate.net/publication/282686878> ARTICLE · FEBRUARY 2015

BIOGRAPHIES



Dr. K N Bhanuprakash received the BE degree in Electrical and Engineering at NIE, Mysore University and ME in Power System at NIE, Mysore University. He received Ph.D at VTU Belgaum. He headed many institutions as principal and has a rich knowledge in administration. He has published many National and International papers.



Supreetha Patel T P received the BE degree at KIT, Tiptur in the department of Information Science and Engineering and M.Tech in the department of Computer Network and Engineering at SIT, Tumkur. Her research areas include Wireless Sensor Networks, Artificial Intelligence, Image Processing and so on.



Nikshepa T received the BE degree at KIT, Tiptur in the department of Information Science and Engineering and M.Tech in the department of Computer Science and Engineering at JNNCE, Shimoga. Her research areas include Wireless Sensor Networks, Artificial Intelligence, Image Processing and so on.



Santosh C pursuing his BE degree at KSSEM, Bangalore in the department of Electrical and Electronics Engineering. His research areas include Artificial Intelligence and Expert Systems, Network Analysis and so on.