Long-Term Load Forecasting Using System Type Neural Network Architecture

Nathaniel J. Hobbs, Byoung H. Kim, Kwang Y. Lee, Fellow, IEEE

Abstract—**This paper presents a methodology for long-term electric power demands using a semigroup based system-type neural network architecture. The assumption is that given enough data, the next year's loads can be predicted using only components from the previous few years. This methodology is applied to recent load data, and the next year's load data is satisfactorily forecasted. This method also provides a more in depth forecasted time interval than other methods that just predict the average or peak power demand in the interval.**

Index Terms—**Decomposition, load forecasting, neural network, system-type architecture.**

I. Introduction

ACCURATE load forecasting is very important for electric utilities in a competitive environment created by the electric industry deregulation. In order to supply high quality electric energy to the customer in a secure and economic manner, an electric company faces many economical and technical problems in operation, planning, and control of an electric energy system [1]. Load forecasting helps an electric utility to make important decisions including decisions on purchasing and generating electric power, load switching, and infrastructure development. Load forecasting is also important for energy suppliers, financial institutions, and other participants in electric energy generation, transmission, distribution, and markets [2].

Load forecasting can be divided into three categories: short-term forecasting which is usually from one hour to one week, medium-term forecasting which is from a week to a month, and long-term forecasting which can extend to several months or years. The short-term forecast is needed for control and scheduling of power system, and also as inputs to load flow study or contingency analysis [1]. In addition, short-term load forecasting can help to estimate load flows and to make decisions that can prevent overloading. The long-term and medium-term forecasts are used to determine the capacity of generation and transmission, distribution system additions, and the type of facilities required in transmission expansion planning, annual hydrothermal maintenance scheduling, etc.

The load is a non-stationary process which is affected by two main factors: time of the day and weather conditions. The time dependence of the load reflects the existence of a daily load pattern, which may vary for different weekdays and seasons. Temperature is the primary weather factor affecting the load. Humidity and wind speed are some of the other factors that may also influence power consumption. For the models including weather variables, the total load may be decomposed into the weather sensitive load and the non-weather sensitive load. The weather sensitive load is mostly predicted using correlation techniques [1].

Most forecasting methods use statistical techniques or artificial intelligence algorithms such as regression, neural networks, fuzzy logic, and expert systems [1]. A variety of methods, which include various regression models, time series, neural networks, statistical learning algorithms, fuzzy logic, and expert systems, have been developed for long-term forecasting. The downside to these methods is that they usually only predict a single value; average, peak, or total power, for any given month or time interval. The proposed method, however, provides broader feel for the forecasted load in that it predicts every hour of every day for the next year.

In general, the load has two distinct patterns: weekday and weekend patterns. Weekday patterns include Tuesday through Friday and weekend patterns include Sunday through Monday. In addition, holiday patterns are different from non-holiday patterns. In this paper there is no distinction made between holidays and non-holidays.

Section II of this paper provides the general background to the load forecasting problem. Section III describes the proposed method. Section IV applies the method to the forecasting problem, and in section V general conclusions are drawn from the results.

II. General Background

Forecasting methods can generally be divided into two broad categories: parametric methods and artificial intelligence based methods. The parametric methods formulate a mathematical or statistical model of load by examining qualitative relationships between the load and the factors affecting the load. The assumed model parameters are then estimated from historical data and the adequacy of the model

---

This work was supported in part by the U.S. National Science Foundation under Grant ECS-0501305.

Nathaniel Hobbs and Byoung H. Kim are with the Electrical Engineering Department, the Pennsylvania State University, University Park, PA 16802 USA (e-mail: njh152@psu.edu; bxk232@psu.edu)

K. Y. Lee is with the Electrical and Computer Engineering Department, Baylor University, Waco, Texas 76798 USA (phone: (254) 710-4195, e-mail: Kwang_Y_Lee@baylor.edu).

is verified by analysis of forecast errors. Artificial intelligence based methods use artificial neural network as a load model. For either of these methods, to perform long-term load forecasting, several factors should be considered, such as the time factor, weather data, and possible customers' classes. The time factors include the season of the year, the day of the week, and the hour of the day. There are differences in load between weekdays and weekends. For example, Mondays and Sundays being adjacent to weekends, may have structurally different loads than Tuesdays through Fridays. Obviously the electric loads are very much dependent upon weather conditions. The load models, however, which include weather variables, are limited in use by problems such as inaccuracy of weather forecasts and difficulties in modeling the weather-load relationship [12]. In this paper, the weather effects on the electric load are not explicitly considered.

Most long-term forecasting is done using artificial intelligence techniques. Many methods have previously been proposed using artificial neural networks (ANN), fuzzy logic, or some combination of the two [6]. Neural networks have become increasingly popular in the past few years because of their abilities to model non-linear and very complex systems. Other proposed methods propose modeling the overall load pattern by multiple linear regression models [7]. Still others have proposed methods that decompose the systems using wavelet decomposition with very good results [9]. However, most of these approaches only forecast the average, total, or peak values for a given interval of time.

III. THE PROPOSED METHOD

Recently, a shift has occurred in the overall architecture of neural networks from simple or component-type networks to system-type architectures. The most popular architecture seems to be the one advocated by Jacobs and Jordan [11], called the "Modular Connectionist Architecture". The most serious flaw in the design of system-type neural networks is the lack of a cohesive discipline in the architectural design and in the design of the learning algorithm. Virtually, the entire design is done on an intuitive basis. To illustrate the lack of a cohesive discipline, in [14], the partitioning of components corresponds to separation of variables, which works if the variables are separated and does not work if the variables are not separated [3]-[5].

A. The System-type Neural Network Method

In previous papers [3]-[5], a system type neural network was proposed which implemented extrapolation. In this method, the distributed parameter system (DPS) surface determined by a given data set was expanded along one axis. Rather than thinking of the load as, $Load = f(Day, Hour, Weather, Customer\ classes)$, this approach considers the $Load = f(Day, Hour)$, parameterized by weather and customer classes. Other parameters might be population growth and special events such as Olympics. That is, the role of the parameters is that they determine the transformation form one load surface to another surface. In the next section, it will be shown that the load for any year can be represented in the following form:

$$L(Day, Hour) = C(Day)E(Hour) \tag{1}$$

This entire method hinges on the assumption that the basis vector set remains very similar from year to year. On account of this, a basis set can be chosen and used for multiple years. Thus, only the coefficient vector for any other year must be known in order to reconstruct the electric power demand.

Neural networks are being used for systems described by PDE's [8]. The system-type attribute of the neural network architecture is shown in Fig. 1, implementing an arbitrary function $L(D,H)$. Unlike conventional neural network architectures that would attempt to achieve the mapping $L(D,H)$ with one neural network, the proposed architecture reflects a system-type approach using two neural network channels, a Function Channel and a Semigroup Channel, in an adaptation of the connectionist architecture (Fig. 1). During use, the Semigroup Channel supplies the function channel with a coefficient vector $C(D)$ as a function of the index D. The coefficient vector, when applied to the basis set $E(H)$ of the function channel, causes the function channel to operate as one specific function from within a vector space of functions. Jointly, these two channels realize a semigroup-based implementation of the mapping $L(D,H)$.

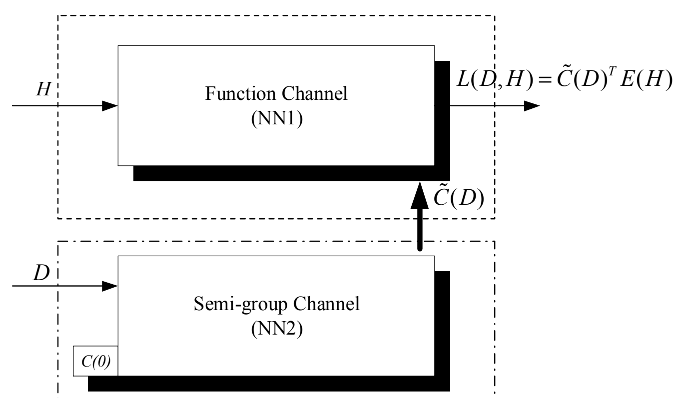


Fig. 1: System-type architecture.

The function channel can have a Radial Basis Function (RBF) architecture [11]. It consists of n RBF networks, each one of which implements one orthonormal vector of an n-dimensional basis set of vectors, $E(H)$. The dimensionality, n, is chosen as the minimum number of vectors which, when recombined with the coefficient vectors, will result in a reconstructed load demand within a given error tolerance. The outputs of the orthonormal vectors are (internally) linearly summed so that the channel spans an n-dimensional function space. The coefficients, which determine the linear sum and thereby define the specific function being implemented is supplied by the Semigroup Channel. Up to this point, the operation of the RBF channel parallels the idea used by Phan and Frueh [15].

One of the essential differences between their approach and the present proposed approach is that the former requires prior engineering knowledge for selecting the basis vectors, and the latter approach requires no such knowledge. One advantage that RBF networks have over other architectures is that their functionality can be given an explicit mathematical expression in which the neuron activation functions act as Green's functions. This makes these networks amenable to design rather than training. Another advantage is that they function as universal approximators [16]. The Semigroup Channel can be adapted from the Diagonal Neural Network (DRNN) or the Elman architecture [13], in which the input is split into a dynamic scalar component D and one static vector component, the vector $C(0)$. The output is a vector, $C(D)$, which is related to the dynamic input D and to the static input $C(0)$ by the semigroup property:

$$C(D) = \Phi(D)C(0)\text{, where } \Phi(D_1 + D_2) = \Phi(D_1)\Phi(D_2) \quad (2)$$

B. Learning Algorithm of the Proposed System-type NN

The first component of the system, namely the Function Channel, since it is composed of RBF components, can be designed, rather than trained. The second component, the Semigroup Channel, can be trained in the new way illustrated in Fig. 2. During training, the Semigroup Channel receives as input a preliminary coefficient vector $C(D)$ and produces a smoothed coefficient vector, $\widetilde{C}(D)$. That is, the primary objective of training is to replicate (and, if necessary, to smoothen) the vector $C(D)$ with a vector $\widetilde{C}(D)$ which has the following semigroup property:

$$\widetilde{C}(D) = \Phi(D)\widetilde{C}(0), \quad (3)$$

where and $\Phi(D)$ is an nxn matrix that satisfies:

$$\Phi(D_1 + D_2) = \Phi(D_1)\Phi(D_2) \quad (4)$$

However, there is a secondary objective of training; the channel must also "replicate" the semigroup property of the trajectory by gradually acquiring a semigroup property of its own, in the weight space. The existence of this acquired semigroup property in the weight space becomes the basis for extrapolation [3]. In order to elicit this gradual acquisition of the semigroup property, it is necessary that the training in this second step (semigroup tracking) occur in a gradual manner, as shown in Fig. 2. In Fig. 2, the entire trajectory is split into successively-longer sub-trajectories. The network is trained on each of these consecutive sub-trajectories until the weights converge.

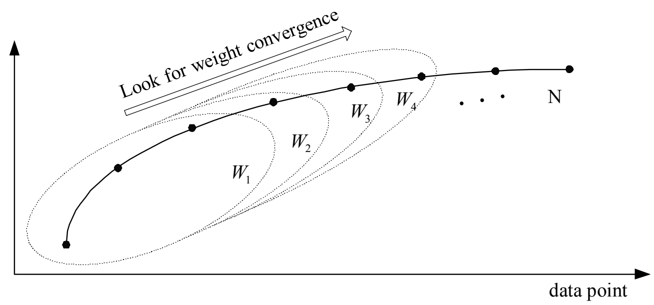


Fig. 2: Overview of new training algorithm.

IV. Simulation and Comparison

Since each day has its own unique load pattern, the load data from 2000 – 2004, provided by Korea Electric Power Corporation (KEPCO), was separated into different days of the week. Wednesday was arbitrarily chosen for extrapolation and each year of Wednesday data was decomposed into a primary basis set and coefficient set of dimensionality n, where n is set to six. Dimensionalities of four and eight were also tried. Four basis vectors did not provide satisfactory computed loads, while eight vectors did not result in significant improvement over six vectors. A common basis set was obtained from the year 2000 and was used as the basis for the other years as well. The data from the year 2004 was set aside for later comparison with forecasted data obtained by extrapolating the first three years. To illustrate the validity of the basis vectors and coefficient vectors, the empirical data for the year 2004 is shown, Fig. 3, for comparison with the computed load, Fig. 4 which was obtained using the rule:

$$L(Day, Hour) = C(Day)E(Hour)\,. \quad (5)$$

Here, Day is a sequential number for a day of the week within the year, where $Day = 1...52$, as there are generally 52 of a given day of the week in a year.

Fig. 3: Empirical load data for year 2004.

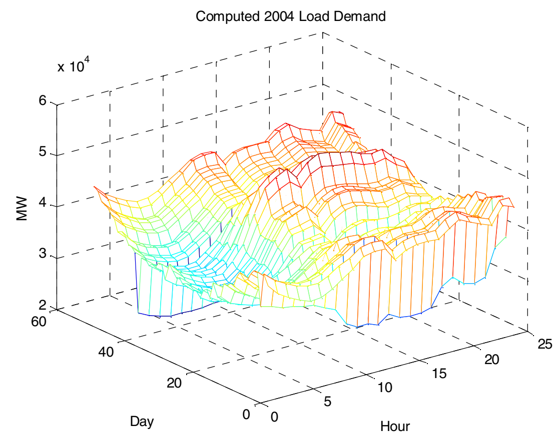


Fig. 4: Load computed from primary vector sets.

Ordinarily, when using the proposed method, extrapolation of the coefficient vectors from the year 2003 would have been attempted. After the coefficient vectors had been extended by the number of Wednesdays in the year 2004, they would have been recombined with the primary basis set of the year 2003, and the year 2004 would have been forecasted. However, it was found when the empirical data was decomposed that the coefficient vectors were highly non-smooth. This non-smooth property of coefficient vectors, Fig. 5, made it impossible to directly extrapolate the coefficient vectors.

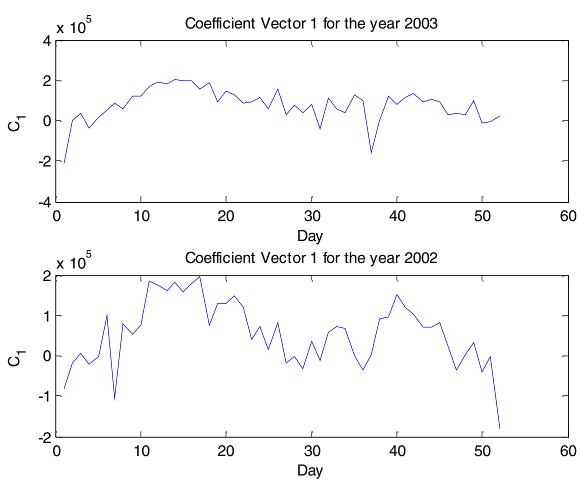


Fig. 5: Sample primary coefficient vector trajectories illustrating non-smoothness.

It was seen that all the coefficient vectors from each year were well correlated with their counterparts in the other years (for example, see Fig. 1 for the primary coefficient vector trajectory). Therefore, each coefficient vector trajectory from each year is stacked with its counterparts from the remaining years. This created new three dimensional data sets, $\overline{C}_i$, where i represents the set of stacked coefficient vectors trajectories i. The new data sets can also be decomposed into a secondary basis and coefficient set. Furthermore, because of the correlation it is expected that the secondary coefficient vectors will be smooth. This new DPS is decomposed as:

$$\overline{C}_i(Year, Day) = C_i^2(Year)E_i^2(Day), \quad (6)$$

where the superscript two indicates that this is the secondary decomposition. Each these new three dimensional data sets was decomposed into a set of four basis vectors, $E_i^2(Day)$, and four coefficient vectors, $C_i^2(Year)$. By choosing the appropriate basis vectors, smooth secondary coefficient vectors were obtained, and their extrapolation became possible. Now, as a result of the secondary decomposition, it is not the day which is being extrapolated. Rather, it is the year that is being extrapolated. That is, by extrapolating the secondary coefficient vectors from the set, $C_i^2(Year)$, along the year axis, the primary coefficient vector, i, for the next year is being predicted.

Training a simple recurrent network (SRN) with the proposed progressive training algorithm, a good fitting smoothed vector was found, shown in Fig. 6, and the semigroup channel acquired a semigroup property of its own. Thus the weights of the neural network were replaced with a weight change sequence calculated from the actual weight changes of the neural network within the observation window.

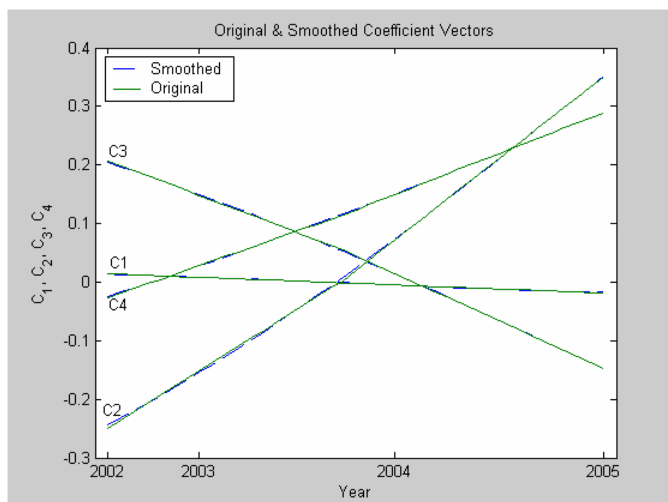


Fig. 6: Comparison of original and smoothed secondary coefficient vectors.

Note that these continuous coefficient vector trajectories represent discrete data points. In order for the weight changes to converge, however, the neural network needed a continuous training path. Thus, the continuous curves were fit to the discrete points. In other words, the coefficient vector trajectories are only valid at the points 2003, 2004, and 2005 on the year axis. The data used in training the network included only data from the last half of the year 2002 and all the data from the year 2004.

Extrapolating the smoothed coefficient vectors from the observation window into the test window confirmed that the weight change sequence was valid. The differences, as seen in Fig. 7, between the smoothed vectors and the extrapolated vectors are minimal. Extending the weight change sequence into the next region the final extrapolated vectors were obtained and shown in Fig. 8.

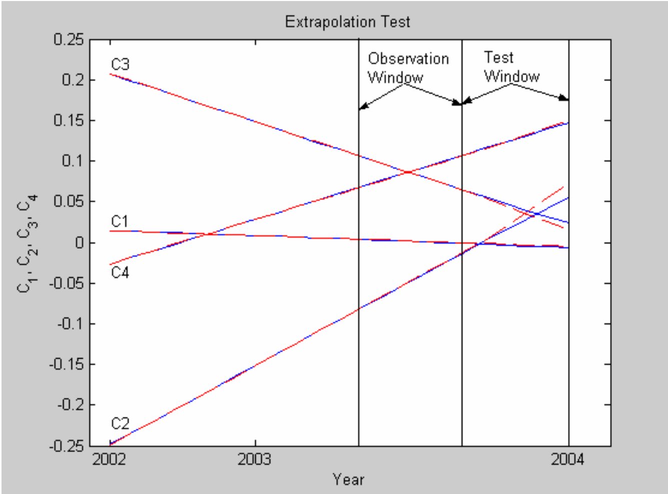


Fig. 7: Extrapolation tests for secondary coefficient vectors from primary coefficient vector set $\overline{C}_1$.

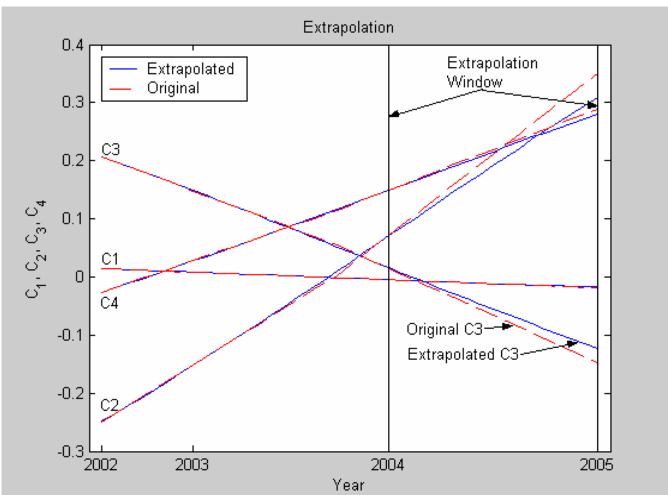


Fig. 8: Extrapolation for secondary coefficient vectors from primary coefficient vector set $\overline{C}_1$.

This extrapolation testing and final extrapolation was then performed for each coefficient vector set within the secondary decomposition. The extended secondary coefficient vectors, when recombined with their respective basis sets, result in a predicted coefficient vector for the year 2004, Fig. 9.

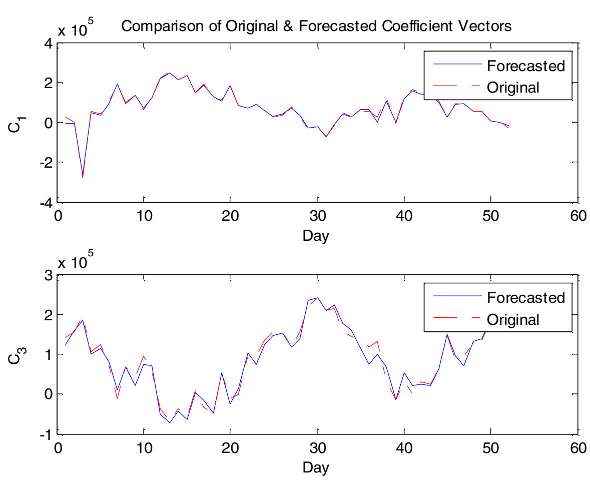


Fig. 9: Comparison of actual and predicted primary coefficient vectors for the year 2004.

For this paper only primary coefficient vectors one and three were shown. They were selected, because they were the most non-smooth, and thus the hardest to predict. As can be seen from Fig. 9, the predicted coefficient vector is very similar to the actual coefficient vector. Since the proposed method of extrapolation was successful for both these, it is expected that it will also be successful for the rest of the primary coefficient vectors.

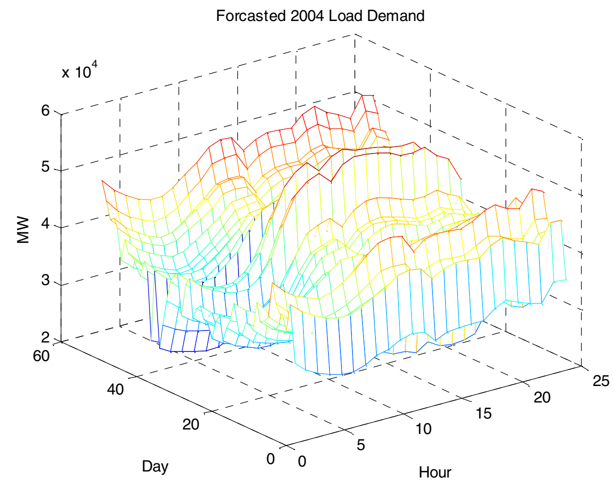


Fig. 10: Forecasted load for year 2004.

The forecasted load shown, Fig. 10, is very similar in shape to the empirical load demand in Fig. 3. Though there appears to be significant error around day 40. Fig. 11 shows a graphical representation of the data in Table 1.

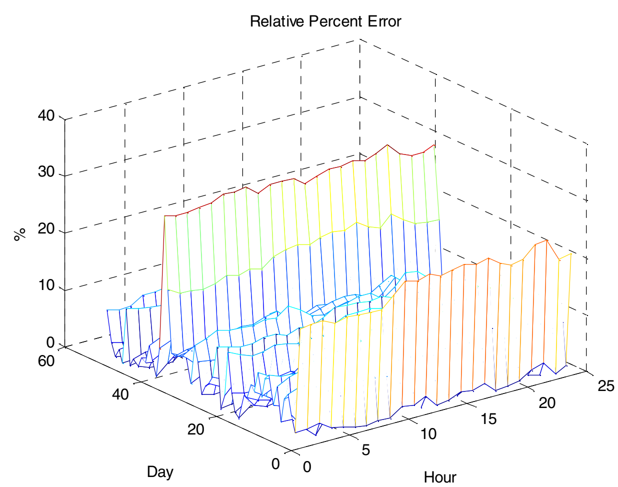


Fig. 11: Relative percent error.

The maximum relative error was 31.32%, and the average was 6.21%. Note that in the original load, Fig. 3, there is a significant negative peak on days 36 and 37. These are likely holidays and certainly deviations from the normal Wednesday load pattern, which results in a high error. Though the error is relatively high compared to that of short term forecasting, which is generally below 2%, it must be remembered that the purpose of long-term forecasting is not to precisely predict the load at any given time, but rather to predict the general trend in the demands over an interval of time. The proposed method has several advantages over other long-term forecasting techniques. First, by using the proposed method, the load pattern can be seen for many different times of day, seasons of the year, etc. This is a distinct advantage over methods which only predict the average, peak, or total demand for a month. Secondly, the proposed method provides the viewer with a more intuitive grasp of the overall load pattern.

To summarize the procedure, data from any day of the week is algebraically decomposed into a basis set and a coefficient set. Since the n vectors comprising the coefficient set are most likely non-smooth, n new three dimensional functions are built by stacking the i^{th} coefficient vector from each year. These in turn are decomposed in an attempt to find a smooth secondary coefficient set. If they are found, the vectors are extrapolated using the neural network system to predict the next year's coefficient vector. Once all n coefficient vectors have been predicted, they are combined with the primary basis set and the forecasted year is obtained.

V. Conclusions

In this paper, a methodology was proposed to perform long-term electric power demand forecasting. This method was then applied to an empirical data set, and the year 2004 was forecasted by predicting its primary coefficient vectors. It was shown that the proposed method achieved satisfactory results although there was no attempt to separate holidays from non-holidays. If the holidays were separated out, it is expected that the error would be significantly lower. This method also provides a more in depth forecasted time interval, rather than just predicting the average or peak power demand in the interval.

This paper also addressed the problem of extrapolating highly non-smooth coefficient vectors, by a secondary decomposition.

This method, though applied to Wednesday in this paper, can be applied to any other day of the week including weekends. Moreover, it is expected that similar results can be obtained for any day.

VI. Acknowledgment

The authors thank the Korea Electric Power Corporation (KEPCO) for providing the load data.

VII. References

[1] K. Y. Lee, Y. T. Cha, and J. H. Park, "Short-term load forecasting using an artificial neural network," *IEEE Trans. on Power Systems*, vol. 7, pp. 124-132, Feb. 1992.

[2] J. H. Chow, *Applied Mathematics for Restructured Electric Power Systems: Optimization, Control, and Computational Intelligence*. New York: Spinger-Verlag, 2005, ch. 12.

[3] K. Y. Lee, J. P. Velas, and B. H. Kim, "Development of an Intelligent Monitoring System with High Temperature Distributed Fiberoptic Sensor for Fossil-Fuel Power Plants," *IEEE Power Engineering Society General Meeting*, pp. 1350-1355, Jun 6-10, 2004.

[4] B. H. Kim, J. P. Velas, and K. Y. Lee, "Development of intelligent monitoring system for fossil-fuel power plants using system-type neural networks and semigroup theory," *IEEE Power Engineering Society General Meeting*, pp.2949-2954, 2005.

[5] Byoung-Hee Kim, Velas, J.P., and Lee, K.Y., "Semigroup based neural network architecture for extrapolation of enthalpy in a power plant," in *Proc. 2005 International Conference on Intelligent Systems Application to Power Systems,* pp. 291-296

[6] M.A Farahat, "Long-term industrial load forecasting and planning using neural networks technique and fuzzy inference method," *39th International Universities Power Engineering Conference*, 2004. Vol. 1, pp.368 - 372

[7] H.M. Al-Hamadi and S.A. Soliman, "Long-term/mid-term electric load forecasting based on short-term correlation and annual growth," in *Electric Power Systems Research* Vol. 74, Issue 3 , June 2005, Pages 353-361

[8] T.Q.D. Khoa, L.M. Phuong, P.T.T. Binh, and N.T.H. Lien, "Application of wavelet and neural network to long-term load forecasting," in *Power System Technology Conference 2004*, Vol. 1, pp. 840 - 844

[9] T.Q.D. Khoa, L.M. Phuong, P.T.T. Binh, and N.T.H. Lien, "Application of wavelet and neural network to long-term load forecasting," in 2004 International Conference on Power System Technology, Vol. 1, pp.:840 - 844

[10] Q. Ahsan and M. Uddin, "A Probabilistic Approach of Electrical Energy Forecasting," in *IEEE Proc. 2005 Instrumentation and Measurement Technology Conference*, Vol. 2, pp.1070 – 1074

[11] R. Jacobs and M. Jordan, "A competitive modular connectionist architecture," *Advances in Neural Information Processing Systems*, Vol. 3, p. 767-773, 1991.

[12] J. H. Park, Y. M. Park, and K. Y. Lee, "Composite modeling for adaptive short-term load forecasting," *IEEE Trans. on Power Systems*, vol. 6, pp. 450-457, May. 1991.

[13] B. H. Kim, J. P. Velas, and Kwang Y. Lee, "Short-Term Load Forecasting Using System-Type Neural Network Architecture", in *International Joint Conference on Neural Networks 2006*, pp. 2619 - 2626

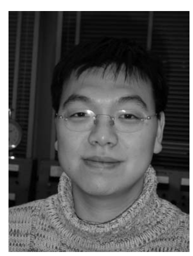
[14] A. Atiya, R. Aiyad, and S. Shaheen, "A Practical Gated Expert System Neural Network," *IEEE International Joint Conference on Neural Networks*, Vol. 1, pp. 419-424, 1998.

[15] M. Q. Phan and J. A. Frueh, "Learning Control for Trajectory Tracking using Basis Functions," *Proceedings of the 35th IEEE Conference on Decision and Control*, pp. 2490-2492, Dec. 1996.

[16] S. Haykin, *Neural Networks*, 2nd ed., Prentice Hall: N.J., 1999.

VIII. Biographies

Nathaniel J. Hobbs received his B.S. degree in Electrical Engineering from the University of Missouri-Rolla in 2006. He is currently enrolled in the Ph.D program in Electrical Engineering at the Pennsylvania State University, University Park, Pa. His research interests are in neural networks, intelligent and classical control, and industrial automation.



Byoung-Hee Kim received his B.S. and M.S degrees in Control and Instrumentation Engineering from University of Ulsan in 1998 and 2000, respectively. He received his Ph.D degree in Electrical Engineering in 2007 from the Pennsylvania State University, University Park, Pa. His research interests are in neural network, intelligent control, power plant control, and network based control systems.

Kwang Y. Lee received his B.S. degree in Electrical Engineering from Seoul National University, Korea, in 1964, M.S. degree in Electrical Engineering from North Dakota State University, Fargo, in 1968, and Ph.D. degree in System Science from Michigan State University, East Lansing, in 1971. He has been with Michigan State, Oregon State, Univ. of Houston, Penn State, and Baylor University, where he is now a Professor and Chair of Electrical and Computer Engineering. His interests include power system control, operation, planning, and intelligent system applications to power systems. Dr. Lee is a Fellow of IEEE, Associate Editor of IEEE Transactions on Neural Networks, and Editor of IEEE Transactions on Energy Conversion. He is also a registered Professional Engineer.

TABLE I: RELATIVE PERCENT ERROR OF FORECASTED RESULTS

D \ Hr	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	21	21	21	20	20	20	20	20	22	24	24	24	23	24	24	24	24	23	22	22	24	24	21	21
2	2.0	2.1	2.6	1.3	0.8	0.3	0.1	0.2	0.1	1.8	1.4	2.3	0.2	0.6	1.4	1.1	1.7	0.0	0.3	0.4	1.9	2.5	0.0	1.0
3	5.0	2.1	0.3	3.3	6.2	8.0	7.6	7.6	7.2	5.7	4.0	0.0	4.8	1.3	0.4	1.7	3.1	11	8.7	3.0	0.7	0.2	6.5	8.3
4	8.4	8.9	8.9	7.2	6.4	5.5	5.2	5.3	6.2	9.0	8.7	9.1	7.9	8.5	9.3	8.8	8.5	5.3	6.4	6.4	8.9	9.6	7.3	6.9
5	7.1	7.5	7.0	5.5	5.2	4.7	4.3	4.2	4.8	7.5	7.5	8.5	7.2	7.6	8.3	7.8	7.7	4.5	4.7	5.3	7.4	8.1	6.0	6.4
6	2.8	2.7	2.3	1.1	0.6	0.2	0.8	0.6	0.5	1.2	0.7	1.0	0.1	0.3	0.9	0.2	0.3	2.2	1.4	1.0	1.0	1.8	0.1	0.9
7	8.6	9.7	10	11	11	12	13	13	14	13	13	13	14	15	14	15	15	17	16	16	13	12	13	11
8	5.1	4.4	3.6	2.4	2.6	2.3	1.6	1.1	2.0	3.1	3.4	3.7	3.1	3.7	3.8	3.6	3.8	0.2	0.7	1.7	3.6	3.4	2.2	3.7
9	10	9.5	9.2	8.2	8.0	7.3	7.2	7.7	8.3	11	11	12	12	12	13	12	12	8.6	8.0	9.1	11	11	8.9	9.8
10	9.3	8.3	7.3	6.0	6.3	6.2	5.8	6.2	7.2	8.8	9.3	10	10	10	11	9.9	11	7.5	6.0	7.9	9.8	9.3	7.2	7.7
11	5.7	6.7	7.3	6.8	4.8	4.0	5.4	5.2	6.9	7.2	7.0	6.4	6.2	7.3	6.9	7.3	6.1	7.2	8.5	8.5	7.3	7.1	6.3	2.1
12	9.6	8.4	6.9	6.6	7.6	7.9	7.0	7.1	7.6	9.2	9.5	10	11	10	11	9.8	9.8	8.2	6.2	7.1	9.6	8.9	8.0	9.2
13	6.0	4.5	3.1	2.6	3.8	4.3	3.6	3.2	2.7	4.2	4.8	5.0	5.2	4.8	5.8	4.4	4.1	3.3	1.2	2.8	4.8	3.9	3.7	5.1
14	6.9	5.2	4.0	3.3	4.5	4.6	3.9	3.8	3.7	5.3	5.7	6.3	6.4	6.3	6.6	5.4	5.9	4.6	1.9	3.9	5.8	5.0	4.7	5.5
15	4.3	3.2	2.7	2.5	4.1	4.9	4.0	3.1	3.0	3.9	5.0	4.1	4.7	4.6	5.0	4.0	4.3	4.2	1.8	3.2	4.0	3.1	4.0	5.2
16	4.3	2.2	1.4	1.6	2.6	3.5	2.4	1.9	1.3	2.4	3.4	3.7	2.5	2.8	3.8	2.9	3.9	3.1	1.2	3.4	3.7	2.0	2.2	2.3
17	0.0	1.8	3.5	4.3	3.6	3.8	4.7	5.3	6.1	4.9	4.3	3.9	4.8	4.6	4.2	6.6	5.5	6.9	10	7.2	3.8	4.3	4.2	3.2
18	0.8	2.1	4.0	5.2	4.7	5.1	6.5	7.7	7.7	5.1	3.0	2.4	3.7	3.5	3.0	4.5	5.2	7.8	11	8.0	2.6	3.6	3.8	5.0
19	3.9	3.3	3.3	3.6	4.6	4.9	4.3	4.2	3.5	3.6	4.7	4.7	4.3	4.0	4.5	3.7	4.6	4.1	2.9	3.1	3.8	3.6	4.0	5.6
20	11	9.8	9.0	9.1	9.8	9.5	9.0	9.3	9.3	11	12	12	12	11	11	10	11	11	8.3	8.6	11	10	10	10
21	2.9	4.9	6.2	6.6	6.8	7.6	8.1	9.5	10	8.8	7.4	7.1	8.6	8.6	8.2	9.1	9.9	11	12	11	6.5	7.0	7.2	7.9
22	0.1	0.4	1.0	0.3	0.0	0.3	0.3	0.4	1.8	1.0	0.2	0.2	0.7	0.8	0.5	1.5	0.1	0.9	2.7	2.3	0.4	0.7	0.3	1.0
23	11	10	9.7	9.8	10	10	9.5	10	9.9	11	12	12	12	11	12	11	12	11	9.4	9.5	11	11	11	11
24	5.1	4.1	3.6	3.6	4.0	3.6	3.4	3.6	3.1	4.2	5.2	5.0	5.0	5.1	5.1	4.4	5.0	4.0	2.3	3.3	4.3	4.6	4.2	5.0
25	3.7	2.7	2.5	3.1	3.2	3.2	2.4	2.8	2.8	3.1	3.5	3.8	3.3	3.1	3.4	3.1	4.5	3.4	2.2	3.1	3.6	3.0	2.7	3.7
26	3.0	3.5	4.2	3.9	3.5	3.8	4.2	3.1	3.9	3.4	2.8	2.6	2.4	2.7	2.5	2.9	2.1	3.0	4.6	3.7	3.2	3.0	3.3	2.5
27	1.3	0.7	0.8	0.9	1.3	1.2	0.9	1.1	0.2	1.2	1.3	0.7	1.0	0.5	0.7	0.3	1.4	0.9	0.0	0.4	1.1	0.7	0.8	1.5
28	6.7	6.2	6.0	6.0	6.2	6.1	5.7	6.6	5.9	7.1	7.5	7.5	7.7	7.7	7.6	7.1	8.1	6.8	5.5	6.3	7.0	6.8	6.7	7.4
29	9.1	9.9	11	10	9.9	9.9	10	10	12	10	8.7	8.4	9.1	9.4	10	9.9	8.5	9.9	12	11	9.7	9.4	9.4	8.6
30	2.3	3.1	3.4	3.5	2.9	3.5	3.8	3.5	4.0	2.7	2.0	2.0	1.8	2.4	2.3	2.5	2.0	3.2	4.8	4.0	3.8	2.1	2.6	1.7
31	6.3	4.5	3.1	2.4	2.6	3.1	3.5	3.8	2.3	3.1	3.6	3.4	4.1	3.6	3.3	3.5	3.0	2.4	1.3	2.9	3.6	3.5	3.0	3.3
32	6.5	7.7	8.1	8.5	8.5	8.8	8.7	9.2	9.7	7.8	6.8	6.5	7.0	7.7	8.2	7.6	6.5	7.9	10	8.4	7.9	7.2	7.9	8.0
33	6.1	6.4	6.6	7.0	6.6	7.4	7.6	8.5	9.3	7.4	7.1	7.2	7.7	8.3	8.7	8.5	7.5	8.9	10	8.5	8.9	8.0	7.0	7.7
34	5.9	6.4	6.7	6.8	6.4	6.4	7.1	5.4	8.9	8.1	7.4	7.1	8.0	7.6	7.8	7.5	6.6	7.9	8.9	6.2	7.7	8.0	7.3	6.3
35	6.0	4.9	4.0	3.8	4.6	5.4	4.8	4.7	3.8	5.3	6.4	6.2	5.9	6.3	6.9	6.1	7.3	5.9	4.3	6.7	5.3	5.3	5.5	5.0
36	17	16	16	15	16	17	16	17	16	18	18	19	18	19	19	19	20	18	18	20	18	17	16	16
37	30	29	29	29	30	31	30	31	29	30	30	30	28	29	30	30	30	30	30	31	29	28	28	29
38	5.8	6.9	7.3	6.9	6.0	5.2	5.7	6.5	9.2	9.4	9.6	9.7	11	10	9.6	10	9.3	9.9	8.2	6.9	8.5	9.4	8.8	7.2
39	0.8	2.6	4.4	4.4	4.4	2.3	2.2	3.4	4.6	5.3	3.7	3.6	4.7	3.4	3.1	2.9	3.6	4.2	0.9	1.7	1.2	2.9	4.6	4.9
40	3.3	4.4	4.8	4.5	3.0	1.9	2.6	3.3	5.7	5.7	5.9	5.9	7.5	6.5	5.6	6.6	5.7	6.3	2.8	3.9	5.1	5.5	5.3	4.8
41	5.5	6.4	6.5	6.3	4.8	3.6	3.9	5.0	7.3	7.3	7.2	7.6	8.7	8.4	7.4	8.4	7.1	7.9	4.1	6.5	6.5	7.0	6.8	5.9
42	2.3	1.9	1.7	1.8	3.5	4.3	4.3	3.6	2.4	3.1	3.3	3.5	3.0	3.0	4.0	3.0	3.2	3.1	5.4	2.9	2.8	3.2	2.8	3.2
43	2.7	2.1	1.4	1.6	3.5	4.7	4.0	3.4	2.4	2.8	3.2	3.2	2.1	2.9	3.6	2.6	2.7	3.5	5.2	3.0	3.5	3.1	2.2	2.7
44	0.9	0.1	0.0	0.0	1.9	2.9	2.1	1.7	0.9	1.1	1.5	1.5	0.3	0.5	1.1	0.3	1.8	2.6	3.0	1.1	1.8	1.7	0.1	1.2
45	1.9	2.5	2.5	1.9	0.1	1.6	0.5	0.5	1.4	1.1	0.8	1.0	2.0	1.6	1.3	2.1	0.0	0.7	0.1	0.7	1.1	1.3	1.5	0.1
46	0.6	0.8	1.5	1.3	0.5	1.3	0.4	1.2	1.6	0.5	0.5	0.5	2.2	2.4	1.8	2.5	1.3	0.3	0.6	1.8	0.7	0.2	1.3	0.6
47	10	9.5	9.2	9.1	11	11	11	11	11	13	13	13	13	12	13	12	13	14	13	12	13	13	10	10
48	0.3	0.4	0.7	0.9	0.1	0.3	0.3	0.7	1.3	0.3	0.5	0.4	1.6	2.0	1.2	2.1	0.6	0.4	0.9	1.3	0.1	0.1	1.7	0.4
49	2.3	2.7	2.7	3.2	2.2	1.9	2.7	3.1	3.4	2.2	2.6	2.5	3.8	4.2	3.7	4.5	2.6	1.1	3.1	3.4	2.5	1.6	3.7	3.0
50	0.7	0.3	1.5	1.7	0.7	0.5	1.4	1.6	2.1	1.1	1.4	1.0	2.1	2.4	1.8	2.6	0.9	0.9	2.7	2.0	1.3	0.8	2.2	0.6
51	3.0	2.4	3.1	3.6	3.1	3.1	4.1	4.6	4.5	4.3	4.3	3.9	5.2	5.4	5.0	5.3	4.2	4.7	5.9	5.5	4.6	3.6	4.8	2.5
52	8.5	7.8	7.9	9.2	9.4	9.7	10	11	11	10	10	9.4	11	11	11	11	10	11	13	12	11	9.7	10	9.4