Short-Term Daily Load Forecasting in an Intelligent Home with GA-Based Neural Network

S. H. Ling, F. H. F. Leung, H. K. Lam and P. K. S. Tam
Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Abstract – **Daily load forecasting is essential to improve the reliability of the AC power line data network and provide optimal load scheduling in an intelligent home system. In this paper, a short-term daily load forecasting realized by a GA-based neural network is proposed. A neural network with a switch introduced to each link is employed to minimize forecasting errors and forecast the daily load with respect to different day types and weather information. Genetic algorithm (GA) with arithmetic crossover and non-uniform mutation is used to learn the input-output relationships of an application and the optimal network structure. Simulation results on a short-term daily load forecasting in an intelligent home will be given.**

I. INTRODUCTION

Nowadays, homes can have smart features to ensure a high degree of security and comfort. Reliable channels for the communications among electrical appliances and users can be realized. Moreover, with a home network, electrical appliances can be used in an efficient way such that the wastage of energy can be reduced. This paper is based on an intelligent home system [15]. In this system, the AC power line network is used not only for supplying electric power, but also serving as the data communication channel among electrical appliances. With this AC power line data network, a short-term load forecasting can be realized. An accurate load forecasting can bring the following benefits to the intelligent home.

1) Increasing the reliability of AC power line data network. One problem faced by the power line data network is the possible low impedance of the power line in the operating bandwidth [16-17] for data transmission. When the line impedance is too low, the maximum transmission rate will be affected, and the reliability and throughput of the AC power line data network will decrease. The attenuation of the data signal in an AC power line is proportional to the loads connected to it. The reliability of the power line data network can be enhanced if the load is kept at an optimal level through forecasting and balancing. We can also adaptively set a suitable data transmission rate based on the forecasted load condition in order to reduce the overhead owing to retransmissions of data.

2) Optimal load scheduling - At present, the peak demands of electricity are met by operating costly auxiliary generators, or by purchasing power from other utility companies. The cost for supplying peak power is therefore much higher than that for supplying the average power. A reduction in the peak value of electricity demand can be achieved if we can realize load forecasting, and schedule the demands on the utility company accordingly. This has to be supported by batteries installed in the intelligent home that are responsible for sharing the load demand.

Computational intelligence techniques have been applied in daily load forecasting. Artificial neural networks have been considered as very promising tools to short-term load forecasting [18-25]. However, the gradient-descent (GD) algorithm for parameter training of the neural networks suffers from the common problems of convergence to local minima and sensitivity to initial values of the parameters. Global search technique such as Genetic Algorithm (GA) may solve these problems. Genetic algorithm (GA) is a powerful searching algorithm to handle optimization problems [1-3]. It is particularly useful for complex optimization problems with a large number of tuned parameters. Applications of GA can be found in fuzzy control [6-10, 12], path planning [9], greenhouse climate control [10], modeling and classification [11], etc.

Neural network with fixed structure trained by GA for short-term load forecasting was reported in [26]. Other application areas of neural networks include prediction [4-5], system modelling and control [13]. Thanks to its particular structure, neural networks are very good in learning [2] using some learning algorithms such as GA [1] and back propagation [2]. Usually, the structure of a neural network is fixed for a learning process. However, a fixed structure may not provide the best performance within a given training period. If the neural network structure is too complicated, the training period will be long and the implementation cost will be high.

In this paper, a three-layer neural network with a switch introduced in each link is proposed to facilitate the tuning of the optimal network structure. GA with arithmetic crossover and non-uniform mutation [5] is used to help tuning the structure as well as the parameters of the proposed neural network. The proposed neural network is then used to forecast the daily load in an intelligent home. Simulation results will be given to illustrate the performance of the proposed neural network.

This paper is organized as follows. The GA-based neural network with link switches is introduced in section II. A short-term daily load forecasting by using the proposed GA-based neural network is presented in section III. A conclusion will be drawn in section IV.

II. GA-BASED NEURAL NETWORK WITH LINK SWITCHES

In this section, a neural network with link switches is presented. The tuning of the network parameters and structure using GA will also be formulated in this section.

A. Neural Network with Link Switches

The proposed three-layer network is shown in Fig. 1. Specifically, a unit step function is introduced to each link. This unit step function is defined as,

$$\delta(\alpha) = \begin{cases} 0 \text{ if } \alpha < 0 \\ 1 \text{ if } \alpha \geq 0 \end{cases}, \ \alpha \in \Re \tag{1}$$

The introduction of the step function is equivalent to adding a switch to each link of the neural network. Referring to Fig. 1, the input-output relationship of the proposed multiple-input-multiple-output three-layer neural network is given by,

$$y_k(t) = \sum_{j=1}^{n_h} \delta(s_{jk}^2) w_{jk} logsig\left[\sum_{i=1}^{n_{in}} \left(\delta(s_{ij}^1) v_{ij} z_i(t) - \delta(s_j^1) b_j^1\right)\right] - \delta(s_k^2) b_k^2$$
$$, k = 1, 2, \ldots, n_{out} \tag{2}$$

$z_i(t)$, $i = 1, 2, \ldots, n_{in}$, are the inputs which are functions of a variable t; n_{in} denotes the number of inputs; v_{ij}, $i = 1, 2, \ldots, n_{in}$, $j = 1, 2, \ldots, n_h$, denotes the weight of the link between the i–th input and the j-th hidden node. n_h denotes the number of the hidden nodes; s_{ij}^1 denotes the parameter of the link switch from the i-th input to the j-th hidden node. s_{jk}^2, $j = 1, 2, \ldots, n_h$; $k = 1, 2, \ldots, n_{out}$, denotes the parameter of the link switch from the j-th hidden node to the k-th output; n_{out} denotes the number of outputs of the proposed neural network. b_j^1 and b_k^2 denote the biases for the hidden nodes and output nodes respectively. s_j^1 and s_k^2 denote the parameters of the link switches of the biases to the hidden and output layers respectively. $logsig(\cdot)$ denotes the logarithmic sigmoid function:

$$logsig(\alpha) = \frac{1}{1+e^{-\alpha}}, \ \alpha \in \Re \tag{3}$$

$y_k(t)$, $k = 1, 2, \ldots, n_{out}$, is the k-th output of the proposed neural network. The weights v_{ij} and the switch states are to be tuned. In can be seen that the weights of the links govern the input-output relationship of the neural network while the switches of the links govern the structure of the neural network.

B. Tuning

In this section, the proposed neural network is employed to learn the input-output relationship of an application using GA with arithmetic crossover and non-uniform mutation. The input-output relationship is described by,

$$\mathbf{y}^d(t) = \mathbf{g}\left(\mathbf{z}^d(t)\right), t = 1, 2, \ldots, n_d \tag{4}$$

where $\mathbf{y}^d(t) = \left[y_1^d(t) \quad y_2^d(t) \quad \cdots \quad y_{n_{out}}^d(t)\right]$ is the desired output corresponding to the input $\mathbf{z}^d(t) = \left[z_1^d(t) \quad z_2^d(t) \quad \cdots \quad z_{n_{in}}^d(t)\right]$ of an unknown nonlinear function $\mathbf{g}(\cdot)$ respectively. n_d denotes the number of input-output data pairs. The fitness function is defined as,

$$fitness = \frac{1}{1+err} \tag{5}$$

$$err = \frac{1}{n_d}\sum_{t=1}^{n_d}\frac{1}{n_{out}}\sum_{k=1}^{n_{out}}\frac{\left|y_k^d(t) - y_k(t)\right|}{y_k^d(t)} \tag{6}$$

The objective is to minimize of mean absolute percentage error (MAPE) (6) using GA by setting the chromosome to be $\left[s_{jk}^2 \quad w_{jk} \quad s_{ij}^1 \quad v_{ij} \quad s_j^1 \quad b_j^1 \quad s_k^2 \quad b_k^2\right]$ for all i, j, k. The range of (5) is from 0 to 1. A larger value of the fitness function indicates a smaller value of (6).

III. Short-Term Daily Load Forecasting System

The proposed short-term daily load forecasting system in an intelligent home will be discussed in this section. The idea of daily load forecasting is to construct seven multi-input multi-output neural networks, one for each day. Each neural network has 24 outputs representing the expected hourly load for a day. The most important work in the short-term daily load forecasting in an intelligent home is the selection of the input variables. In this forecasting system, there are three main kinds of input variables:

1. Historical loads data: hourly loads were collected and used as historical load inputs. The historical load data reflect the habit of the family on power consumption.
2. Temperature inputs: the average temperature of the previous day and the present day are used as two inputs in this forecasting system. The value of the average temperature of the present day is got from the temperature forecasting of the weather observatory, assuming that their forecasting accuracy is good.
3. Rainfall index inputs: the average rainfall index of the previous day and the present day (again predicted by the weather observatory) are used as two inputs in this forecasting system. The range of the rainfall index is from 0 to 1. 0 represents no rain and 1 represents heavy rain.

A diagram of the daily load forecasting system is shown in Fig. 2. Each of the seven neural networks (for Monday to Sunday) has 28 inputs, 24 outputs with link switches. Among the 28 inputs nodes, the first 24 input nodes (z_1, …, z_{24}) represent the previous 24 hourly loads [21] and is denoted by $z_i = L_i^d(t-1)$,where i = 1, 2, ..., 24. Nodes 25 (z_{25}) and nodes 26 (z_{26}) represent the average temperature of the previous day and present day respectively. Nodes 27 (z_{27}) and Nodes 28 (z_{28}) represent the average rainfall index at previous day and present day respectively. The output layer consists of 24 output nodes that represent the forecasted 24 hourly loads of a day and is denoted by $y_k(t) = L_i(t)$,i = 1, 2, ..., 24. There are two methods of realizing the daily load forecasting system. One is by training the forecasting neural network off-line and the other is by training on-line. The offline training is a time consuming processing. However, once trained, the system can make the forecast quickly (as a lower number of iterations is needed). In this example, we use 12 sets historical data for off-line training. Once trained off-line, the forecasting system operates in an on-line mode, and the weights of neural network will be updated day by day with 500 iterations. From (2), the proposed neural network used for the daily load forecasting is governed by,

$$y_k(t) = \sum_{j=1}^{n_h} \delta(s_{jk}^2) w_{jk} logsig\left[\sum_{i=1}^{28}\left(\delta(s_{ij}^1) v_{ij} z_i(t) - \delta(s_j^1) b_j^1\right)\right] - \delta(s_k^2) b_k^2 \,,$$

k=1, 2, ..., 24. (7)

The number of hidden nodes (n_h) is changed from 14 to 19 in order to compare the learning performance. GA is employed to tune the parameters and structure of the neural network of (7). The fitness function is defined as follows,

$$fitness = \frac{1}{1+err} \tag{8}$$

$$err = \frac{1}{12}\sum_{t=1}^{12}\frac{1}{24}\sum_{k=1}^{24}\frac{\left|y_k^d(t) - y_k(t)\right|}{y_k^d(t)} \tag{9}$$

The objective is to maximize the fitness function of (8). A larger value of the fitness function indicates a smaller value of *err* of (9). The best fitness value is 1 and the worst one is 0. The population size used for the GA is 10. The lower and the upper bounds of the link weights are defined as $\frac{-3}{\sqrt{n_h}} \geq v_{ij}, w_{jk}, b_j^1, b_1^2 \geq \frac{3}{\sqrt{n_h}}$ and, $-1 \geq s_{j1}^2, s_{ij}^1, s_j^1, s_1^2 \geq 1$, i = 1, 2, ..., 3; j = 1, 2, ...,n_h , k = 1, 2, ...,24 [14]. The chromosomes used for the GA are $\left[s_{j1}^2 \quad w_{jk} \quad s_{ij}^1 \quad v_{ij} \quad s_j^1 \quad b_j^1 \quad s_k^2 \quad b_1^2\right]$ for all i, j, k. The initial values of the link weights are set at 0.03. The probability of crossover and mutation are set at 0.8 and 0.03 respectively and the shape parameter of the non-uniform mutation operation is set at 5. The number of the iterations to train the neural network is 2000. The simulation results are tabulated in Table I and Table II. Table I shows the simulation results of daily load forecasting on Wednesday and Table II shows the daily load forecasting on Sunday. These tables show the fitness value, the average training error, the average forecasting error (in term of MAPE) and the percentage of reduction of the number of links of the neural network. Referring to these two tables, the best result is obtained when the number of hidden node is equal to 16. From Table I, n_h=16, the number of link reduced is 111 after learning (the number of links of a fully connected network is 872). It is about 12.73% reduction of the links. The average training error and forecasting error in term of MAPE from Monday to Sunday are shown in Table III. The average errors of training and forecasting are 2.2403 and 2.8677 respectively. Fig.3 and Table IV show the simulation results of the daily load forecasting on Sunday using the proposed network. In Fig.3, the solid line represents the forecasted result and the dashed line is the actual load. Referring to Table IV, we can observe that the percentage error for each hour is within 0 to 3.8% (the average error is 2.3485%).

IV CONCLUSION

By introducing a switch to each link, a neural network that facilitates the tuning of its structure has been proposed. By employing GA with arithmetic crossover and non-uniform mutation, a proposed GA-based neural network is able to learn both the input-output relationship of an application and the optimal network structure. As a result, a given fully connected neural network will become a partly connected neural network after learning. This implies a lower cost of implementation. A short-term daily load forecasting in an intelligent home using the proposed GA-based neural network has been presented. The performance of the proposed network is satisfactory.

ACKNOWLEDGEMENT

The work described in this paper was substantially supported by a Research Grant of The Hong Kong Polytechnic University (project numbers G-V954 and A420).

REFERENCES

[1] J. H. Holland, *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[2] D. T. Pham and D. Karaboga, *Intelligent optimization techniques, genetic algorithms, tabu search, simulated annealing and neural networks*, Springer, 2000.

[3] Z. Michalewicz, *Genetic Algorithm + Data Structures = Evolution Programs*, second, extended edition, Springer-Verlag, 1994.

[4] M. Li, K. Mechrotra, C. Mohan, S. Ranka, "Sunspot Numbers Forecasting Using Neural Network," *5th IEEE International Symposium on Intelligent Control, 1990. Proceedings*, pp.524-528, 1990.

[5] T. J. Cholewo, J. M. Zurada, "Sequential network construction for time series prediction," *International Conference on Neural Networks*, vol.4, pp.2034-2038, 1997.

[6] B. D. Liu, C. Y. Chen and J. Y. Tsao, "Design of adaptive fuzzy logic controller based on linguistic-hedge concepts and genetic algorithms," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 31 no. 1, Feb. 2001, pp. 32-53.

[7] Y. S Zhou and L. Y, Lai "Optimal design for fuzzy controllers by genetic algorithms," *IEEE Trans., Industry Applications*, vol. 36, no. 1, Jan.-Feb. 2000, pp. 93-97.

[8] C. F. Juang, J. Y. Lin and C. T. Lin, "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design," *IEEE Trans., Systems, Man and Cybernetics, Part B*, vol. 30, no. 2, April, 2000, pp. 290–302.

[9] H. Juidette and H. Youlal, "Fuzzy dynamic path planning using genetic algorithms," *Electronics Letters*, vol. 36, no. 4, Feb. 2000, pp. 374-376

[10] R. Caponetto, L. Fortuna, G. Nunnari, L. Occhipinti and M. G. Xibilia, "Soft computing for greenhouse climate control," *IEEE Trans., Fuzzy Systems*, vol. 8, no. 6, Dec. 2000, pp. 753-760.

[11] M. Setnes and H. Roubos, "GA-fuzzy modeling and classification: complexity and performance," *IEEE. Trans, Fuzzy Systems*, vol. 8, no. 5, Oct. 2000, pp. 509–522.

[12] Belarbi, K.; Titel, F., "Genetic algorithm for the design of a class of fuzzy controllers: an alternative approach," *IEEE Trans., Fuzzy Systems*, vol. 8, no. 4, Aug. 2000, pp. 398-405.

[13] M. Brown and C. Harris, *Neuralfuzzy adaptive modeling and control*, Prentice Hall, 1994.

[14] L.F.F. Wessels and E. Barnard, "Avoiding false local minima by proper initialization of connections," *IEEE Trans., Neural Network*, vol. 3, no.6, 1992, pp. 899-905.

[15] L. K. Wong, S. H. Ling, F.H.F. Leung, Y. S. Lee, S. H. Wong, T. H. Lee, H. K. Lam, K.H Ho, D. P. K. Lun, T. C. Hsung, "An intelligent home," in *Proc. Workshop n Service Automation and Robotics, Hong Kong*, June 2000, pp. 111-119.

[16] G. Schickhuber and O. McCarthy, "Control using power lines: a European view," *Computing and Control Engineering Journal*, Aug.1997, pp.180-184.

[17] D. Liu, E. Flint, B. Gaucher, and Y. Kwark, "Wide band AC power line characterization," *IEEE Transactions on Consumer Electronics*, Vol. 45, no.4, Nov.1999, pp. 1087-1097.

[18] K.Y. Lee, Y.T. Cha, J.H. Park, "Short-term load forecasting using an artificial neural network," *IEEE Transaction on Power Systems*, Vol.7, no.1, 1992, pp.124-132.

[19] Y.Y. Hsu, C.C. Yang, "Design of artificial neural networks for short-term load forecasting. Part 1: Self-organizing feature maps for day type identification," *IEE Proceedings-C*, Vol.138, no.5, 1991, pp. 407-418.

[20] I. Drezga, S. Rahman, "Short-term load forecasting with local ANN predictors," *IEEE Transactions on Power System*, Vol. 14, no. 3, Aug. 1999, pp. 844-850.

[21] J.A. Momoh, Y. Wang, M. Elfayoumy, "Artifical neutal network based load forecasting," *IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation*, Vol. 4, 1997, pp. 3443-3451.

[22] D. Part, M. El-Sharkawi, R. Marks, L. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," *IEEE Transactions on Power System*, Vol. 6, no. 2, 1991, pp.442-449.

[23] C.N. Lu, H.T. Wu and S. Vemuri, "Neural network based short-term load forecasting", *IEEE Transaction on Power Systems*, Vol.8, no.1, Feb.1993, pp.336-342.

[24] A.P. Rewagad and V.L. Soanawane, "Artificial neural network based short term load forecasting," *TENCON '98. 1998 IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control*, Vol. 2, 1998, pp.588 -595.

[25] A.G. Bakirtzls, V. Petridls, S.J. Klartzis, M.C. Alexladls, and A.H. Malssls, "A neural network short term load forecasting model for Greed power system", *IEEE Transaction on Power Systems*, vol.11, no.2, May1996, pp.858-863.

[26] E.T.H. Heng, D. Srinivasan and A.C. Liew, "Short term load forecasting using genetic algorithm and neural networks", *1998 International Conference on Energy Management and Power Delivery*, vol. 2, 1998, pp.576-581.

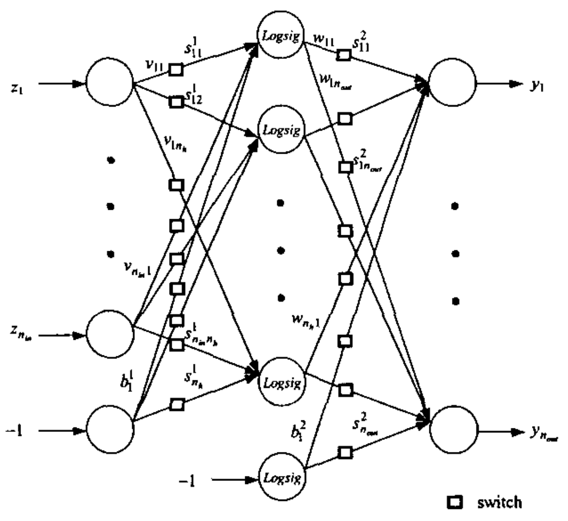

Fig. 1. The proposed 3-layer neural network.

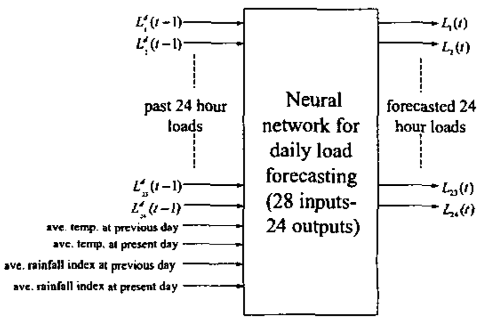
$L^d_1(t-1)$
$L^d_2(t-1)$
past 24 hour loads
$L^d_{23}(t-1)$
$L^d_{24}(t-1)$
ave. temp. at previous day
ave. temp. at present day
ave. rainfall index at previous day
ave. rainfall index at present day
Neural network for daily load forecasting (28 inputs-24 outputs)
$L_1(t)$
$L_2(t)$
forecasted 24 hour loads
$L_{23}(t)$
$L_{24}(t)$

Fig. 2. Neural network for daily load forecasting.

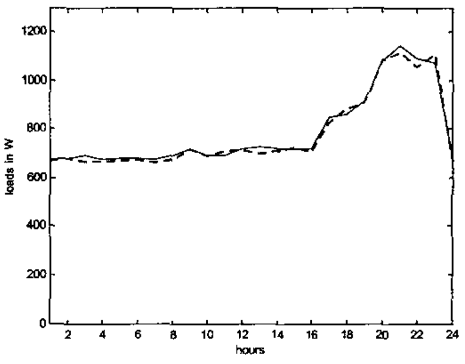

Fig.3. Load forecasting results for Sunday (Week13) given by the proposed network (solid line) and the actual load (dashed line).

n_h	Fitness Values	Ave. training error MAPE(%)	Ave. forecasting error MAPE(%)	Ratio of the no. of links reduced to the total number of links (Percentage of reduction)
14	0.977000	2.3541	2.8217	91/766 (11.88%)
15	0.976537	2.4027	3.1310	121/819 (14.77%)
16	0.981352	1.9002	2.5977	111/872 (12.73%)
17	0.977047	2.3493	2.9259	142/925 (15.35%)
18	0.978904	2.1551	2.8034	139/978 (14.21%)
19	0.969785	3.1156	3.7413	163/1031 (15.81%)

Table I. Learning results of the daily load forecaster for Wednesday.

n_h	Fitness Values	Ave. training error MAPE(%)	Ave. forecasting error MAPE(%)	Ratio of the no. of links reduced to the total number of links (Percentage of reduction)
14	0.969899	3.1035	3.0288	116/766 (15.14%)
15	.0.973920	2.6778	2.6630	127/819 (15.51%)
16	0.977285	2.3243	2.3485	165/852 (18.92%)
17	0.977035	2.3405	2.3621	138/925 (14.92%)
18	0.972765	2.7997	2.8900	186/978 (19.02%)
19	0.974062	2.6629	2.4359	152/1031 (14.74%)

Table II. Learning results of the daily load forecaster for Sunday.

Day type	Ave. training error MAPE(%)	Ave. forecasting error MAPE(%)
Monday	2.1769	2.6963
Tuesday	2.3396	2.9289
Wednesday	1.9002	2.5977
Thursday	2.2937	3.5016
Friday	2.4001	3.2759
Saturday	2.2472	2.7249
Sunday	2.3243	2.3485
Average error:	2.2403	2.8677

Table III. Training error and forecasting error represented in MAPE at different day type (Mon-Sun) with n_d =16.

Sunday Hours	Normalized Loads		% error
Time of the day	Predicted	Desired	Percentage Error (%)
01:00	0.4509	0.4483	0.59
02:00	0.4513	0.4526	0.28
03:00	0.4595	0.4451	3.25
04:00	0.4502	0.4436	1.5
05:00	0.4513	0.4467	1.04
06:00	0.4517	0.4488	0.64
07:00	0.4504	0.4414	2.04
08:00	0.4595	0.4490	2.35
09:00	0.4771	0.4766	0.12
10:00	0.4595	0.4595	0
11:00	0.4595	0.4731	2.87
12:00	0.4769	0.4753	0.32
13:00	0.4822	0.4646	3.8
14:00	0.4783	0.4726	1.21
15:00	0.4773	0.4836	1.3
16:00	0.4772	0.4691	1.74
17:00	0.5637	0.5517	2.18
18:00	0.5714	0.5861	2.52
19:00	0.6067	0.6055	0.2
20:00	0.7218	0.7218	0
21:00	0.7636	0.7451	2.48
22:00	0.7262	0.7043	3.1
23:00	0.7143	0.7393	3.38
24:00	0.4595	0.4463	2.96

Table. IV. Sunday hourly loads: predicted and actual figures.