# One Month Ahead Prediction of Suburban Average Electricity Load

Jelena Milojković and Vančo Litovski

*Abstract*— **One month ahead prediction of suburban average electricity load, based on short time series, is presented. It will be shown here first that for the subject of short term prediction of electricity load, even though a large amount of data may be available, only the most recent of it may be of importance. That gives rise to prediction based on limited amount of data. We here propose implementation of some instances of architectures of artificial neural networks as potential systematic solution of that problem as opposed to heuristics that are in use. To further rise the dependability of the predicted data averaging of two independent predictions is proposed. A specific approach to the choice of the number of hidden neurons will be implemented. Example will be given related to monthly forecasting of the electricity load at suburban level. Prediction is carried out on real data taken from the literature. Prediction errors lower than two percent were obtained.**

*Index Terms*— **forecast, load prediction, electricity, artificial neural networks**

## I. Introduction

In an inspired paper [1] Prof. Mendel' claims: "Prediction of short time series is a topical problem. Cases where the sample length *N* is too small for generating statistically reliable variants of prediction are encountered every so often. This form is characteristic of many applied problems of prediction in marketing, politology, investment planning, and other fields." Further he claims: "Statistical analysis suggests that in order to take carefully into account all components the prediction base period should contain several hundreds of units. For periods of several tens of units, satisfactory predictions can be constructed only for the time series representable as the sum of the trend, seasonal, and random components. What is more, these models must have a very limited number of parameters. Series made up by the sum of the trend and the random component sometimes may be predicted for even a smaller base period. Finally, for a prediction base period smaller than some calculated value *N*min, a more or less satisfactory prediction on the basis of observations is impossible at all, and additional data are required".

Among the fields not mentioned in [1], dealing with really small set of data or "prediction base period", we will discuss here monthly short-term prediction of electricity loads at suburban level or on the level of a low voltage transformer station. In fact, the amount of data available in this case is large enough to apply any other forecasting method [2,3,4] but looking to the load diagram i.e. monthly load-value curves, we easily recognize that past values of the consumption are not very helpful when prediction is considered. That stands even for data from the previous month and for data from the same month in the previous year. Accordingly, we propose the problem of prediction of the load value in the next month to be performed as a deterministic prediction based on very short time series. To help the prediction, however, in an appropriate way, we introduce past values e.g. load for the same month but in previous year. That is in accordance with existing experience claiming that every month in the year has its own general consumption profile [2].

Having all that in mind we undertook a project of developing an artificial neural network (ANN) based method that will be convenient for systematic implementation in stationary time series prediction with reduced set of data. Our first results were applied to prediction of environmental as well as technological data and published in [5, 6, 7]. Analysis as to why neural networks are implemented for prediction may be found in [5]. The main idea implemented was the following. If one wants to create neural network that may be used for forecasting one should enable this property during ANN‘s training. In addition, the ANN used has to have such a structure to accommodate to the training process for prediction.

Following these considerations new forecasting architectures were developed. Namely, prediction is an activity that is always related to uncertainty. One is supposed to have at least two solutions for them to support each other. The structures developed were named Time Controlled Recurrent (TCR) and Feed Forward Accommodated for Prediction (FFAP). Both were implemented successfully for prediction in modern developments in microelectronics [7] as well as in other areas including load predict ion on yearly basis [8].

The goal of this paper is to put the new methods into a broader context of implementation of ANNs for short term forecasting of electricity loads on monthly basis. Namely, the monthly load curve at a suburban (transformer station) level is influenced by several factors the main being the time of the year. Accordingly a predictor is to be capable to approximate two curves concurrently. To meet that we upgraded our original TCR and FFAP ANN structures to accommodate for implementation in the field of short term electricity load forecasting on hourly basis. The results obtained were published in [9] and [10], for feed-forward and for recurrent ANNs, respectively. That ideas will now be implemented for monthly prediction. In addition we here we propose an ave-

Jelena Milojković, ICAT, Niš, Bulevar Nikole Tesle 61, lokal 5, 18000 Niš, (email: jelena.milojkovic@icnt.rs)

Vančo Litovski, Independent researcher, Vojvode Mišića 60/18 Niš, Serbia, (email: vanco@elfak.ni.ac.rs)

raging method that will use both predictions in order to smooth the prediction error so making the final result as dependable as possible. Finally, we propose a method for finding the proper number of hidden neurons in both networks.

The structure of the paper is as follows. After general definitions and statement of the problem we will give a short background related to ANNs application to forecasting. Then we will describe two solutions for possible applications of ANNs aimed to the same forecasting task. Finally short discussion of the results and consideration related to future work will be given.

## II. PROBLEM FORMULATION AND SOLUTION

A time series is a number of observations that are taken consecutively in time. A time series that can be predicted precisely is called deterministic, while a time series that has future elements which can be partly determined using previous values, while the exact values cannot be predicted, is said to be stochastic. We are here addressing only deterministic type of time series.

Consider a scalar time series denoted by $y_i$, $i$=1,2, … $m$. It represents a set of observables of an unknown function , taken at equidistant time instants separated by the interval $\Delta t$ i.e. $t_{i+1}=t_i+\Delta t$. One step ahead forecasting means to find such a function $\hat{y} = \hat{f}(t)$, that will perform the mapping

$$y_{m+1} = f(t_{m+1}) = \hat{y}_{m+1} + \varepsilon , \qquad (1)$$

where $\hat{y}_{m+1}$ is the desired response, with an acceptable error ε.

The prediction of a time series is synonymous with modeling of the underlying physical or social process responsible for its generation. This is the reason of the difficulty of the task. There have been many attempts to find solution to the problem. Among the classical deterministic methods we may mention the $k$-nearest-neighbor [11], in which the data series is searched for situations similar to the current one each time a forecast needs to be made. This method asks for periodicity to be exploited that, as already discussed, here is not of much a help.

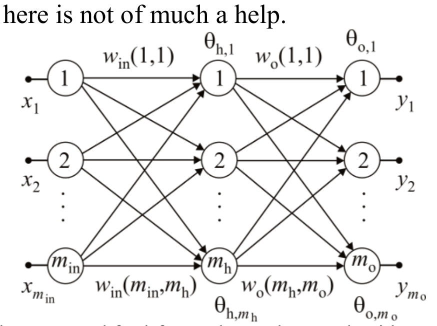

Figure 1. Fully connected feed-forward neural network with one hidden layer and multiple outputs

In the past decades ANNs have emerged as a technology with a great promise for identifying and modeling data patterns that are not easily discernible by traditional methods. A comprehensive review of ANN use in forecasting may be found in [12]. Among the many successful implementations we may mention [13]. A common feature, however, of the existing application is that they ask for a relatively long time series to become effective. Typically it should be not shorter then 50 data points [12]. In the case under consideration it means at least five years backward. This is due to the fact that they all look for periodicity within the data. Very short time series were treated [13]. Here additional "nonsample information" was added to the time series in order to get statistical estimation from deterministic data.

That is why we went for a search for topological structures of ANN that promise prediction based on short time series. In the next, we will first briefly introduce the feed-forward neural networks that will be used as a basic structure for prediction throughout this paper.

The network is depicted in Fig. 1. It has only one hidden layer, which has been proven sufficient for this kind of problem [14]. Indices: "in", "h", and "o", in this figure, stand for input, hidden, and output, respectively. For the set of weights, $w(k,l)$, connecting the input and the hidden layer we have: $k$=1,2,..., $m_{in}$, $l$=1,2,..., $m_h$, while for the set connecting the hidden and output layer we have: $k$=1,2, ...$m_h$, $l$=1,2,..., $m_o$. The thresholds are here denoted as $\theta_{x,m_r}$ where $r$=1,2, …, $m_h$ or $m_o$, with $x$ standing for "h" or "o", depending on the layer. The neurons in the input layer are simply distributing the signals, while those in the hidden layer are activated by a sigmoidal (logistic) function. Finally, the neurons in the output layer are activated by a linear function. The learning algorithm used for training is a version of the steepest-descent minimization algorithm [15]. The number of hidden neurons, $m_h$, is of main concern. To get it we applied a procedure that is based on proceedings given in [16] but here further developed.

In prediction of time series, in our case, a set of observables (samples) is given (approximately every fifteen minutes) meaning that only one input signal is available being the discretized time [17]. To get the average monthly consumption we averaged the data for every month of the year. According to (1) we are predicting one quantity at a time meaning one output is needed, too. The values of the output are numbers (average power for a period of one month). To make the forecasting problem numerically feasible we performed transformation in both the time variable and the response. The time was reduced by $t_0$ so that

$$t=t^*-t_0. \qquad (2)$$

Having in mind that $t^*$ stands for the time (in months), this reduction gives the value of 0 to the time ($t_0$) related to the first sample. The samples are normalized in the following way

$$y=y^* -M \qquad (3)$$

where $y^*$ stands for the current value of the target function, $M$ is a constant (for example $M$=596,8595, being the average monthly consumption for a year).

If the architecture depicted in Fig. 1 was to be implemented (with one input and one output terminal) the following series would be learned: ($t_i$, $f(t_i)$), $i$=1,2,....

Starting with the basic structure of Fig. 1, in [6] possible solutions were investigated and two new architectures were suggested to be the most convenient for the solution of the forecasting problem based on short prediction base period. Here, however, having in mind the availability of data related to previous year, these architectures will be properly accommodated.

The first one, named *time controlled recurrent* (TCR) was inspired by the time delayed recurrent ANN. It is a recurrent architecture with the time as input variable so controlling the predicted value. Our intention was to benefit from both: the generalization property of the ANNs and the success of the recurrent architecture. Its structure is depicted in Fig. 2a. We extend, now, this architecture so that we allow for the values of the power consumption, at a given time per day, but of the same month in the previous year, to control the output.

Hence, the term extended will be added. The resulting architecture is depicted in Fig. 2b. It will be referred from now on to as the Extended Time Controlled Recurrent (ETCR) architecture. Here in fact, the network is learning a set in which the output value representing the average power consumption for a given month in a given year is controlled by the present time and by its own previous instances:

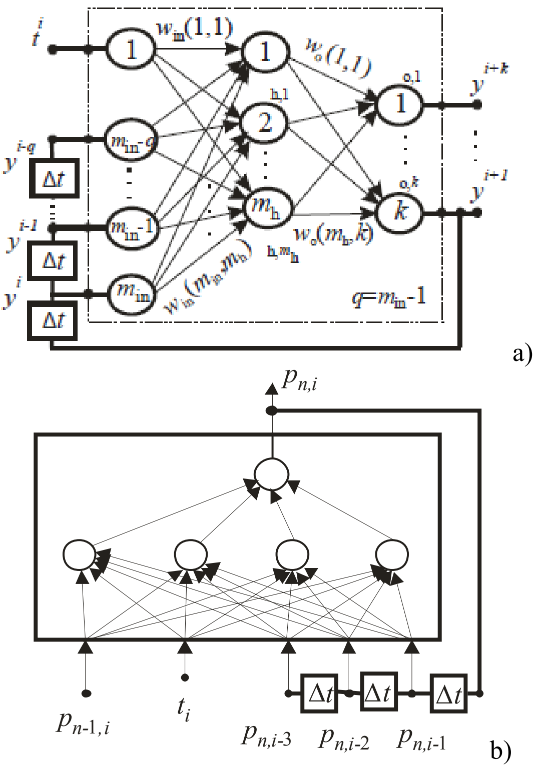


Figure 2. a) Time controlled recurrent ANN and b) ETCR. Extended time controlled recurrent ANN

$$p_{n,i} = f(t_i, p_{n,i-1}, p_{n,i-2}, p_{n,i-3}, p_{n-1,i}) \quad i = 1,2,3... \quad (4)$$

Here $n$ stand for the number of the month (in the year). In that way the values indexed with $n$ are from the actual year, while the value indexed $n$-1 is from the previous year. $i$ stands for the $i$-th sample in the year selected. The actual value $p_{n,i}$ is unknown and should be predicted. Incrementing $i$, in fact, means moving the prediction window one step ahead. These quantities are illustrated in Fig. 3. It represents the load curve for two years. Note the x-axis is reduced to the first month available while the y-axis is reduced by the average monthly value of the load.

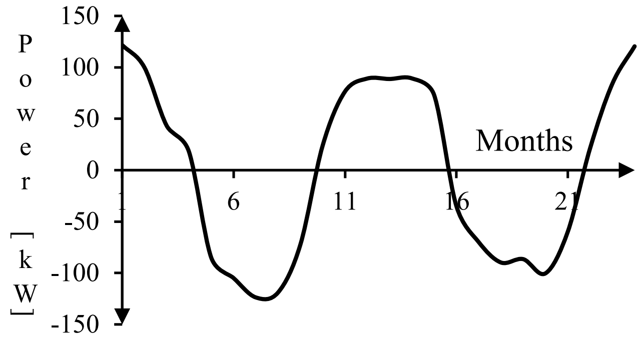


Fig. 3. Average power (reduced by 596,8595) versus time (months)

The second structure was named *feed forward accommodated for prediction* (FFAP) and depicted in Fig. 4a. Our idea was here to force the neural network to learn the same mapping several times simultaneously but shifted in time. In that way, we suppose, the previous responses of the function will have larger influence on the $f(t)$ mapping.

In this architecture there is one input terminal that, in our case, is $t_i$. The *Output*3 terminal, or the *future* terminal, in our case, is to be forced to approximate $y_{i+1}$. In cases where multiple-step prediction is planned *Output*3 may be seen as a vector. *Output*2 should represents the *present* value i.e. $y_i$. Finally, *Output*1 should learn the *past* value i.e. $y_{i-1}$. Again, if one wants to control the mapping by a *set* of previous values, *Output*1 may be seen as a vector.

As an example we may express the functionality of the network as

$$\{y_{i+1}, y_i, y_{i-1}, y_{i-2}\} = f(t_i), \quad i=3,4,... \quad (5)$$

where *Output*1=$\{y_{i+1}, y_i, y_{i-1}, y_{i-2}\}$, meaning that: one future ($i$+1), one present ($i$), and two previous ($i$-1, $i$-2) responses are to be learned.

It is our experience that the FFAP architectures produce better results than the TCR. Nevertheless, we regularly implement both of them and use the results obtained as reference to each other when choosing the forecast that makes most sense. That allows avoidance of solutions that represent local minima in the optimization process representing the training of the ANN.

In the case of hourly prediction of power consumption we extended the FFAP architecture exactly in the same way as we did with the TCR. In that way for the approximation function we may write the following

$$\{p_{n,i+1}, p_{n,i}, p_{n,i-1}, p_{n,i-2}\} = f(t_i, p_{n-1,i}\} \quad i=1.2.3... \quad (6)$$

The new network is approximating the future (unknown) value $p_{n,i+1}$, based on the actual time $t_i$, the actual consumption $p_{n,i}$, the past consumption values for the given year ($p_{n,i-k}$, $k$=1,2), and the past consumption values for the same month at the actual time of the previous year ($p_{n-1,i}$). The new architecture is referred to as extended feed forward accommodated for predict ion (EFFAP). It is depicted in Fig. 4b.

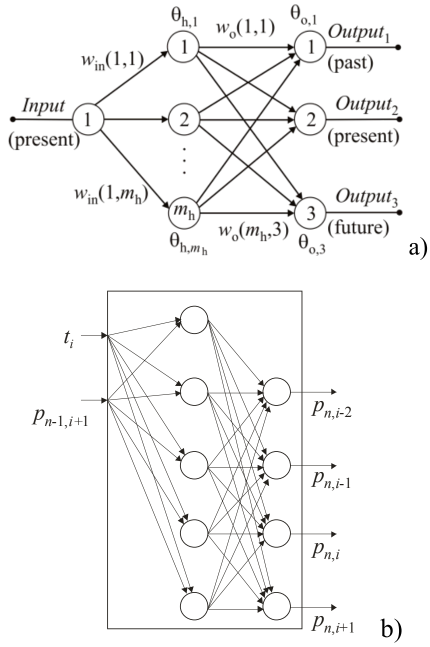


Figure 4. a) Feed forward ANN structure accommodated for predict ion (FFAP), and b) The Extended feed forward accommodated for predict ion ANN (EFFAP) according to (6)

In the next the procedure of implementation of ETCR and EFFAP network will be described. It consists of the following steps.

STEP 1. For a given month ($i$th month) a training table is constructed for both ANN structures. These constructs are illustrated in Table I and Table II, for the ETCR and EFFAP network, respectively, for $i$=18.

STEP 2. Both network are repeatedly trained with the same training data but with increased complexity i.e. with increased number of hidden neurons. We start with $m_h$=3 and end with $m_h$=10. The number of neurons is chosen to be "small" since the problem under consideration is not a difficult one. One is not to forget that an ETCR ANN, like the one depicted in Fig. 2, having 10 hidden neurons, will have 70 free parameters which is much above the need to approximate the curve given in Fig. 3.

STEP 3. To find the proper ETCR and EFFAP number of hidden neurons, the predicted values are compared. Namely, we consider the prediction as a step in darkness and to get an authentic prediction, we think, one needs at least two solutions supporting each other (The well known medical "second opinion"). In that way we choose two among the eight ETCR and eight EFFAP solutions (each from a kind) that are the most similar.

4. Since the ETCR and the EFFAP solutions just chosen are of the same importance, as the final result, we adopt their average.
5. Then we proceed to the next month

## III. Implementation Example

The diagram depicted in Fig. 3 is created from the UNITE competition data [17]. Since there are data for two years only we created 24 instances as shown in Fig. 3. Having in mind, however that our method asks for a value of the load for the same month in the previous year, the first 12 instances are to be reserved. Furthermore, to start the prediction we need some values of the previous months. For these reasons we started the prediction with the fourth part of the data i.e. from the 19th month.

Table I and Table II are examples of the training set for the first prediction. The rest of the training set is obtained by "sliding" down the table of the load as a function of the month.

TABLE I One training session for ETCR

| Inputs | | | | | Outputs |
|---|---|---|---|---|---|
| $t_n$ | $p_{n-1,i}$ | $p_{n-2,i}$ | $p_{n-3,i}$ | $p_{n,i-1}$ | $p_{n,i}$ |
| 13 | 88.9537 | 76.16484 | 23.58744 | 121.6963 | 88.54376 |
| 14 | 88.54376 | 88.9537 | 76.16484 | 99.9508 | 89.14276 |
| 15 | 89.14276 | 88.54376 | 88.9537 | 43.22303 | 73.22104 |
| 16 | 73.22104 | 89.14276 | 88.54376 | 18.00998 | -34.8074 |
| 17 | -34.8074 | 73.22104 | 89.14276 | -85.0241 | -69.965 |
| 18 | -69.965 | -34.8074 | 73.22104 | -104.965 | -89.8928 |
| 19 | -89.8928 | -69.965 | -34.8074 | -123.849 | ? $=p_{n,19}$ |

TABLE II One training session for EFFAP

| Inputs | | Outputs | | | |
|---|---|---|---|---|---|
| $t_i$ | $p_{n-1,i}$ | $p_{n,i-2}$ | $p_{n,i-1}$ | $p_{n,i}$ | $p_{n,i+1}$ |
| 12 | 121.6963 | 23.58744 | 76.16484 | 88.9537 | 88.54376 |
| 13 | 99.9508 | 76.16484 | 88.9537 | 88.54376 | 89.14276 |
| 14 | 43.22303 | 88.9537 | 88.54376 | 89.14276 | 73.22104 |
| 15 | 18.00998 | 88.54376 | 89.14276 | 73.22104 | -34.8074 |
| 16 | -85.0241 | 89.14276 | 73.22104 | -34.8074 | -69.965 |
| 17 | -104.965 | 73.22104 | -34.8074 | -69.965 | -89.8928 |
| 18 | -123.849 | ? | ? | ? | ?$=p_{n,19}$ |

TABLE III The most similar ETCR and EFFAP solutions on transformed input data

| $t_n$ | ECTR | | EFFAP | | Average | Expected |
|---|---|---|---|---|---|---|
| | $m_h$ | $p$ | $m_h$ | $p$ | ($p$) | |
| 19 | 3 | -95.5011 | 4 | -83.9625 | -89.7318 | -86.6497 |
| 20 | 9 | -94.309 | 3 | -103.751 | -99.03 | -100.462 |
| 21 | 7 | -85.2832 | 7 | -33.3583 | -59.3208 | -60.1296 |
| 22 | 4 | 2.09448 | 7 | 17.3844 | 9.73944 | 20.94362 |
| 23 | 4 | 87.2713 | 7 | 84.9242 | 86.097 | 83.82734 |
| 24 | 4 | 100.231 | 3 | 122.701 | 111.466 | 120.2991 |

TABLE IV The most similar ETCR and EFFAP solutions on restored original input data

| $t_n$ | ECTR | | EFFAP | | Average | Expected |
|---|---|---|---|---|---|---|
| | $m_h$ | $p$ | $m_h$ | $p$ | ($p$) | |
| 19 | 3 | 501,358 | 4 | 512,897 | 507,128 | 510,2098 |
| 20 | 9 | 502,551 | 3 | 493,109 | 497,830 | 496,3975 |
| 21 | 7 | 511,576 | 7 | 563,501 | 537,539 | 536,7299 |
| 22 | 4 | 598,954 | 7 | 614,244 | 606,600 | 617,8031 |
| 23 | 4 | 684,131 | 7 | 681,784 | 682,957 | 680,6868 |
| 24 | 4 | 697,091 | 3 | 719,561 | 708,326 | 717,1586 |

The results of STEP 3 described in the previous paragraph, Table III was created. While its content is selfexplainable we will here stress again that among the predictions for a given month, the two most similar were saugth. So, for example, for the twentisecond month the prediction of the ETCR ANN built by four hidden neurons and the EFFAP ANN built by seven neurons were the most similar ones. These two were chosen and the average calculated.

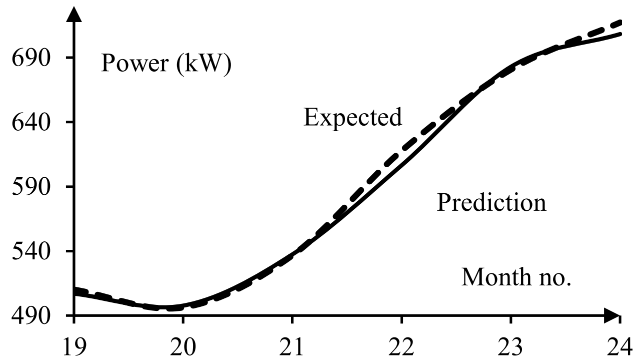


Fig. 5. Visualization of the last two columns of Table IV

To complete the prediction the values produced by (3) were to be restored . That practicaly meant that all entries of Table III were to be incremented by 596,8595. In that way Table IV was created. Fig. 5 depicts the two last columns of Table IV. Namely the expected and the predicted values are drawn together.

TABLE V PREDICTION ERROR

| $t_n$ | Error (%) ECTR | Error (%) EFFAP | Error (%) Average |
|---|---|---|---|
| 19 | 1,735 | -0,5267 | 0,604 |
| 20 | -1,240 | 0,6625 | -0,289 |
| 21 | 4,687 | -4,988 | -0,151 |
| 22 | 3,051 | 0,576 | 1,813 |
| 23 | -0,506 | -0,161 | -0,334 |
| 24 | 2,798 | -0,335 | 1,232 |

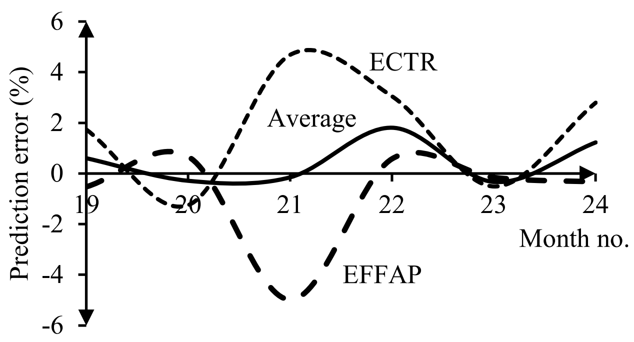


Figure 6. Prediction error (in %) of the STCR, EFFAP and the averaged solution (Graphical depiction of Table V)

Finally, in order to get even better insight into the results, the prediction error was calculated and depicted in Table V. As can be seen the error of the average value compared with the expected one is less than 2% in all six cases. A graphical representation of Table V is given in Fig. 6.

It is interesting to note that the prediction errors of the ETCR and the EFFAP ANNs are much larger (less than 6%). That means that the worst prediction would never exceed that value. By good luck, however, in this case, cancellation occurred during the computation of the average which led to an extraordinary good result.

## IV. CONCLUSION

One month ahead prediction of suburban average electricity load, based on short time series, was presented. It was shown first that for the subject of short term prediction of electricity load, even though a large amount of data may be available, only the most recent of it may be of importance. That gives rise to prediction based on limited amount of data. We here proposed implementation of some instances of architectures of artificial neural networks as potential systematic solution of that problem as opposed to heuristics that are in use. To further rise the dependability of the predicted data averaging of two independent predictions was proposed. A specific approach to the choice of the number of hidden neurons was implemented. Example was given related to monthly forecasting of the electricity load at suburban level. Prediction was carried out on real data taken the literature. Prediction errors lower than two percent were obtained.

## ACKNOWLEDGMENT

This research was partially supported by the Ministry of Education Science and technological Development of Serbia within the project TR32004.

## REFERENCES

[1] A. S., Mandel', "Method of Analogs in Prediction of Short Time Series: An Expert-statistical Approach", *Automation and Remote Control*, Vol. 65, No. 4, April 2004, pp. 634-641.
[2] P., Murto, "Neural Network Models for Short -Term Load Fore-casting", M S Thesis, Helsinki University of Technology, 1998.
[3] F., Cavallaro, "Electric load analysis using an artificial neural network", *Int. J. of Energy Research*, Vo l. 29, 2005, pp. 377–392.
[4] H., Hahn, S., Meyer-Nieberg, and S., Pickl, "Electric load fore-casting methods: Tools for decision making", *European J. of Operational Research*, Elsevier, Vol. 199, 2009, pp. 902–907.
[5] J. Milojković, V. B. Litovski, "New methods of prediction implemented for sustainable development", Proc. of the 51th Conf. ETRAN, Herceg Novi, Monte Negro, June 2007, Paper no. EL1.8 (in Serbian).
[6] J. Milojković, V. B. Litovski, "Comparison of some ANN based forecasting methods implemented on short time series", Proc. of the 9th Symp. NEUREL-2008, Belgrade, ISBN 978-1- 4244-2903-5, Sept. 2008, pp. 175-178.
[7] J. Milojković, V. B. Litovski, "Short term forecasting in Electronics", *Int. J. of Electronics*, Vol. 98, No. 2, 2011, pp. 161-172.
[8] J. Milojković, V. B. Litovski, O., Nieto-Taladriz, and S., Bojanić, "Forecasting Based on Short Time Series Using ANNs and Grey Theory – Some Basic Comparisons", In Proc. of the 11th Int. Work-Conf. on Artificial Neural Networks, IWANN 2011, June 2011, Torremolinos-Málaga (Spain). J. Cabestany, I. Rojas, and G. Joya (Eds.): Part I, LNCS 6691, pp. 183–190, 2011, © Springer-Verlag, Berlin, Heidelberg.
[9] J. Milojković, V. B. Litovski, „Dynamic Short-Term Fore-casting Of Electricity Load Using Feed-Forward ANNs", *Int. J. of Engineering Intelligent Systems for Electrical Engineering and Communication*, Vol. 17, No. 1, March 2009, pp. 38-48.

[10] J. Milojković, V. B. Litovski, "Short -term Forecasting of Electricity Load Using Recurrent ANNs", "*Electronics*", *ISSN*: 1450-5843, Vol. 14, No. 1, June 2010, pp. 45-49.

[11] E.A., Plummer, "Time series forecasting with feed-forward neural networks: guidelines and limitations", M.S. Thesis, University of Wyoming, Laramie, USA, July 2000.

[12] B.G., Zhang, "Forecasting with artificial neural networks: The state of the art", *Int. J. of Forecasting*, Vol. 14, No. 1, March 1998, pp. 35-62

[13] K., Brännäs, and J., Hellström, "Forecasting based on Very Small Samples and Additional Non-Sample Information", Umeǻ Economic Studies 472, Umeǻ University , Sweden, 1998

[14] T., Masters, "*Practical Neural Network Recipes in C++*", Academic Press, San Diego, 1993.

[15] Z., Zografski, "A novel machine learning algorithm and its use in modeling and simulation of dynamical systems", in Proc. of 5th Annual European Computer Conf., COMPEURO '91, Hamburg, Germany, 1991, pp. 860-864.

[16] E.B., Baum, and D., Haussler, "What size net gives valid generalization", *Neural Computing*, 1989, Vol. 1, pp. 151-160.

[17] -,World-wide competition within the EUNITE network, http://neuron.tuke.sk/competition.