# SHORT TERM LOAD FORECASTING WITH MULTILAYER PERCEPTRON AND RECURRENT NEURAL NETWORKS

**Muhammad Riaz Khan — Čestmír Ondrůšek** *

The ability of the Elman recurrent neural network (RNN) to model the short-term load forecasting (STLF) problem is investigated in this paper. Its performance in a competition is then contrasted with that of a multilayer perceptron (MLP) network. It is postulated that the load can be modeled as the output of some dynamic system, influenced by a number of weather, time and other environmental variables. RNN exhibiting inherent dynamic behavior can thus be used to construct a forecasting model for this dynamic system. Due to a nonlinear dynamic nature of this model, the behavior of the load prediction system can be captured in a compact and robust representation. This is illustrated by the performance of the Elman RNN model for the short-term forecasting of the nation-wide load for the Czech Electric Power Utility (ČEZ). Both techniques have been trained and tested on the data provided by ČEZ and promising results have been obtained.

K e y w o r d s: Short-term load forecasting, multilayer perceptron, Elman recurrent neural network, and backpropagation learning algorithm.

## 1 INTRODUCTION

Short-term load forecasting (STLF) plays a major role in economic optimization and reliable operation of electric utility companies. It is a vital tool to predict power system loads. Particularly, STLF has an influence on maintenance planning, economic operation of generating units and transmission systems, energy transfer scheduling and peak power supply. Because of the shrinking spinning reserve and the seasonal high-energy demand, utilities are dependent more than ever on short-term forecasting for daily energy transactions.

Accuracy in forecasting can provide lower operational costs, which can lead to savings passed onto the customers. The accuracy of the forecasted load influences decision-making in unit commitment, load dispatch, hydrothermal coordination, fuel allocation, maintenance and off-line network analysis.

The dependence of the load on several changeable factors, such as weather, social events and other random factors, make load forecasting a challenging job. Conventional STLF models can not adapt easily to rapid changes of the load variation pattern, they have deficiencies, especially in geographically diverse areas and with abrupt changes in environmental or sociological variables. Recent progress in the application of artificial neural network (ANN) to load forecasting problem provides a potential technique to deal with the above difficulties. ANN has a key feature in not relying on the explicitly expressed relationship between input variables and forecasted load. When using ANN for load forecasting, one needs only to consider the selection of variables as the network-input variables. A training process will formulate the relationship between the input variables and the predicted load.

Different ANN architectures are reported for load forecasting problem, selected ones are Multi Layer Perceptron (MLP), Not Fully Connected MLP, Recurrent, Functional Links and Kohonen Maps as shown in Fig. 1. Currently, dynamic and recurrent architectures constitute also an interesting research area. Some of the papers [1–7] report applications of these models to load forecasting problem. Incorporating feedback connections into an MLP network results in significant changes in the operation and learning processes of the networks as compared to their static counterparts. They have an increased computational power over the conventional MLP networks. RNN can perform mappings that are functions of time and/or space or converge to one of a number of limit points. As a result, they are capable of performing more complex computations than static feedforward networks. For example, they are capable of learning temporal pattern sequences, *ie*, sequences of patterns that are context or time dependent.

Neural networks typically exhibit two types of behavior. If no feedback loops connect neurons, the signal produced by an external input moves in only one direction and the output of the network is just the output of the last group of neurons in the network. In this case, the network behaves mathematically like a nonlinear function of the inputs. This feedforward type of network is most often used in time series forecasting, with past time series values as the inputs and the desired future value as the output. The second type of network behavior is observed when there are feedback loops in the neuron connections. In this case, the network behaves like a dynamical system, so the outputs of the neurons vary with time [8]. The neuron outputs can then oscillate, or settle down into steady

* Department of Power Electrical and Electronic Engineering, Faculty of Electrical Engineering and Computer Science, Brno University of Technology, Technická 8, 616 00, Brno, Czech Republic, e-mail: khan@pti.sk.ca

ISSN 1335-3632 

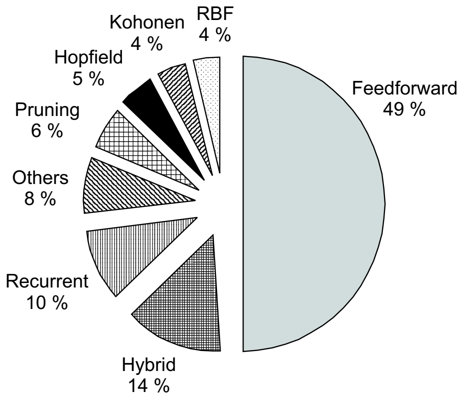


**Fig. 1.** Percentage of using different types of ANN for load forecasting.

state values, or, since the threshold function introduces nonlinearity into the system, they can become chaotic.

## 2 LOAD PATTERN

The power load variations occurring during each weekly cycle are attributed to such factors as:

- *The day of the week*: The load characteristics on Saturday and Sunday are different from the usual weekdays, which is attributed to the fact that most business and industries are closed over the weekend, thus giving rise to an overall lower load demand. Weekend effect is also seen in the first part of Monday, which varies from regular weekdays as shown in Fig. 2.
- *The time of the day*: Power consumption during the night is much lower than at day time; furthermore, power consumption during the day-time varies with the time of the day. For example, the morning rush hour has a different load demand than the lunch hour or the afternoon period.

The load variations over a year follow the same pattern as shown in Fig. 3, with differences seen in the maximum peak load that can be attributed to factors such as:

- *Weather factors*: The variations in weather parameters such as temperature, relative humidity, wind speed, visibility, illumination level, thunderstorms, and cloud coverage have a major effect on the behavior of the load consumption throughout the year. The most important of the weather parameters are the temperature variables, representing the strongest correlation with weather-related load variations. Temperatures, in general, can be measured to a higher degree of accuracy relative to any of the other weather variables. Weather sensitive load, as the name implies, is caused primarily by temperature, and it is the amount of load demand which is generated at each temperature range. As we expect, the weather sensitive load is large at the two extremes of temperature, and reaches its minimum when the temperature is within a comfortable range. Therefore, the weather sensitive load is a structure that models how a certain area reacts to changes in temperature in terms of load demand. On the other hand, the reaction to temperature changes is not the same throughout the year.
- *Holiday effects*: The load of public holidays, which are resting/relaxing days for most people, are lower than regular weekdays. Such days are identified and presented separately in the forecasted model.

All of these factors make the STLF problem highly nonlinear with an infinite number of modeling characteristics and model data input combinations.

A broad spectrum of other factors affect the system's load level, such as human activities, seasonal variations, changes in industrial activities and economic growth, load management, pricing strategy and electricity tariff structures, shifts to day light-saving time, and vacation periods. In addition, the total system load is subjected to random disturbances caused by a sudden increase of large loads or outages and special events [9]. Thus the load

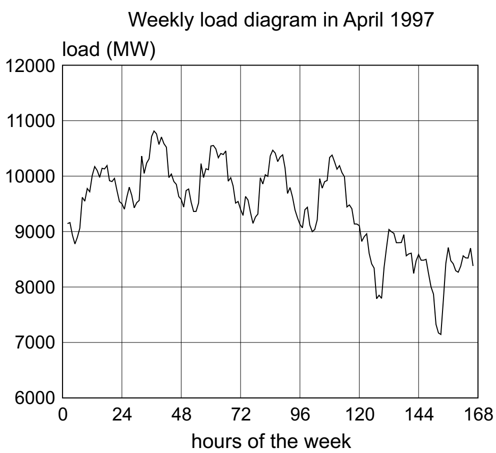


**Fig. 2.** Weekly electric load diagram.

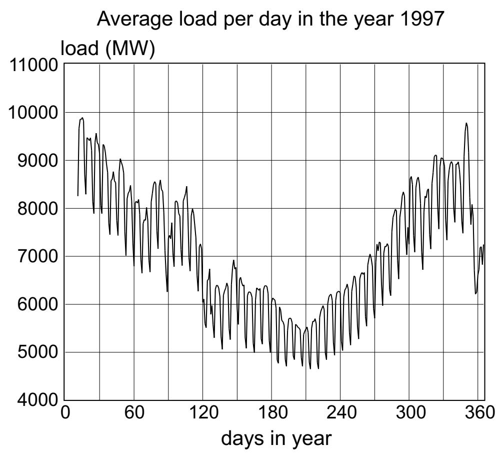


**Fig. 3.** Average daily electric load diagram for the year 1997.

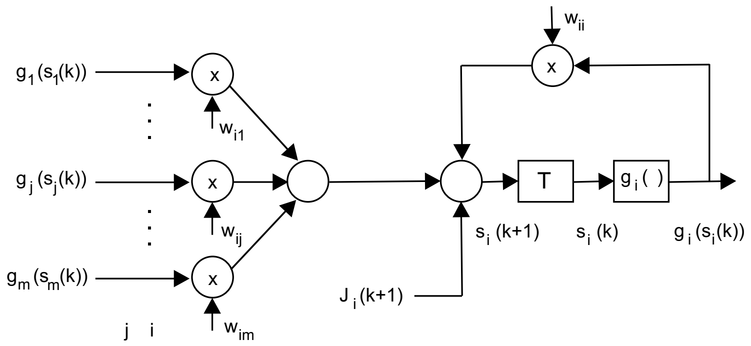


**Fig. 4.** Structure of the individual neuron for the discrete-time, fully connected RNN.

profile is dynamic in nature with temporal, seasonal and annual variations.

It is evident that the nature of the load is dynamic, rather than static. The change in the load is not only influenced by the external weather and time variables, but is also highly dependent on the past and current load states. Recurrent neural networks, being members of a class of neural network models exhibiting inherent dynamic behavior, can thus be used to construct empirical models for the load as a dynamic system [7]. Because of the nonlinear nature of these models, the behavior of the load prediction system can be captured in a compact, robust and more natural representation.

## 3 TRAINING OF MLP NETWORK

In this paper, the error backpropagation algorithm is used to train the MLP network. Presenting an input pattern to the network produces an output vector. According to the difference between the produced and target outputs, the network's weights ($W_{ij}$) are adjusted to reduce the output error. The error at the output layer propagates backward to the hidden layer, until it reaches the input layer. This algorithm is also known as "Generalized Delta Rule (GDR)".

The output from neuron $i$, '$O_i$' is connected to the input of neuron $j$ through the inter- connection weight '$W_{ij}$'. Unless neuron $k$ is one of the input neurons, the state of neuron $k$ is given as

$$O_k = f\Big(\sum\nolimits_i W_{ik} O_i\Big) \tag{1}$$

where $f(x) = 1/\big(1+e^{-x}\big)$, and the sum is over all neurons in the adjacent layer. Let the target state of the output neuron be $t$. Thus, the error at the output neuron can be defined as

$$E = \frac{1}{2}\,(t_k - O_k)^2 \tag{2}$$

where neuron $k$ is the output neuron. The gradient descent algorithm adapts the weights according to the gradient error, *ie*,

$$\Delta W_{ij} \propto -\frac{\partial E}{\partial W_{ij}} = -\frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial W_{ij}}\,. \tag{3}$$

Specifically, we define the error signal as

$$\delta_j = -\frac{\partial E}{\partial O_j}\,. \tag{4}$$

With some manipulation, we can get the following GDR as

$$\Delta W_{ij} = \epsilon\,\delta_j O_i \tag{5}$$

where $\epsilon$ is an adaptation gain. $\delta_j$ is computed based on whether or not neuron $j$ is in the output layer. If neuron $j$ is one of the output neurons, then

$$\delta_j = \big(t - O_j\big)O_j\big(1 - O_j\big)\,. \tag{6}$$

If neuron $j$ is not in the output layer, then

$$\delta_j = O_j\big(1 - O_j\big)\sum\nolimits_k \delta_k W_{jk}\,. \tag{7}$$

In order to improve the convergence characteristics, we can introduce a momentum term with momentum gain $\alpha$ to Eq. 5, thus

$$\Delta W_{ij}(n+1) = \epsilon\delta_j O_j + \alpha\Delta W_{ij}(n) \tag{8}$$

where $n$ represents the iteration index.

Once the neural network is trained, it produces a very fast output for given input data. It only requires a few multiplications, additions, and calculations of sigmoid function [9].

The training process is a vital part to build an intelligent system capable of predicting the hourly load for one day-ahead precisely. The error back propagation algorithm was utilized for training the proposed network in order to get a well-trained network. The following steps were taken before starting the training process:

- The error level was set to a relatively small value ($10^{-4}$) that could be decreased to a smaller level, but the results show satisfactory predicted loads. Also setting the training accuracy to a higher level will take a much longer training time.
- The hidden neurons were varied from 10 to 80. The optimal number of 24 neurons in the hidden layer were obtained experimentally by changing the network design and running the training process several times until a good performance was obtained.

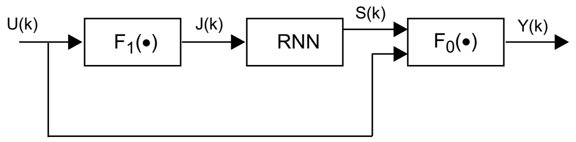


**Fig. 5.** General description of the recurrent neural network system.

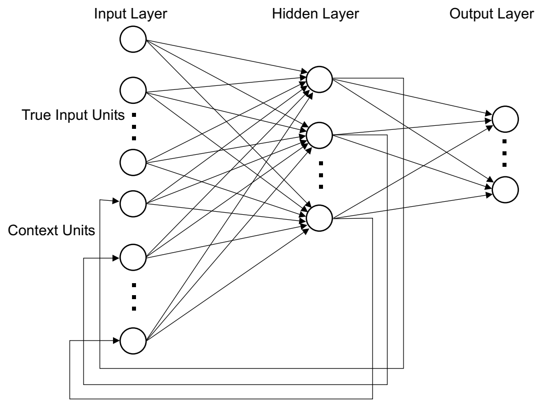


**Fig. 6.** An Elman recurrent neural network.

- When the network faces local minima (false wells), new ones to escape from such false wells replace the whole set of network weights and thresholds. Actually, a random number generator was used to assign the initial values of weights and thresholds with a small bias as a difference between each weight connecting two neurons together since similar weights for different connections may lead to a network that will never learn.
- Our efforts were devoted to provide the network with a sufficient number of training patterns containing different load features for the day under study to enhance the network experience. Using the historical load and temperature data collected from (ČEZ) for the years 1994 through 1996 were fed to the network for training purpose.

## 4 RECURRENT NEURAL NETWORK (RNN)

Recurrent neural networks are members of a class of neural network models exhibiting inherent dynamic behavior. Each neuron $i$ is connected to every other neuron $j$, including itself, via the appropriate weights ($W_{ij}$). The recursive equation describing the dynamic behavior of the neuron state can be derived from the construction of the neuron in Fig. 4 to be

$$S_i(k+1)=\sum_{i=1}^{m} W_{ij} g_j(S_j(k))+J_i(k+1)\,,\ \ i=1,2,\ldots,m\,. \tag{9}$$

where $m$ is the number of neurons in the network, $k$ is the discretized time index, $S_i(k)$ is the state of neuron $i$ at time step $k$, $J_i(k)$ is the input to neuron $i$ at time step $k$, $W_{ij}$ is the connection weight from neuron $j$ to neuron $i$, and $g_i(\cdot)$ is the neuron activation function of the $i$th neuron. The most common of these are the linear and sigmoidal transfer functions. In vector notation form, Eq. (9) becomes

$$S(k+1)=W_g(S(k))+J(k+1)\,,\ \ k_o\le k\le k_f \tag{10}$$

where $[k_o,k_f]$ is the simulation range of interest.

To generalize the recurrent neural network description, input and output equations can be defined as

$$J(k)=F_l[U(k)] \tag{11}$$

$$Y(k)=F_o[S(k),U(k)] \tag{12}$$

respectively, where $U(k)\in\mathcal{R}^p$ is the external input to the system, and $Y(k)\in\mathcal{R}^n$ is the output of the system, $p$, $n\le m$. This generalization results in the recurrent neural network system as shown in Fig. 5, the dynamic behavior of which is uniquely determined by Eqs. (10), (11), (12) and $Y(k_o)$, the initial output of the system (this is true because the initial state of the recurrent neural network, $S(k_o)$, can be obtained from $Y(k_o)$ and the initial input, $U(k_o)$, by the inverse of Eq. (12).

### 4.1. Elman Recurrent Neural Networks

The feedforward multilayer perceptron (MLP) network is used frequently in time series prediction. MLP network, however, has a major limitation that it can only learn an input-output mapping which is static. Thus, it can be used to perform a nonlinear prediction of a stationary time series. A time series is said to be stationary, when its statistics does not change with time. In many real world problems, however, the time when certain feature in the data appears contains important information. More specifically, the interpretation of a feature in data may depend strongly on the earlier features and its appearance time.

Neural networks must contain memory in order to process temporal information. There are two basic ways to build memory into the neural network [11]. The first one is to introduce time delays in the network and to adjust their parameters during learning phase. The second way is using a positive feedback, which is making the network recurrent.

The Elman network, which is known as partial recurrent network or simple recurrent network (SRN) is shown in Fig. 6. In an SRN, the outputs of the hidden layer are allowed to feedback onto itself through a buffer or "context" layer. These are the only feedback connections in the network and the weights from the hidden layer to the context layer are constant values. All other connections are feedforward with adjustable weights. Although simple in structure, this network is capable of learning to perform powerful tasks. The Elman network has a large depth, low resolution memory, since the context units

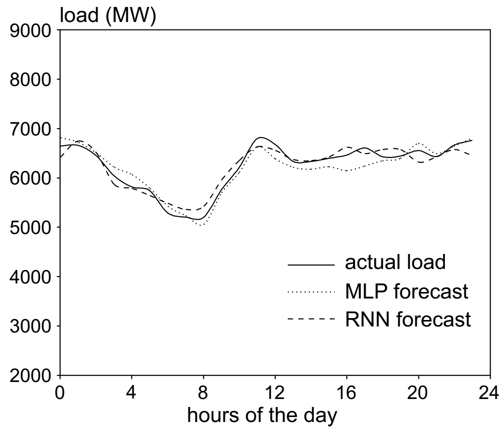


**Fig. 7.** Actual, and recurrent and feedforward forecasted load profiles for $24^{\text{th}}$ December,1997 (Christmas day).

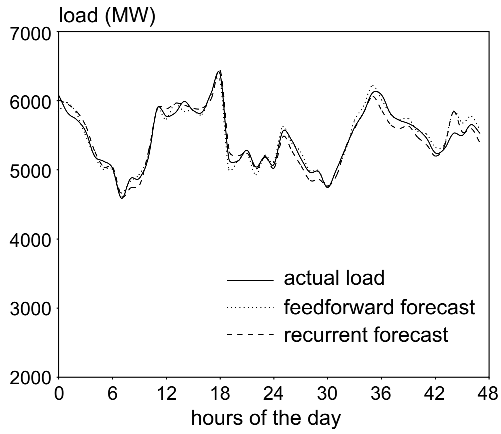


**Fig. 8.** Actual, and recurrent and feedforward forecasted load profiles for weekend (Saturday and Sunday) in $1^{\text{st}}$ week of November 1997.

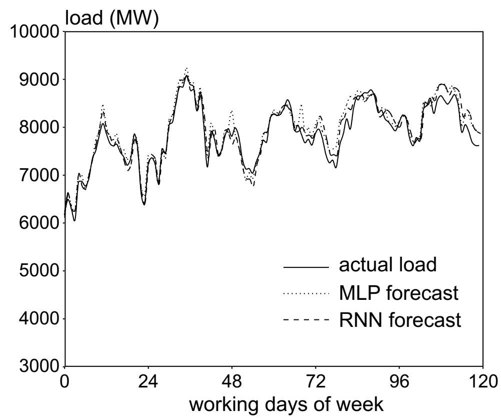


**Fig. 9.** Actual, and recurrent and feedforward forecasted load profiles for working days of a week in November 1997.

keep an exponentially decreasing trace of the past hidden neuron output values. In this network, signals are processed in two time steps. During the first step at time $t-1$, signals from the input and context layers, which are fully connected to the hidden layer, are distributed to the hidden-layer units. The pattern of activation outputs from the hidden layer are then computed and passed onto the output layer for processing at time $t$. At the same time, the hidden-layer outputs are copied back onto a set of context units. Outputs from the context units then combine together with new input signals on the next cycle to feed the hidden units again at time $t+1$. Thus, the external inputs are being mixed with the previously computed inputs "in context" to give recurrent combinations of transformed inputs to the output layer. The weights on the feedback connections from the hidden to the context layer are fixed, typically as unit valued weights. All other weights learn to encode sequences of input patterns during the training process. The activation functions are nonlinear differentiable functions, although the output activation function is normally linear.

## 5 LOAD FORECASTING USING ANN

Two different methods of application of ANN are presented for the short-term load forecasting. The first method is a *"static approach"* in the sense that the 24-hour load vector is forecasted simultaneously using the previous load patterns. The second method is a *"dynamic approach"* in the sense that the 24-hour load is forecasted sequentially using the previous-time forecasts. However, when the recurrent neural network is used to the second method, the recurrent state memorizes the previous state, which makes this approach truly a *dynamic approach*.

### 5.1 Selection of Input Variables

The most important task in constructing our network was the selection of input variables. The dynamic behavior of the network is highly dependent on the chosen inputs, the load has to exhibit a strong degree of statistical correlation with these variables. Prevalent weather patterns have a significant impact on the nature of the load profile. Thus, the inclusion of weather variables in the network inputs can significantly improve the prediction performance. The following variables were used as inputs to the proposed network:

- *Weather related inputs*: Temperature is the most important weather variable, representing the strongest correlation with weather-related load variation. The maximum, minimum and average values of the daily temperature were used as three inputs in the model.
- *Historical loads*: 24 hourly-recorded values of historical load were used as 24 inputs to the proposed network.

### 5.2 Scaling the Network Variables

Due to the nature of sigmoidal transfer function, the outputs of the neurons in the hidden layer are limited to values between $[0,1]$ or $[-1,1]$. Allowing large values for the neuron input variables will cause the threshold functions to be driven into saturation frequently, resulting in an inability to properly train the network. In practical implementations, scaling the network inputs and outputs to an appropriate range, usually between $[-1, 1]$ by applying the following normalization formula solves this problem.

$$Normalized\ value = \frac{actual\ value - minimum\ value}{maximum\ value - minimum\ value} . \tag{13}$$

Care has to be taken not to destroy vital relationships between different network variables by the scaling techniques employed. Thus, as a rule of thumb, if two network variables represent the same physical parameter (*eg* temperature), although for different time instances or geographical locations, they should be scaled according to the same strategy, using the same scaling parameters.

### 5.3 Construction of MLP and RNN Architectures

The network architectures of both MLP and RNN for one-day ahead load forecasting were as follows:

- The input layer consists of 27 neurons. The first 24 nodes represent the past 24-hour loads, nodes 25, 26 and 27 represent minimum, maximum and average values of the daily temperatures, respectively.
- The hidden layer consists of 24 neurons in case of MLP, while 60 neurons in case of RNN (this number was determined from studying the network behavior during the training process taking into consideration some factors like convergence rate, error criteria, *etc*).
- The output layer consists of 24 neurons each represent the predicted hourly load covering 24 hours of the day.

Three networks were constructed in each case corresponding to working days, weekends and special days (holidays) of the year in the Czech Republic.

### 5.4 Evaluation of Prediction Performance

In this stage, the assessment of the forecasting performance of the trained networks was done. Quantifying the prediction error obtained on an independent data set typically does this evaluation. If training was successful, the networks will be able to generalize, resulting in a high accuracy in the forecasting of unknown patterns (provided the training data is sufficiently representative of the forecasting situation). Various forecasting error measures between the actual and forecasted load are defined, but the one most commonly adopted by load forecasters is the mean absolute percentage error (MAPE) defined by

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{P_{actual,i} - P_{predicted,i}}{P_{actual,i}}\right] \times 100 \tag{14}$$

where $P_{actual}$ and $P_{predicted}$ are the actual and forecasted loads, respectively. $N$ is the number of hours in the data set.

## 6 TEST RESULTS

Both a recurrent and purely feedforward neural networks were trained properly in order to predict the short-term load. Learning was made on the load and temperature data for the years 1994 to 1996 and test/evaluation on the year 1997. Both feedforward and recurrent networks employed a single hidden layer, and were trained according to the error backpropagation algorithm, using conjugate gradient descent optimization. In all cases, the number of neurons in the hidden layer was varied until the prediction performances were satisfactory on both the training and evaluation data sets. These performances were quantified in terms of the mean absolute percentage error.

Examples of the feedforward and recurrent predicted profiles, together with the actual load are given in Figs. 7, 8 and 9 for holidays, one weekend and working days of a week, respectively. The forecasting results for the feedforward and recurrent network models are compared in Figs. 10 to 12 for weekend, five working days and ten special days of the year in the Czech Republic, respectively.

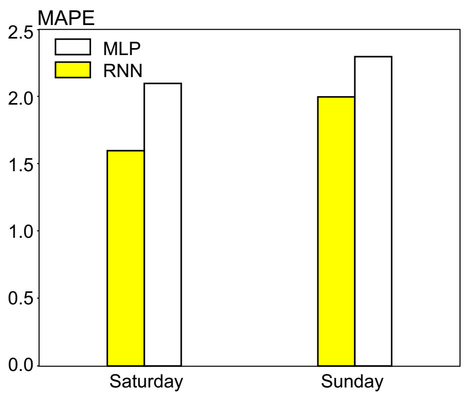


**Fig. 10.** Mean absolute percentage error (MAPE) for the 1st weekend in November 1997 using MLP and recurrent neural networks.

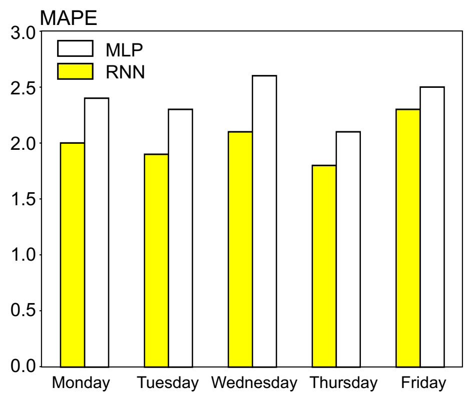


**Fig. 11.** Mean absolute percentage error (MAPE) for five working days of a week in November 1997 using MLP and recurrent neural networks.

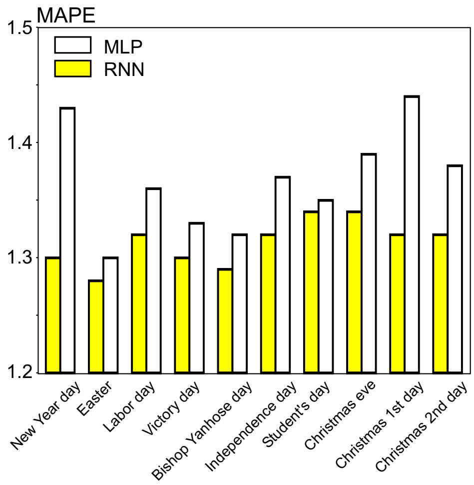


**Fig. 12.** Mean absolute percentage error (MAPE) for ten special days/holidays in the Czech Republic using MLP and recurrent neural networks.

## 7 CONCLUSIONS

In this paper, the emphasis was put on the comparison of recurrent and feedforward neural networks. The recurrent network architecture has been applied to the short-term load forecasting problem. It successfully captured the dynamic behavior of the load, resulting in a more compact and natural internal representation of the temporal information contained in the load profile than possible with a normal feedforward network. The training phase of the Elman recurrent network was time-consuming depending on the training data size and the number of network parameters. However, it must be remarked that the evaluation stage was considerably fast.

Even though the results are promising, more research still needs to be done before any conclusive statements concerning the appropriateness of each of these models can be made. This also includes the investigation of different other recurrent network topologies and training paradigms, together with an analysis of conditions for stability. Thus, it can be concluded that recurrent neural networks yield even more better and effective results, if good data selection strategies and network input and output representations have to be determined properly. The superior performance of the recurrent network over the feedforward network is time-series dependent.

### Acknowledgements

The authors would like to thank the Czech Power Company (ČEZ), Prague, Czech Republic for providing the electrical load and temperature data used in this work. This research was possible with the support of Výzkumný Záměr CEZ: J22/98:262200010, Czech Republic.

## References

[1] CONNOR, J. T., *et al*: Recurrent Neural Networks and Load Forecasting, Proc. 1st International Forum on Applications of Neural Networks to Power System, 1991, 22–25.

[2] LEE, K. Y., *et al*: Short-Term Load Forecasting Using Diagonal Recurrent Neural Network, Proc. 2nd International Forum on Applications of Neural Networks to Power System, 1993, 227–32.

[3] CZERNICHOW, T., *et al*: Improving Recurrent Network Load Forecasting, Proc. ICNN'95 — International Conference on Neural Networks, Perth, Australia, November 1995.

[4] CZERNICHOW, T.—MUNOZ, A.: Variable Selection Through Statistical Sensibility Analysis: Application to Feedforward and Recurrent Neural Networks, Technical Report 95-07-01 INT — Signal and Image Processing Department, July 1995.

[5] CZERNICHOW, T., *et al*: Load Forecasting and Recurrent Networks, Neural Network World **6** (1993), 895–906.

[6] KHAN MUHAMMAD, R.—ONDRŮŠEK, Č.: Recurrent Neural Network Technique for One-Day Ahead Load Forecasting, Proc. 3rd International Conference on Prediction (NOSTRADAMUS), pp. 93–99, Technical University, Faculty of Technology, Zlín, Czech Republic, October 2–3, 2000.

[7] VERMAAK, J.—BOTHA, E. C.: Recurrent Neural Networks for Short-Term Load Forecasting, IEEE Transactions on Power Systems **13** No. 1 (1998), 126–132.

[8] VEMURI, V. R.—ROGERS, R. D.: Artificial Neural Networks: Forecasting Time Series, IEEE Computer Society Press, Los Alamitos, 1994.

[9] DJUKANOVIC, M., *et al*: Unsupervised/Supervised Learning Concept for 24-Hour Load Forecasting, IEE Proceedings-C **140** No. 4 (1993), 311–318.

[10] PARK, D. C.—MOHAMMED OSAMA: Artificial Neural Network Based Electric Peak Load Forecasting, IEEE Proc. South East Conference, Vol. 1, USA, 1991, 225–228.

[11] KOSKELA, T., *et al*: Time Series Prediction with Multilayer Perceptron, FIR and Elman Neural Networks, WCNN'96 — World Congress on Neural Networks. International Neural Network Society 1996 annual meeting, USA, 491–495.

[12] PATTERSON, D. W.: Artificial Neural Networks — Theory and Applications, Institute of System Sciences, National University of Singapore, February 1995.

Received 9 November 2000

**Muhammad Riaz Khan** (Ing) was born in Mardan, Pakistan in 1971. He received BSc (Hons) and MSc electrical engineering degrees from N-W.F.P. University of Engineering and Technology, Peshawar, Pakistan in 1992 and 1997, respectively. From 1993 to 1997, he worked as Assistant Director (Technical) in HV and SC Testing Laboratory, WAPDA, Rawat, Islamabad, Pakistan. He was involved in commercial testing of all electrical equipment. Presently, he is a postgraduate student in the Department of Power Electrical and Electronic Engineering, Faculty of Electrical Engineering and Computer Science, Brno University of Technology, Brno, Czech Republic. His research interests include electrical load forecasting, artificial neural networks and fuzzy logic.

**Čestmír Ondrůšek** (Doc, Ing, CSc) was born in Uherské Hradiště, Czech Republic in 1941. He received the Ing, CSc and Doc degrees from Military Academy, Brno in 1965, 1976 and 1983, respectively. From 1965 to 1990, he was attached with the Military Academy Brno. Since 1991 till now, he has been a staff member in VUT, FEI Brno, Czech Republic. Presently, he is the head of the Department of Power Electrical and Electronic Engineering. His research interests include electrical machines, electric drives, mechatronics and artificial intelligence.