Recurrent Neural Network Based Gating for Natural Gas Load Prediction System

Petr Musilek, *Member, IEEE*, Emil Pelikán, Tomáš Brabec and Milan Šimůnek

Abstract— **Prediction of natural gas consumption is an important element in gas load management aimed to better utilize the facilities of a gas distribution system. The major challenges faced by developers of prediction systems are the variety and volatility of consumer profiles, strong seasonal dependency and dependency on climatic conditions, and lack of extensive and reliable historical data. In this paper, the problem of seasonal dependency is tackled with a recurrent neural network used as a gate for a statistical mixture model. Historical consumption data along with climatic conditions and other auxiliary descriptors are combined with expert delineation of heating season boundaries to provide training data. The resulting gating system is capable of reliable identification of the start and end of the heating season and, combined with the statistical models, of accurate predictions of gas load.**

I. Introduction

There is a strong economical aspect of natural gas consumption forecasting. Accurate prediction of natural gas consumption can significantly contribute to improvement of gas load management. Subsequently, facilities of a gas distribution system can be better utilized and gas purchases managed, leading to significant savings. Compared to human experts, quality of load predictions can be improved using automated forecasting systems, usually based on statistics and/or computational intelligence (CI). In addition to the direct use of the load predictions, they can also be used as support tools aiding experts (analysts, economists, dispatchers) in their decisions. Another advantage of CI based forecasting systems is their ability to learn from past experience and to adapt to changing distribution environment, both without the need of their explicit description.

The major challenges faced by the developers of prediction systems are the variety and volatility of consumer profiles, and strong seasonal and climatic dependence of actual consumption. These difficulties usually limit the maximum accuracy possible to achieve by a single-model forecasting system. To overcome this limitation, multiple models can be combined into so called *mixture models* based on a divide-and-conquer approach to solving complex problems. Mixture models are used when it is either not possible or too difficult to solve a problem at once. They use several solution modules that deal with simpler problems and a gating module for deciding which of the solution modules is responsible for a certain part of the problem.

In this paper, divide-and-conquer approach is applied to the problem of gas load forecasting. The approach is used to extend the forecasting system ELVIRA, a modular system developed and used for utility load predictions in various time horizons [8]. The current system provides various statistical and CI methods that can be used to build a single global prediction model for given forecasting task. To improve accuracy of the system, it is proposed to build several local models and a gating system responsible for applying a local model appropriate to actual situation. The proposed gating system is build using a recurrent neural network. Historical consumption information along with climatic data and other descriptors are combined with expert delineation of heating season boundaries to obtain training data. The resulting gating module is capable of reliable identification of start and end of heating season and, combined with the prediction models provided by ELVIRA, of more accurate predictions of gas load.

The paper is organized as follows. Section II introduces the prediction system ELVIRA, and provides a brief overview of mixture models and recurrent neural networks. The gating module itself is described in Section III, while the results of experiments using this module for load prediction are presented and analyzed in Section IV. Finally, Section V brings main conclusions and provides possible directions of future research.

II. Background

A. The Forecasting System

The presented gating module extends the commercial forecasting system ELVIRA [8]. This module can be, however, used with other prediction modules and/or systems. For this reason, the following description is limited to general characteristics of the system and properties of the prediction method employed for short-term (1–day) forecasts of gas consumption, used as an illustrative example in this paper.

ELVIRA is a complex modular system providing prediction and decision-support information to utility production and distribution companies. Within the system, there are various prediction methods available for deployment, such as:

- Box-Jenkins models
- Case-based reasoning (CBR) models
- Rule-based systems

Petr Musilek is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (phone: 780-492-5368; fax: 780-492-1811; email: musilek@ece.ualberta.ca). Currently, he is a visiting scientist with the Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic.

Emil Pelikán, Tomáš Brabec and Milan Šimůnek are with the Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod vodárenskou věží 2, 182 07 Prague 8, Czech Republic. (email: emil, bendoch, simunek@cs.cas.cz).

- Artificial neural networks
- Decomposition time series

Depending on particular tasks and available data, the system recommends the most appropriate method and performs forecast for a desired time horizon. General structure of the system is depicted in Figure 1.

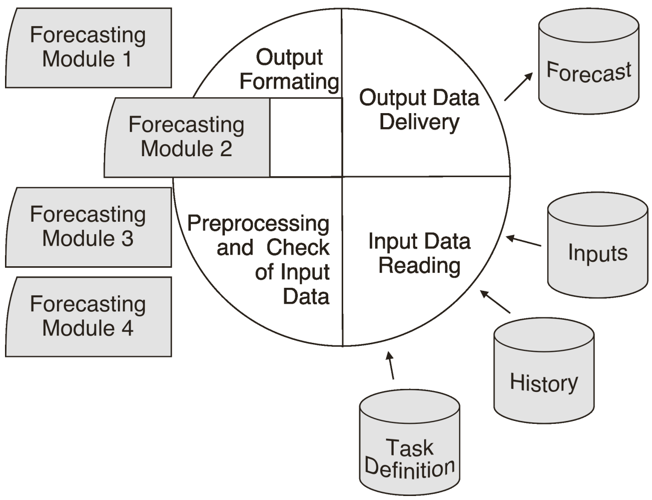

Fig. 1. Modular structure of the forecasting system ELVIRA

As an illustration, consider the short-term load forecasting module that performs prediction of the next day value of gas load based on the real values of consumption and temperatures, as well as the weather outlook. The module uses a nonlinear multiple regression model which takes into account current and past values of temperature, calendar and seasonal effects, as well as an autoregressive term correcting the past prediction error. The module provides a number of parameters that can be tailored to particular customer. The use of this module is illustrated in Figure 2.

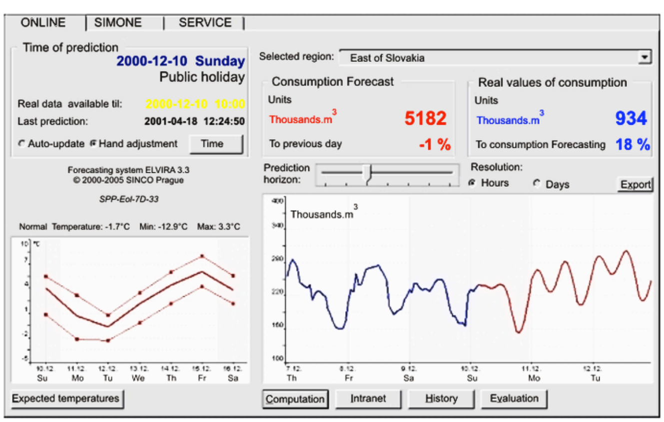

Fig. 2. ELVIRA – results of short-term load forecast

Although this module provides highly accurate predictions for the time periods that can be considered either heating or non-heating season, it is unable to effectively cope with transient “between-season” periods. This led to the development of two separate modules: a summer module that is able to predict the summer load including the decreasing gas usage in the spring; and a winter module that is able to predict the winter load as well as the increasing gas consumption in the fall. The use of two (or more) modules has the potential of significantly improving the quality of load predictions. Indeed, an appropriate method for combining the results of individual modules is required.

B. Mixture Models

The basic idea behind mixture models is to divide a complex problem to several subproblems that can be solved using simpler partial/local models. The solutions to the subproblems are then combined to yield a solution to the overall problem. Algorithms of this type typically consist of two classes of modules [4]. A gating module that selects a module of the other class to solve a particular part of a problem, and combines the partial results; and two or more solution modules providing the partial results to specific parts of the problem space.

Two main problems to be solved by a mixture model are: (i) splitting of the input space into a number of regions, and (ii) finding an appropriate solution for each region which is simple compared to a global solution. Depending on the way in which the input space is partitioned, there are two major classes of mixture models. Hard splitting algorithms [2], [9] assign each point from the input space to only one region, while soft splitting algorithms [7], [13] allow a point to belong partially to more than one region. Soft splitting can be generally based on probabilistic or possibilistic (fuzzy) principles. Although the current gating module provides hard splitting of the input space, its fuzzy-based soft splitting extension is planned for the near future.

Mixture model approach to forecasting has a number of advantages. First, each partial model can specialize on a different region of the input space and their combination may yield better results than possible with a global model. This specialization can have different forms, such as adapting to different noise levels [13]. In the case described in this paper, the specialization is concerned with different seasons (heating/non–heating), effectively eliminating problems associated with non–stacionarity of underlying time series. Second, the gating system itself can provide important additional information about the nature of the input space (e.g. relevance of individual features or their combinations, discovery of hidden dependencies, etc.).

C. Recurrent Neural Networks

Conventional feedforward neural networks can be used to approximate any spatially finite function given a sufficient set of hidden nodes [5]. Recurrent neural networks (RNN) are fundamentally different from feedforward architectures in the sense that they operate, in addition to an input space, on an internal state space representing what already has been processed by the network [1]. The state space enables representation of temporally/sequentially extended dependencies over long intervals. Trained RNNs can thus exhibit virtually unlimited temporal dynamics and as such are suitable for processing complex time series, including non-stationary series corresponding to gas load processes.

There have been many different types of RNNs proposed during the past three decades, including Elman [3] and Jordan [6] networks. Based on its superior performance in initial experiments, Jordan RNN is used to predict the start of heating season in this study. This network is equipped with a set of *context units* that receive a copy of the network's outputs and have self-recurrent connections as shown in Figure 3. The number of context units corresponds to the number of outputs. This form of recurrence is a compromise between the simplicity of a feed-forward network and the complexity of a fully recurrent neural network.

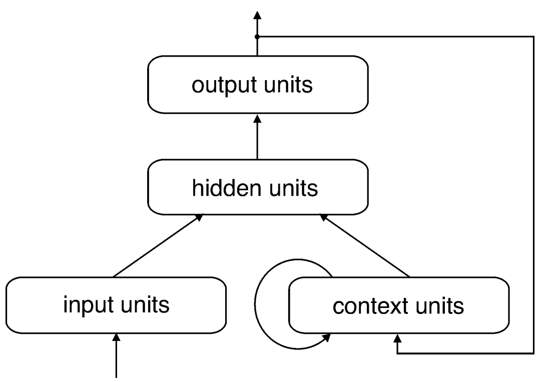

Fig. 3. Architecture of Jordan network

The state neurons are activated according to the following expression:

$$s_i(t) = \lambda s_i(t-1) + y_i(t-1), \quad (1)$$

where $y_i(t-1)$ is the activation of the i–th output neuron at time $t-1$, and λ is a positive coefficient of self-recurrent connections. In general, $\lambda < 1$, and its actual value determines how much of the past output activation is considered in current network operation.

The outputs of the context neurons are then fed back to the hidden neurons. Thus, the neurons of the hidden layer receive a vector of values formed by concatenation of the inputs and states of the network

$$\mathbf{z} = [\underbrace{z_1, z_2, \ldots, z_{n-1}, z_n}_{\text{from input units}}, \underbrace{z_{n+1}, z_{n+2}, \ldots, z_{n+m}}_{\text{from context units}}],$$

where components $z_{n+i} = s_i, i = 1, \ldots, m$, are derived from the network's output. This way, the behavior of Jordan's recurrent network can be simulated with a simple feedforward network that receives the state not implicitly through recurrent links, but as a part of the input vector [14]. Consequently, backpropagation training algorithm [11] and its derivatives can be easily modified for such networks.

When used for prediction or classification of univariate time series, the network has only one output neuron carrying the predicted value or class label, respectively. The number of input neurons is driven by the number of features used

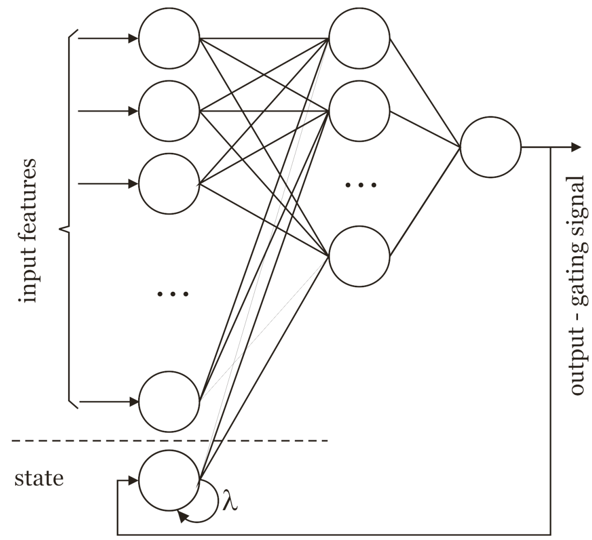

Fig. 4. General architecture of the gating network

for given task, while the number of hidden neurons depends on the structural and temporal complexity of the task and should be determined experimentally. General structure of such neural network is shown in Fgure 4.

III. Gating Module

To overcome the difficulties associated with using a single global forecasting model, two separate models have been constructed. Summer module to predict the summer load including the decreasing gas usage in the spring; and winter module that is able to predict the winter load as well as the increasing gas consumption in the fall. To schedule appropriate application of either model, a gating module has been designed based on the concept of mixture models summarized in the previous section. The overall scheme of the forecasting system system is shown in Figure 5. In the figure, $x_g, x_1, x_2 \subseteq \mathbf{x}$ are the input vectors of the gating module and the forecasting modules 1 and 2, respectively, y_1, y_2 are the values predicted by each forecasting module and gated by the gating signals g_1 and g_2, and y is the overall prediction produced by the system.

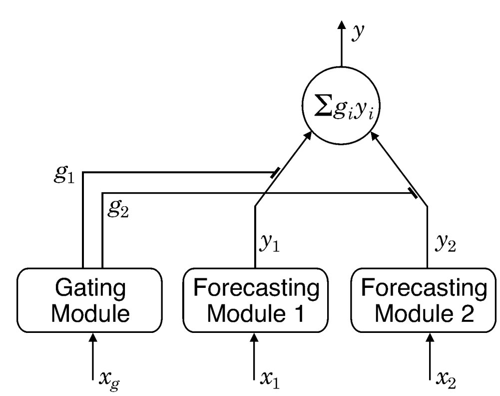

Fig. 5. Structure of the gating system

The gating system is currently designed in a binary form, i.e. forecast of only single module is considered at a time, not their combination. It is, however, possible to extended the system to soft/fuzzy case.

The gating module itself must process sequential information related to the climatic and other conditions of the environment where the forecasting system is used. This requirement of sequential processing led to the choice of recurrent neural networks as the implementation platform.

A. *Feature selection*

There are many features that could potentially have influence on forecasting model construction. Preliminarily, the following features have been selected

- $T_{D-h}, h = 1, \ldots, 5$; temperatures h days prior the day D for which forecast is performed;
- D^{week}, day of week, to capture short-term changes in load (e.g. weekdays, weekends, holidays, etc.);
- D^{year}, day of year, to capture long-term changes in load (e.g. seasons).

General validity of this initial feature set has been confirmed using correlation analysis. Later on, a set of experiments has been performed to either confirm all these features as relevant, or to get a more compact yet sufficient set of features. Results of these experiments are reported in Section IV.

B. *Training and testing data*

The data for training and testing of both forecasting modules and the gating module have been compiled from the following sources (all for the period of four years 2001-2004)

- temperature records (daily mean temperatures);
- calendar records (weekdays, holidays);
- measured gas load (daily consumption);
- expert estimates of start/end of heating season;

The data sets have been merged and preprocessed to represent the features identified earlier. The temperature data has been directly grouped into sequences of five consecutive values using the method of sliding window.

To distinguish not only between working and non–working days but also among different types of these days (e.g. working day before weekend vs. working day after weekend), so called WN coding has been adopted. In this coding, character of the day for which forecast is performed, D, is provided in the context of the surrounding (i.e. immediately preceding and immediately following) days. Using labels 1 (working) and 0 (non–working), eight possibilities shown in Table I can be encoded.

TABLE I
WN CODING

Code	Example
111	Tuesday (standard week)
110	Friday (standard week)
101	mid–week holiday
100	Saturday (standrad week)
011	Monday (standard week)
010	e.g. Monday December 24
001	Sunday (standard week)
000	long weekend

The day of year could be coded directly using its sequential number in year (a.k.a. Julian code). However, this would bring problems associated with the abrupt change of such code at the turn of year. For this reason, trigonometric encoding scheme [12] has been adopted that provides a smooth coding of the day of year. In this coding scheme, each day is encoded using two values, $D^{year}_{\sin}$ and $D^{year}_{\cos}$, derived from the Julian code as follows

$$D^{year}_{\sin} = \sin \frac{2\pi D^{year}}{366},$$

$$D^{year}_{\cos} = \cos \frac{2\pi D^{year}}{366},$$

where D^{year} is the Julian code of the day. Finally, the expert estimates of the start/end of heating season have been encoded using values 0 (non-heating) and 1 (heating).

Each data point is thus represented by an 11–dimensional vector

$$[T_{D-5}, T_{D-4}, T_{D-3}, T_{D-2}, T_{D-1}, \ \ldots$$
$$\ldots \ D^{week}_{D-1}, D^{week}_{D}, D^{week}_{D+1}, D^{year}_{\sin}, D^{year}_{\cos}, \overline{g}_D],$$

where $\overline{g}_D$ is the desired value of gating signal (season indicator) for day D.

A total of 1461 data points from the period 2001–2004 have been collected this way. Approximately 75% of the data (years 2001–2003) has been used for the gating module development and 25% of the data (year 2004) has been segregated for testing. Table II contains information regarding the temperature ranges and heating days percentages for the training and testing data sets.

TABLE II
COMPOSITION OF THE DATA SETS

Data set	Calendar year	Heating days	Daily averages [°C]		
			min	max	avg
Training	2001	65.5%	-10.1	25.3	8.9
	2002	63.0%	-13.1	26.6	9.9
	2003	58.9%	-9.7	28.2	9.9
Testing	2004	68.6%	-11.0	24.9	9.3

C. *Gating Network*

The overall structure of the gating network is depicted in Figure 6. The input layer contains ten input neurons labeled 1.1-10 and one context neuron labeled 1.11. While the input neurons receive the external inputs corresponding to the features identified in the previous subsection, the context neuron receives the output of the network - gating signal indicating presence of conditions corresponding to the heating season and thus the need to use appropriate forecasting module. The input neurons have linear activation function as they serve only to distribute the input signals to the neurons of the hidden layer. All remaining neurons of the network, including hidden neurons 2.1-6, context neuron 1.11 and output neuron 3.1, have unipolar logistic activation function

$$y = \frac{1}{1 + e^{-u}},$$

where u represents the net input to the neuron and y is the neuron's output.

The network has been constructed in Stuttgart Neural Network Simulator [14] and trained using Rprop learning algorithm [10] with the following parameters: initial step-size $\Delta_0 = 0.3$, maximum step size $\Delta_{\max} = 30.0$, weight decay exponent $\alpha = 4.0$ and teacher forcing 50% (see [14] for description of the learning parameters).

IV. Experimental Results

In this section, the results of two sets of experiments are presented. The first set was designed to determine the most appropriate set of features that should be included as the inputs of the gating module. The second set of experiments was performed using the actual forecasting modules, to assess the the accuracy of the overall prediction system.

A. Feature set size determination

To determine the most appropriate set of features for the gating module, the recurrent neural network from Figure 6 was modified to accommodate appropriate number of inputs. The size of hidden layer was kept at about 60% of the size of the input layer. The experiments started with only 3-dimensional input vector containing the temperatures from 3 days preceding day D. Other features were added, individually or in groups, as indicated by x-marks in Table III.

TABLE III

Feature Selection

T_{D-5}			x	x	x	x
T_{D-4}		x	x	x	x	x
T_{D-3}	x	x	x	x	x	x
T_{D-2}	x	x	x	x	x	x
T_{D-1}	x	x	x	x	x	x
D^{week}				x		x
D^{year}					x	x
Training	72.6%	78.2%	88.6%	92.3%	94.3%	98.9%
Testing	76.0%	92.9%	92.8%	94.5%	95.4%	94.7%

For each of the six sizes of the feature set, ten independent trials have been conducted involving initialization and training of the network, and running the trained network to obtain results for the testing test. The results reported in the table represent the percentage of days that have been identified correctly as heating/non–heating, averaged over the ten trials. Based on the results reported in Table III, all ten features have been used to build the gating network.

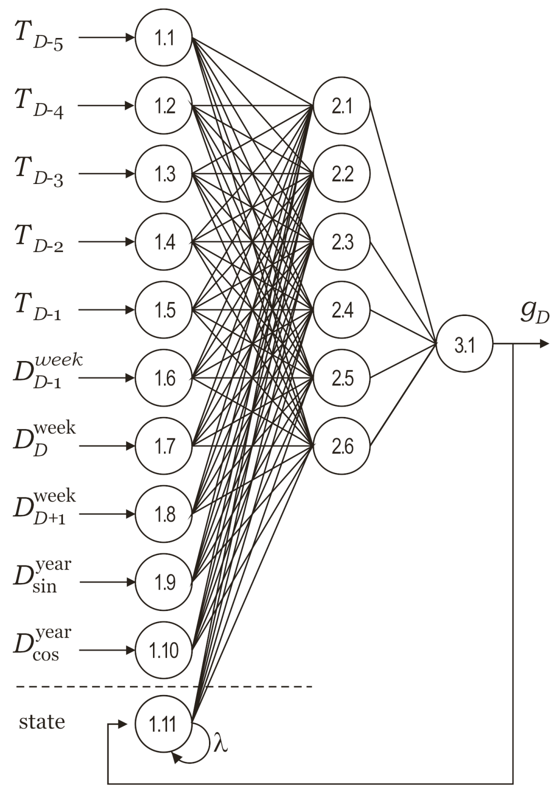

Fig. 6. Final architecture of the gating network

B. Load forecasting results

After the gating network was constructed and trained, its output was used to gate the two forecasting modules of ELVIRA. The results obtained this way were compared to the actual (measured) values of gas load, both for the period covered by the training and testing sets. Measures used to compare the results are mean absolute prediction error (MAPE), and percentage of correct predictions at the levels of 25% and 10% (Pred(25) and Pred(10)). Results of these experiments are reported in Table IV.

TABLE IV

Results of gas load forecasting system obtained with global and gated models

Model	Set	MAPE	Pred(25)	Pred(10)
Global	Training	8.4%	89.5%	78.9%
	Testing	6.3%	95.7%	80.3%
Gated	Training	3.8%	99.0%	93.9%
	Testing	3.6%	99.8%	94.0%

It can be seen that the use of gated model further improves accuracy of the forecasting system. Looking at the results received with the testing set, the mean absolute prediction error was been lowered by almost 3%, while the proportion of load values predicted within 10% of the actual values increased from 80.3% to 94.0%.

V. Conclusions and Future Work

In this paper, a new mixture model has been introduced to improve results of a statistical system for gas load forecasting. The gating module of the model is based on Jordan's partial recurrent neural network and, as such, it is able to learn and predict time series that exhibit complex non-stationary behavior.

The current gating system provides binary output used for switching between the two available forecasting modules based on actual climatic and seasonal conditions. It is planned to extended the current model to provide soft boundaries between the seasons and to blend the results of the forecasters instead of just considering one and ignoring the rest.

Another expected continuation of the present work is its extension to predictions with different time horizons (i.e. medium– and long–term predictions). An interesting aspect of such extension will be sensitivity analysis of the system necessary due to the need of using weather outlook information hindered by uncertainty.

Acknowledgment

This work was supported in part by the Grant Agency of the Academy of Science of the Czech Republic under grant No. 1ET4003005e13, and by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

References

[1] M. Bodn, *A guide to recurrent neural networks and backpropagation*, Report from the NUTEK-supported project AIS-8: Application of Data Analysis with Learning Systems, 1999-2001. Holst, A. (ed.), SICS Technical Report T2002:03, SICS, Kista, Sweden. 2002.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, 1984.

[3] J. L. Elman, Finding structure in time, *Cognitive Science*, 14, 1990, pp. 179–211

[4] C. Gilde,*Time Series Analysis And Prediction Using Recurrent Gated Experts*, MSc Thesis, University of Skövde, Sweden, 1996

[5] K. Hornik, M. Stinchcombe, H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, 2, 1989, pp. 359–366.

[6] M. I. Jordan, "Attractor dynamics and parallelism in a connectionist sequential machine," In *Proc. Proceedings of the Eighth Conference of the Cognitive Science Society*, 1986, pp. 531–546

[7] M. I. Jordan, R. Jacobs, "Hierarchical mixtures of experts and the EM algorithm", *Neural Computation*, 6, 1994, pp. 181–214

[8] E. Pelikan, M. Simunek , T. Brabec, "Load Forecasting using the System Elvira," In *Proc. Proceedings of the 3rd SIMONE workshop*, Cesky Krumlov, Czech Republic, May 11-14, 2004 (CD proceedings)

[9] J. R. Quinlan, "Induction of decision trees", *Machine Learning*, 1(1), pp. 81–106

[10] M. Riedmiller, H. Braun, "RPROP – a fast adaptive learning algorithm." In *Proc. of ISCIS VII*,1992

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Representations by Back-Propagating Errors," *Nature*, Vol. 323, 1986, pp. 533–536

[12] N. H. Viet, J. Mandziuk, "Neural and Fuzzy Neural Networks for Natural Gas Consumption Prediction", In *Proc. 2003 IEEE XIII Workshop on neural Networks for Signal Processing*, (NNSP03), Toulouse, France, pp. 759–768

[13] A. S. Weigend, M. Manegad, A. N. Srivastava,"Nonlinear gated experts for time series: discovering regimes and avoiding overfitting", *International Journal of Neural Systems*, 6, 1995, pp. 373–399

[14] A. Zell et al. *SNNS: Stuttgart Neural Network Simulator.* User manual, University of Stuttgart, Germany, 1994