LINEAR AND NON-LINEAR MODELS FOR SIGNAL PREDICTION

A. Pavelka and A. Procházka

Institute of Chemical Technology, Department of Computing and Control Engineering

Abstract

The focus of this study is in the proposal of a suitable prediction model and comparison of results of different methods. Methods presented include polynomial models, autoregressive models, linear neural networks, adaptive linear element, feed-forward neural networks, the Elman neural networks and a recurrent neural networks with a real time recurrent learning algorithm. These models have been modified by a selection of a suitable set and order of parameters by SVD and QRcp algorithms. The created algorithms have been then applied for processing of real data representing gas consumption in the Czech Republic with a focus on the data collected during the winter season. The main part of the work has been devoted to the description of the modifications proposed to the models, calculation of conditions and comparison of all the models mentioned above. All calculations have been done with the usage of MATLAB Distributed Computing Toolbox in a cluster of several computers.

1 Introduction to Model Structure

A wide range of linear and non-linear methods can be used for the prediction of gas consumption in the Czech Republic. The main focus has been devoted especially to neural networks models. All models used for prediction and their main modifications are summarized in the Tab. 1. Most of prediction approaches have been slightly modified to gain better results or find out new prediction ways.

Table 1: All models used for prediction of gas consumption

Model group	**Adaptivity**	**Linear**	**Non-Linear**
Classical	Non-Adaptive	UDP AP ARF ARO ARR	*none*
Neural Networks	Non-Adaptive	NL NLs	FF FFs ELM ELMs
	Adaptive	ADAL ADALs	RTRL RTRLs

Each prediction approach due to faster identification has been named by specific shortcut, explained below:

UDP - model created and used at BC[1] (ÚPD)[2]
AP - polynomial model
ARF - full autoregressive model
ARO - subset autoregressive model with original parameters
ARR - subset autoregressive model with recalculated parameters
NL - linear neural network
NLs - linear neural network with input data set optimized by SVD[3]
ADAL - adaptive linear element
ADALs - adaptive linear element with input data set optimized by SVD
FF - feed-forward neural network
FFs - feed-forward neural network with input data set optimized by SVD
ELM - Elman neural network
ELMs - Elman neural network with input data set optimized by SVD
RTRL - recurrent neural network with real time recurrent learning algorithm
RTRLs - recurrent neural network with real time recurrent learning algorithm with input data set optimized by SVD

2 Common Properties of Models

One of the main goals of the research was to find the most suitable model for prediction of gas consumption in the Czech Republic. For model comparisons it has been necessary to have some of parameters of prediction models the same. Description of "tuning" specific models follows.

2.1 Inputs of Models

The main unification element among calculated models is the number of inputs and outputs. Not only total number of inputs is important, but also its structure and composition play its own role in the model inputs. Input had tree parts:

- data of gas consumption (*gas*) – input had 0, 1, 2, 3, 4, 5, 7, 9, 12, 14 and 16 values following each other
- daily temperature (*temp*) – input had 0, 1, 2, 3, 4, 5, 7, 9 and 12 values following each other
- information about a day in a week – generally optional input, that has been used in all models.

The number of inputs of gas consumption or daily temperature come out of periodicity occurred in processed signals, see Fig. 1 and Fig. 2. Reducing higher numbers of inputs speeded up all calculations and reduced number of models. The values representing information about a day in a week have following logic. Every normal day in a week has been represented by specific weight: Monday 1.01, Tuesday, Wednesday and Thursday 1.00, Friday 0.95, Saturday 0.80 and

[1] Bilanční Centrum (Bilance Center)
[2] Ústřední Plynárenský Dispečink (Gas Control Center)
[3] Singular Value Decomposition

Sunday 0.85. Holidays have been weighted according to the occurrences in a week respecting human behavioral and average evaluation of the gas consumption during a week. Whole algorithm is described in Tab. 2 in detail.

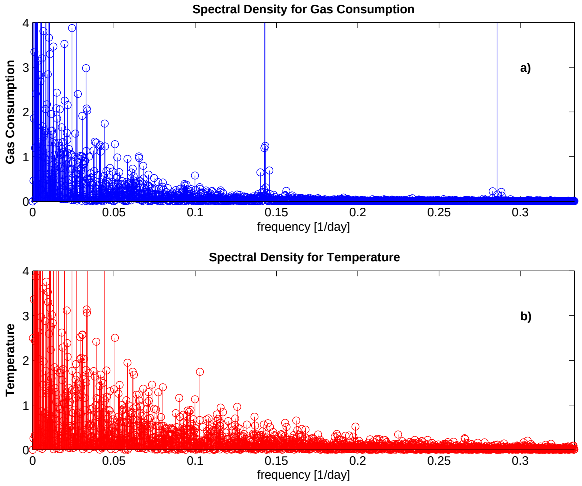

Figure 1: Spectral density of gas consumption and average daily temperature

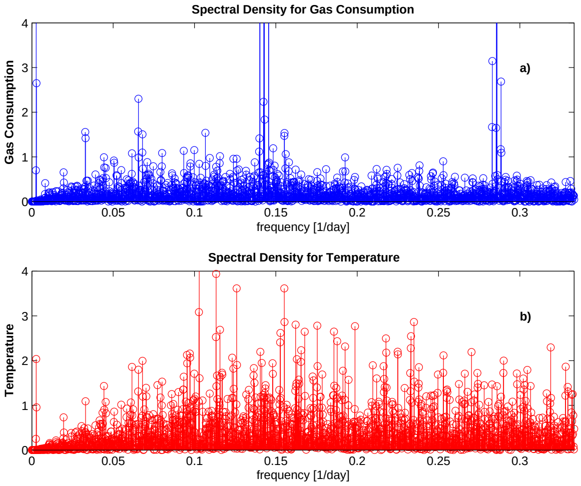

Figure 2: Spectral density of gas consumption and temperature, modified by difference of logarithms

Table 2: Influence of holidays to evaluation of week

	One holiday					Two holidays				Three holidays			
1.00	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	1.00
0.95	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	0.95
0.80	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	0.80
0.85	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	0.85
1.01	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	1.01
1.00	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	1.00
1.00	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	1.00
1.00	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	1.00
0.95	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	0.95
0.80	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	0.80
0.85	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	0.85
1.01	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	1.01
1.00	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	1.00
1.00	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	1.00
1.00	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	1.00
0.95	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	Fri	0.95
0.80	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	Sat	0.80
0.85	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	Sun	0.85
1.01	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	Mon	1.01
1.00	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	Tue	1.00
1.00	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	Wen	1.00
1.00	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	Thu	1.00

How does the influence of holidays to evaluation works? The gray cells in Tab. 2 represent holiday and color of the name's day is weight that is in the first and in the last column. Let us demonstrate the practical use of this table in example, let us have two holidays in a week on Tuesday and on Wednesday. The whole week before these holidays remains untouched. The change become on Monday before holidays. Monday has the same weight as Friday, because people feels that working week is shorter so their productivity slightly decrease. On the Tuesday, the first holiday will be the weight of Saturday and on the Wednesday the situation is like on Sunday. Both days will have the same evaluation as weekend. When two days holidays ends, the Thursday and next days will have the same weight as they have in a normal week. In this situation won't be a working increase as increase days after weekend usually. Situation with different combination of holidays in a week, that is shown on Fig. 2, has not been solved, because only these 12 combination occurs in the Czech calendar.

The pattern matrix has been composed, as described above, from data of gas consumption, then data of daily temperature and finally from data of day information. Models with name modified by suffix *s* (NLs, FFs, ELMs, ADALs and RTRLs) had different structure of pattern matrix, the number of parameters for these models has been reduced. The original pattern matrix has been reduced by SVD algorithm and reorganized by QRcp algorithm. For reducing the number of parameter two different methods based on analysis of diagonal matrix **S** have been used. The first was separation according to the highest difference in diagonal matrix **S** excluding the first highest value. The second approach was to calculate 20% of values of the second element in the diagonal matrix **S** and then to look for the latest value that is before 20% descent. Finally both methods have been compared and the method with lower number number won.

As gas consumption is higher during winter period and has higher correlation with temperature, as have been shown on Fig. 3 and on Fig. 4, the general input data set covered beginning of the winter period. Main goal has been set to predict gas consumption on 11$^{\text{th}}$ November 2005 and further. As input has been selected 10 differently longer periods that are summarized in Tab. 3. The selection of lengths of input data set has been done under influence of suitable input consistency and with remark that models with high number of parameters needs to have a lot of data to create over-determined system. This type of system makes results more reliable, than models created with under-determined system and of course reproducible. Using under-determined system is during creating model architecture one of the most common mistakes. But the large amount of input data has negative influence to calculation performance, that has been in case of 10 years data set very low.

Table 3: Lengths of input data set used for prediction - *hist.* The ending date has been one day before target prediction day 11. 11. 2005

Starting date	Length of Input Data Set		
	days	months	years
11. 10. 2005	31	1	1/12
11. 09. 2005	61	2	1/6
11. 08. 2005	92	3	1/4
11. 05. 2005	184	6	1/2
11. 11. 2004	365	12	1
11. 11. 2003	731	24	2
11. 11. 2002	1096	36	3
11. 11. 2000	1826	60	5
11. 11. 1998	2557	84	7
11. 11. 1995	3653	120	10

2.2 Outputs of Models

All prediction models have been constructed for one step prediction to predict gas consumption on 11th November 2005 and further. It does mean that data with sampling period one day have been predicted one sample ahead, that is in our case one day. To extend prediction length two techniques have been combined.

The first is expansion of model outputs (*out*) from 1 to 2, 3, 5, 7 or 10 elements. When model have 3 outputs the first output predict one day ahead, second two and third output tree days ahead. The architecture of multi-output model gives us the opportunity to receive prediction for arbitrary number of samples ahead in one prediction step. The next method is based on gradual prediction steps. After each prediction step the data set used for prediction is shifted one step ahead and the length of data set remains the same. This is called length of prediction time. In our models length of prediction time (*time*) was 1, 2, 3, 5, 7 and 10.

But usually due to certain un-precisions in prediction the prediction error start increasing and the results are not good and this is one of reasons why this pure prediction method is rarely used. The long time prediction can be solved by predicting data with lower sampling period. For example for the prediction of gas consumption for month the data with sampling period one week can be used.

For prediction of gas consumption in Czech Republic has been used combination of both methods. The largest models have been used for predicting 19 samples (days) ahead.

2.3 Hidden Layers in Neural Networks

Number of neurones in the hidden layer has been one of the common parameters that has been the same for feed-forward and recurrent neural networks. This parameter has high influence to calculation time of neural networks and significantly increases the complexity of neural network as well as the number of values needful for the over-determined system. For prediction of daily gas consumption in Czech Republic 2, 4, 6 and 8 hidden neurones for appropriate models have been used.

2.4 Models Excluded from Calculations

When all desired combinations of parameters have been set, the problem with huge amount of models occurred. The target was to calculate models with parameters consisting of 11 different number of inputs for gas consumption, 9 for daily temperature, 6 for prediction (model) outputs, 6 for length of prediction, 5 for hidden neurones. All listed parameters has been used in calculation with 10 different data sets. This conditions aim to calculate 178 200 sets of parameters for 15 models which gives 2 673 000 models and this could take more than 5 years of calculation, when calculation of one single model takes only one minute. The algorithm for model reduction has been therefore applied according to the following equations:

$$\begin{aligned} length\ prediction\ outputs &> gas + temp \qquad (1)\\ length\ prediction\ outputs &> hide \qquad (2)\\ length\ history &< unknown\ parameters * 2 - 1 \qquad (3)\end{aligned}$$

$$\begin{aligned} unknown\ parameters &= (gas + temp + day + \\ &+ length\ prediction\ outputs + hidden) * hidden \end{aligned} \qquad (4)$$

Where variable *gas* represents number of input values of gas consumption, *temp* are values of daily temperature, *day* is value of weight of day (identification what day is in a week), *length prediction outputs* says how many output has calculated model and *hidden* is number of hidden neurones in calculated model. For this calculation all models without hidden neurones or elements have value *hidden* equal to one.

Conditions described above had strong influence to model size reduction. From the total number of 178 200 investigated combinations of parameters 122 910 combinations have been removed during the calculations and 55 290 combinations gained some results. All calculations performed on MATLAB cluster of 8 computers took almost one year. Not only time spare has been reason for reduction. The models that have been calculated and will be analysed later in this text, comes from over-determined system that keeps final results more reliable.

3 Specific Properties of Models

3.1 UDP Model

The main idea of UDP[4] model has been created in BC, but for our purposes it has been sightly changed. The first change has been dynamical recalculation of specific weight of day in a week under condition if there is a holiday during a week or not. Next change has been modification of model for multi-step – multi-output prediction

$$\begin{aligned} &\hat{g}(n+N) + \ldots + \hat{g}(n+1) + \hat{g}(n) = \\ &\qquad = a_0 + a_1 t(n) + a_2\big(t(n) - t(n-1)\big) + \\ &\qquad\quad + a_3\big(t(n-1) - t(n-2)\big) + \\ &\qquad\quad + a_4 i(n) + a_5 \frac{1}{M} \sum_{j=n-1}^{n-M} g_j \end{aligned} \qquad (5)$$

where n is day for which is the consumption calculated, $\hat{g}$ is new predicted gas consumption, N is the length of multi-step – multi-output prediction, $t(n)$ is average daily temperature, $t(n) - t(n-1)$ is difference of today and yesterday daily temperature, $t(n-1) - t(n-2)$ is difference of yesterday and day before yesterday daily temperature, $i(n)$ is information what is day in week, the last element of Eq. (5) calculate arithmetic mean of gas consumption g_j for last M days, $\{n-1, \ldots, n-M\}$ and $a_0, \ldots, a_5$ are model parameters.

[4] The correct shortcut should be UPD of course, but due to mistype in the development author has been forced to propagate the “wrong” shortcut in to the final text to avoid complete misunderstanding and mishmash in methods’ naming.

3.2 Polynomial Model

The polynomial model is one of the simplest models that has been used for prediction of gas consumption and it is based on presumption that gas consumption has polynomial dependence with daily temperature, or with time, or with information what is day in week, or with day in a month, or with month in a year, or with difference of today and yesterday daily temperature, or with difference of yesterday and day before yesterday daily temperature, or with difference of today and yesterday gas consumption, or with difference of yesterday and day before yesterday gas consumption, and so on. From all these possibilities the dependence with daily temperature shown on Fig 3 with mutual correlation on Fig. 4 has been selected and used for prediction of gas consumption.

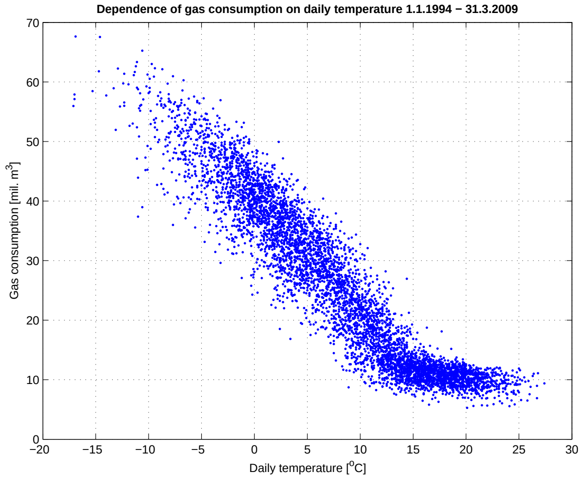

Figure 3: Correlation between gas consumption and daily temperature (red and blue bars), normalized gas consumption (black line), normalized and inverted daily temperature (yellow line)

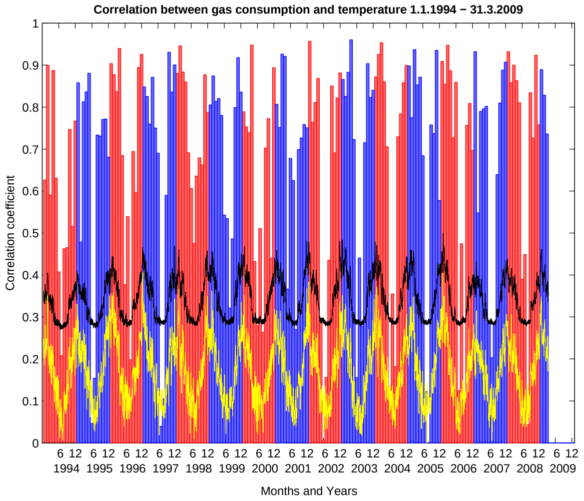

Figure 4: Dependence of gas consumption on daily temperature

3.3 Autoregressive Model

There is no specific configuration for input or output data set. Important for autoregressive models is the part about algorithm of composing the pattern matrix and application of SVD and QRcp algorithm to reduce number of parameters in calculated model. The calculation of autoregressive model has been split into three approaches: ARF – calculation of full autoregressive model with no parameter reduction, this model can be called the base or full autoregressive model, two subset models follows: ARO – autoregressive model with reduced number of parameters, that have not been recalculated. And finally ARR – model with reduced re-calculated parameters. Re-calculation means that parameters of reduced autoregressive model has been calculated again with appropriate order of reduced and permutated pattern matrix.

3.4 Linear Neural Network

Algorithm of one layer linear neural network from has been used with no specific configuration for input or output data set.

3.5 Adaptive Linear Element

Algorithm of adaptive linear element has been used with no specific configuration for input or output data set. Because adaptive models have no learning or verification data set both sets have been put together and used as one processed time line. In the ADALINE architecture has been included bias and calculation has been done with adaptive statistical predictor or so-called slave filter, see section 3.9.

3.6 Feed-Forward Neural Network

The model used for prediction of gas consumption in the Czech Republic had arbitrary number of inputs and outputs, according to the common parameters set. Number of hidden neurones in one hidden layer was 2, 4, 6 or 8. Before the first calculation step it is necessary to have some values in the weight matrix. For initialization of the weight matrix the matrix of random values is often used. Every feed-forward neural network has been calculated with 101 different random weight initialization. After learning phase networks with singular or nearly singular matrix has been removed and the other models have been verified on verification data set. After that the best results with network parameters have been stored. All feed-forward neural networks have been calculated without bias, the first transfer function has been hyperbolic tangent sigmoid and in the second layer linear transfer function has been used. As the network training function algorithm of Levenberg-Marquardt backpropagation has been used.

3.7 Elman Neural Network

Architecture of Elman neural network is very close to the feed-forward neural network excluding the recurrent loop. That is the reason why calculation philosophy has been very close to calculation of feed-forward networks. There could be negative number of hidden neurones used for finding of the most suitable number of hidden neurones as well. Analysing of 101 weight initialization and selection of the best one has been made to. Same as feed-forward network the Elman network has one hidden layer, no bias and hyperbolic tangent sigmoid has been the first transfer function and as the second linear transfer function has been used. As the network training function the algorithm of gradient descent with momentum and adaptive learning rate backpropagation has been used.

3.8 Recurrent Neural Network with RTRL Algorithm

The last calculated model has been recurrent neural network with RTRL algorithm and architecture shown on Fig. 5, This type of neural network is a member of adaptive models. In the network architecture calculations have been done with slave filter. The RTRL algorithm has been programmed according to the algorithm listed in [7, 8, 1]. Hyperbolic tangent sigmoid function has been used for recurrent neural network with real-time recurrent learning algorithm as transfer function. But the algorithm of standard `tansig` function used in the MATLAB environment had to be improved. First improvement has been ability to set own slope s and range r of the sigmoid function and the second has been decreasing of calculation time at least two times. Calculation of `tansig` takes about 0.19 ms and calculation of new `tansigfree_fast` with default slope $s = 1$ and range $r = 2$ takes about 0.09 ms. For recurrent neural network with RTRL algorithm has been used slope $s = 40$ and range $r = 100$. The slope s represents the dynamic of increasing part of sigmoid function. The higher slope values implicate slower increase. Difference between slower and faster increase is shown on Fig. 6 for $s = 10$ and $s = 40$. The range r represents maximal range of sigmoid function, for $r = 100$ the minimal value of function is -50 and maximal 50.

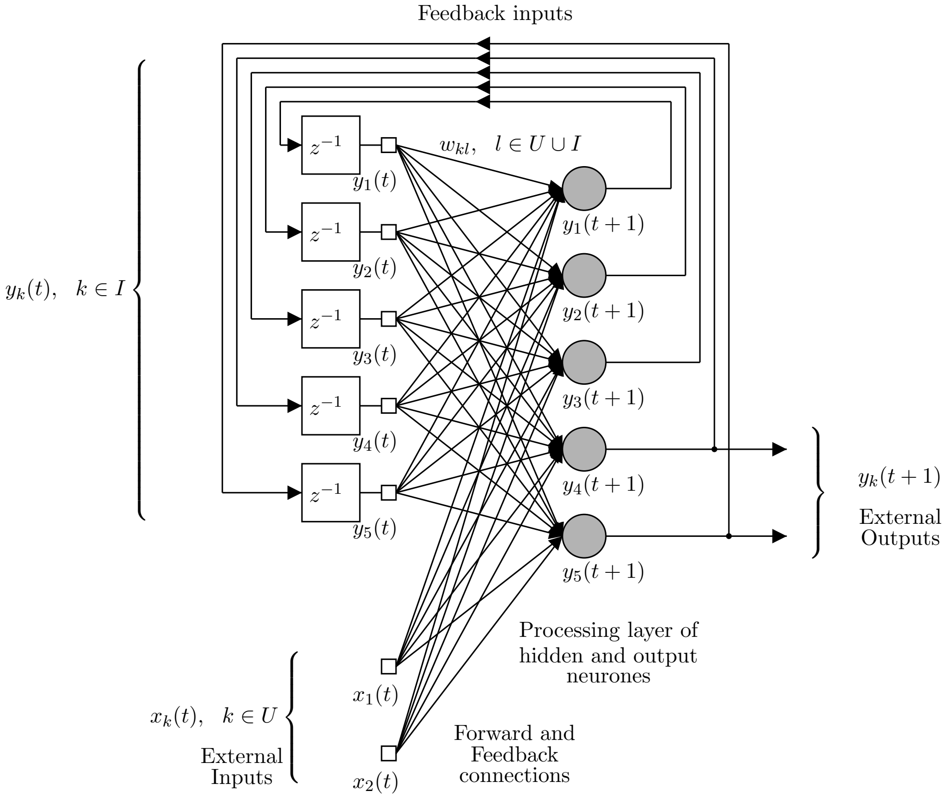

Figure 5: Architecture of the recurrent neural network without subsidiary output layer, simple recurrent neural network

3.9 Adaptive Statistical Predictor

Adaptive statistical predictor shown in Figure 7 is not model used for prediction, but represents very useful algorithm that can be part of the prediction model. The main idea of this predictor is to copy weights from adaptive filter to the slave filter.

The input signal $P(t)$ is split into three parts. The first part is delayed and represent input $P(t-1)$ to the adaptive filter. Output $y(t)$ from this filter, eq.(6) is compared with desired result $T(t) \approx P(t)$ that is the second part of input signal $P(t)$ and producing the error signal $e(t)$. The third part is the input to the slave filter. Output of the slave filter is $y_s(t+1)$, eq.(7) that

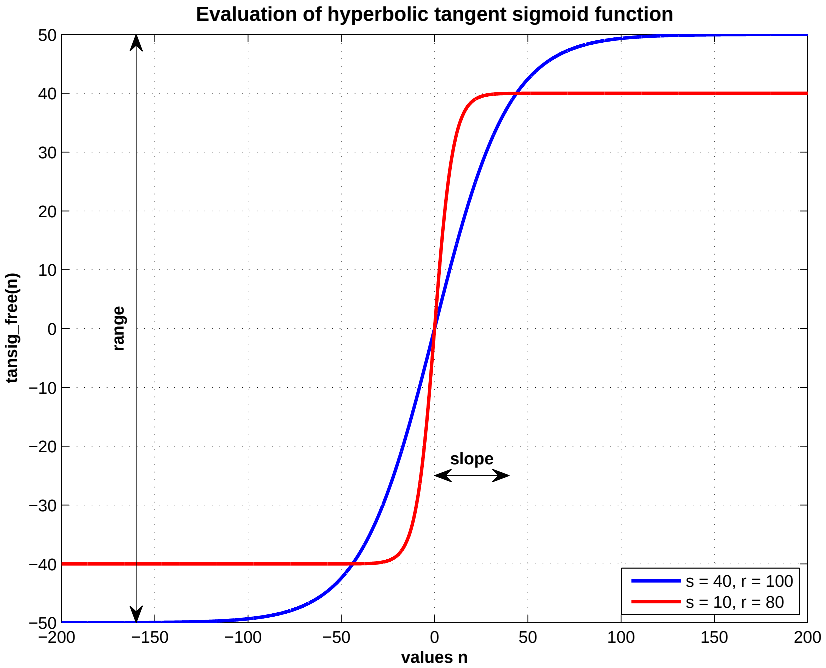

Figure 6: Evaluation of hyperbolic tangent sigmoid function, the higher slope values implicate slower increase

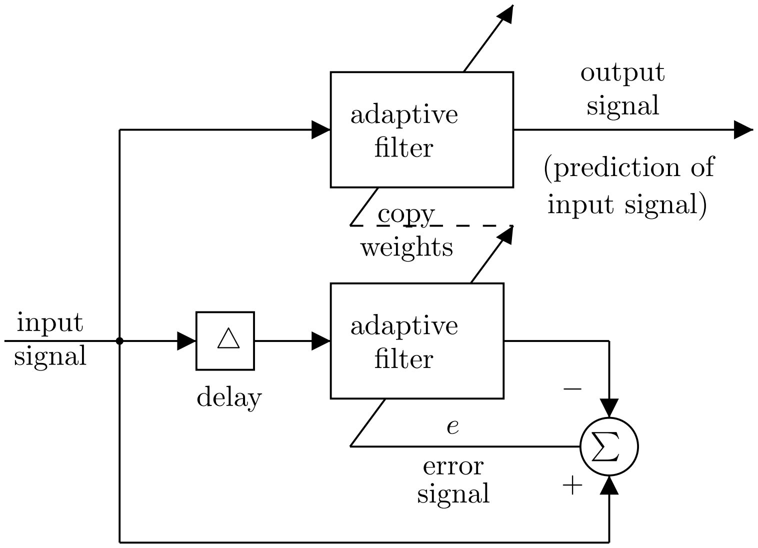

Figure 7: Schema of the Adaptive Statistical Predictor

is real predicted value of the original input signal. Weights $W(t-1)$ in the adaptive filter are set by minimizing of the error signal $e(t)$. Weights of the slave filter $W_s(t)$ are copied from the adaptive filter.

The calculation of Adaptive Statistical Predictor is performed by

$$y(t) = W(t-1)^{\mathrm{T}} P(t-1), \tag{6}$$

then the error signal is calculated $e(t) = T(t) - y(t)$. Weights are adapted by using LMS algorithm by Widrow and Hoff, for more details see [6], $W_s(t) = W(t-1) + \alpha e(t)P(t)$ and the new weights are used in the slave filter $W_s(t) = W(t)$, that finally predict

$$y_s(t+1) = W_s(t)^{\mathrm{T}} P(t). \tag{7}$$

The key feature of this predictive approach is that in real-time the model in time t is adapted only by known values, input values are from time $t-1$ or older and the values from time t are used for model update as desired output. The new weights freshly updated in the adaptive filter are then used in the slave filter that has as the input values from time t or older, this mean actual values. Output of the slave filter are then new predicted values. There is a possibility to compare those predicted values in the real-time calculation in the next time step, because if now is time t we do not exactly know values from future time $t+1$.

4 Selection of Best Models

The final step in prediction of the gas consumption in Czech Republic included the comparison of all calculated models. For this task I have used the multicriteria decision, that has been based on pareto optimality and AIA technique. For all model 25 different information criteria listed in Tab. 4 have been calculated. The criteria are marked in column MCD by "check mark ✓" showing which criteria have been used for selection of the best model by multicriteria decision making and which not. Column labeled "Best is" gives us the information whether the minimal or maximal values of corresponding criteria are desired for the best performance of this criteria.

Table 4: All calculated information criteria, in column MCD are criteria used for selection of the best model by multicriteria decision making

MCD	Information Criteria	Best is
	mean error	min
✓	summed squared error	min
✓	mean squared error	min
✓	SE1	min
✓	SE2	min
✓	SE3	min
✓	0.25 Quantile	min
✓	0.50 Quantile	min
✓	0.75 Quantile	min
	mean - median	min
	minimal error	min
	maximal error	max
✓	standard deviation	min
✓	R-squared	max
✓	skewness	min
✓	kurtosis	min
	in5% absolute value	max
	out5% absolute value	min
✓	in5% relative value	max
	in2.5% absolute value	max
	out2.5% absolute value	min
✓	in2.5% relative value	max
	in1.25% absolute value	max
	out1.25% absolute value	min
✓	in1.25% relative value	max

Most of used criteria, are the well-known, but the last set of criteria in5% absolute/relative value is new type of criteria used for model comparison. In5% criteria shows how many predicted values have ±5% deviation from real absolute value at most. It means that if the real value is in example 100 then the predicted values must be in range from 95 till 105. Of course that out5% criteria is the rest of values that does not fit into in5% criteria. Three variations of this criteria has been used, in5% where is allowed 5% prediction error at most, in2.5% with 2.5% prediction error and at last in1.25% with 1.25% prediction error.

Selection of the best models has been provided in three steps. In the first step the MCD has been applied to the all models according to the model's main properties (common parameters), that are represented by *hist* – length of input data set, *gas* – number of values of gas consumption in the model's input, *temp* – number of values of daily temperature in the model's input, *out* – number of the model's outputs, *time* – length of prediction time, *hide* – number of hidden neurones and also applied according to each model type (UDP, AP, ARF, ARO, ARR, NL, NLs, ADAL, ADALs, FF, FFs, ELM, ELMs, RTRL, RTRLs). Models have been compared according to their values of 15 selected information criteria. The output of this step have been tables with the best models, but there were groups with more that one suitable model, so the more detail selection had to take place. The second selection step used as an input for the MCD values form the main models properties (common parameters) instead of information criteria. In this step the best model have been looked for according to the model's structure, number of *hist*, *gas*, *temp*, *time*, *hide* and not according to the values of information criteria as in previous step. The best models were models with minimal number of values in input parameters *hist*, *gas*, *temp*, *hide* and with maximal number of values in output parameters *out* and *time*. After this step a few combination remains only with more than one best model. The algorithm of last selection step has been based on mutual comparison between models for number inputs of gas consumption (lower is better) or for comparison of information criteria in5% or for R-square.

5 Programming Tools

All calculations have been programmed in the MATLAB R2007b environment with usage of some toolboxes, especially the Neural Network Toolbox, Statistics Toolbox and Distributed Computing Toolbox. Other used software has been Microsoft Excel 11.0 and TEXLive 2007.

For time consuming problems, that was in our case the calculation of all possible combination of parameters, we have possibility to separate the whole calculation into smaller elements and process them separately. In the MATLAB environment we can provide it using the Distributed Computing Toolbox (DCT). This system is important for communication between the user's computer and the calculation cluster created by the MATLAB Distributed Computing Engine (DCE). The DCT and the MATLAB DCE enable us to coordinate and execute independent MATLAB operations simultaneously on a cluster of computers, speeding up execution of large MATLAB jobs [3, 4].

More detail about distributed computing can be found in [5, 2]. Direct side effect of calculation of prediction models is fully functional MATLAB cluster of eight workers and one job manager providing distributed computing in the Department of Computing and Control Engineering at Institute of Chemical Technology, Prague. As we have been informed this system as been one of the first operating MATLAB cluster in the Czech Republic.

6 Numerical Comparison of Calculated Models

The numerical comparison of calculated models have been done by comparison of the resulting data, that were represented by predicted values of gas consumption. But for comparison of models the information criteria have been used. Criteria have been calculated from predicted values and grouped according to the type of models (UDP, AP, ARF, ARO, ARR, NL, NLs, ADAL, ADALs, FF, FFs, ELM, ELMs, RTRL, RTRLs) and common parameters (*hist*, *gas*, *temp*, *out*, *time*, *hide*). The algorithm of selection of the best model based on multicriteria technique applied to the information criteria is described in section 4 in detail and final result are presented on Tab. 5. Next tables with best models are listed in the thesis. Rows of each table represent different types of models and columns are values of individual common parameters. The last row or column then shows the best model in corresponding group (row or column). Description of the model is in following tables. The "ID" of the best model is composed as *hist-gas-temp-out-time-hide*. The best models for all common parameters including their information criteria are show in Tab. 6 and in Tab. 7.

Tab. 6 and Tab. 7 summarize the absolutely best models from Tab. 6 and in Tab. 7. Detail description of information criteria listed in Tab. 7 is described in section 4. The close analysis of listed models reveals that the most successful algorithm for gas prediction in our study has been artificial feed-forward neural network and the UDP model created at Bilance Center. Very interesting is fact that two identical models has been the best models in different comparison scenarios. The FF model 0031-00-04-01-02-2 has been the best model for the fixed *hist* and fixed *gas*. The UDP model 1826-01-00-01-02-0 has been the best model for fixed *out* and *hide*.

The best prediction results have been achieved for model with one output, that is used for two step prediction (*out* is 1 and *time* is 2). This fact is obvious not only from the model architecture but also from Tab. 7. The sum of criteria in5% and out5% in absolute values gives us the total number of predicted values. And when the relative form of in5% is 100% then all predicted values have prediction error in ±5% area. Models using 4 values of daily temperature and 2 hidden neurones have been successful too. From results it is obvious that complicated and highly sophisticated models (ELM, RTRL) or models with large numbers of input values could not beat easier models. Same statement can be said about some simple models (AP, AR, NL, ADAL).

Table 5: The best models according to the length of the learning data set - reduced table contents

hist	**31**	61	92	184	365	Total
UDP	0031-01-00-01-01-0	0061-01-00-01-03-0	0092-01-00-01-02-0	0184-01-00-01-02-0	0365-01-00-01-02-0	0031-01-00-01-01-0
AP	0031-01-00-01-03-0	0061-01-00-01-02-0	0092-01-00-01-02-0	0184-01-00-01-05-0	0365-01-00-01-01-0	0031-01-00-01-03-0
ARF	0031-09-03-01-03-0	0061-12-00-01-02-0	0092-05-12-01-02-0	0184-16-00-01-02-0	0365-05-05-01-02-0	0184-16-00-01-02-0
ARO	0031-05-02-01-05-0	0061-05-02-01-02-0	0092-03-02-01-02-0	0184-02-03-01-02-0	0365-07-12-01-02-0	0031-05-02-01-05-0
ARR	0031-07-03-01-03-0	0061-09-12-01-02-0	0092-07-00-01-02-0	0184-14-07-01-02-0	0365-16-07-01-02-0	0092-07-00-01-02-0
NL	0031-09-01-01-03-0	0061-05-12-01-02-0	0092-05-12-01-02-0	0184-05-00-01-02-0	0365-05-00-01-02-0	0731-16-00-01-02-0
NLs	0031-03-01-01-02-0	0061-16-00-01-02-0	0092-04-02-01-02-0	0184-16-03-01-02-0	0365-09-02-01-02-0	0031-03-01-01-02-0
ADAL	0031-00-01-01-07-0	0061-00-01-01-05-0	0092-00-01-01-05-0	0184-00-02-01-02-0	0365-00-01-01-01-0	0061-00-01-01-05-0
ADALs	0031-00-02-01-07-0	0061-02-02-01-10-0	0092-00-01-01-07-0	0184-05-04-01-02-0	0365-01-02-01-01-0	0031-00-02-01-07-0
FF	**0031-00-04-01-02-2**	0061-04-04-01-02-2	0092-01-03-01-02-2	0184-01-00-01-02-2	0365-16-03-01-02-4	**0031-00-04-01-02-2**
FFs	0031-02-01-01-02-2	0061-07-00-01-02-2	0092-16-01-01-02-2	0184-01-00-01-02-2	0365-16-04-01-02-2	0731-03-00-01-02-8
RTRL	0031-00-03-02-01-2	0061-00-02-02-10-2	0092-03-00-02-05-2	0184-00-03-02-07-4	0365-05-00-02-02-2	0031-00-03-02-01-2
RTRLs	0031-02-01-02-01-2	0061-00-09-02-10-2	0092-02-12-02-01-2	0184-02-01-02-02-4	0365-04-04-02-02-8	0031-02-01-02-01-2
ELM	0031-02-00-01-02-2	0061-04-01-01-02-2	0092-12-07-01-02-2	0184-03-07-01-02-4	0365-14-00-01-02-2	0061-04-01-01-02-2
ELMs	0031-00-02-01-02-2	0061-02-03-01-02-2	0092-01-04-01-02-4	0184-04-09-01-02-4	0365-09-02-01-02-2	0031-00-02-01-02-2
Total	**0031-00-04-01-02-2**	0061-04-01-01-02-2	0092-01-03-01-02-2	0184-01-00-01-02-2	0365-16-03-01-02-4	**0031-00-04-01-02-2**

The "ID" of the best model is composed as *hist-gas-temp-out-time-hide*. Where *hist* represents length of input data set, *gas* number of values of gas consumption in the model's input, *temp* number of values of daily temperature in the model's input, *out* number of model's outputs, *time* length of prediction time, *hide* number of hidden neurones. Used prediction models have been *UDP* model created at BC (ÚPD), *AP* polynomial model, *ARF* full autoregressive model, *ARO* subset autoregressive model with original parameters, *ARR* subset autoregressive model with recalculated parameters, *NL* linear neural network, *NLs* linear neural network with input data set optimized by SVD, *ADAL* adaptive linear element, *ADALs* adaptive linear element with input data set optimized by SVD, *FF* feed-forward neural network, *FFs* feed-forward neural network with input data set optimized by SVD, *ELM* Elman neural network, *ELMs* Elman neural network with input data set optimized by SVD, *RTRL* recurrent neural network with real time recurrent learning algorithm, *RTRLs* recurrent neural network with real time recurrent learning algorithm with input data set optimized by SVD.

Table 6: The best models for all common parameters

Common Parameters	Algorithm	Model
hist	FF	0031-00-04-01-02-2
gas	FF	0031-00-04-01-02-2
temp	FF	0061-04-04-01-02-2
out	UDP	1826-01-00-01-02-0
time	FF	0365-05-09-01-05-6
hide	UDP	1826-01-00-01-02-0

Table 7: Information criteria of the best models for all common parameters

	hist	gas	temp	out	time	hide
mean	-0.0105	-0.0105	0.0316	-0.0094	0.0609	-0.0094
SSE	0.00206	0.00206	0.00616	0.000478	0.0117	0.000478
MSE	0.00103	0.00103	0.00308	0.000239	0.00587	0.000239
SE1	0.0303	0.0303	0.0456	0.0123	0.0609	0.0123
SE2	0.0321	0.0321	0.0555	0.0155	0.0766	0.0155
SE3	0.0336	0.0336	0.0614	0.0172	0.0853	0.0172
Quantile 0.25	-0.0408	-0.0408	-0.0141	-0.0217	0.0143	-0.0217
Quantile 0.50	-0.0105	-0.0105	0.0316	-0.0094	0.0609	-0.0094
Quantile 0.75	0.0199	0.0199	0.0772	0.00288	0.107	0.00288
mean - median	0	0	0	0	0	0
min	-0.0408	-0.0408	-0.0141	-0.0217	0.0143	-0.0217
max	0.0199	0.0199	0.0772	0.00288	0.107	0.00288
std	0.0429	0.0429	0.0645	0.0174	0.0658	0.0174
R-square	0.996	0.996	0.989	0.999	0.98	0.999
skewness	0	0	0	0	0	0
kurtosis	1	1	1	1	1	1
in5%	2	2	2	2	2	2
out5%	0	0	0	0	0	0
in5% [%]	100	100	100	100	100	100
in2.5%	2	2	2	2	2	2
out2.5%	0	0	0	0	0	0
in2.5% [%]	100	100	100	100	100	100
in1.25%	2	2	2	2	2	2
out1.25%	0	0	0	0	0	0
in1.25% [%]	100	100	100	100	100	100

7 Behaviour of Predicted Models

Graphical comparison of the four best models is show on Fig. 8. Next figures with best models are listed in the thesis. Study of lower Fig. 8 in the light blue prediction area reveals precise prediction, because real and predicted values are nearly the same. The dark green line covers blue line and both lines are in gray in5% area. Very good result is obtained by long time prediction shown in Fig. 8, that is whole in the in5% area along its evolution.

Modification of daily temperature on Fig. 8 has been used for presentation purpose only, to show daily temperature and predicted gas consumption on one scale. The increase of daily temperature is caused by the inverse presentation of daily temperature. The real temperatures in days 4333–4337 (11–15 Nov 2005) are shown in Tab. 8 and have decreasing trend. In addition,

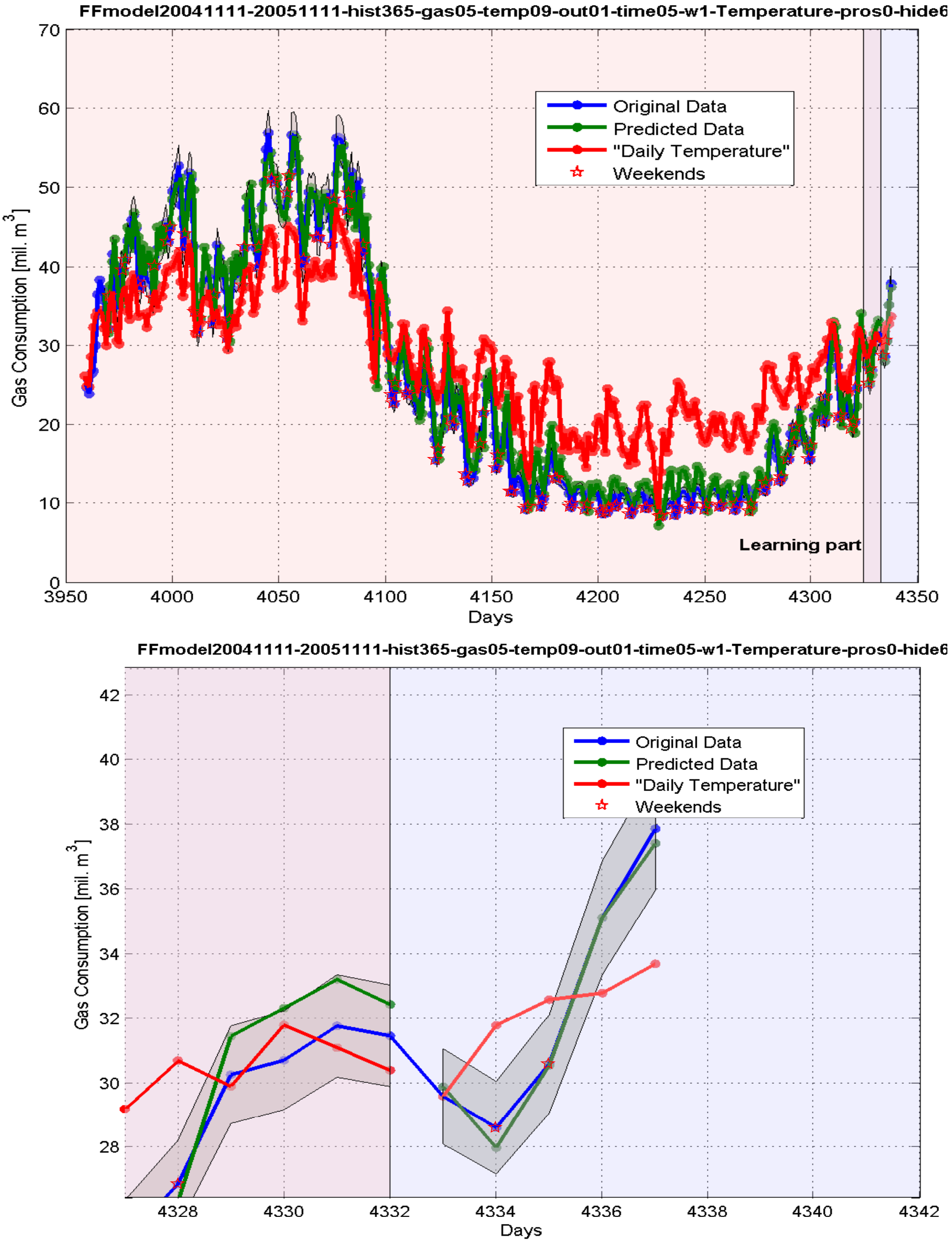

Figure 8: Prediction model FF *hist*365-*gas*05-*temp*09-*out*01-*time*05-*hide*6 has been the best model for common parameter *time* Prediction in learning part is filled by light pink color, verification part is filled by light blue , real data of gas consumption are **blue line**, predicted values are **dark green line**, **red line** are data of daily temperature normalized and inverted (in reality the evolution of temperature is completely opposite), two sequential red stars ☆ representing two days during weekend – Saturday and Sunday, the area of in5% criteria is show in light gray color mixed with corresponding background.

trend lines of gas consumption and daily temperature are in conformity with mutual correlation. This is caused by the fact that gas consumption usually increases when days are getting colder.

Table 8: Real values of gas consumption and daily temperature on 11–15 Nov 2005

absolute day	4333	4334	4335	4336	4337
day in a month	11	12	13	14	15
day in a week	Fri	Sat	Sun	Mon	Tue
daily temperature	6.5	4.3	3.5	3.3	2.4
gas consumption	29.55	28.58	30.56	35.08	37.81

8 Proposed Model for Prediction

The deep analysis of all gained results reveals, that the most suitable model for one or multiple day prediction of gas consumption in Czech Republic is the artificial feed forward neural network. According to the architecture of the best models summarized in Tab. 6 for the good prediction result the structure of the neural network's input needs to have more input values of daily temperature than values of gas consumption. One output neurone with repeating prediction gives the good results too. It is suitable to have number of values used for learning of the neural network in range from one month to one year when speaking about data with sampling period one day. Currently used UDP model from BC has also very good results.

Not recommended are complicated or highly sophisticated models (ELM, RTRL) as well as models with large numbers of input values or simple models (AP, AR, NL, ADAL). Good results have not been received also by adaptive models (ADAL, RTRL).

9 Conclusions

For the prediction of gas consumption in the Czech Republic different linear and non-linear models have been used. The classical and adaptive predicted approach have been compared and the large portfolio of model's parameters has been tested. The following models have been used: model created and used at BC (ÚPD), polynomial model, full autoregressive model, subset autoregressive model with original parameters, subset autoregressive model with recalculated parameters, linear neural network, linear neural network with input data set optimized by SVD, adaptive linear element, adaptive linear element with input data set optimized by SVD, feed-forward neural network, feed-forward neural network with input data set optimized by SVD, Elman neural network, Elman neural network with input data set optimized by SVD, recurrent neural network with real time recurrent learning algorithm and recurrent neural network with real time recurrent learning algorithm with input data set optimized by SVD.

All listed models have same conditions for calculation:

- model's input has been composed from various number of values of gas consumption, daily temperature and information about a day in a week
- neural networks with hidden layers have different number of hidden neurones
- model's output has variability in number of elements for receiving results of the multistep prediction

For the same reason prediction steps have been repeated in different number of cycles.

Prediction has been focused to the forecasting of gas consumption for 11$^{\text{th}}$ November 2005 and further to the future. For this prediction different lengths of learning data sets have been used. Totaly it has been investigated 178 200 sets of parameters for 15 models giving 2 673 000 models. From this set the best models have been selected by the multicriteria decision. For the prediction of gas consumption it has been recommended to use feed forward neural network, which had more input values of daily temperature than values of gas consumption in the structure of the neural network's input. Further networks had one output neurone with repeating prediction and number of values used for learning of the neural network have been recommended to have in range from one month to one year, in case of prediction of data with sampling period one day. Very good results have been observed in the UDP model from BC, too.

Newly defined information criteria in5% had an important role in the result comparisons. This criterion shows how many predicted values have the ±5% deviation from real absolute value at most.

For the calculations MATLAB Distributed Computing Toolbox has been used giving us the opportunity to calculate large and time consuming task in the cluster of several computers.

10 Selected Source Codes

Selected source code for MATLAB Version 7.5.0.342 (R2007b) and later. Listed source codes are cited in the text, rest of codes is located on website `http://dsp.vscht.cz/pavelka`

```matlab
function out1 = tansigfree_fast(in1,in2,in3,in4)
%TANSIG Hyperbolic tangent sigmoid transfer function.
% Modified by Ales Pavelka - modification of my tansigfree for faster
% calculation
%
% Syntax
%
%   A = tansigfree_fast(N,FP)
%   dA_dN = tansigfree_fast('dn',N,A,FP)
%
% Description
%
%   TANSIGFREE_FAST is a neural transfer function.  Transfer functions
%   calculate a layer's output from its net input.
%
%   TANSIGFREE_FAST(N,FP) takes N and optional function parameters,
%     N - SxQ matrix of net input (column) vectors.
%     FP - Struct of function parameters (ignored).
%   and returns A, the SxQ matrix of N's elements squashed into [-1 1].
%
%   TANSIGFREE_FAST('dn',N,A,FP) returns derivative of A w-respect to N.
%   If A or FP are not supplied or are set to [], FP reverts to
%   the default parameters, and A is calculated from N.
%
% Examples
%
%   Here the code to create a plot of the TANSIGFREE_FAST transfer function.
%
%     n = -50:0.1:50;
%     fp.slope = 10;
%     fp.range = 20;
%     a = tansigfree_fast(n,fp);
%     plot(n,a)
%
%   Here we assign this transfer function to layer i of a network.
%
%     net.layers{i}.transferFcn = 'tansigfree_fast';
%
% Algorithm
%
%     a = tansigfree_fast(n) = 2/(1+exp(-2*n))-1
%
%   This is mathematically equivalent to TANH(N).  It differs
%   in that it runs faster than the MATLAB implementation of TANH,
%   but the results can have very small numerical differences.  This
%   function is a good trade off for neural networks, where speed is
%   important and the exact shape of the transfer function is not.
%
% See also SIM, DTANSIG, LOGSIG, TANSIGFREE_FAST, TANSIG.

switch nargin
    case 1
        fp.slope = 1;
        fp.range = 2;
        out1 = apply_transfer(in1,fp);
    case 2
        out1 = apply_transfer(in1,in2);
    case 4
        out1 = derivative(in2,in3,in4);
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Apply Transfer Function
function a = apply_transfer(n,fp)
a = fp.range ./ (1 + exp(-2/fp.slope*n)) - fp.range/2;
i = find(~isfinite(a));
a(i) = sign(n(i));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Derivative of Y w/respect to X
function da_dn = derivative(n,a,fp)
da_dn = 2*fp.range./(1+exp(-2/fp.slope*n)).^2/fp.slope.*exp(-2/fp.slope*n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

ALGORITHM 1: Function tansigfree_fast.m: modified calculation of hyperbolic tangent sigmoid transfer function

11 Acknowledgements

The work has been supported by the research grant of the Faculty of Chemical Engineering of the Institute of Chemical Technology, Prague No. MSM 6046137306.

12 Bibliography

[1] Jonathon A. Chambers and Danilo P. Mandic. *Recurrent Neural Networks For Prediction: learning algorithms, architectures, and stability.* John Wiley & Sons, Ltd., Chichester, 2001.

[2] J. Krupa, A. Pavelka, O. Vyšata, and A. Procházka. Distributed Signal Processing. In *Proc. of 15th Annual Conf. Technical Computing Prague 2007*, pages 81/1–6, 2007.

[3] The MathWorks. *Distributed Computing Toolbox, Users Guide.* The MathWorks, Inc., Natick, MA, version 2 edition, March 2006.

[4] The MathWorks. *MATLAB Distributed Computing Engine, System Administrators Guide.* The MathWorks, Inc., Natick, MA, version 2 edition, March 2006.

[5] A. Pavelka and A. Procházka. Distributed Computing in Data Processing. In *Proc. of 14th Ann. Conf. Technical Computing 2006*, pages PP/1–12, 2006.

[6] Bernard Widrow and Michael A. Lehr. 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9):1415–1442, September 1990.

[7] Ronald J. Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2:490–501, 1990. http://www.ccs.neu.edu/home/rjw/pubs.html.

[8] Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1:270–280, 1989. http://www.ccs.neu.edu/home/rjw/pubs.html.

Ing. Aleš Pavelka, Prof. Aleš Procházka
Institute of Chemical Technology, Prague
Department of Computing and Control Engineering
Technická 1905, 166 28 Prague 6
Phone.: 00420-2-2435 4198, Fax: 00420-2-2435 5053
E-mail: ales.pavelka@volny.cz, A.Prochazka@ieee.org
WWW: http://dsp.vscht.cz