Enhanced MIA-GMDH Algorithm

Petr Buryan[1]

[1]*Gerstner Laboratory, Department of Cybernetics,*
Czech Technical University, Technicka 2, 166 27 Prague, Czech Republic.

buryan@labe.felk.cvut.cz

Abstract. *The paper presents a new methodology which is an enhancement of MIA algorithm of self-organizing polynomial Group Method of Data Handling (GMDH). Classical MIA algorithm suffers mainly by quick loss of the layer diversity resulting in almost homogenous layer output which strongly prohibits reaching any improvement in the next layers of the network and has negative impacts on the stability of transfer functions of the nodes (gained by least mean squares method). The impact is mainly non-stable behaviour and loss in quality of the output of the model as a whole. Several specific improved features were therefore applied in order to improve the behaviour of the algorithm. The enhancements described bellow are mainly semi-randomized selection approach to layer pruning, coefficient rounding and thresholding schemes. The usefulness of proposed enhancements is supported by experimental results of time series analyses.*

Keywords

Inductive modelling, GMDH algorithm, time series prediction, self-organizing polynomial networks.

1 Introduction

Despite the fact that the ability of human brain to gather and process information is permanently developing, it simply cannot compete with the technological boom of the last decades. Due to this information explosion of today people are getting more than overwhelmed by large amounts of data that contemporary technology can measure or just gather as a side effect of its activity. This ability to collect precise and valuable datasets has in many cases turned into an obstacle prohibiting users from gaining information about the processes and systems generating the data sets. The most frequently used data analysis methods are based on statistical approach and are either too simple to discover full structure of generating processes or too complex and time consuming to carry out. These challenges open up new areas for methods of machine learning which seem to be viable alternative to classical approaches bringing automated analysis that does not require special knowledge from the user. This allows also considering and seeking for more complicated models within given datasets. Enahnced algorithm of Group Method of Data Handling (GMDH) used in this paper presents one of such approaches.

2 Theoretical Part

2.1 Classical MIA GMDH approach

The algorithm of Group Method of Data Handling (GMDH) [1, 2] was first introduced by Ivachknenko in 1966. Its main purpose is identification of relations in large complex non-linear multidimensional systems, their approximation and prediction. It searches for optimal structure within the space of multi-polynomial functions g: $\boldsymbol{R}^n \rightarrow \boldsymbol{R}$ which it realizes as a multilayered polynomial network. So the main idea is to obtain a mathematical model of the analyzed object (which can be later used e.g. for further predictions). Ivakhnenko came from the Kolmorov-Gaborov sentence, which proves that every function $y_n = f(X)$ can be can be represented by an infinite Volterra-Kolmogorov-Gabor (VKG) polynomial of the form:

$$y_n = a_0 + \sum_{i=1}^{M} a_i x_i + \sum_{i=1}^{M}\sum_{j=1}^{M} a_{ij} x_i x_j + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M} a_{ijk} x_i x_j x_k \ldots \qquad (1)$$

where $X(x_1, x_2, ..., x_M)$ is the vector of input variables and $A(a_1, a_2, ..., a_M)$ is the vector of coefficients

or weights. This is the discrete-time analogue of a continuous time Volterra series and can be used to approximate any stationary random sequence of physical measurements.

The main function of GMDH is based on forward propagation of signal through nodes of the net similar to principle used e.g. in classical neural nets [7] – input signal is passed to input nodes the outputs of which are then distributed through the structure to upper layers where appropriate mathematical combinations are carried out. Every layer consists of simple nodes each of which performs its own polynomial transfer function and passes its output to nodes in the next layer. The output of the last layer (almost always consisting of only one node) is the output of the whole net.

The coefficients of nodes' transfer functions are estimated in the learning phase during which the whole structure is being automatically built up. This inductive approach to the model structure determination reduces the amount of a priori knowledge required from the user and allows for automatic selection of a structure that follows best given dataset. Classical approaches [1,2] employ linear regression to set these coefficients. Units with the smallest error of the output signal form the final layer, the rest is cut off.

It is clear that not prunning the layers quickly leads to great expansion of state space to be searched and causes the algorithm to be suitable only for simpler models. Measures taken to prevent such expansion of MIA-GMDH algorithm include mainly implementation of node selection on each layer according to selected criterion (e.g MSE). Only the best nodes for each MIA-GMDH layer are retained and their outputs form input to the next layer. Beginning from first layer, the process is continued until the best approximation of the last layer is worse than the best approximation of the previous one or limit of number of layers is reached. The final network is created by connecting nodes surviving prunning from layer to layer up to the output layer. As a result a feedforward polynomial network (like e.g. on Fig.1) is created.

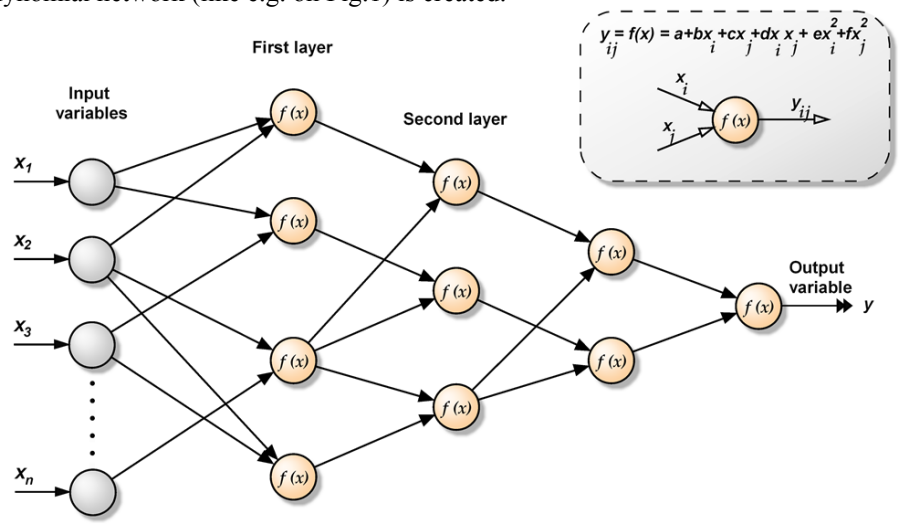

Fig.1 Prunned MIA-GMDH net

2.2 Enhanced MIA algorithm

Enhancements were applied in order to improve accuracy and also robustness of the algorithm in work [3]. One of the main cause of problems can be seen in the prunning process within the run of algrotihm. The selection strategy based upon selecting the nodes only according to their quality based purely on the evaluation criterion (most often MSE) brings along following two important drawbacks.

1. Throwing away nodes that might play an important role in the model should their outputs (and through them their inputs as well) be combined on the higher level results in "loosing" these inputs.
2. Quick unification of the layer output (outputs of all nodes in the layer) leads to bad performance of regression on next level (small diversity of the layer outputs causes unstable regression - due to *very* large value of condition number of matrix $X^T X$ to be inverted.

It seems to be obvious that finding out and employing some heuristics that could drive the prunning process better than evaluation criterion gradient should lead to improvements of the algorithm as a whole. A similarity to such layer pruning can be seen e.g. in evolutionary algorithms, where also only some of the members form last reached population are selected to survive, the rest being thrown away. By seeing the problem from this point of view, another point comes into focus. It is the automatical selection that can be utilised more just to

reduce the time demands of the algorithm. One can simply implement various functions into the nodes and let the algorithm itself choose, which one is the best for current problem..

However, a well known fact is, that the selection of the best members only *does not* make sure that the best solution will be reached in general. Keeping the population diversity on acceptable level is one of the main issues of evolutionary techniques and might be an inspiration for GMDH as well. Realising this fact, enhanced MIA GMDH algorithm (eMIA-GMDH) was proposed to deal with introduced problems. Trying to offer solution to problems with selection, specific improved features of the eMIA-GMDH-type network are described in subsequent subsections of the paper. Following major features of the modified algorithm considering this diversity problem are discussed in the paper:

- a new selection strategy based on semi-randomised selection approach to pruning (in which only a part of the best nodes is selected while the rest is chosen randomly, which gives much better pruning solution in each layer);
- coefficient rounding and thresholding scheme (in which, coefficients that are very large or smaller than a specified threshold are rejected to help to stabilize the regression, limit overfitting and improve genral robustness of the final model);
- non-linear regression possibility for the nodes‘ transfer functions.

Employing these enhancements brings along considerable improvements in the net behaviour, not only in regression stabilisation but in the quality and robustness of the final output as well. This fact was proven also by experiments with time series that are described in the rest of the paper. The whole algorithm [3] along with other features such as graphical interface was developped in Matlab environment and an example of structure of final model can be found in Figure 3.

3 Enhancements of the MIA GMDH algorithm

3.1 Nodes selection during layer pruning

During pruning, each node is evaluated by selected criterion. The final evaluation is calculated as a weighted average of the criterion not only over testing data but also over training part of the dataset. The weights correspond with relative sizes of both parts, i.e.

$$C_{tot} = \alpha \cdot C_{train} + (1-\alpha) \cdot C_{test}, \qquad (2)$$

where $\alpha = \frac{size(\text{training data set})}{size(\text{whole data set})}$, C_{train} and C_{test} are values of selected criterion over relevant part of the dataset.

By selecting only the best nodes from each layer, the output of the layer gets almost homogenous which holds especially for the higher levels. The result of such diversity loss is that almost all the nodes have almost the same output objective function value. Consequently, the least square method gets unstable rejecting enormous amount of coefficients during regression and the whole GMDH net shows a tendency to overfitting.

3.1.1 Diversity problem

Consider following problem of linear regression using least squares. A linear model in the form of $Y = X.b$ is expected, where vector b is the vector of regression coefficients, X is a matrix of input data with rows corresponding to observations and columns to predictor variables and Y is a vector of time series observations. A commonly used approach to determine the unknown vector b utilizes pseudoinverse matrix:

$$b = inv(X^T.X).X^T.Y\ , \qquad (3)$$

which works well in most of the cases. Problems come, when the values of columns in X get close to each other. As a result the condition number of matrix X gets huge as well as do the values of b produced by regression. According to [8] the condition number κ is a measure of stability or sensitivity of a matrix (or the linear system it represents) to numerical operations (stated in other words, it is not wise to trust the results of computations on an ill-conditioned matrix with a very high κ).

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}, \tag{4}$$

where $\sigma_{max}(A)$ and $\sigma_{min}(A)$ are maximal and minimal singular values of A respectively.

This is exactly what happens when selecting just the best nodes on each layer according to some fitness function based on error criterion only. As the nodes with output closest to expected value are preferred the output of higher layers (starting often as low as on layer 3) gets almost unique which leads to ill conditioned matrices, that are constructed for regression from these outputs on higher layers.

Although regression methods are not the only approach of setting the parameters of node transfer functions (see e.g. [9, 10]), discussion not much different from this can be held for any of optimization techniques used because all of them would have to face this *diversity problem*. The main principle of the problem is that it is not possible to gain new information from any analysis of two time series with almost the same values (e.g. differences comparable to the level of measurement noise).

The problem of diversity has been already adressed e.g. in [11]. Several approaches to solving the problem can be seen mainly in:

- modifications to the fitness criterion – modifying the error criterion by some additional criteria (e.g. complexity criterion - Akaiake, Minimum Description Length etc.);
- modifications to selection principles (discussed in following paragraphs).

3.1.2 Selection strategies

Considering the diversity problem, three new pruning techniques, which are embedded in the enhanced MIA-GMDH structure along with the classical "best-of" selection, are discussed in this paper, all of them having been inspired by the evolutionary algorithms:

1) classical (MIA)-GMDH "best-of" approach

2) roulette wheel selection;

3) semi-randomised selection (random factor selection);

4) totally randomised selection.

Classical multilayered iterative algorithm (MIA)-GMDH approach

The approach here is the classical selection of best chosen number of nodes only. This approach is utilized by classical MIA but produces very poor choice for the next layer because it leads to selection of nodes with almost the same output.

Roulette selection inspired by evolutionary algorithm (EA) techniques

Employing the *roulette wheel* selection approach inspired by EA techniques (such as in genetic algorithms, etc.) presents an effort to solve the problem of ensuring variability of selection of nodes on the next level.

Semi-randomised selection (random factor selection).

In work [3] the best found solution to the problem of variability is to enrich the selection of the best nodes by randomized selection in the ratio of 1:1. A conclusion was drawn in [3] that best-of selection is needed so that the learning process can follow some clear optimization rule while the randomised selection ensures the layer diversity, thereby stabilizing the regression and retaining information from nodes that would be otherwise lost for future potential use.

Another effect of keeping the "best-of" approach is the fact, that the unsuitable nodes, that might be left in the layer by random selection process may be cut off post facto – at following layers the nodes these will be connected to might show too big errors and be removed which means removing their input nodes as well.

Totally randomised selection

The totally randomized selection approach represents another effort to solve the problem of ensuring variability of selection of nodes on the next level. This type of selection that can be considered as a special case

of semi-randomised selection, was implemented mainly in order to prove the fact, that the choice of at least some of the best nodes has sense (e.g. in the last layer the selection of the best node only is more than reasonable). The best node is allways kept in the layer.

3.1.3 Semi-randomised selection

Previous exhaustive research experimentation [3] show that for semi-randomised selection approach to pruning, the best pruning solution in each layer is gained when the layer after selection constitutes approximately from half of the best nodes while the second half of the layer is randomly selected from the rest of the nodes (no node being selected twice). Finally, a node with unitary transfer function ($y = x_1$) is added to each layer enabling the ouput of the best node of the previous layer to be used also in the following one.

Consequently, this is the approach that has been used in the enhanced GMDH algorithm presented in this paper. The main principle of semi-randomised selection is as follows - when constructing the layer, its size is limited by user defined limit – desired value of nodes to survive in the layer (can be defined either as an absolute number or as a proportion of the total count of nodes constructed from outputs of the previous layer). The reason of limitting number of layer nodes is to prevent unfeasible expansion of nodes on upper layers and to ensure finishing the algorithm in reasonable time. However, this limit has of course its impact not only on the time of obtaining final model but on its accuracy as well.

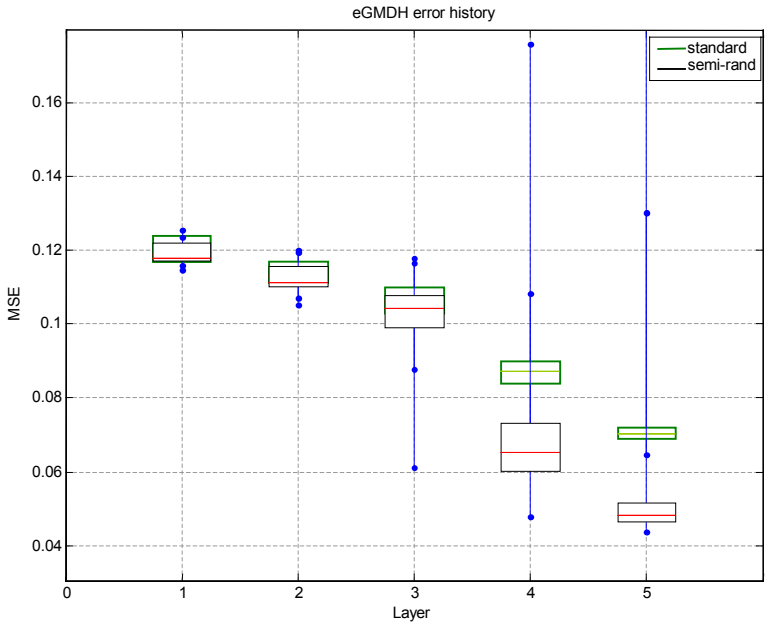

Figure 2 – Box-plot comparison of standart and semi-randomised selection

The effect of employing semi-randomised selection and the differences between it and the best-of approach surely depends on many factors - the problem that is solved (number of intputs, characteristics of the dependencies etc.), setting of the constraints (maximal number of layers, maximal number of nodes in layer) are only some of those. However, a typical situation is depicted on Figure 2 (maximum of five layers i n the net, 100 nodes per layer, 7 initial inputs). A classical box-plot graph shows that semi-randomised selection helps to improve the quality of the layers as a whole – the node errors are smaller and the layer is also more diverse (the boxes are bigger i.e. the nodes are not that simillar – according to chosen criterion - to each other as it results from classical best-of selection). Another impact is limitting the occurance of nodes with unstable transfer function (the worst error of classical approach on layer 5 is out of axis limits).

3.2 Coefficient rounding and thresholding

In the coefficient rounding and thresholding scheme coefficients that are in absolute value very large or smaller than specified thresholds are rejected in order to help stabilize the regression. As discussed earlier, there is high pressure to select nodes the outputs of which are as close to desired output as possible. This consequently leads to joining very similar data as inputs into one node which makes the least squares method work with ill-conditioned input matrix. Final effect are enormous node coefficients (several orders higher than maximal value in input or output data) which are likely to have pairiwise the same value differing in sign.

Unfortunately, this is the reason why from final node output data ill-conditioned input cannot be recognized because the effects of opposite huge coefficients zeroize upon same inputs and the output vector gets even slightly closer to desired data. Evaluation upon the testing dataset might often not discover such behaviour because the testing data are often of the same nature as the training dataset and it is probable that this node might be left within the model (actually splitting data into training and testing dataset presumes simillar nature of both datasets – it makes no sense to train model on totally different data than it shoul be used for).

On the other hand, an ability of generalistion is expected from any modelling method to consider it useful and a node such as the one described in previous paragraph clearly fails on data out of training range as it shows strong signs of overfitting.

Enhanced MIA-GMDH faces this problem with coefficient tresholding and rounding scheme. Such overtrained nodes (and especially their outlying coefficient values) clearly bring no new information to the model. Therfore, large coefficients get tresholded to user defined limits, small coefficients (often arising as a result of huge coefficients in the previous layer) get rounded to zero and it is left to the selection mechanism to judge if such node is still good enough to be kept in the layer. Using rounded polynomial for the node transfer function is another example of emplying the advantages offered by the selection procedure.

Another positive impacts of this operation are lowering the sensitivity of the net to small changes of nodes' parameters (e.g. by rounding off lower orders of the coefficients).This makes easier also the potential final usage of the net – generally speaking, only the transfer function of the GMDH model defined by the coefficients can be used instead of implementing the whole network structure. For the end user, a polynomial (or modified polynomial) function is for sure much easier to implement and can be realised even on standard office software.

3.3 Other enhancements

Other new features were implemented into the eGMDH algorithm in order to make it more flexible and to better accomodate to the use it was developped for - time series analysis. Therefore, two other polynomials can be used on for the node transfer function (apart of the classical quadratic polynomial), non-linear functions were implemented to be used within the node transfer function and several evaluation criterions were implemented as well.

3.3.1 Regression polynomials

There are three types of regression polynomials that are available in the enhanced MIA-GMDH as shown lower. For two inputs to a node following forms of the transfer function polynomial are availible:

(1) Bilinear $a_0 + a_1x_1 + a_2x_2$

(2) Biquadratic $a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2$

(3) Modified biquadratic $a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2$

3.3.2 Types of regression polynomial

Eight types of regression polynomials have been implemented in the enhanced GMDH as shown in Table 1. These were implemented not just to improve the accuracy of the algorithm but also to show that many various alternatives can be incorporated into one net. The choice is thereafter left to the selection process, whole model can finally have form e.g. like the one depicted of Figure 3.

Table 1 Node types (objective functions)

	Polynomial type	**Transfer function (example for two node inputs)**
1	polynomial	$f(X) = f(x1,x2) = a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + gx_2^2$
2	harmonic (cosine)	$f(X) = f(x1,x2) = \cos(a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + gx_2^2)$
3	square root	$f(X) = f(x1,x2) = (a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + gx_2^2)^{1/2}$
4	inverse polynomial	$f(X) = f(x1,x2) = (a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + gx_2^2)^{-1}$
5	logarithmic	$f(X) = f(x1,x2) = \ln(a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + gx_2^2)$
6	exponential	$f(X) = f(x1,x2) = \exp(a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + gx_2^2)$
7	arc tangent	$f(X) = f(x1,x2) = \operatorname{atan}(a + bx_1 + cx_2 + dx_1x_2 + ex_1^2 + gx_2^2)$
8	rounded polynomial	$f(X) = f(x1,x2) = (a_r + b_rx_1 + c_rx_2 + d_rx_1x_2 + e_rx_1^2 + g_rx_2^2)$ *)

*) coefficients for linear combination gained from the regression as in classical polynomial (row 1) are rounded e.g. $a_r = round(a)$

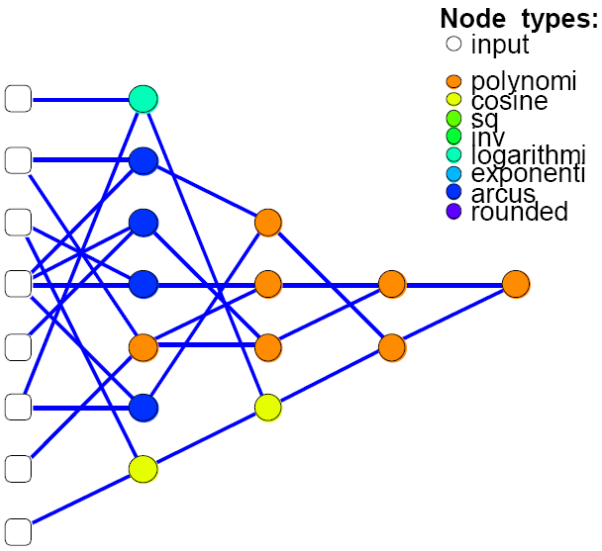

Fig.3 An example of eMIA-GMDH model structure

3.3.3 Node evaluations (objective functions)

Eight optimisation criteria (node fitness functions) were implemented in the enhanced MIA-GMDH and these are listed in Table 2.

Table 2 Implemented node fitness functions

Name	**Definition**
Mean Square Error	$MSE = \frac{\sum (y(t) - \hat{y}(t))^2}{N} = \frac{\sum e^2(t)}{N},$
Mean Absolute Error	$MAE = \frac{\sum \lvert y(t) - \hat{y}(t) \rvert}{N} = \frac{\sum \lvert e(t) \rvert}{N},$
Variation Accuracy Criterion (Ivachnenko's δ^2)	$\delta^2 = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}$
Mean Absloute Percental Error	$MAPE = \frac{\sum \frac{\lvert y(t) - \hat{y}(t) \rvert}{y(t)}}{N}.100\% = \frac{\sum \frac{\lvert e(t) \rvert}{y(t)}}{N}.100\%,$
Maximal Error	$MXE = \max(\lvert y(t) - \hat{y}(t) \rvert)$
Maximal Percental Error	$MXPE = \max(\frac{\lvert y(t) - \hat{y}(t) \rvert}{y(t)})$
Negative Pearsons Coefficient	$r = \frac{SS_{x\hat{x}}}{\sqrt{SS_{xx} SS_{\hat{x}\hat{x}}}},$ **)
R squared	$R^2 = 1 - \frac{\sum_{t=1}^{T} (y(t) - \hat{y}(t))^2}{\sum_{t=1}^{T} (y(t) - \bar{y}(t))^2}$

Where $\hat{y}(t)$ stands for the predicted value, *y(t)* is the actual value and $\bar{y}(t)$ is the average value

**) $SS_{xx} = \sum (y(t) - \bar{y}(t))^2$, $SS_{\hat{x}\hat{x}} = \sum (\hat{y}(t) - \bar{\hat{y}}(t))^2$, $SS_{x\hat{x}} = \sum (y(t) - \bar{y}(t)).(\hat{y}(t) - \bar{\hat{y}}(t))$

4 Experimental Results

Several experiments were carried out with eMIA-GMDH network from which Mackey-Glass series analysis and exchange rate series analysis are presented in this paper. Several classical techniques were used for comparison. These include:

- *classical linear regression* as a representative of most widely used techniques (no special data treatment has been carried out for this method),
- *NC-model (non-changing model)* returning the last-known value as prediction (same as one sample delay),
- classical *backpropagation-trained neural network* with 15 nodes in hidden layer and standard node sigmoid transfer function.

4.1 Mackey-Glass series

The Mackey-Glass series, based on the Mackey-Glass differential equation [8] is widely regarded as a benchmark for comparing the generalization ability of different methods. This series is a chaotic time series generated from the following time-delay ordinary differential equation:

$$\frac{dx(t)}{dt} = -bx(t) + a\frac{x(t-\tau)}{1+x(t-\tau)^{10}}. \tag{5}$$

Following the majority of studies, the series has been generated using the next values for the parameters: a=0.2, b=0.1, and T = 17. As in the mentioned studies, the task is to predict the value of the time series at point x[t+P] from the earlier points (x[t],x[t-6],x[t-12], x[t-18]). If the number of sample steps P is 50, the function to be learned -whose dimension is 4- is:

$$x(t) = f(x(t-50),x(t-50-6), x(t-50-12),x(t-50-18)),$$

while the initial 3500 samples is to be discarded in order to avoid the initialization transients.

Yhe Mackey-Glass series modelling results were summed up in table 3. Due to the fact, that training of neural network and GMDH of the used implementation was influenced by random factor (e.g. random splits of the dataset into training and testing part, random initialization of neural network, semi-randomized selection of eMIA-GMDH nodes), the results in table 3 present always an average of 10 runs of the algorithm over the dataset.

Table 3 Prediction results for Mackey-Glass time series (input history length = 5)

Model	MSE
classical MIA - GMDH, *average of 10 runs* *5 layers, max. 100 nodes per layer, **best-of selection,** polynomial transfer function only*	3.25 E-07
eMIA – GMDH, *average of 10 runs* *5 layers, max.100 nodes per layer, **semi-randomised selection**, polynomial transfer function only,*	3.06 E-07
eMIA – GMDH, *average of 10 runs* *5 layers, max.100 nodes per layer, **roulette selection**, polynomial transfer function only,*	3.15 E-07
eMIA – GMDH, *average of 10 runs* *5 layers, max.100 nodes per layer, **random selection**, polynomial transfer function only,*	3.45 E-07
eMIA – GMDH, *average of 10 runs* *5 layers, max.100 nodes per layer, semi-randomised selection, polynomial transfer function only, coefficient values limited to 1000 (in absolute value) and three decimal places*	3.12 E-07
eMIA - GMDH, *average of 10 runs* *5 layers, max. 100 nodes per layer semi-randomised selection, all transfer functions possible, coefficient values limited to 1000 (in absolute value) and three decimal places*	2.92 E-07
Neural Network, average of 10 runs *Backpropagation, sigmoid transfer function, 15 nodes in hidden layer*	3.34 E-06
Linear Regression	4.56 E-07
NC model	1.21 E-03

Following conclusions can be drawn from the table:

1) Utilising only criterion based selection principle and thereby selecting just the best n nodes on each layer does not lead to the best results of the final model. So, there is sense in looking for a heuristics or – generally speaking – a more complex method that could maybe even predict the potential contribution of each node to the model as a whole (if the node would be kept). Among all four implemented selection strategies, semi-randomised has scored the best.

2) Utilisation of non-linear elements (table 1) brings along additional improvement of accuracy. However, one has to be aware of the fact, that network with non-polynomial elements loses one of potential advantages of low-layered GMDH nets – transparent transfer function.

3) Limiting maximal values of coefficients does not bring along severe deterioration of GMDH model accuracy and can be used to make the whole transfer function more transparent and also as an element that helps stabilizing the regression and improves robustness of the model's response..

4.2 Exchange rate predictions

Exchange rate analysis is also one of popular prediction tasks for modern algorithms. This is mainly due to high level of problem complexity as exchange rates change in subject to many influencing factors which simply cannot be incorporated fully into the analyses [5, 6]. However, perhaps the main reason, why the exchange rates balance on the endge predictability is the fact, that these factors are often random in their nature.

Time series of exchange rates of Japanese Yen and U.S. Dollar has been chosen as the subject of analysis. The selection was made mainly due to availibility of data – series of financial indicators of Japan and USA are available long enough and almost in all frequencies. This exchange rate was also not burdened by various transient performances caused by political reasons simillar to those that happened in Europe quite recently (implementation of Euro, broadening of Euro-area, opening of the Eastern Europe, joinder of East and West Germany etc.).

4.2.1 Ex-ante predictions

In order to get closer to real utilisation of the model, ex-ante predictions were constructed instead of ex-post predictions that are often stated in research papers. Having series $x_1, .. x_n$, ex-post predictions usually choose (often at random) some numbers m_i, such that $0 < m_i \leq n$, and remove them from the input dataset. The rest of the data is uhen used for model estimation and quality of model is finally judged by evaluation of model predictions for those "hidden" data. However, this approach leads to the fact that samples occuring later than "unknown" samples in the series are used for the model estimation as well. It is clear that this would be not possible in reality as we simply cannot know what will happen after the time we conststrut the prediction for.

Ex-ante approach that was used in this paper to model the time series works in a more natural way. For each sample that is to be predicted all data coming after it are condemned as unknown as well. Therefore, no bias is added to the model estimation process. On the other hand such strategy is much more time consuming because at each step (for each predicted sample) all models are to be recalculated again and again.

Several types of dependencies can be sought for within financial series. These depend mainly on the sampling frequency of analysed series:

- Exchange rate series of shorter periods seem to be interesting mainly due to perspective of gain from short term speculations. What is complicated about these is the fact that short term exchange rates seem to behave in a rather unpredictable way and are often modelled by a random walk model [4, 5].

- On the other hand, series with longer periods (monthly, quartely) are interesting mainly for the financial managers of companies, because they can help them in their efforts to avoid exchange rate risks. Simply put, the longer the period of the series the stronger dependencies with other economical indicators appear.

Non-statonarity of the daily exchange rates was proven by short autocorrelation analysis (see Figures 4). The graphs show, that characteristics of daily JPY/USD exchange rate series are almost the same as those of random walk process.

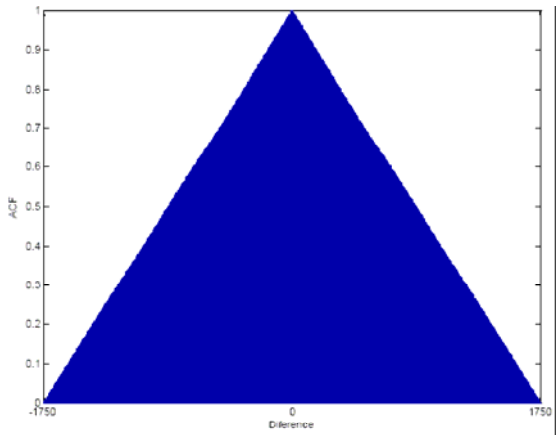

Fig.4a) Autocorrelation function of daily JPY/USD series

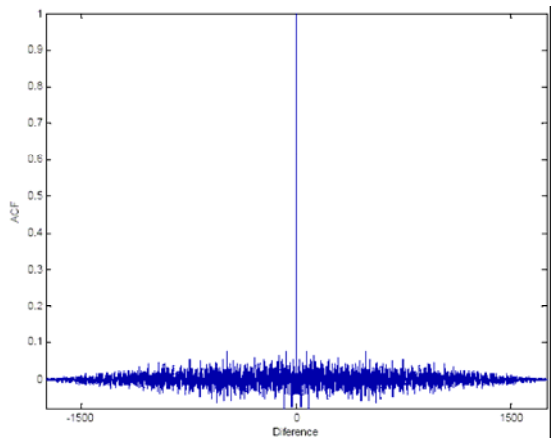

Fig.4b) Autocorrelation function of first differencies of daily JPY/USD

4.2.2 Daily series

Results of daily series predictions were summed up into table 4. Series of approx 9000 samples from 1.1.1971 to 28.2.2007 was used as input data for the analysis. Predictions were evaluated upon interval from 1.1.2007 to 28.2.2007. Several prediction horizons and several lengths of input time windows were used for prediction. As expected, the results were not really stimulative as the NC model (non changing model) was as good as other more sophisticated approaches.

Tab.4 Daily JPY/USD exchange rate prediction, normalised series

Model	MSE . 10^{-4}					
	$XR_{4d}(t+1)$	$XR_{7d}(t+1)$	$XR_{7d}(t+2)$	$XR_{7d}(t+3)$	$XR_{14d}(t+1)$	$XR_{21d}(t+1)$
NC model	0,18	0,18	0,35	0,53	*0,18*	*0,18*
Random walk*)	0,36	0,36	0,54	0,72	*0,36*	*0,36*
Linear regression	0,18	0,18	0,35	0,53	0,18	0,18
Neural network*)	0,18	0,18	0,36	0,55	0,18	0,18
eMIA - GMDHp1) *)	0,18	0,17	0,34	0,52	0,17	0,17
eMIA - GMDHnp2) *)	0,17	0,17	0,33	0,52	0,17	0,17

XR_{Dd} (t+***k***) .. JPY/USD exchange rate series, prediction of ***k*** samples into future from history time window of ***D*** days, normalised series

1)eMIA - GMDHp – 5 layers, 100 nodes per layer (max), only polynomial transfer function

2)eMIA - GMDHp – 4 layers, 100 nodes per layer (max), with non-polynomial transfer function elements,

*) average value from 10 runs

4.2.3 Monthly series

More promising results were expected from analyses of longer-term series. Other relevant series (see table 5) were used as inputs for the analyse (unfortunatelly, monthly average forward exchange rates for such long period were not availible). Analysed dataset consists of average values from the period from January 1980 till February 2007, predictions were made for period from January 2001 until February 2007

Tab.5 Monthly JPY/USD exchange rate prediction, input series

Series	Frequency	Unit
Average monthly exchange rate JPY/USD	monthly	JPY/USD
Discount rates of Japan and USA central banks	monthly	%
Concumer price indicies of Japan and USA	monthly	% vers. 1960 (USA) resp. 1970 (Japan)
GNP of Japan and USA *)	quaterly	current prices, mil. USD
Balance of trade payments of Japan and USA *)	quaterly	current prices, mil. USD

*) converted to monthly sampling by step function

Results of the modelling are to be found in table 6. Apart from already introduced approaches, often praxis from time series analyses - prediction from first differencies - was also utilised. The results of modelling might show that GMDH is really the proper algorithm for the problem as the average percentual error gets near to 1%. This could mean a very good performance, however, the results of NC model (one-step-delay) show, that we deal here with a quite stable series, that do not change much sample to sample. Graphical presentation of the results follow in figure 3.

Tab.6 Monthly JPY/USD exchange rate prediction

Model	MSE	MPE [%]
NC model	6,06	1,64
Random walk *)	23,90	3,37
Linerar regression	9,34	2,08
Neural network *)	6,59	1,75
MIA- GMDH *) 1)	6,51	1,72
eMIA – GMDH 2) *)	**5,27**	**1,21**
eMIA - GMDH – 1st diference *) **)	6,34	1,51

1)MIA - GMDHp – 5 layers, 100 nodes per layer (max),
2)eMIA - GMDHp – 4 layers, 100 nodes per layer (max), with non-polynomial transfer function elements,
*) average value from 10 runs
**) Prediction coming out from the first diference of the series

As can be seen from Figure 5, all predictions fo both eMIA – GMD and neural network for the analysed period from January 2001 until February 2007 fell into the range of 2% deviationfrom the real exchange rate (marked grey the figure), while the results of eMIA-GMDH were slightly better than those of back-propagation neural network or classical MIA-GMDH approach.

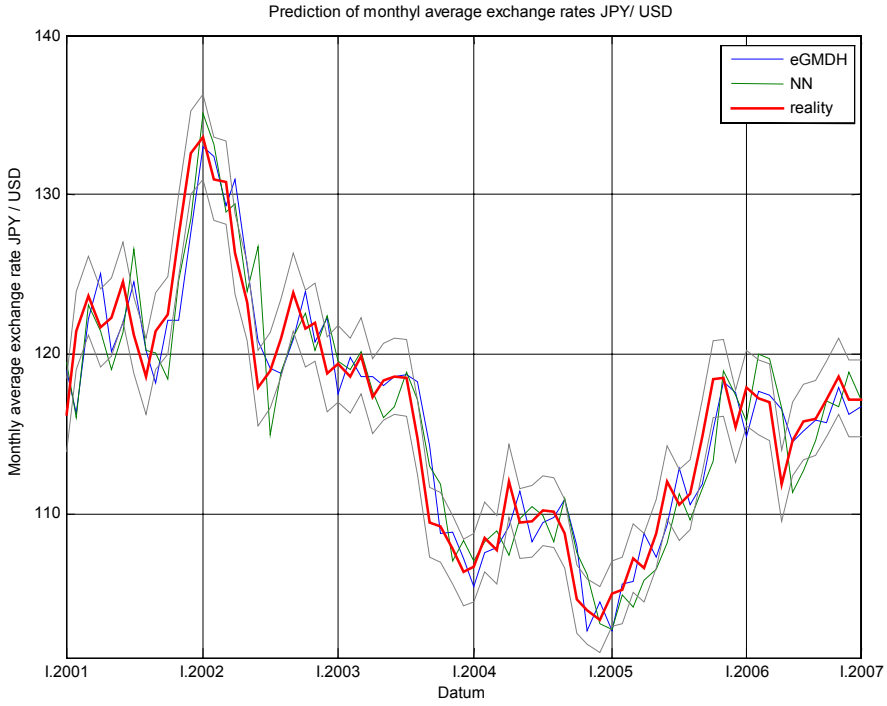

Fig.5 – Results of predicting monthly JPY/USD exchange rate,
(for comparison 2% difference range marked grey)

5 Conclusions

In this paper an enhanced MIA-GMDH-type algorithm was presented along with some time series experiments. The specific features of the enhanced MIA-GMDH-type network presented in this paper include:

- coefficient rounding and thresholding scheme (in which, coefficients that are very large or smaller than a specified threshold are rejected to help stabilize the regression);
- heuristic prunning - semi-randomised selection approach (in which half of the best nodes are selected while the other half are randomly selected, resulting in best pruning solution in each layer);
- high level of flexibility as each node can have a different number of input variables as well as exploit a polynomial of different order;
- various objective functions available.

Attention was paid to selection within the prunning phase in order to stabilize the regression used for node transfer function determination. One of great advantages of the GMDH algorithm is the fact that it is very easy to implement various alternatives into the process of model construction and leave to the automatic selection mechanism to judge if such node is still good enough to be kept in the layer. Using rounded polynomial for the node transfer function is an example of employing the advantages offered by the selection procedure.

Results of studies carried out on Mackey-Glass time series and also JPY/USD exchange rates (daily and monthly series) show that introduced enhancements helped to improve prediction capabilities of the GMDH algorithm which reveals the potential of self-organizing network methodology for use in prediction applications.

References

[1] Ivakhnenko A. G.: Polynomial theory of complex systems. IEEE Transactions on Systems, Man, and Cybernetics, SMC-1(1):364378, 1971.

[2] H. R. Madala, A. G. Ivakhnenko, Inductive Learning Algorithms for Complex Systems Modeling, CRC Press, Boca Raton, 1994

[3] P. Buryan, Time Series Analysis by Means of Enhanced GMDH Algorithm, Dissertation Thesis, CTU Prague, 2006.

[4] Taylor, S., Modeling Financial Time Series, Wiley, New York, NY, 1986

[5] Arlt, J. Finanční časové řady, Grada,Praha 2003

[6] Rosenberg, M.: Currency Forecasting: A Guide to Fundamental and Technical Models of Exchange Rate Determination, Irwin Professional Publishing, 1996

[7] Kröse, B., Smagt, P. van der, An introduction to neural networks. University of Amsterdam, November 1996

[8] Mackey, M. C., and L. Glass, Oscillation and chaos in physiological control systems*, Science,* vol. 197, pp. 287-289, 1977.

[9] PlanetMath [online] http://planetmath.org/encyclopedia/MatrixConditionNumber.html, latest revision 23-05-2007

[10] Kordík P. Group of Adaptive Models Evolution, Technical Report, ČVUT Praha 2005. [online] http://neuron.felk.cvut.cz/game/index.html, latest revision 23-05-2007

[11] Maintaining Diversity in Population of Evolved Models, Drchal J., Šnorek M., Kordík P. In Proceedings of 40th Spring International Conference MOSIS 06, Modelling and Simulation of Systems. Ostrava: MARQ, 2006, vol. 1, p. 113-120. ISBN 80-86840-21-2.