DOI: 10.18267/j.pep.398

EXCHANGE RATE PREDICTIONS IN INTERNATIONAL FINANCIAL MANAGEMENT BY ENHANCED GMDH ALGORITHM

Josef Taušer, Petr Buryan*

Abstract:
Exchange rate forecasting is an important financial problem that is receiving increasing attention nowadays especially because of its difficulty and host of practical applications in globalising world of today. The paper presents an enhanced MIA-GMDH-type network, discusses its design methodology and carries out some numerical experiments in the field of exchange rate forecasting. The method presented in this paper is an enhancement of self-organizing polynomial Group Method of Data Handling (GMDH) with several specific improved features - coefficient rounding and thresholding schemes and semi-randomized selection approach to pruning. The experiments carried out include exchange rate prediction and hedging case study where the predictions were used for financial management decision simulation of a virtual company. The results indicate, that the method shows promising potential of self-organizing network methodology. This implies that the proposed modelling approaches can be used as a feasible solution for exchange rate forecasting in financial management.

Keywords: GMDH, self-organizing polynomial networks, time series analysis, exchange rate prediction, FX hedging

JEL Classification: G17, F37

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 by contemporary technology can be measured or just gathered as a side effect of their activity. This ability to collect precise and valuable data has in many cases turned into an obstacle prohibiting users from gaining information about the processes and systems generating the data.

Exchange rate movements and their forecasting became an important financial problem in the world of today and increasing effort of many researchers is focused in this direction. Behaviour of exchange rates is suspected to be of highly non-linear

[*] Josef Taušer, University of Economics, Prague (tauser@vse.cz); Petr Buryan, Czech Technical University, Prague (buryan@labe.felk.cvut.cz). The paper was elaborated within the Research Plan MSM6138439909 “Governance in the Context of Globalized Economy and Society” of the Faculty of the International Relations, University of Economics, Prague.

DOI: 10.18267/j.pep.398

nature and predictions in such environment present very challenging problem to be solved. One of the main difficulties lies in the fact, that exchange rates are affected not only by economic measurable factors, but also non-measurable elements of political or even psychological character plays their important role. These factors interact in a very complex fashion and mainly exchange rate series of short period (daily, weekly) exhibit high volatility, complexity and noise.

As a consequence, many models describing exchange rate movements have been constructed over past decades. Potential utilisation of such predictions does not lie only in the field of scientific research and business speculations. As more and more companies are entering global markets nowadays, the demand for applicable and efficient predictive tools that could give groundwork for international financial management decisions is growing.

The gravity point of current approaches to exchange rate modelling lies nowadays mainly in statistics and econometrics. Exhaustive research was conducted and is still going on allowing to uncover at least some of the complexity of the problem. Results can be seen *e.g.* in extended autoregressive models such as the autoregressive random variance (ARV) model or autoregressive conditional heteroscedasticity (ARCH) model. However, a growing interest in utilisation of machine learning methods such as neural networks cannot go unnoticed. A recent review of neural networks based exchange rate forecasting can be found *e.g.* in Lai, Yu &, Wang (2007).

This paper presents employment of enhanced version of GMDH polynomial network algorithm into the field of financial predictions. The Group Method of Data Handling (or GMDH) was first introduced by Ivakhnenko in the late 1960s (Ivakhnenko, 1971) as a means of identifying non-linear relations between input and output variables. GMDH has since been improved and applied to a host of practical situations (Mueller, Lemke & Ivakhnenko, 1998) which showed that this class of multilayered polynomial networks has proved effective in the field for a wide spectrum of applications for both modelling and prediction. In this study, we adopt enhanced GMDH algorithm and include some experimental studies in exchange rate modelling.

The remaining sections are organized as follows. Section 2 presents the GMDH framework which is the basis for the enhanced MIA-GMDH architecture that is the focus of this article. Section 3 describes the detailed enhancements that resulted in the enhanced MIA-GMDH presented in this article. Section 4 presents the results of experiments carried out, the conclusions from this study are given in Section 5.

2. Group Method of Data Handling (GMDH) Framework

The algorithm of Group Method of Data Handling (GMDH) was firstly introduced by Ivakhnenko in 1966 (Ivakhnenko, 1971). Its main purpose was identification of relations in large complex non-linear multidimensional systems, their approximation and prediction. GMDH searches for optimal structure within the space of multi-polynomial functions $g\colon \boldsymbol{R}^n \to \boldsymbol{R}$ which it realizes as a multilayered polynomial network. The main idea behind the algorithm is obtaining a mathematical model of the analyzed object produced in an automated heuristic driven self-organizing learning process.

Ivakhnenko employed the approach that every function $y_n = f(X)$ can be represented by an infinite Volterra-Kolmogorov-Gabor (VKG) polynomial of the form:

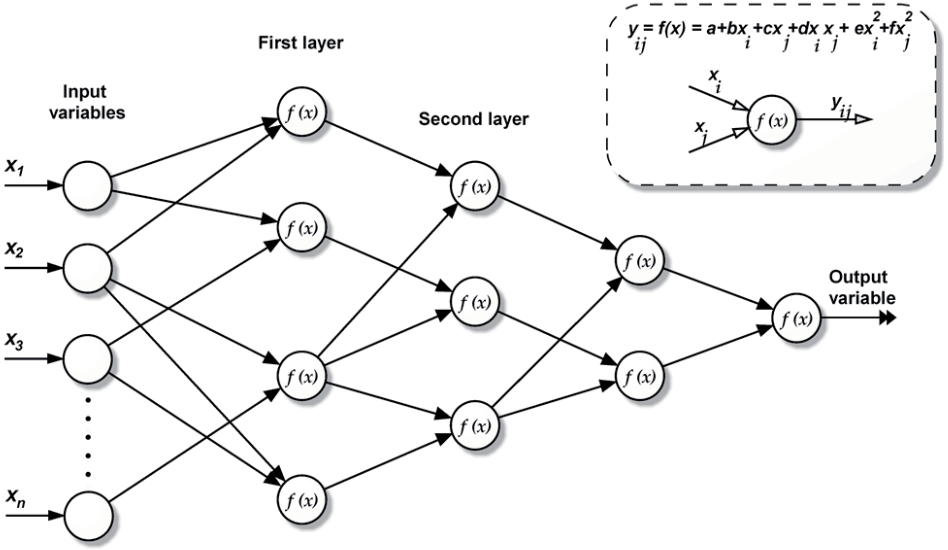
DOI: 10.18267/j.pep.398

$$y_n = a_0 + \sum_{i=1}^{M} a_i x_i + \sum_{i=1}^{M}\sum_{j=1}^{M} a_{M+i+j} x_i x_j + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M} a_{M+M^2+i+j+k} x_i x_j x_k \ldots \qquad (1)$$

where $X(x_1, x_2, ..., x_M)$ is the vector of input variables and $A(a_1, a_2, ..., a_R)$, $\mathrm{R} \subset \aleph$ 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.

In the GMDH algorithm, the Volterra-Kolmogorov-Gabor (VKG) series is estimated by a cascade of second order polynomials using only pairs of variables (Madala & Ivakhnenko, 1994). The corresponding network can be constructed from simple polynomial and delay elements. The main function of the model is based on forward propagation of signal through nodes of the GMDH net similar to principle used *e.g.* in classical neural nets – input signal is applied to input nodes, the outputs of which are then distributed through the structure to upper layers where appropriate mathematical combinations are carried out. Each layer consists of simple nodes each performing its own polynomial transfer function and passing its output to nodes in the next layer. The output of the last layer (consisting of only one node) is the output of the whole net. A typical structure of MIA GMDH network can be seen on Figure 1.

Figure 1
MIA-GMDH-type Network Structure



Two salient features of GMDH polynomial class of networks are:

– evolving structure of self-organizing networks during training process – therefore, there is no need for initial guessing of the number of layers or the number of neurons needed in each layer of the network;

– resistance to over-fitting – self-organizing networks possess the inherent feature of good immunity to over-fitting, mainly due to the division of the data set before launching the learning process into two sub-sets: the Training Set, and the Testing Set.

DOI: 10.18267/j.pep.398

The coefficients of nodes' transfer functions are estimated during the learning phase within of which the whole structure is being automatically built up. This inductive approach to the model structure determination notably reduces the amount of *a priori* knowledge required from the user and allows selecting structure that follows best given dataset. During the evolution of the learning procedure, network branches that do not contribute significantly to the specific output can be pruned, thereby allowing only the dominant causal relationship to evolve.

The GMDH network model is constructed during the learning process by the following five procedures: 1) Separating the original data into the training and test sets, 2) Generating combinations of the node input variables in each layer, 3) Calculating the optimum partial descriptions, 4) Selecting the intermediate variables and 5) Considering stopping the multi-layered iterative computation.

The GMDH algorithm generates optimal structure of the model through successive generations of partial descriptions (PDs) of data, which are described by quadratic regression polynomials most often with two input variables (though more inputs can also be allowed). The complexity of the network increases with each training and selection cycle through addition of new layers until little or no further improvement in fidelity is achieved. Although GMDH provides for a systematic procedure of system modelling and prediction, it has also a number of shortcomings. Among the most problematic can be stated:

- a tendency to generate quite complex polynomial (since the complexity of the network increases with each training and selection cycle through addition of new layers) for relatively simple systems (data input);
- an inclination to producing overly complex network (model) when dealing with highly nonlinear systems owing to its limited generic structure (quadratic two-variable polynomial).

Looking for the best structure of the GMDH net during its learning process can be considered as a state space search. The original learning process as presented in Ivakhnenko (1971) goes through the whole state space of possible combinations by iteratively creating new layers of nodes connected with the previous ones. When a solution is found that seems to be the locally best (*i.e.* best within its local neighbourhood formed by solutions of other already existing nodes), the model is constituted by cutting off nodes that are not connected to the chosen one. It is clear that this 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 is mainly implementation of node selection on each layer. In order to prevent expansion of the net and to reach the model in feasible time, only some of the constructed nodes are selected to constitute the layer. In the original MIA algorithm, only a defined number of n nodes with best evaluation criterion were selected to form the layer, the rest being cut off.

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. The final network is the connections of the best nodes from layer to layer until the output layer. However, this selection strategy brings along two important drawbacks:

DOI: 10.18267/j.pep.398

- throwing away nodes that might play an important role in the model if their outputs (and through them their inputs as well) would be combined on the higher level results in ”loosing” their inputs;
- fast 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^TX to be inverted during regression).

A similarity to such layer pruning can be seen *e.g.* in evolutionary algorithms utilised for GMDH *e.g.* in Park, Kim & Oh (2004), where also only some of the members that form current population are selected to survive, the rest being thrown away. A well known fact is that the selection of the best members *does not* lead to gaining the best solution. 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. Realizing this fact, eMIA-GMDH algorithm was proposed in Buryan (2006) to deal with introduced problems.

3. Enhanced MIA-GMDH Algorithm

Trying to offer solution to stated problems with selection, specific improved features of the enhanced MIA-GMDH-type network were proposed (Buryan, Onwubolu & Lemke, 2007) and are described in subsequent subsections. Amongst the major features of the enhanced MIA-GMDH-type network are:

- semi-randomized 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 stabilize the regression);
- high level of flexibility as each node can have non-linear transfer function, a different number of input variables as well as exploit a different order of polynomial.

Table 1

Node Types (objective functions), Illustration for Two Inputs to a Node

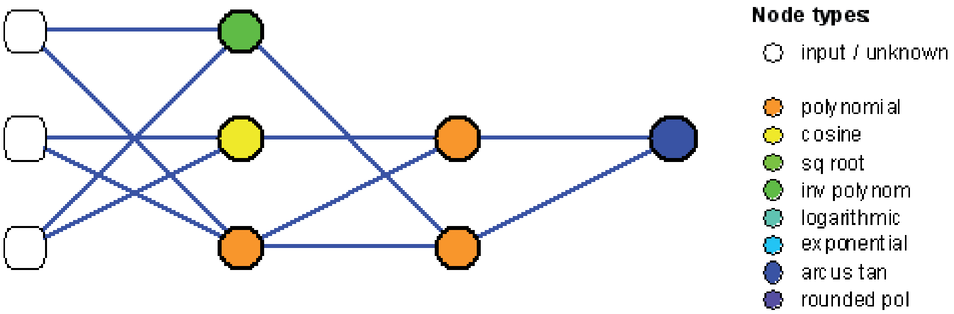
	Polynomial type	**Transfer function (example for two node inputs)**
1	**polynomial**	$f(X) = f(x1,x2) = a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2 + a_5x_1^2 + a_6x_2^2$
2	**harmonic (cosine)**	$f(X) = f(x1,x2) = cos(a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2 + a_5x_1^2 + a_6x_2^2)$
3	**square root**	$f(X) = f(x1,x2) = (a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2 + a_5x_1^2 + a_6x_2^2)^{1/2}$
4	**inverse polynomial**	$f(X) = f(x1,x2) = (a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2 + a_5x_1^2 + a_6x_2^2)^{-1}$
5	**logarithmic**	$f(X) = f(x1,x2) = ln(a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2 + a_5x_1^2 + a_6x_2^2)$
6	**exponential**	$f(X) = f(x1,x2) = exp(a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2 + a_5x_1^2 + a_6x_2^2)$
7	**arc tangent**	$f(X) = f(x1,x2) = atan(a_1 + a_2x_1 + a_3x_2 + a_4x_1x_2 + a_5x_1^2 + a_6x_2^2)$
8	**rounded polynomial**	$f(X) = f(x1,x2) = (a_{r1} + a_{r2}x_1 + a_{r3}x_2 + a_{r4}x_1x_2 + a_{r5}x_1^2 + a_{r6}x_2^2)$ [*)]

[*)] coefficients for linear combination gained from the regression as in classical polynomial (row 1) are rounded e.g. $a_{ri} = round(a_i)$

DOI: 10.18267/j.pep.398

Apart from implementing polynomials of different order, seven other types of transfer (objective functions) that have been used in the enhanced GMDH as shown in Table 1 allowing the nodes to have also non-polynomial transfer function. Nonlinear regression is used to estimate the parameters of non-polynomial nodes. The network final structure can be *e.g.* as shown in Figure 2.

Figure 2
Graphical-representation of eMIA-GMDH Network Connections after Pruning



3.1 Diversity Problem

By selecting only the best nodes from each layer, the output of the layer gets almost homogenous especially in 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. Employing suitable heuristics in the selection process could therefore bring positive effects on the net precision.

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 get the unknown vector b utilizes pseudo inverse matrix:

$$b = \mathrm{inv}(X^T.X).X^T.Y \ , \qquad (2)$$

which works well in most of the cases. Problems arise, 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 the values of b produced by regression. The condition number 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).

This is exactly what happens when selecting only 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

DOI: 10.18267/j.pep.398

of node transfer functions, 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 about the relation of two time series with almost the same values (*e.g.* differences comparable to the level of measurement noise).

Several ways of solving the problem can be seen:

- modifications to the fitness criterion – modifying the error criterion by some additional criteria, *e.g.* complexity criterion – Akaiake, Minimum Description Length (Iba, de Garis & Sato, 1994) *etc*.
- modifications to selection principles (discussed in following paragraphs); for different approach see *e.g*. Kordík, Šnorek & Genyk-Berezovskyj (2004).

3.2 Semi-Randomised Selection

In the work reported in Buryan (2006) the best found solution to the problem of variability is to utilise semi-randomised approach *i.e*. to enrich the selection of the best nodes by randomized selection in the ratio of 1:1. The main principle of this heuristics 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 limiting 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.

Previous exhaustive research experimentation (Buryan, 2006) shows that for semi-randomized 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 the layer enabling the output 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.

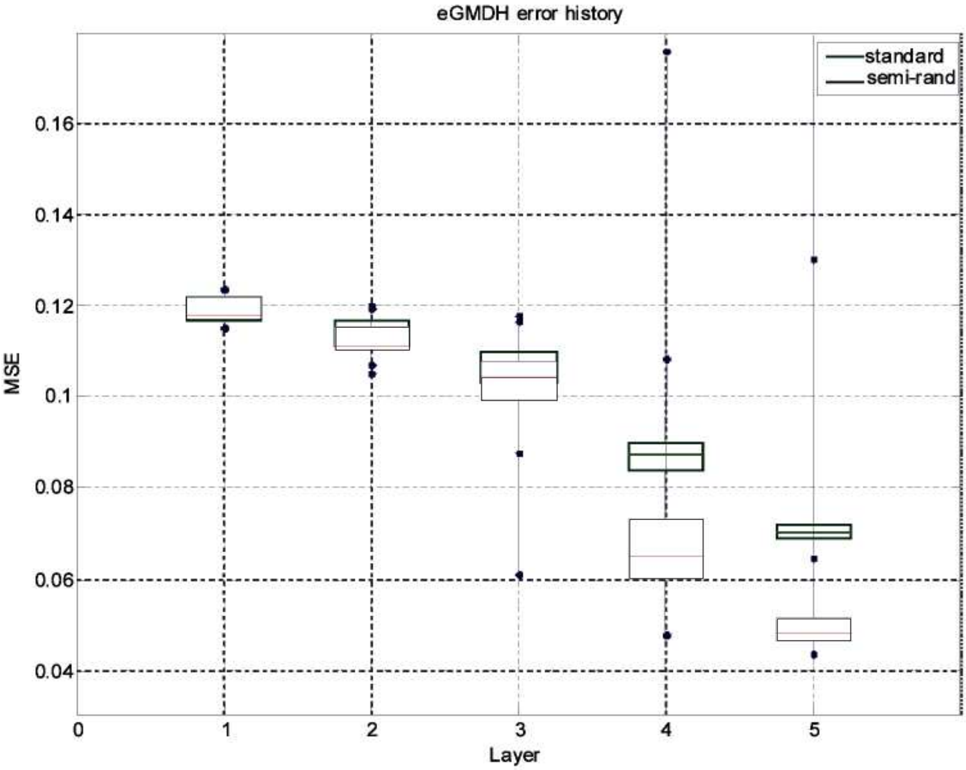
A conclusion was drawn in Buryan (2006) that best-of selection is needed so that the learning process can follow some clear optimization rule while the randomized selection ensures the layer diversity, thereby performing local stochastic space search, stabilizing the regression and retaining information 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.

The typical effect of employing semi-randomised selection can be easily seen in box-plot in Figure 3 characterising the error distribution within the nodes of each layer. The figure presents results of experiments carried out in Buryan (2006) upon Mackey-Glass time series (input history length of 5 preceding samples, polynomial transfer

DOI: 10.18267/j.pep.398

functions only). The limits of the boxes are set to lower and upper quartile, the line splitting the box indicates median error of the layer, blue dots connected with the box symbolise outliers *i.e.* the biggest or the smallest error value respectively.

Figure 3
Comparison of Standard and Semi-randomized Selection



As depicted in the figure, the utilisation of 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 similar to each other as during classical best-of selection). Another impact is limiting the occurence of nodes with unstable transfer function (the worst error of classical approach on layer 5 is out of axis limits). It is clear, that an effort to improve the selection strategy can bring positive effects into GMDH modelling and future research on possible heuristics seems to be promising.

3.3 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

DOI: 10.18267/j.pep.398

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 effects are enormous node coefficients (several orders higher than maximal value in input or output data) which are likely to have pair-wise 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 similar nature of both datasets – it makes no sense to train model on totally different data than it should be used for).

On the other hand, an ability of generalization is expected from a 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 over fitting.

Enhanced MIA-GMDH faces this problem with coefficient tresholding and rounding scheme. Such overstrained nodes (and especially their outlying coefficient values) clearly bring no new information to the model. Therefore, 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 employing the advantages offered by the selection procedure.

Of course, thresholding is not the only approach admissible. Another way to deal with this problem can be found *e.g.* in coefficient statistical significance tests. However, performing these tests for every coefficient of every node seems to complicate the learning procedure and in comparison with simple rounding/thresholding lengthens the run of the algorithm while requires also disposal of more mathematical background and its software implementation for the algorithm realisation. The weak point of the thresholding approach currently lies in setting of threshold levels. Possible methods of setting the threshold levels were not studied much so far and represent an area of potential further research. However, the experiments showed that the optimal levels lie in tens in case of dataset normalized into both <0; 1> and <-1; 1> interval.

Another positive impact of this operation is making easier 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. Providing the end user a polynomial (or modified polynomial) function is much more comfortable for him to implement (can be realized even on standard office software).

4. Experimental Results

Exchange rate analysis is 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 (Mentzel, 1998 & Lai, Yu, & Wang, 2007). However, perhaps the main reason, why

DOI: 10.18267/j.pep.398

the exchange rate series balance on the edge predictability is the fact, that these factors are often random in their nature.

Exchange rate series with mid-term periods (monthly, quarterly) are not often under focus of analyses as they do not offer immediate yields as the daily rates do while, on the other hand, the long-term evolution of exchange rate (periods of year and more) is often subject to classical regression analysis or cointegration mainly due to strong linear relationships with other macroeconomical indicators. However, these mid-term exchange rate series remain utmost interesting *e.g.* for the financial managers of companies, because they can help them in their efforts to avoid exchange rate risks. Therefore, exchange rates with monthly frequency were selected for the analyses.

Two types of experiments are described in this paper demonstrating utilisability of the eGMDH algorithm for time series prediction. First we shortly illustrate performance of the enhanced GMDH network by comparing its results to other known approaches predicting monthly average JPY/USD exchange rates. The second experiment is a case study modelling hedging decision process based on the cash flow of a virtual Czech internationally active company and real market data and prices.

4.1 Ex-ante Predictions

In order to get closer to real application 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 remaining part of the data (training dataset) is then used for model estimation and quality of model is finally judged by evaluation of model predictions for those “hidden” data (testing dataset). However, this approach leads to the fact that samples occurring 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 construct 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 values coming after it are condemned as unknown as well. Therefore, no bias is added to the model estimation process.

4.2 Monthly Average JPY/USD Exchange Rate Prediction

In this experiment we evaluate prediction power of the eGMDH algorithm by comparing its results to other known approaches. Monthly average JPY/USD exchange rate time series was selected as the object of the predictions. The analysed dataset contained time series described shortly in Table 2 from period starting January 1980 until December 2006; the predictions were constructed for period from January 2001 until January 2007.

DOI: 10.18267/j.pep.398

Table 2
Input Data for Monthly Average JPY/USD Exchange Rate Prediction

Time series	Frequency	Unit
Monthly average JPY/USD	Monthly	JPY/USD
Discount rates in Japan and USA	Monthly	%
CPI of Japan and USA	Monthly	% of 1960 (USA) resp. 1970 (Japan)
GDP of Japan and USA [*)]	Quarterly	current prices, mil. USD
Balance of payments of Japan and USA [*)]	Quarterly	current prices, mil. USD

[*)] step function utilised to convert quarterly to monthly frequency

Modelling results of the series have been summed up in Table 3 and illustrated also on following Figure 4. Several classical modelling techniques were used for comparison with the eGMDH approach:

- 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),
- random walk model – variability of random number equal to variability of first differences of the time series,
- classical back propagation-trained neural network with 15 nodes in hidden layer and standard node sigmoid transfer function,
- standard generic MIA algorithm with 5 layers, max. 100 nodes per layer.

The eGMDH model was set-up as follows: 5 layers, max. 100 nodes per layer, all 8 types of transfer function available, 66% of data available was used as training data, the rest as test dataset.

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. Quality of model prediction was assessed by standard mean square error (MSE) criterion.

$$MSE = \frac{\sum (x(t) - \hat{x}(t))^2}{N} = \frac{\sum e^2(t)}{N}, \tag{3}$$

N .. total length of prediction (number of examples),
$x(t)$ .. the true value,
$\hat{x}(t)$ .. the value predicted and
$e(t)$ .. error of the model.

DOI: 10.18267/j.pep.398

Table 3
Prediction ex ante of Monthly Average JPY/USD Exchange Rates

Model	MSE
NC model	6,06
Random walk	23,90
Linear regression	9,34
NN	6,59
MIA GMDH	6,32
eGMDH	**5,27**

Following conclusions for the eGMDH algorithm can be drawn from the table:

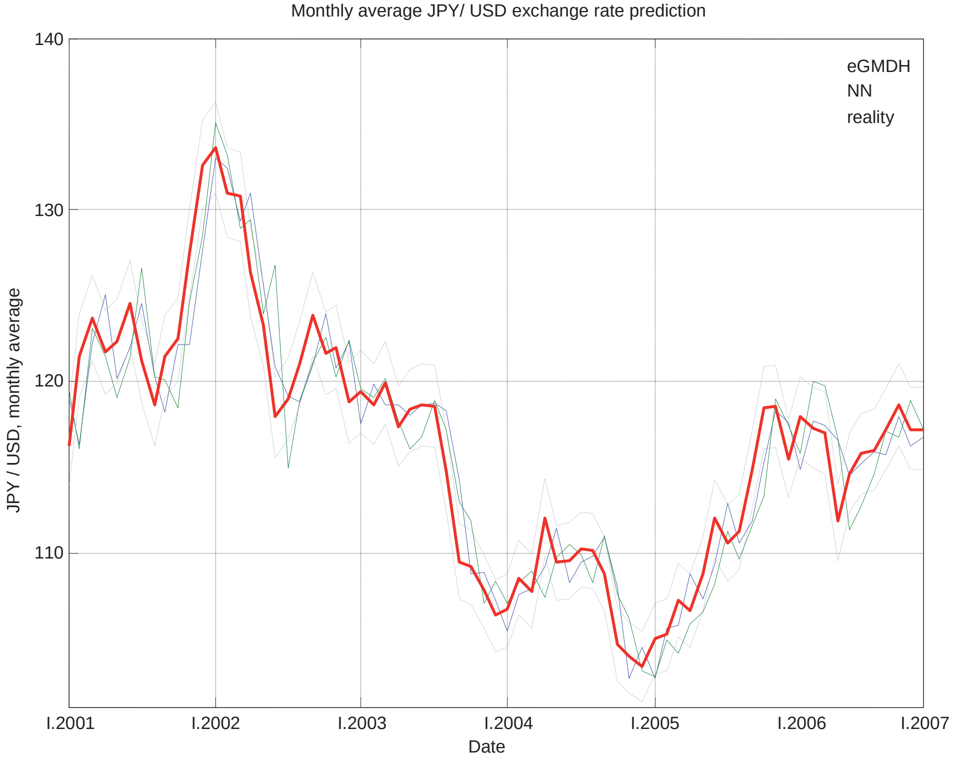
- The eGMDH algorithm performed to have better accuracy than other models.
- Following only the criterion and thereby selecting just the best *n* nodes on each layer during pruning phase does not necessarily lead to reaching the best results of the final model. So, there is sense in looking for a heuristics or – generally speaking – a more complex method that could predict the potential contribution of each node to the model as a whole (if the node would be kept).
- Utilisation of non-linear elements 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.
- 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.

4.3 Financial Flows and Transaction Exposure - Simulation Analysis

Simulation presented in this paper was based on the cash flow of a virtual firm and real market data and prices in the years 2005-2006. The firm sold merchandise on open account for €400,000 each month. The payment was always made in 1 month. The firm was exposed to the exchange rate transaction risk arising from the risk that euro would weaken and the seller would receive less Czech korunas.

DOI: 10.18267/j.pep.398

Figure 4
Prediction ex ante of Monthly Average JPY/USD Exchange Rates, Comparison of bp-NN and eGMDH, deviation Interval of +/- 2% (boundary lines)



4.3.1 Prediction Models

The focus of the simulation analysis was the implementation of exchange rate predictions based on outputs algorithm eGMDH into the modelled decision process of company's financial management. In order to get close to real situation, all of the predictions were constructed by *ex-ante* approach, *i.e.* all the models were based only on data accessible to the time point of prediction. For each month, a group of ten GMDH models were constructed, each of these being restricted to 5 layers consisting of maximally 80 nodes. The final prediction was selected by the quality of each model's prediction over historical data using classical mean-squared error (MSE) criterion.
The analysed input dataset consisted of following relevant time series:

- CZK/EUR spot exchange rate (5 preceding historical values),
- 3- and 6-months forward CZK/EUR exchange rates (2 preceding historical values),
- actual discount interest rates in Czech Republic and EU15.

DOI: 10.18267/j.pep.398

4.3.2 Hedging Portfolio

Financial management of the company disposed of portfolio consisting of five instruments – forward contract, plain vanilla option, the option strategy “Vertical Spread” and the option strategy “Ration Forward”.

Forward Contract

“*Forward contract specifies the amount of a particular currency that will be purchased or sold at a predetermined point in the future.*” (Madura, 2006). Thus, if the aim is to hedge receivables in a foreign currency the corporation may close the position by selling the specific amount of foreign currency on the forward market.

In the simulation, forward rates were derived from the covered interest rate parity condition using LIBOR (London Interbank Offered Rate) and PRIBOR (Prague Interbank Offered Rate) as the reference interest rates (see Taušer, 2007).

Plain Vanilla Option

The virtual firm may also close the position by buying a put option, which gives the right to sell $400,000 at a predetermined strike price. Since the option gives only the right, and not the obligation, to sell a given amount of foreign currency at a predetermined strike price, the company has to pay the option premium to the bank.

Option Strategy “Vertical Spread”

Option strategies generally represent a mixture of different option positions. The most popular ones are the so called “zero costs strategies” that represent mix of two or more option positions with the same amount of paid and received option premiums.

The so called “Vertical Spread” strategy consists of two basic option positions with the same amount of underlying currency and with different strikes. If the aim is to hedge the receivables, the corporation buys one put option, which gives it the right to sell a specific currency at a predetermined strike price within a specific period of time, and simultaneously it sells a call option, which grants the right to the bank to buy the same amount of foreign currency within the same period of time. Both options have different strikes, but, the same option premiums. The corporation sets a strike price of the put option and the bank calculates the strike price of the call option, both option premiums being equal.

Option Strategy “Ratio Forward”

The “Ratio Forward” option strategy represents a combination of two basic option positions. In this case corporation buys a put option and simultaneously sells a call option with different strikes and different amounts of underlying currency.

Thus, the virtual firm buys a put option that gives the right to sell €400,000 at some point in the future, and sets the strike price that is, for example, CZK 27.700 *per* euro. At the same time the firm sells a call option, which grants the right to the

DOI: 10.18267/j.pep.398

bank to buy, for instance, €600,000 at some point in the future. The bank calculates the strike price of the put option so that the option premiums are the same. Since the amount of underlying currency of the put option is smaller than the amount of underlying currency of the call option, the strike price of the call option shall be greater in comparison with the "Vertical Spread" strategy.

The corporation may use that strategy if it generally expects a depreciation of the domestic currency, but small fluctuations due to the intervention of central banks are possible as well. The strategy provides hedging against the appreciation of the domestic currency and simultaneously enables participation in the slight depreciation of the domestic currency. If the depreciation is, however, too strong, the revenues will start to fall.

4.3.3 Hedging Strategy

The decision of the strategy was made on the basis of predictions offered by neural networks and GMDH algorithm. Standard parameters of the instruments that are offered by the banks to smaller companies on the Czech market were used in the simulation.

In the simulation, forward rates were derived from the covered interest rate parity condition including the transaction costs in terms of average spreads for small companies on the Czech market. Strike prices of the put options were set about 1% below the forward rates, and the option premiums as well as the strike prices of the call options within the option strategies were obtained from one of the largest Czech commercial banks. Thus, the prices of financial instruments used in the simulation reflected real market prices of these instruments for small companies.

As for the exchange rate forecasting two machine learning methods – the enhanced Group Method of Data Handling (GMDH) and the back propagation Neural Network were employed. The relevant algorithms were created in MATLAB programming environment.

There were two main reasons for using machine learning methods for exchange rate forecasting. Firstly, although machine learning provides highly sophisticated non-linear modelling of exchange rate behaviour, financial managers can easily use these models on the "black box" principle. Therefore, as soon as an expert creates the model, it is relatively easy to use.

Secondly, the machine learning models provide short-term predictions of time series future values. This was very important for our decision-making process simulation, because managers of virtual firms could use the predictions not only as recommendations whether to hedge or not, but they could use the forecasts also as instructions on which instrument to use. Thus, the process of hedging was fully automated.

The hedging strategy was defined as follows:

- If the predicted exchange rate is lower than the forward rate, the virtual firm sells €400,000 on the forward market.
- If the predicted exchange rate lies between the forward rate and the strike price of the call option within the "Vertical Spread" strategy, the virtual firm uses the "Vertical Spread" strategy.

DOI: 10.18267/j.pep.398

- If the predicted exchange rate lies between the strike price of the call option within the "Vertical Spread" strategy and the strike price of the call option within the "Ratio Forward" strategy, the virtual firm uses the "Ratio Forward" strategy.
- If the predicted exchange rate is higher than the strike price of the call option within the "Ratio Forward", but lower than this strike price plus two thirds of the paid plain vanilla put option premium, including relevant interest[1], the virtual firm stays with the "Ratio Forward" strategy.
- If the predicted exchange rate is higher than the strike price of the call option within the "Ratio Forward," plus, two thirds of the paid plain vanilla put option premium including relevant interest, then, the virtual firm buys the plain vanilla put option.

The last rule can be formally derived as follows:

$$\mathrm{S} - \mathrm{P} \cdot (1+\mathrm{i}) > \mathrm{X} - (\mathrm{S} - \mathrm{X}) \cdot 0.5, \tag{4}$$

so that

$$1.5 \cdot \mathrm{S} > 1.5 \cdot \mathrm{X} + \mathrm{P} \cdot (1+\mathrm{i}),$$

and finally

$$S - X > \frac{2}{3} P \cdot (1+i) \tag{5}$$

S … spot exchange rate on the expiry day;
P … put option premium;
X … call option strike price;
i … reference interest rate on a monthly basis.

4.4 Simulation Analysis Results

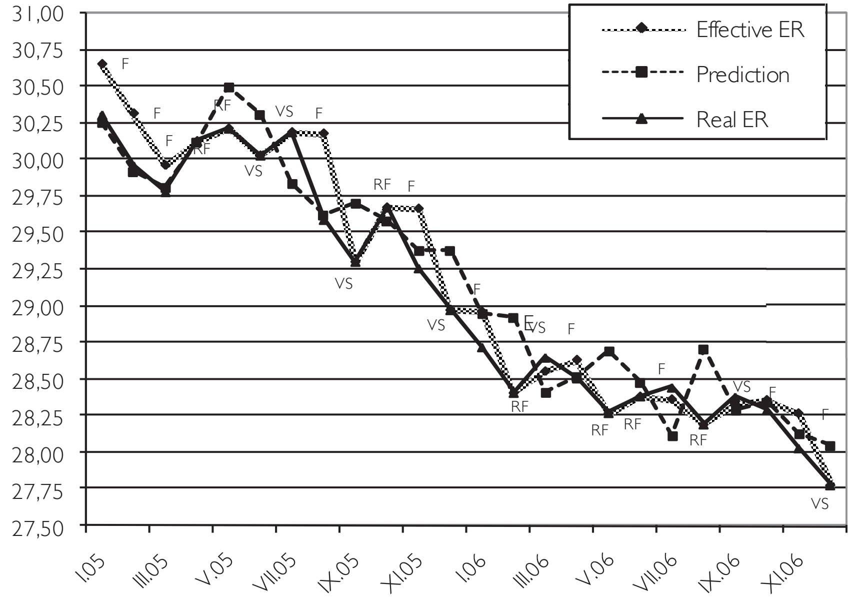
In the simulation, the financial managers of the virtual firm hedged the receivables of €400,000 each month during the years 2005–2006. In total, 24 decisions were made. Hedging procedure was quite automatic, because all decisions were made with respect to the predefined strategy and the relevant exchange rate forecasts were based on machine learning models.

Figure 5 illustrates the hedging process. Three different exchange rates are plotted in the graph – the *ex-ante* predicted exchange rate, the real exchange rate and the effective exchange rate achieved through the hedging strategy. It is also noticeable which instrument was used in each month – forward (F), "Vertical Spread" strategy (VS) and "Ratio Forward" strategy (RF).

[1] In the Czech Republic the option premiums are usually paid at the beginning of the option contracts. Thus, we have to calculate the interest for the period between the forming of the contract and the expiry day.

DOI: 10.18267/j.pep.398

Figure 5
Hedging Process



The first moth the predicted exchange rate for the next month was CZK 30.250 *per* euro, 1-month forward rate was CZK 30.656 *per* euro, strike price of the put option was set at CZK 30.312 *per* euro and strike price of the call option equalled CZK 30.828 *per* euro for the "Vertical Spread" strategy and CZK 31.065 *per* euro for the "Ratio Forward" strategy. With respect to the defined hedging strategy we decided for forward hedge.

In one month, the real exchange rate was CZK 29.960 *per* euro. Thus, the savings from hedging amounted to €27,809, because thanks to the forward we could sell €400,000 at the higher rate (CZK 30.656 *per* euro) *etc*.

In total, the forward was used ten times. In seven cases, the open position was hedged with the "Vertical Spread" strategy. The "Ratio Forward" strategy was applied seven times as well. The plain vanilla put option was not used at any time. In comparison with no hedging, the revenues of the virtual firm were higher by about CZK 939, 896. Thus, the hedging process was quiet successful.

5. Conclusions

In this study, we presented an enhanced MIA-GMDH-type network, discussed main points of its design methodology and carried out experiments of exchange rate prediction and implementation of the algorithm in the field of financial hedging. The e-GMDH approach presented in this paper is more flexible than the conventional GMDH approach due to specific improvements features analyzed in the paper.

It is clear mainly from the results of employing the semi-randomised selection

DOI: 10.18267/j.pep.398

strategy that an effort to improve the selection strategy can bring positive effects into GMDH modelling and future research on possible heuristics seems to be promising.

Experiments reported include standard exchange rate prediction task and a real-data based simulation analysis in which a simulation of financial flow of a internationally active medium-sized company based upon real market data was performed. The hedging portfolio consisted of five main instruments – forward contract, plain vanilla option, the option strategy "Vertical Spread" and the option strategy "Ration Forward" while the decision about strategy chosen was made on the basis of ex-ante predictions of given algorithm. Comparing with situation when no hedging would be used, the revenues of the virtual company were significantly higher.

Enhanced GMDH algorithm outperformed classical GMDH algorithm as well as other widely used methods in predicting monthly JPY/USD exchange rate. The results of performed simulation analysis show that eGMDH algorithm represents a proper tool that can be used as a decision support instrument in the financial management process of companies.

References

Buryan, P. (2006), "Time Series Analysis by Means of Enhanced GMDH Algorithm." Dissertation Thesis, CTU Prague.

Buryan, P. Onwubolu, G. C., Lemke, F. (2007), "Modeling Tool Wear In End-Milling Using Enhanced Gmdh Learning Networks." *International Journal Of Advanced Manufacturing Technology*, Springer Verlag 2007, Doi 10.1007/S00170-007-1296-1.

Iba, H., de Garis, H., Sato, T. (1994), "Genetic Programming Using a Minimum Description Length Principle," in *Advances in Genetic Programming*, Kinnear, K. E. Jr. (ed), Cambridge: MIT, pp. 265-284.

Ivakhnenko, A. G. (1971), "Polynomial Theory of Complex Systems." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-1, No. 4, pp. 364-378.

Kordík P., Šnorek M., Genyk-Berezovskyj M. (2004), "Hybrid Inductive Models: Deterministic Crowding Employed." Proceedings of the International Joint Conference on Neural Networks; Piscataway: IEEE, pp. 2343-2346; ISBN 0-7803-8360-5.

Lai K. K., Yu L., Wang S. (2007), *Foreign-Exchange-Rate Forecasting with Artificial Neural Networks*. Springer Verlag, ISBN: 978-0-387-71719-7.

Madala, H. R., Ivakhnenko A. G. (1994), *Inductive Learning Algorithms for Complex Systems Modeling*. CRC Press, Boca Raton.

Madura, J. (2006), *International Corporate Finance*. 8th Edition. Thomson South-Western. ISBN 0-324-32382-4.

Mentzel, S. M. (1998), *Real Exchange Rate Movements*. Springer Verlag, ISBN: 978-3-7908-1081-3.

Mueller, J. A., Lemke, F., Ivakhnenko A. G. (1998), "GMDH algorithm for complex systems modelling." *Mathematical Modelling of Systems*, No. 4.

Park, H. S., Park, B. J., Kim, H. K., Oh, S. K. (2004), "Self-organizing Polynomial Neural Networks Based on Genetically Optimized Multi-layer Perceptron Architecture." *International Journal of Control, Automation, and Systems*, 2(4), pp. 423-434.

Taušer, J. (2007)**,** *Měnový kurz v mezinárodním podnikání*. VŠE Praha, ISBN 978-80-245-1165-8.

Žamberský, P. (2003), *Ekonomie měnového kurzu I*. Praha : Nakladatelství Oeconomica, ISBN 80-245-0637-8.