DEEP HYBRID SYSTEM OF COMPUTATIONAL INTELLIGENCE FOR TIME SERIES PREDICTION

Yevgeniy Bodyanskiy, Iryna Pliss, Dmytro Peleshko, Olena Vynokurova

***Abstract**: In this paper the deep hybrid system of computational intelligence for time series forecasting is proposed. As the layer of such hybrid system we propose to use the hybrid generalized additive type-2 fuzzy-wavelet-neural network. Proposed deep stacking forecasting hybrid system of computational intelligence is enough simple in computational implementation due to parallelizing process of implemented computations, has high learning rate convergency due to dismissal from errors backpropagation and using the rapid adaptive algorithms for tuning its parameters, has also flexibility due to possibility of tuning the activation-membership functions. Proposed system is aimed for solving the wide range Data Stream Mining tasks, such as forecasting, identification, emulation, classification and pattern recognition in online mode.*

***Keywords**: deep learning, computational intelligence, time series forecasting, hybrid type-2 fuzzy system.*

***ACM Classification Keywords**: I.2 ARTIFICIAL INTELLIGENCE, I.5 PATTERN RECOGNITION.*

Introduction

Nowadays hybrid systems of computational intelligence [Rutkowski, 2008; Du and Swamy, 2014; Mumfold and Jain, 2009] are wide spread for solving different type of Data Mining problems under uncertainty conditions when the investigated objects or processes are nonstationary, nonlinear, stochastic, chaotic, etc. Among these problems the most complicated task is forecasting because it has highest level of uncertainty, which is defined by multivariance of future facts and events. Specifically, hybrid systems have shown the best results for solving such type of tasks due to universal approximation properties and capability to learning of artificial neural networks [Haykin, 1999; Nelles, 2001], transparency and interpretability of obtained results and high learning rate of tuning synaptic weights of neuro-fuzzy systems [Jang and Mizutani, 1997; Bodyanskiy et all, 2001; Bodyanskiy et all, 2005; Bodyanskiy et all, 2008], including type-2 fuzzy neural networks [Aliev and Guirimov, 2014], the possibility of effective description of nonstationary signal local characteristics due to wavelet neural networks [Alexandridis and Zapranis, 2014] and flexibility of wavelet-neuro-fuzzy systems [Abiyev and

Kaynak, 2008; Bodyanskiy and Vynokurova, 2013], including type-2 fuzzy-wavelet neural networks [Bodyanskiy et all, 2015].

Recent years deep neural networks [LeCun et all, 2015; Schmidhuber, 2015; Goodfellow et all, 2016] attract the attention of many researchers. There networks provide higher quality of data processing including forecasting problem [Langkvist Längkvist et all, 2014; Huang et all, 2014; Setlak et all, 2016] in comparison with the conventional shallow neural networks. But the learning process of deep networks is enough time-consuming and demands heavy span time, leading to impossibility of information processing in real time. Nevertheless, such tasks are appeared quite often, especially in the case when data are fed to the processing in on-line mode and time for processing is strictly constrained. These problems have appeared in the context of Dynamic Data Mining [Lughofer, 2011] and Data Stream Mining [Aggarwal, 2007; Bifet, 2010] and even accelerated deep systems [Bodyanskiy et all, 2016], which are constructed based on conventional architectures, cannot manage to adjust their parameters in a time between two next observations.

In the connection with that topical problem is synthesis of forecasting deep hybrid system of computational intelligence, which allows to process the nonlinear nonstationary time series in on-line mode.

Architecture of deep forecasting hybrid system

Figure 1 shows the architecture of proposed forecasting deep systems of computational intelligence, which is a hybrid of cascade-correlation neural network of Fahlman-Lebiere [Fahlman and Lebiere, 1990], cascaded neuro-fuzzy systems [Bodyanskiy et all, 2014; Bodyanskiy and Tyshchenko, 2016; Bodyanskiy et all, 2016a], stacking deep neural networks [Schmidhuber, 2015] and hybrid generalized additive type-2 fuzzy-wavelet-neural network [Bodyanskiy et all, 2015], that form layers of deep system. At that number of layers can be increased during learning process of system.

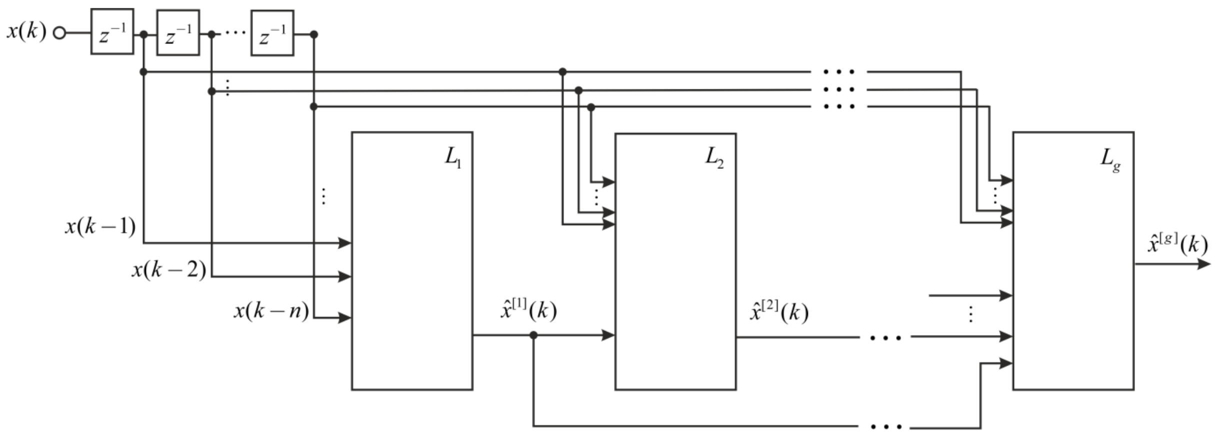


Figure 1. Forecasting deep hybrid system of computational intelligence

The zero layer of system is formed by elements with pure delay z^{-1}, at that the current observation of processed time series $x(k)$ is fed to input of first element. Thus, the zero layer forms prehistory of process $X(k) = (x(k-1), x(k-2), \ldots, x(k-n))^T \in R^n$, which is fed to input of the first hidden layer L_1. The output of this layer is estimated $\hat{x}^{[1]}(k) = f^{[1]}(X(k))$, where $f^{[1]}(X(k))$ is nonlinear transformation which is implemented by HGAT2FWNN [Bodyanskiy et all, 2015]. Input of second layer L_2 is signal $X^{[2]}(k) = (X^T(k), \hat{x}^{[1]}(k))^T \in R^{n+1}$, and its output is $\hat{x}^{[2]}(k)$; to the input of third layer L_3 the signal $X^{[3]}(k) = (X^T(k), \hat{x}^{[1]}(k), \hat{x}^{[2]}(k))^T = (X^{[2]T}(k), \hat{x}^{[2]}(k))^T \in R^{n+2}$ is fed, and finally, the vector $X^{[g]}(k) = (X^T(k), \hat{x}^{[1]}(k), \hat{x}^{[2]}(k), \ldots, \hat{x}^{[g-1]}(k))^T = (X^{[g-1]T}(k), \hat{x}^{[g-1]}(k))^T \in R^{n+g-1}$ is fed to input of $g-$th layer L_g, and the signal-forecast $\hat{x}^{[g]}(k)$ is formed in the output of system. Such architecture allows to avoid using the errors backpropagation procedures, which are implemented in batch mode and doesn't allow to realize online data processing. In our case the learning process is involved sequentially layer by layer and each layer is learned by high-speed adaptive algorithms. At that each next layer is distinct from previous layer only additional input, and output signal from previous layer is fed to this input.

Hybrid generalized additive type-2 fuzzy-wavelet-neural network

Figure 2 shows the architecture of hybrid generalized additive type-2 fuzzy-wavelet-neural network (HGAT2FWNN) in first hidden layer L_1 with n inputs $x_1(k) \equiv x(k-1),\ x_2(k) \equiv x(k-2), \ldots, x_n(k) \equiv x(k-n)$ and single output $\hat{x}^{[1]}(k)$.

This system consists of four layers of information processing; the first and second layers are similar to the layers of TSK-neuro-fuzzy system [Takagi and Sugeno, 1985; Sugeno and Kang, 1988; Takagi and Hayashi, 1991]. The only difference is that the odd wavelet membership functions "Mexican Hat", which are "close relative" of Gaussians, are used instead of conventional bell-shaped Gaussian membership function in the first hidden layer

$$\phi_{li}(x_i(k)) = (1 - \tau_{li}^2(k)) \exp(-\tau_{li}^2(k)/2) \qquad (1)$$

where $\tau_{li}(k) = (x_i(k) - c_{li})\sigma_{li}^{-1}$; c_{li}, σ_{li} are the centre and width of the corresponding membership function implying that $\underline{c} \le c_{li} \le \overline{c}$; $\underline{\sigma} \le \sigma_{li} \le \overline{\sigma}$; $i = 1, 2, \ldots, n$; $l = 1, 2, \ldots, h$; n is the input number; h is the membership function number.

It is necessary to note that using the wavelet functions instead of common bell-shaped positive membership functions gives the system more flexibility [Mitaim and Kosko, 1996], and using odd wavelets for the fuzzy reasoning does not contradict the ideas of fuzzy inference, because the negative values of these functions can be interpreted as non-membership levels [Mitaim and Kosko, 1997].

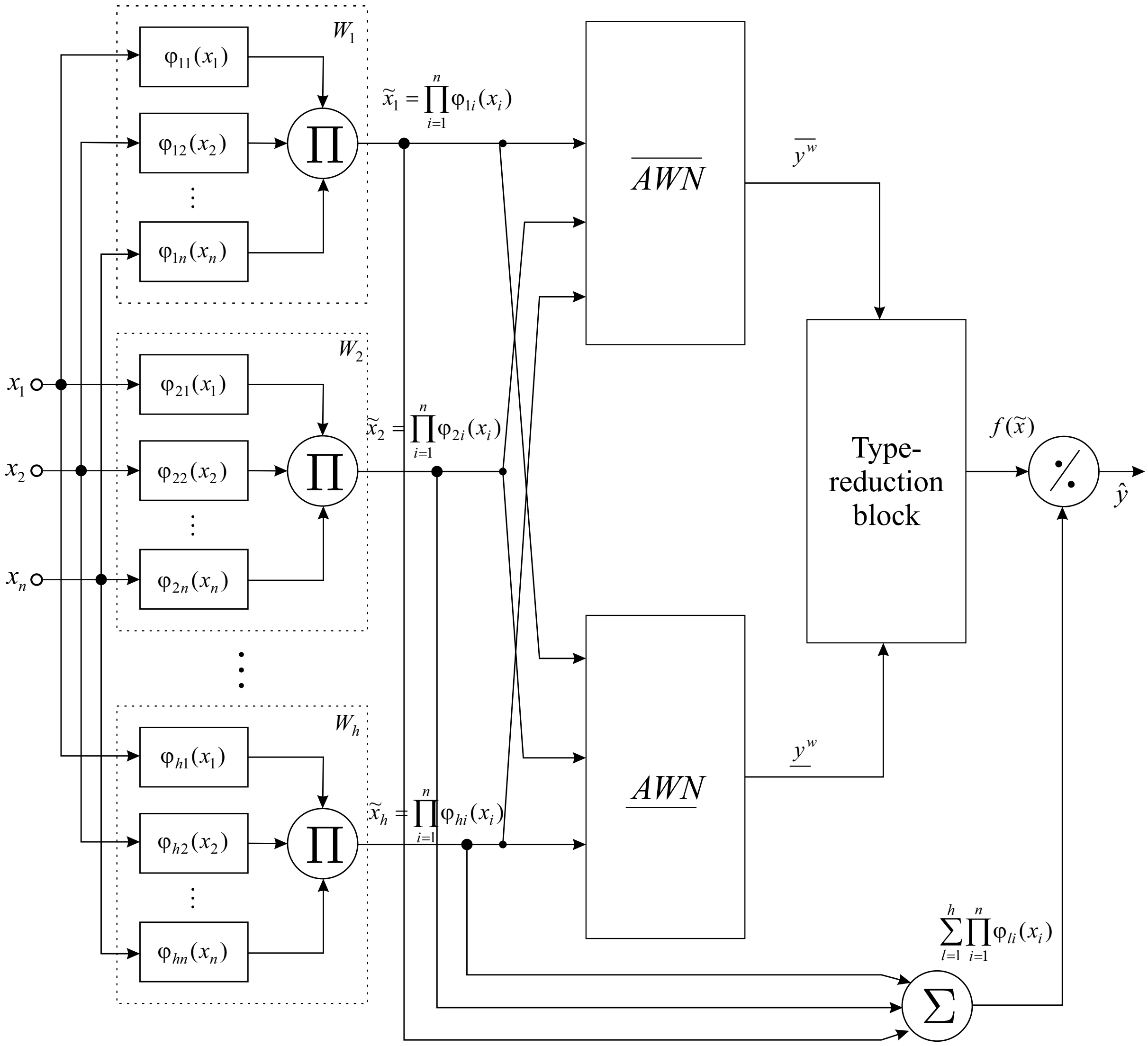


Figure 2. Architecture of HGAT2FWNN

Thus, if the input vector $X(k)$ is fed to the system input, then in the first layer the hn levels of membership functions $\phi_{li}(x_i(k))$ are computed and in the hidden layer h vector product blocks perform the aggregation of these memberships in the form

$$\tilde{x}_l(k) = \prod_{i=1}^{n} \phi_{li}(x_i(k)). \tag{2}$$

This means that the input layers of HGAT2FWNN transform the information similarly to the neurons of the wavelet neural networks [Alexandridis and Zapranis, 2014; Bodyanskiy et all, 2005a], which form the multidimensional activation functions

$$\prod_{i=1}^{n}(1-\tau_{li}^{2}(k))\exp(-\tau_{li}^{2}(k)/2) \tag{3}$$

providing a scatter partitioning of the input space.

As a result, the signals in the output of the second layer can be written in the form

$$\tilde{x}_l(k)=\prod_{i=1}^{n}\left(1-\frac{(x_i(k)-c_{li})^2}{\sigma_{li}^2}\right)\exp\left(-\frac{(x_i(k)-c_{li})^2}{2\sigma_{li}^2}\right). \tag{4}$$

To provide the required approximation properties, the third layer of system is formed based on type-2 fuzzy wavelet neuron (T2FWN) [Bodyanskiy and Vynokurova, 2011; Bodyanskiy et all, 2012]. This neuron consists of two adaptive wavelet neurons (AWN) [Bodyanskiy et all, 2005a; Bodyanskiy et all, 2014a], whose prototype is a wavelet neuron of T. Yamakawa [Yamakawa et all, 1998]. Wavelet neuron differs from the popular neo-fuzzy neuron [Yamakawa et all, 1992; Uchino and Yamakawa, 1997] that uses the odd wavelet functions instead of the common triangular membership functions. The use of odd wavelet membership functions, which form the wavelet synapses $WS_1,\ldots,WS_l,\ldots,WS_h$, provides higher quality of approximation in comparison with nonlinear synapses of neo-fuzzy neurons. Figure 3 shows the architecture of wavelet-neuron.

In such a way the wavelet neuron performs the mapping in the form

$$y^w(\tilde{x}(k))=\sum_{l=1}^{h}f_l(\tilde{x}_l(k)) \tag{5}$$

where $\tilde{x}(k)=(\tilde{x}_1(k),\ldots,\tilde{x}_l(k),\ldots,\tilde{x}_h(k))^T$, $y^w(\tilde{x}(k))$ is the scalar output of wavelet neuron. Each wavelet synapse WS_l consists of p wavelet membership functions $\tilde{\phi}_{jl}(\tilde{x}_l)$, $j=1,2,\ldots,p$ (p is a wavelet membership function number in the wavelet neuron) and the same number of the tuning synaptic weights w_{jl}. Thus, the transform that is implemented by each wavelet synapse WS_l in the k-th instant of time, can be written in form

$$f_l(\tilde{x}_l(k))=\sum_{j=1}^{p}w_{jl}(k-1)\tilde{\phi}_{jl}(\tilde{x}_l(k)) \tag{6}$$

(here $w_{jl}(k-1)$ is the value of synaptic weights that are computed based on previous $k-1$ observations), and the general wavelet neuron performs the nonlinear mapping in the form

$$y^w(\tilde{x}(k))=\sum_{l=1}^{h}\sum_{j=1}^{p}w_{jl}(k-1)\tilde{\phi}_{jl}(\tilde{x}_l(k)) \tag{7}$$

i.e., in fact, this is the generalised additive model [Hastie and Tibshirani, 1990] that is characterised by the simplicity of computations and high approximation properties.

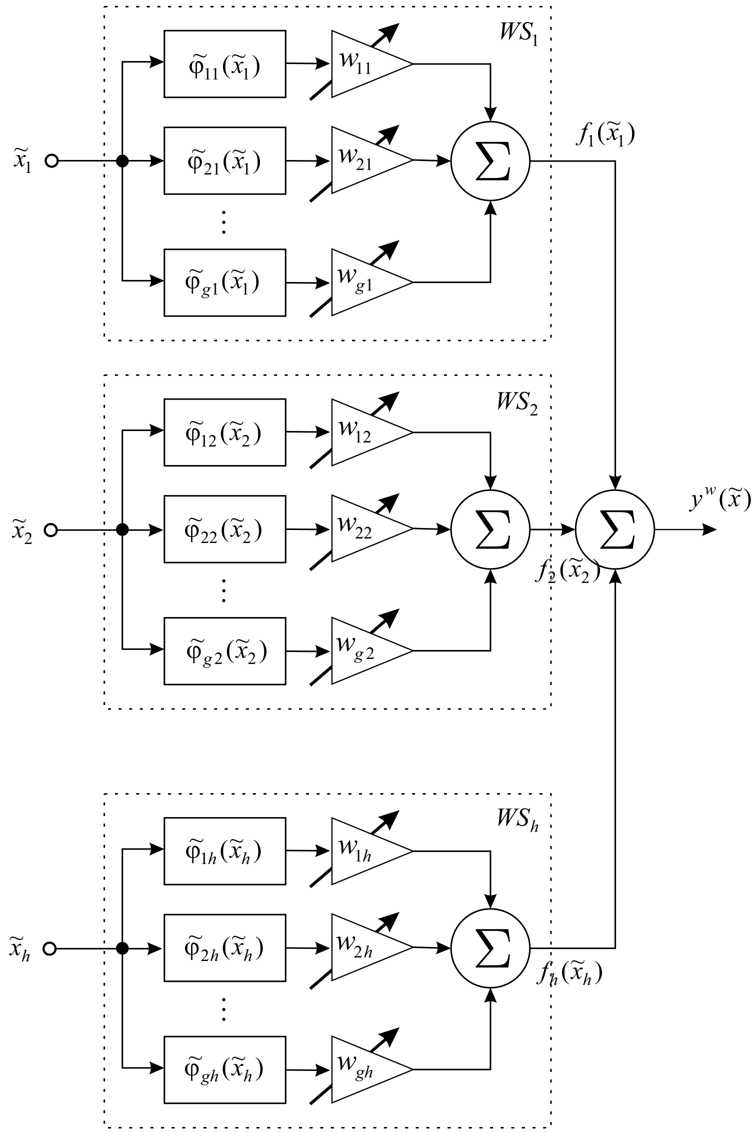
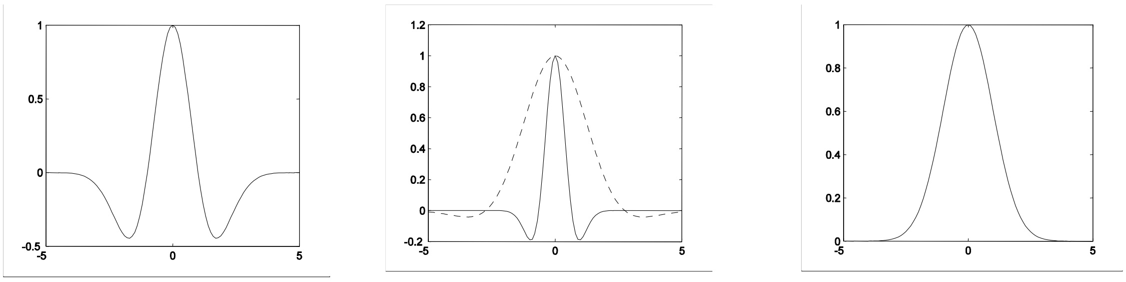


Figure 3. Wavelet-neuron architecture

Under uncertain, stochastic or chaotic conditions, it is more effective to use the adaptive wavelet neuron (AWN) instead of common wavelet neuron. The adaptive wavelet neuron is based on the adaptive wavelet function in the form

$$\tilde{\phi}_{jl}(\tilde{x}_l(k)) = (1-\alpha_{jl}\tau_{jl}^2(k))\exp(-\tau_{jl}^2(k)/2) = \left(1-\alpha_{jl}\frac{(\tilde{x}_l(k)-c_{jl})^2}{\sigma_{jl}^2}\right)\exp\left(-\frac{(\tilde{x}_l(k)-c_{jl})^2}{2\sigma_{jl}^2}\right) \quad (8)$$

where $0 \le \alpha_{jl} \le 1$ is the shape parameter of adaptive wavelet function, if $\alpha_{jl} = 0$ it is conventional Gaussian, if $\alpha_{jl} = 1$ it is the wavelet “Mexican Hat”, and if $0 < \alpha_{jl} < 1$ it is some hybrid activation-membership function (see Figure 4).

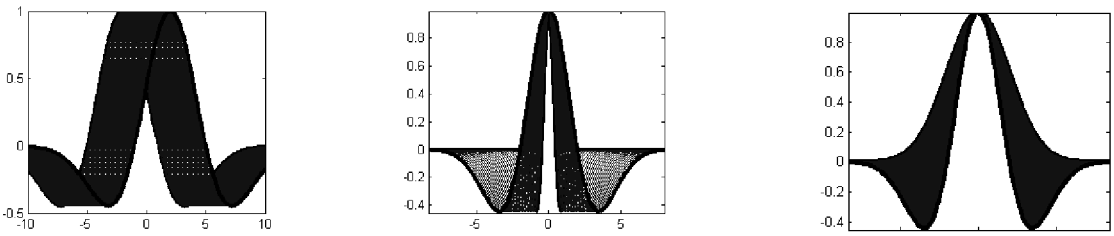


a) b) c)

Figure 4. Adaptive wavelet function

Here it should be noted that if the learning process of wavelet neuron of T. Yamakawa is the tuning of the synaptic weights w_{jl}, then the learning process of adaptive wavelet neuron consists of the tuning not only synaptic weights but also centres c_{jl}, widths σ_{jl} and shape parameters α_{jl} of wavelet functions. However, if synaptic weights w_{jl} can be tuned using the second-order optimisation algorithms, such as a recurrent least squares method, then the optimisation of the operation speed in the gradient learning algorithms of c_{jl}, σ_{jl}, α_{jl} is significantly difficult.

To overcome this difficulty, we can set the boundary of possible changes of adaptive wavelet function parameters $\underline{c}_{jl} \le c_{jl} \le \bar{c}_{jl}$, $\underline{\sigma}_{jl} \le \sigma_{jl} \le \bar{\sigma}_{jl}$, $\underline{\alpha}_{jl} \le \alpha_{jl} \le \bar{\alpha}_{jl}$ and introduce the type-2 fuzzy wavelet membership functions. These functions form the type-2 fuzzy wavelet neuron (T2FWN) and are shown in Figure 5.



a) uncertain centre b) uncertain width c) uncertain function shape parameter

Figure 5. Type-2 fuzzy-wavelet membership function with different type uncertainties

T2FWN consists of two AWNs, where the signal $\tilde{x}(k)$ is fed to the inputs of it. Thus, one AWN uses low boundary values $\underline{c}_{jl}$, $\underline{\sigma}_{jl}$, $\underline{\alpha}_{jl}$, while the other AWN uses high boundary values $\bar{c}_{jl}$, $\bar{\sigma}_{jl}$, $\bar{\alpha}_{jl}$. It is important to notice that neurons $\underline{AWN}$, $\overline{AWN}$ are trained independently of each other based on common reference signal $x(k)$.

As a result, the transformation that is implemented by $\underline{AWN}$ can be written in the form

$$\underline{y}^{w}(\tilde{x}(k)) = \sum_{l=1}^{h}\sum_{j=1}^{p} \underline{w}_{jl}(k-1)\tilde{\underline{\phi}}_{jl}(\tilde{x}_l(k)) \tag{9}$$

and $\overline{AWN}$ -

$$\bar{y}^{w}(\tilde{x}(k)) = \sum_{l=1}^{h}\sum_{j=1}^{p} \bar{w}_{jl}(k-1)\bar{\tilde{\phi}}_{jl}(\tilde{x}_l(k)) \tag{10}$$

where

$$\tilde{\underline{\phi}}_{jl}(\tilde{x}_l(k)) = \left(1-\underline{\alpha}_{jl}\frac{(\tilde{x}_l(k)-\underline{c}_{jl})^2}{\underline{\sigma}_{jl}^2}\right)\exp\left(-\frac{(\tilde{x}_l(k)-\underline{c}_{jl})^2}{2\underline{\sigma}_{jl}^2}\right), \tag{11}$$

$$\bar{\tilde{\phi}}_{jl}(\tilde{x}_l(k)) = \left(1-\bar{\alpha}_{jl}\frac{(\tilde{x}_l(k)-\bar{c}_{jl})^2}{\bar{\sigma}_{jl}^2}\right)\exp\left(-\frac{(\tilde{x}_l(k)-\bar{c}_{jl})^2}{2\bar{\sigma}_{jl}^2}\right). \tag{12}$$

In the type-reduction block, the signals $\underline{y}^{w}(\tilde{x}(k))$ and $\bar{y}^{w}(\tilde{x}(k))$ are united in the simplest way and form the output signal of T2FWN

$$f(\tilde{x}(k)) = c(k)\bar{y}^{w}(\tilde{x}(k)) + (1-c(k))\underline{y}^{w}(\tilde{x}(k)) \tag{13}$$

where $c(k)$ is the tuning parameter that defines closeness of signals $\underline{y}^{w}(\tilde{x}(k))$ and $\bar{y}^{w}(\tilde{x}(k))$ to reference signal $x(k)$.

Finally, the forth (output) layer of system that consists of elementary sum and division blocks implements the defuzzification in the form

$$\hat{x}^{[1]}(k) = \frac{f(\tilde{x}(k))}{\sum_{l=1}^{h}\tilde{x}_l(k)} = c(k)\frac{\bar{y}^{w}(\tilde{x}(k))}{\sum_{l=1}^{h}\tilde{x}_l(k)} + (1-c(k))\frac{\underline{y}^{w}(\tilde{x}(k))}{\sum_{l=1}^{h}\tilde{x}_l(k)} = \tag{14}$$

$$= c(k)\frac{\sum_{l=1}^{h}\sum_{j=1}^{p}\bar{w}_{jl}(k-1)\bar{\tilde{\phi}}_{jl}(\tilde{x}_l(k))}{\sum_{l=1}^{h}\tilde{x}_l(k)} + (1-c(k))\frac{\sum_{l=1}^{h}\sum_{j=1}^{p}\underline{w}_{jl}(k-1)\underline{\tilde{\phi}}_{jl}(\tilde{x}_l(k))}{\sum_{l=1}^{h}\tilde{x}_l(k)} =$$

$$= c(k)\sum_{l=1}^{h}\sum_{j=1}^{p}\bar{w}_{jl}(k-1)\frac{\bar{\tilde{\phi}}_{jl}(\tilde{x}_l(k))}{\sum_{l=1}^{h}\tilde{x}_l(k)} + (1-c(k))\sum_{l=1}^{h}\sum_{j=1}^{p}\underline{w}_{jl}(k-1)\frac{\underline{\tilde{\phi}}_{jl}(\tilde{x}_l(k))}{\sum_{l=1}^{h}\tilde{x}_l(k)} =$$

$$= c(k)\sum_{l=1}^{h}\sum_{j=1}^{p}\bar{w}_{jl}(k-1)\bar{\tilde{\psi}}_{jl}(\tilde{x}_l(k)) + (1-c(k))\sum_{l=1}^{h}\sum_{j=1}^{p}\underline{w}_{jl}(k-1)\underline{\tilde{\psi}}_{jl}(\tilde{x}_l(k)) =$$
$$= c(k)\bar{w}^T(k-1)\bar{\tilde{\psi}}(\tilde{x}(k)) + (1-c(k))\underline{w}^T(k-1)\underline{\tilde{\psi}}(\tilde{x}(k))$$

where

$\underline{w}(k-1) = (\underline{w}_{11}(k-1), \underline{w}_{21}(k-1), \ldots, \underline{w}_{p1}(k-1), \underline{w}_{12}(k-1), \ldots, \underline{w}_{jl}(k-1), \ldots, \underline{w}_{ph}(k-1))^T$,

$\underline{\tilde{\psi}}(\tilde{x}(k)) = (\underline{\tilde{\psi}}_{11}(\tilde{x}(k)), \underline{\tilde{\psi}}_{21}(\tilde{x}(k)), \ldots, \underline{\tilde{\psi}}_{p1}(\tilde{x}(k)), \underline{\tilde{\psi}}_{12}(\tilde{x}(k)), \ldots, \underline{\tilde{\psi}}_{jl}(\tilde{x}(k)), \ldots, \underline{\tilde{\psi}}_{ph}(\tilde{x}(k)))^T$,

$\bar{w}(k-1) = \left(\bar{w}_{11}(k-1), \bar{w}_{21}(k-1), \ldots, \bar{w}_{jl}(k-1), \ldots, \bar{w}_{ph}(k-1)\right)^T$,

$\bar{\tilde{\psi}}(\tilde{x}(k)) = \left(\bar{\tilde{\psi}}_{11}(\tilde{x}(k)), \bar{\tilde{\psi}}_{21}(\tilde{x}(k)), \ldots, \bar{\tilde{\psi}}_{jl}(\tilde{x}(k)), \ldots, \bar{\tilde{\psi}}_{ph}(\tilde{x}(k))\right)^T$.

Adaptive Learning of HGAT2FWNN

The learning process of the layer L_1 is the tuning of synaptic weight vectors $\underline{w}(k)$ and $\bar{w}(k)$ of the neurons $\underline{AWN}$, $\overline{AWN}$ and scalar parameter $c(k)$ in the type-reduction block.

Since the output signals of wavelet neurons depend linearly on the synaptic weights, for their settings it is possible to use the exponentially weighted recursive least squares method, which is, in fact, the second-order optimisation procedure

$$\begin{cases} \underline{w}(k) = \underline{w}(k-1) + \dfrac{\underline{P}(k-1)\left(x(k) - \underline{w}^T(k)\underline{\tilde{\psi}}(\tilde{x}(k))\right)}{\beta + \underline{\tilde{\psi}}^T(\tilde{x}(k))\underline{P}(k-1)\underline{\tilde{\psi}}(\tilde{x}(k))}\underline{\tilde{\psi}}(\tilde{x}(k)), \\ \underline{P}(k) = \dfrac{1}{\beta}\left(\underline{P}(k-1) - \dfrac{\underline{P}(k-1)\underline{\tilde{\psi}}(\tilde{x}(k))\underline{\tilde{\psi}}^T(\tilde{x}(k))\underline{P}(k-1)}{\beta + \underline{\tilde{\psi}}^T(\tilde{x}(k))\underline{P}(k-1)\underline{\tilde{\psi}}(\tilde{x}(k))}\right), \end{cases} \tag{15}$$

$$\begin{cases} \bar{w}(k) = \bar{w}(k-1) + \dfrac{\bar{P}(k-1)\left(x(k) - \bar{w}^T(k)\bar{\tilde{\psi}}(\tilde{x}(k))\right)}{\beta + \bar{\tilde{\psi}}^T(\tilde{x}(k))\bar{P}(k-1)\bar{\tilde{\psi}}(\tilde{x}(k))}\bar{\tilde{\psi}}(\tilde{x}(k))), \\ \bar{P}(k) = \dfrac{1}{\beta}\left(\bar{P}(k-1) - \dfrac{\bar{P}(k-1)\bar{\tilde{\psi}}(\tilde{x}(k))\bar{\tilde{\psi}}^T(\tilde{x}(k))\bar{P}(k-1))}{\beta + \bar{\tilde{\psi}}^T(\tilde{x}(k))\bar{P}(k-1)\bar{\tilde{\psi}}(\tilde{x}(k))}\right) \end{cases} \tag{16}$$

where $0 < \beta \le 1$ – the forgetting factor.

To calculate the parameter $c(k)$ we can introduce an optimal adaptive procedure.

For off-line learning, we can write the learning error in the form

$$e(k) = y(k) - \hat{y}(k) = y(k) - c\bar{y}^w(k) - (1-c)\underline{y}^w(k) = y(k) - c\bar{y}^w(k) - \underline{y}^w(k) + c\underline{y}^w(k)$$

and the global learning criterion for the type-reduction block in the form

$$\begin{aligned} E(k) &= \sum_k \left(x(k) - \hat{x}^{[1]}(k) \right)^2 = \sum_k \left(x(k) - c\bar{y}^w - (1-c)\underline{y}^w(k) \right)^2 = \\ &= \sum_k \left(x(k) - \underline{y}^w(k) \right)^2 + 2\sum_k \left(c(x(k) - \underline{y}^w(k))(\underline{y}^w(k) - \bar{y}^w(k)) \right) + \sum_k c^2 \left(\underline{y}^w(k) - \bar{y}^w(k) \right)^2. \end{aligned} \tag{17}$$

Using criterion (1), we can provide optimization in the form

$$\frac{\partial E(k)}{\partial c} = 2\sum_k \left((x(k) - \underline{y}^w(k))(\underline{y}^w(k) - \bar{y}^w(k)) \right) + 2\sum_k c \left(\underline{y}^w(k) - \bar{y}^w(k) \right)^2 = 0 \tag{18}$$

and the expression for calculating parameter c can be written as follows

$$c = -\frac{\sum_k \left((x(k) - \underline{y}^w(k))(\underline{y}^w(k) - \bar{y}^w(k)) \right)}{\sum_k \left(\underline{y}^w(k) - \bar{y}^w(k) \right)^2}. \tag{19}$$

Using the global learning criterion (17), we can obtain the optimal adaptive procedure for tuning parameter c.

Based on (19), we can write the value of parameter c for k and $k+1$ instant time

$$c(k) = -\frac{\sum_{i=1}^{k} \left((x(i) - \underline{y}^w(i))(\underline{y}^w(i) - \bar{y}^w(i)) \right)}{\sum_{i=1}^{k} \left(\underline{y}^w(i) - \bar{y}^w(i) \right)^2} \tag{20}$$

or in the recurrent form

$$\begin{cases} \gamma(k) = \gamma(k-1) + \left(\underline{y}^w(k) - \bar{y}^w(k) \right)^2, \\ c(k) = c(k-1)\dfrac{\gamma(k-1)}{\gamma(k)} + \dfrac{\left(x(k) - \bar{y}^w(k) \right)\bar{y}^w(k)\left(\underline{y}^w(k) - \bar{y}^w(k) \right)}{\gamma(k)}. \end{cases} \tag{21}$$

It is clear that the condition of optimal learning is the closeness of signals $\underline{y}^w(k)$ and $\bar{y}^w(k)$, "subtending" of type-2 fuzzy wavelet membership function to common "Mexican Hat" wavelet and approximating of parameter $c(k)$ to the value 0.5.

The learning process of layers $L_2, L_3, \ldots, L_g$ is the same. At that after the next observation $x(k)$ is fed, all layers tune their parameters sequentially.

For making a one-step forecast, the signal in the form $X(k+1)=(x(k),x(k-1),\dots,x(k-n+1))^T$ is fed to the output tuned system, after that the feature observation $\hat{x}^{[g]}(k+1)$ is appeared.

Conclusions

Proposed deep stacking forecasting hybrid system of computational intelligence is enough simple in computational implementation due to parallelizing process of implemented computations, has high learning rate due to dismissal from errors backpropagation and using the rapid adaptive algorithms for tuning its parameters, has also flexibility due to possibility of tuning the activation-membership functions. Proposed system is aimed at solving the wide range Data Stream Mining tasks, such as forecasting, identification, emulation, classification and pattern recognition in online mode. Experiments have confirmed the effectiveness of proposed system under consideration.

Acknowledgement

The paper is published with partial support by the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua)

Bibliography

[Abiyev and Kaynak, 2008] Abiyev R., Kaynak O., Fuzzy wavelet neural networks for identification and control of dynamic plants – a novel structure and a comparative study. IEEE Trans. on Industrial Electronics, vol. 2, no. 55, pp. 3133–3140, 2008. DOI 10.1109/TIE.2008.924018

[Aggarwal, 2007] C. Aggarwal, Data Streams: Models and Algorithms (Advances in Database Systems). Springer, 2007. DOI 10.1007/978-0-387-47534-9

[Alexandridis and Zapranis, 2014] A.K. Alexandridis and A.D. Zapranis, Wavelet Neural Networks: with Applications in Financial Engineering, Chaos, and Classification. John Wiley & Sons, 2014. DOI 10.1002/9781118596272

[Aliev and Guirimov, 2014] R.A. Aliev and B.G. Guirimov, Type-2 Fuzzy Neural Networks and Their Applications. Switzerland: Springer; 2014. DOI 10.1007/978-3-319-09072-6

[Bifet, 2010] A. Bifet, Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams. Amsterdam: IOS Press, 2010. DOI 10.3233/978-1-60750-472-6-i

[Bodyanskiy et all, 2001] Bodyanskiy Ye., Kolodyazhniy V., Stephan A., An adaptive learning algorithm for a neuro fuzzy network. Ed. by B. Reusch “Computational Intelligence Theory and Applications.” Berlin Heidelberg New York: Springer, 2001, pp. 68-75. DOI 10.1007/3-540-45493-4_11

[Bodyanskiy et all, 2005] Bodyanskiy Ye., Kolodyazhniy V., Otto P., Neuro-fuzzy Kolmogorov’s network for time-series prediction and pattern classification. Lecture Notes in Artificial Intelligence. V. 3698. Heidelberg: Springer –Verlag, 2005, pp. 191-202. DOI 10.1007/11551263_16

[Bodyanskiy et all, 2005a] Bodyanskiy Ye., Lamonova N., Pliss I., Vynokurova O., An adaptive learning algorithm for a wavelet neural network. *Expert Systems*, vol. 22, no. 5, pp. 235–240, 2005. DOI 10.1111/j.1468-0394.2005.00314.x

[Bodyanskiy et all, 2008] Bodyanskiy Ye., Popov S., Rybalchenko T., Multulayer neuro-fuzzy network for short term electric load forecasting. Lecture Notes in Computer Science. V.5010. Berlin Heidelberg: Springer-Verlag, 2008, pp. 339-348. DOI 10.1007/978-3-540-79709-8_34

[Bodyanskiy and Vynokurova, 2011] Bodyanskiy Ye., Vynokurova O., Hybrid type-2 wavelet-neuro-fuzzy network for businesses process prediction. Business Informatics, no. 21, pp. 9–21, 2011.

[Bodyanskiy et all, 2012] Bodyanskiy Ye., Pliss I., Vynokurova O., Type-2 Fuzzy-Wavelet-Neuron for Solving Data Mining Problems. In: Proc. East West Fuzzy Colloquium 2012, 19-th Zittau Fuzzy Colloquium. Zittau / Goerlitz: HS, 2012, pp. 96–103.

[Bodyanskiy and Vynokurova, 2013] Bodyanskiy Ye., Vynokurova O., Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification. Information Science, no. 220, pp. 170–179, 2013. DOI 10.1016/j.ins.2012.07.044

[Bodyanskiy et all, 2014] Bodyanskiy Ye., Tyshchenko O., Kopaliani D., A multidimensional cascade neuro-fuzzy system with neuron pool optimization in each cascade. Int. J. of Information Technology and Computer Science. vol.6, iss. 8, pp. 11-17, 2014. DOI 10.1007/s00500-014-1344-3

[Bodyanskiy et all, 2014a] Bodyanskiy Ye., Kharchenko O., Vynokurova O., Least squares support vector machine based on wavelet-neuron. Information Technology and Management Science, vol. 7, pp. 19–24, 2014. DOI 10.1515/itms-2014-0002

[Bodyanskiy et all, 2015] Bodyanskiy Ye., Pliss I.P., Tatarinova Yu., Vynokurova O. Hybrid Generalized Additive Type-2 Fuzzy-Wavelet-Neural Network in Dynamic Data Mining Tasks. Information Technology and Management Science, 18, pp. 70-77, 2015. DOI 10.1515/itms-2015-0011

[Bodyanskiy and Tyshchenko, 2016] Bodyanskiy Ye., Tyshchenko O., Fast Deep Learning for Cascade Neuro-Fuzzy System. In: 2nd Int. Workshop and Networking Event “Advances in Data Science”. Holny Mejera, Poland, 2016, pp. 18-19.

[Bodyanskiy et all, 2016] Bodyanskiy Ye., Mulesa P., Setlak G., Pliss I., Vynokurova O., Fast learning algorithm for deep evolving GMDH-SVM neural network in data stream mining tasks. In: Proc. of

2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), 2016, pp. 257-262. DOI 10.1109/DSMP.2016.7583555

[Bodyanskiy et all, 2016a] Bodyanskiy Ye., Pliss I., Peleshko D., Vynokurova O., Adaptive Deep Learning of Multivariate Hybrid System of Computational Intelligence. In: VII-th Int. Conf. “Decision-Making Theory”, Uzhhorod, Ukraine, 2016, pp. 58-59. (In Ukrainian)

[Du and Swamy, 2014] K.-L. Du, M.N.S. Swamy, Neural Networks and Statistical Learning. London: Springer-Verlag, 2014. DOI 10.1007/978-1-4471-5571-3

[Fahlman and Lebiere, 1990] Fahlman S.E., Lebiere C., The Cascade-Correlation Learning Architecture. Advances Neural Information Processing Systems, San Mateo, CA: Morgan Kaufman, pp. 524-532, 1990. ISBN:1-55860-100-7

[Goodfellow et all, 2016] I. Goodfellow, Y. Bengio, A. Courville. Deep learning. MIT Press, 2016.

[Hastie and Tibshirani, 1990] T. Hastie, R. Tibshirani, Generalized Additive Models. Chapman and Hall/CRC, 1990.

[Haykin, 1999] S. Haykin, Neural Networks. A Comprehensive Foundation. Upper Saddle River. N.J.: Prentice Hall, 1999.

[Huang et all, 2014] Huang W., Song G., Hong H., Xie K. Deep architecture for traffic flow prediction. Deep belief networks with multitask learning. IEEE Trans. Intell. Transp. Syst. vol. 15, is. 5, pp. 2191-2201, 2014. DOI 10.1109/TITS.2014.2311123

[Jang and Mizutani, 1997] J.-S. R. Jang, C. T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. N.J.: Prentice Hall, 1997. ISBN:0-13-261066-3

[Längkvist et all, 2014] Längkvist, M., Karlsson, L., Loutfi, A. A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognition Letters, vol. 42, is. 1, pp. 11-24, 2014. DOI 10.1016/j.patrec.2014.01.008

[LeCun et all, 2015] LeCun Y., Bengio Y., Hinton G. E., Deep Learning. Nature. 521, pp. 436-444, 2015.

[Mumfold and Jain, 2009] C.L. Mumfold, L.C. Jain, Computational Intelligence Collaboration, Fusion and Emergence. Berlin: Springer-Verlag, 2009. DOI 10.1007/978-3-642-01799-5

[Lughofer, 2011] E. Lughofer, Evolving Fuzzy Systems - Methodologies, Advanced Concepts and Applications. Springer, 2011. DOI 10.1007/978-3-642-18087-3

[Mitaim and Kosko, 1996] Mitaim S., Kosko B., What is the best shape for a fuzzy set in function approximation? In: Proc. 5th IEEE Int. Conf. on Fuzzy Systems “Fuzz-96”, vol. 2, 1996, pp. 1237–1213. DOI 10.1109/fuzzy.1996.552354

[Mitaim and Kosko, 1997] Mitaim S., Kosko B., Adaptive joint fuzzy sets for function approximation. In: Proc. Int. Conf. on Neural Networks "ICNN-97", 1997, pp. 537–542. DOI 10.1109/icnn.1997.611726

[Nelles, 2001] O. Nelles, Nonlinear System Identification. Berlin: Springer, 2001. DOI 10.1007/978-3-662-04323-3

[Rutkowski, 2008] L. Rutkowski, Computational Intelligence: Methods and Techniques. Berlin: Springer-Verlag, 2008. DOI 10.1007/978-3-540-76288-1

[Schmidhuber, 2015] Schmidhuber, J., Deep learning in neural networks: An overview. Neural Networks. 61, pp. 85-117, 2015. DOI 10.1016/j.neunet.2014.09.003

[Setlak et all, 2016] Setlak G., Bodyanskiy Ye., Pliss I.P, Vynokurova O., Deep Evolving GMDH-SVM-Neural Network and its Learning for Data Mining Tasks. In: Proc. of the Federated Conference on Computer Science and Information Systems (Annals of Computer Science and Information Systems, Volume 5), 2016. Gda´nsk, Poland, pp. 147–151. Electronic ISBN: 978-8-3608-1090-3

[Sugeno and Kang, 1988] Sugeno M., Kang G.T., Structure identification of fuzzy model. Fuzzy Sets and Systems, vol. 28, pp. 15–33, 1988. DOI 10.1016/0165-0114(88)90113-3

[Takagi and Sugeno, 1985] Takagi T., Sugeno M., Fuzzy identification of systems and its applications to modeling and control. IEEE Trans. on Systems, Man and Cybernetics, vol 15, no. 1, pp. 116–132, 1985. DOI 10.1109/TSMC.1985.6313399

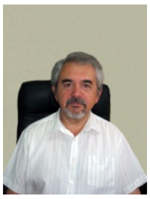
[Takagi and Hayashi, 1991] Takagi H., Hayashi I., NN-driven fuzzy reasoning. Int. J. of Approximate Reasoning, vol. 5, no. 3, pp. 191–212, 1991. DOI 10.1016/0888-613X(91)90008-A

[Uchino and Yamakawa, 1997] Uchino E., Yamakawa T., Soft computing based signal prediction, restoration and filtering. Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks and Genetic Algorithms, Da Ruan Eds., Boston: Kluwer Academic Publisher, pp. 331–349, 1997. DOI 10.1007/978-1-4615-6191-0_14

[Yamakawa et all, 1998] Yamakawa T., A novel nonlinear synapse neuron model guaranteeing a global minimum – Wavelet neuron. In: Proc. 28-th IEEE Int. Symp. on Multiple-Valued Logic. Fukuoka, Japan: IEEE Comp. Soc., 1998, pp. 335-336. DOI 10.1109/ISMVL.1998.679510

[Yamakawa et all, 1992] Yamakawa T., Uchino E., Miki T., Kusanagi H., A neo-fuzzy neuron and its applications to system identification and prediction of the system behaviour. In: *Proc.* 2-nd Int. Conf. on Fuzzy Logic and Neural Networks "IIZUKA-92", Iizuka, Japan, 1992, pp. 477–483.

Authors' Information



Yevgeniy Bodyanskiy – *Dr. Sc., Prof., Control System Research Laboratory, Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine;*

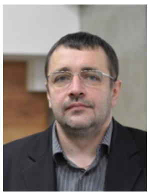
e-mail: yevgeniy.bodyanskiy@nure.ua

Major Fields of Scientific Research: Computational Intelligence, Hybrid Evolving Systems, Data Stream Mining, Clustering, Forecasting.

Iryna Pliss – *Ph.D., Leading reseacher, Control System Research Laboratory, Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine;*

e-mail: iryna.pliss@nure.ua

Major Fields of Scientific Research: Data Mining, Computational Intelligence, Hybrid Evolving Systems, Clustering, Forecasting.



Dmytro Peleshko – *Dr. Sc., Prof., University "IT Step Academy", 83a Zamarstynivs'ka st., 79019, Lviv, Ukraine*

e-mail: dpeleshko@gmail.com

Major Fields of Scientific Research: Computational Intelligence, Pattern Recognition, Computer Vision, Image and Video Processing.

Olena Vynokurova – *Dr. Sc., Prof., Control System Research Laboratory, Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine;*

e-mail: vynokurova@gmail.com

Major Fields of Scientific Research: Computational Intelligence, Hybrid Wavelet Neural Systems, Data Stream Mining, Clustering, Forecasting.