ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 1, July 2015

# Bankruptcy Risk Forecasting under Uncertainty Application of Fuzzy Neural Network NEF Class

Ovi Nafas Aghaei Agh Ghamish (Iran)
Yuriy Zaychenko Kiev Polytechnic Institute, Kiev, Ukraine

***Abstract-* The problem of corporations' bankruptcy risk forecasting under uncertainty is considered in this paper. The application of fuzzy neural network NEFClass-M is suggested. The structure of FNN NEFClass-M is presented the learning algorithms for fuzzy rules base and membership functions are described. The experimental investigations of of FNN NEFClass for bankruptcy risk forecasting for Ukrainian corporations were carried out and their efficiency was estimated and compared with fuzzy neural networks with algorithms of Mamdani, Tsukamoto and classical statistical methods.**

***Index terms-* bankruptcy risk forecasting, FNN NEFClass-M, fuzzy networks of Mamdani and Tsukamoto.**

## I. INTRODUCTION

The task of financial analysis and forecasting of risk of bankruptcy of corporations is one of the most important. The timely identification of signs of possible bankruptcy allows corporate management to make operational decisions to improve the financial condition and prevent a possible bankruptcy. Traditionally, this problem was solved with the use of classical methods of multivariate discriminant analysis (MDA), beginning with the pioneering work of Professor. E. Altman 1968 [2]. Further, this method was developed by Lisa, Daytona - Belikova [3] and other authors. A feature of this problem in terms of Ukraine's economy is the presence of considerable uncertainty caused by the desire to guide corporations to hide the true state sponsored enterprises to obtain loans. This calls for the development and application of new methods and algorithms, allowing working in fuzzy and uncertain information. These methods are based on a system of fuzzy logic and fuzzy neural networks. In our previous studies to the problem of predicting the risk of bankruptcy have been proposed and investigated fuzzy neural network (FNN) and Mamdani Tsukamoto [5] , as well as cascading neo-fuzzy neural networks. There were experimental studies of their effectiveness in relation to the economy of Ukraine and the comparative analysis with classical methods. The aim of this work is the development of the proposed methods and research in FNN NEFClass problem predicting the risk of bankruptcy of corporations.

## II. FNN NEFCLASS, ARCHITECTURE, LEARNING ALGORITHM FOR RULE BASE AND MEMBERSHIP FUNCTIONS

The problem of classifying data is currently one of the most pressing areas of applications of artificial intelligence. To solve this problem have been proposed various approaches and directions, including the most popular acquired solutions that combine neural networks and fuzzy inference system. One of these solutions is the system NEFClass (Neuro-Fuzzy Classifier), based on the generalized architecture of a fuzzy perceptron [4].

Both original and modified model NEFClass are derived from the general model of fuzzy perceptron as described in [4]. Goal model - fuzzy inference rules from the set of data that can be divided into a number of (distinct) disjoint classes.

Vagueness occurs due to inadequate or incomplete measurement properties of the objects to be classified.

Fuzzy rules describing the data have the following form

If $x_1$ and $x_2$ is $\mu_1$ and $\mu_2$ is ... and $x_n$ is the $\mu_n$,,

the sample $(x_1, x_2, ..., x_n)$ belongs to the class i,

where, $\mu_i$ ..., $\mu_n$ - MF fuzzy sets.

Task NEFClass - determine the rules and kind of membership functions for the fuzzy sets. Here it is assumed that the intersection of two different sets is empty.

NEFClass system has 3- layer serial architecture (see. Figure 1). The first layer comprises input neurons, which is the input samples. Activation of neuron usually does not change the input value. The hidden layer comprises a fuzzy rule, and the third layer consists of output neurons for each class. Activation of neurons and the rules for the output layer of neurons with a sample p is calculated as follows:

$$a_R^{(p)} = \min_{x\in U_1}\{W(x,R)(a_x^{(p)})\}, \qquad (1)$$

$$a_c^{(p)} = \sum_{R\in U_2} W(c,R)\cdot a_R^{(p)}, \qquad (2)$$

oralternatively

$$a_c^{(p)} = \max_{R\in U_2}\{(a_R^{(p)})\}, \qquad (3)$$

where W (x, R) - weight compounds fuzzy input neuron to neuron x rules R , and W (R, c) - the weight of the compound neuron fuzzy rules R to the output neuron layer. Instead of using operations of taking the maximum and minimum of other functions can be used t-norms and t-norms respectively.

The rule base is an approximation of the unknown function $\varphi : R^n \to \{0;1\}^m$ and describes the classification problem , where $\varphi(x) = (c_1,...,c_m)$ such that $c_i$=1, c$_j$=0 ( $j = 1...m, \forall \ j \neq i$ ) ,If $x$ belongs to the class $C_i$.

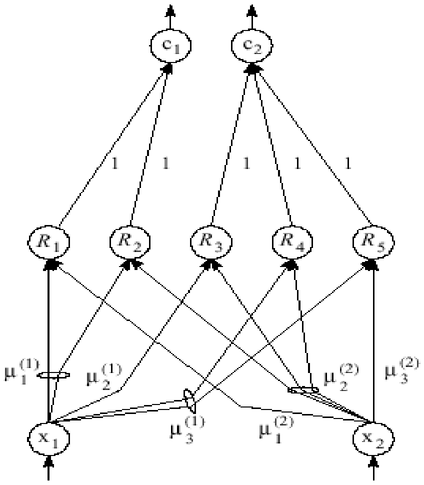

**Fig.1. FNN NEFClass with two inputs and five rules Each fuzzy set is marked by linguistic terms , such as "large" , "small" , "medium" , etc.**

Fuzzy sets and linguistic rules represent an approximation of classifying function and determine the result of NEFClass. They are obtained from the sample by learning. Must necessarily comply with the rules, for each linguistic value (for example, «x1 and large positive"), there can be only one representation of a fuzzy set.

## III. RULE BASE LEARNING

NEFClass system can be constructed by a partial knowledge of the samples. The user must determine the amount of the initial fuzzy sets for each of the attributes of the object and set the value of $k_{max}$ - maximum number of nodes of rules that can be created in the hidden layer . For learning using triangular membership functions. Consider just learning algorithm rule base

**Step 1:** Generation of the rule base

The first phase , which aims to establish rules of the system of neurons $k$ NEFClass, consists of the following steps [4]:

1. Select the following sample (p, t) of L.
2. For each input neuron xi ЄU1 find a membership function $\mu^i_{J_i}$ that

$$\mu^{(p)}_{Ji} = \max_{j \in 1,..q_1} \{\mu^{(p)}_{ji}(p_i)\}, \qquad (4)$$

Where xi=$p_i$.

3. If still rules the number of nodes k is less than $k_{max}$ *and* there is no rule node R , such that

$$W(x_1,R) = \mu_{J1},...,W(x_n,R) = \mu_{Jn},$$

then create a node and connect it to the output node $c_i$, if t$_i$.=1

4. If there are still unprocessed samples L and $k$<=$k_{max}$ , then we go to step 1 , otherwise stop.
5. Determine the rule base for one of the three following procedures:

   a)"simple" learning rules: leave only the first $k$ rules (stop the creation of rules , when it was created $k$=$k_{max}$ rules).

   b) the "best" learning rules : treat samples and accumulate L activation of each neuron rules for each class of samples that have been distributed . If a neuron rules R shows a greater

$$V_R = \sum_{p \in L} a^{(p)}_R \cdot e_p , \qquad (5)$$

$$e_p = \begin{cases} 1, & \text{if p classification equal right} \\ -1, & \text{otherwise} \end{cases}$$

We reserve the neurons $k$ rules with the highest values of $V_R$ and remove rules from other neurons of NEFClass.

c) "Best of each class" learning algorithm: act as in the previous case , but the reserve for each class Cj are the best $\left[\frac{k}{m}\right]$ rules , the investigation of which represent the class C$_j$ (where [$x$] – the integral part of $x$).

## IV. FUZZY SETS LEARNING

The second step is learning the parameters of membership functions (MF) of fuzzy sets . Supervised learning algorithm NEFClass system must adapt its MF of fuzzy sets; algorithm cyclically runs through the entire training set L , by performing the steps listed below, will run until one of the stopping criteria [4]:

1. Select the following sample (p, t) of the sample L, applies it to the input of the system and determine NEFClass output vectors.

2. For each output neuron $c_i$ determine the value $\delta_{C_i}$

$$\delta_{C_i} = t_i - a_{C_i}$$

Where $t_i$– the desired output, $a_{c_i}$ –the actual output of a neuron c$_i$.

3. For each neuron rules $R$, for which the output $a_R$>$0$:

a)We determine the value $\delta_R$ equal to ,

$$\delta_R = a_R \cdot (1 - a_R) \cdot \sum_{C \in U_3} W(R,C)\delta_C . \qquad (6)$$

b)we find the $x'$ , that the

$$W(x',R)(a_{x'}) = \min_{x \in U_1}\{W(x,R)(a_x)\} \qquad (7)$$

DOI:10.17605/OSF.IO/P7KDG

c) for fuzzy set $W(x', R)$ we determine the offset parameters MF $\Delta_a, \Delta_b, \Delta_c$

using the speed of speed learning $\sigma > 0$

$$\Delta_b = \sigma \cdot \delta_R \cdot (c-a) \cdot \mathrm{sgn}(a_{x'} - b)\,, \tag{8}$$

$$\Delta_a = -\sigma \cdot \delta_R \cdot (c-a) + \delta_b\,, \tag{9}$$

$$\Delta_c = \sigma \cdot \delta_R \cdot (c-a) + \delta_b\,. \tag{10}$$

and make changes $W(x', R)$

d) calculate the error rules :

$$E = a_R \cdot (1 - a_R) \cdot \sum_{c \in U_3} (2 \cdot W(R,c) - 1) \cdot |\delta_c|\,. \tag{11}$$

The end of the iteration. Repeat the iteration before the trip condition.

The criteria for stopping can take, for example, as follows:

1. Error for n iterations is not reduced.
2. Stop the training after reaching a certain error (preferably close to zero) values.

## V. MODIFIED FNN NEFCLASS- M, LEARNING ALGORITHMS

The above learning algorithm of fuzzy sets MF an empirical question and its convergence is not proven. This is due to the fact that MF triangular, non-smooth in terms of the operation of intersection of the rules used at least, the composition of output rules of operation is used most . As a result of these operations is a non-differentiable criterion, which does not apply numerical optimization techniques based on the gradient method or Newton optimization. Therefore, modification NEFClass - M was developed in [4].This network uses smooth MF, for example, Gaussian or bell-shaped, crossing conditions are realized in the form of product, and the composition of the outputs of the rules implemented as a weighted sum of [4].As a result, the overall optimization criterion becomes differentiable. We give a description of the learning algorithms FNN NEFClass-M.

## VI. THE GRADIENT LEARNING ALGORITHM

For the first stage of the algorithm - learning rule base used by the first phase of the basic algorithm NEFClass. The second stage uses the gradient algorithm neural network learning direct action, which is described below.

Let the criterion of learning fuzzy neural network, which has 3 layers (one hidden layer), as follows:

$$e(W) = \sum_{i=1}^{M} (t_i - NET_i(W))^2 \to min\,, \tag{12}$$

Where $t_i$ - the desired value of the i-th output neural network.

$NET_i(w)$ the actual value of the i-th neural network output for the weight matrix $W = [W^I, W^0]$

$$W^I = W(x,R) = \mu_j(x), \quad W^0 = W(R,C)$$

That is , criterion $e(w)$ it is the mean square error of approximation.

Let activation function for the hidden layer neurons (neurons rules):

$$O_R = \prod_{i=1}^{N} \mu_{j_i}^{(i)}(x_i)\,, \quad j = 1,\ldots,q_i\,, \tag{13}$$

Where $\mu_{ji}(x)$ membership function , which has the form:

$$\mu_{j_i}^{(i)}(x) = e^{-\frac{(x - a_{ji})^2}{b_{ji}{}^2}}\,, \tag{14}$$

and the activation function of neurons in the output layer (weighted sum):

$$O_C = \frac{\sum_{R \in U_2} W(R,C) \cdot O_R}{\sum_{R \in U_2} W(R,C)}\,, \tag{15}$$

or maximum function:

$$O_C = \max W(R,C) \cdot O_R\,. \tag{16}$$

Consider the gradient learning algorithm fuzzy perceptron

1. Let $W(n)$ the current value of the weights matrix. The algorithm has the following form:

$$W(n+1) = W(n) - \gamma_{n+1} \nabla_w e(W(n))\,, \tag{17}$$

Where $\gamma_n$ step size to –th iteration

$\nabla_w e(W(n))$ Gradient (direction) , which reduces the criterion (12).

2. at each iteration, we first learning (adjust) the input weight W , which dependent on the parameters a and b (see. The expression 14)

$$a_{ji}(n+1) = a_{ji}(n) - \gamma_{n+1} \frac{\partial e(W)}{\partial a_{ji}}\,, \tag{18}$$

$$b_{ji}(n+1) = b_{ji}(n) - \gamma`_{n+1} \frac{\partial e(W)}{\partial b_{ji}}\,, \tag{19}$$

Where $\gamma`_{n+1}$ step size parameter b.

$$\frac{\partial e(W)}{\partial a_{ji}} = -2 \sum_{k=1}^{M} ((t_k - NET_k(w)) \cdot W(R, C_k)) \cdot O_R \cdot \frac{(x - a_{ji})}{b_{ji}{}^2}$$

, (20)

$$\frac{\partial e(W)}{\partial b_{ji}} = -2 \sum_{k=1}^{M} ((t_k - NET_k(w)) \cdot W(R, C_k)) \cdot O_R \cdot \frac{(x - a_{ji})^2}{b_{ji}{}^3}$$

. (21)

3. we find the (learn) output weight :

$$\frac{\partial e(W^0)}{\partial W(R, C_k)} = -(t_k - NET_k(W^0)) \cdot O_R\,, \tag{22}$$

DOI:10.17605/OSF.IO/P7KDG

$$W_k^0(n+1)=W_k^0(n)-\gamma``_{n+1}\frac{\partial e(W^0)}{\partial W(R,C_k\ )} \qquad (23)$$

4. $n:=n+1$ and go to the next iteration
The gradient method is first suggested learning algorithm, it is easy to implement, but has the disadvantages:

a) converges slowly;
b) It is only a local extreme.

***Conjugate Gradient Method For The System Nefclass-M***

Conjugate gradient algorithm, as well as more general algorithm of conjugate directions, was used in the field of optimization thanks to a wide class of problems for which it ensures the convergence to the optimal solution for a finite number of steps [4].

Conjugate directions. The name comes from the use of conjugate vectors. The vector space of dimension N vector set of conjugate directions. The name comes from the use of conjugate vectors. The vector space of dimension N the set of vectors $\{P_1, P_2, \dots, P_D\}$ forms aset of conjugate directions with respect to the matrix A , if $P_i AP_j = 0$ for $j \neq i$
(24)

where *A*- positive definite matrix of size

Vectors that satisfy (24) , called A-conjugate.

In the N-dimensional space have exactly $N-1$ independent vectors , which form the A-conjugate pair with the vector $p_1$ . Thus, we will need only a finite number of directions in order to find the optimal solution

The algorithm of conjugate directions systematically constructs the set of A-conjugate vectors. After a maximum of N steps the algorithm finds the optimal direction and convergence will be provided.. In this problem neural network learning, we do not have an explicit expression for the matrix A, although the gradient error $\nabla E$ It can fulfill this role. We give a brief description of the conjugate gradient algorithm.

0. Assume that $K=1$. Initialize the weight vector $W$ and calculate the gradient $G = grad\ E(W)$
Suppose that the the initial direction vector $p_K = -\frac{G}{\|G\|}$

1.Find a scalar $\alpha^*$ ,which minimizes $E(W+\alpha p)$ ,
How can you use the Fibonacci and "golden section".

$$W(K+1)=W(K)+\alpha^* p(K)\ . \qquad (25)$$

2.if $E(W(K+1)) < \varepsilon_{npun}$ , where $\varepsilon_{npun}$ tolerance on achieving minimum, STOP . Otherwise calculate a new direction:

$$G(k+1)=gradE(W(k+1)). \qquad (26)$$

3.if $mod_N K = 0$ , the new direction vector:

$$P(k+1) = -\frac{G(k+1)}{\|G(k+1)\|}, \qquad (27)$$

Otherwise calculate the parameter $\beta$ :

$$\beta = \frac{G(K+1)^T G(K+1)}{G(K)^T G(K)}\ , \qquad (28)$$

and to calculate a new vector direction

$$P_{k+1} = \frac{-G(k+1)+\beta p(k)}{\|-G(k+1)+\beta p(k)\|}. \qquad (29)$$

4. replace the $p(k)$ on the $p(k+1)$ and $G(k)$ , on the $G(k+1)$ .step 1 go to the next iteration

In this algorithm $G(k)$ calculated for two parameters MF (a,b) sequentially or separately, as depicted in the gradient algorithm. Speed learning for these two parameters is also configured separately. Note that the conjugate gradient method has a significantly higher rate of convergence in comparison with the gradient method.

***Genetic Method for learning System NEFClass-M***

Consider the implementation of a genetic algorithm for learning FNN NEFCLASS .This algorithm is a global optimization algorithm. It uses the following mechanisms

a) crossing of parental pairs ,the generation of offspring
b) mutation (random effect influences) ;
c) the natural selection of the best (selection);

Goal of learning -minimization of the mean square error:

$$E(W) = \frac{1}{M}\sum_{k=1}^{M}(t_k - NET_k(W))^2\ , \qquad (30)$$

Any individual seems appropriate vector of weights $W$ Sets the initial population of $N$ individuals $[W_1(0),\dots,W_i(0),\dots,W_N(0)]$

We calculate the index the suitability (FI), and evaluate the quality of forecasting:

$$FI(W_i) = C - E(W_i) \rightarrow max, \qquad (31)$$

Where *C - constant*

Next comes the crossing of parental pairs. When selecting parents used a probabilistic mechanism

Lets us denote $P_i$ - the probability of selecting the i-th of his father:

$$P_i = \frac{FI(W_i(0))}{\sum_{i=1}^{N} FI(W_i(0))}, \qquad (32)$$

Then he made the crossing of selected pairs. You can apply different mechanisms of crossing .For example: for the first child is taken even components of the vector of the first parent and the odd components of the vector of the other parent, and the second on the contrary:

DOI:10.17605/OSF.IO/P7KDG

Take the $\frac{N}{2}$ parental pairs and generated $N$ descendants.

After the generated descendants of, the mutation acts on the population:

$$w'_{ij}(n)=w_{ij}(n)+\xi(n) \quad (33)$$

Where $a=const\in[-1;+1]$; $\xi(n)=r_0e^{-\alpha n}$

$\alpha$ – mutation rate of extinction; $r_0$– is randomly selected from the interval [-1, 1].

Then, after the effect of mutation breeding occurs in a population, which allows to choose the "fittest" individuals. You can use different mechanisms of selection.

1.complete replacement of the old to the new population.

2. selecting $N$ best of all existing species $N_{parent}+N_{\text{child}}=2N$ by the criterion of maximum max (FI).

After the crossing, mutation and selection of the current iteration ends. The iterations are repeated until, until one of the criteria carried out stop.

The genetic algorithm requires a lot of computational cost, but in contrast to the gradient method allows to find the limit globally optimal solution.

## VII. EXPERIMENTAL RESEARCH OF BANKRUPTCY RISK FORECASTING METHODS

To analyze the different methods to assess the risk of bankruptcy was developed software package , which implements the classical method of discriminant analysis Altman method Davydova, Belikov, fuzzy neural network NEFClass M Using the developed software complex were conducted for predicting bankruptcy of fifty-eight enterprises in Ukraine. 29 of them in 2011 by the arbitral tribunal had been declared bankrupt. The input data for the calculations used the following financial indicators, which are calculated based on data from the accounting statements (balance sheet and income statement) for 2009 and 2010:

- X1 – equity ratio (the ratio of equity to the balance sheet);
- X2 - ratio of current assets to ensure their own means (the ratio of networking capital as current assets);
- X3 –intermediate liquidity ratio (the ratio of cash and receivables to short-term liabilities);
- X4 – absolute liquidity ratio (the ratio of cash to short-term liabilities);
- X5 - the turnover of all assets for the year (the ratio of revenue from sales to the average earnings in the period the value of assets);

X6 - Return on total capital (the ratio of net profit to the average for the period of asset value).

Forecasting was conducted using models Altman, Davydova Belikov and FNN NEF Class-M. In tables 1, 2 presents the results of prediction - the percentage of misclassification of bankruptcy for one year before the bankruptcy Altman statistical method and model Davydova - Belikov, respectively .Tables 3 and 4 show the results of the classification using the FNN NEFClass-M

**Table 1. Results of Prediction by Method of Altman One Year before Bankruptcy**

| | Altman |
|---|---|
| The error of the first type | 0.3 |
| The error of the second type | 0.344 |
| The number of errors of the first type | 9 |
| The number of errors of the second type | 10 |
| The relative number of errors | 0.327 |

**Table 2. Results of risk prediction by model of Davydova-Belikov one year before bankruptcy**

| | **model of Davydova-Belikov** |
|---|---|
| The error of the first type | 0.206 |
| The error of the second type | 0.31 |
| The number of errors of the first type | 6 |
| The number of errors of the second type | 9 |
| The relative number of errors | 0.258 |

**Table 3. Results of prediction using FNN NEFClass-M the year before bankruptcy**

| | Learning sample | The test sample |
|---|---|---|
| The error of the first type | 0.104 | 0.138 |
| The error of the second type | 0.069 | 0.138 |
| The number of errors of the first type | 3 | 4 |
| The number of errors of the second type | 2 | 4 |
| The relative number of errors | 0.094 | 0.138 |

**Table 4. Results of prediction using FNN NEFClass-M for 2 years prior to bankruptcy**

| | Learning sample | The test sample |
|---|---|---|
| The error of the first type | 0.172 | 0.206 |
| The error of the second type | 0.138 | 0.172 |
| The number of errors of the first type | 5 | 6 |
| The number of errors of the second type | 4 | 5 |
| The relative number of errors | 0.094 | 0.138 |

Thus , a method that predicted the bankruptcy with the greatest accuracy for the year before the bankruptcy proved method based on fuzzy neural network NEFClass-M. It showed 86% correct classification of one year before the bankruptcy, and 81% - two years before the bankruptcy. On verification sample. Deviation 14% and 19% due to input data. We do not have 100% confidence in the breakdown of the entire sample for bankrupts and successful businesses. After screening t he sample may have certain inaccuracies that influence the magnitude of the errors In [Ovi Nafas, 2012] have been proposed for predicting the risk of bankruptcy to use FNN with the conclusion of Mamdani and Tsukamoto. In this sample we used the same raw data as in the present experiments. The experimental results were so. FNN Mamdani has shown the best results: one year before the bankruptcy - 90% of correctly classified examples, two years before the bankruptcy,85.71% FNN Tsukamoto showed the following results: one year before the bankruptcy , 84.7% of correct classifications, and two years before the bankruptcy, 82.15%.

Thus, all studied classes of fuzzy neural networks have shown similar results, which are significantly superior to the results of predicting the risk of bankruptcy by the classical method of multivariate discriminant analysis.

(Altman model and Davydovoy- Belikov). But unlike FNN Mamdani in which the rule base designs experts, FNN NEFClass-M rules are generated automatically, and it is possible to optimize it. This results in some cases, certain advantages. High classification accuracy using FNN for two reasons. Firstly, the FNN based on fuzzy logic is used for classification in the face of uncertainty and heterogeneity of data. Secondly, the parameters of the FNN in the learning process have been adapted so that the best approximation of desired relationship between input and output.

## VIII. CONCLUSIONS

The article describes the method for predicting the risk of bankruptcy in the face of uncertainty, using fuzzy neural networks NEFClass-M. Learning algorithms for the rule base and membership functions of the FNN NEFClass-M were developed. Experimental studies of the proposed method and its comparison with the classical methods of discriminant analysis E, Altman and Davydova, Belikov, as well as the with FNN Mamdani Tsukamoto were carried out in the problem of prediction the risk of bankruptcy of enterprises in Ukraine. Studies have shown that the highest prediction accuracy of bankruptcy risk in relation to the economy of Ukraine give fuzzy neural networks.

## REFERENCES

[1] Altman E.I. CorporateFinancialDistress/ Altman E.I. -New York, JohnWiley, 1983. – 192 p

[2] Altman E.I. DiscriminantAnalysisandthepredictionofCorporateBankruptcy. / Altman E.I. //FinancialRatios. //JournalofFinance.-1968.-September.-pp. 589-609.

[3] Davydova G.V, Belikova.Yu. Methods of quantitative risk bankruptcy./Risk management, 1999.- № 3.- p. 13-20.

[4] ZaychenkoYu.P.Fuzzy models and Methods in Intelligent Systems. Kiev. Publishing House"Slovo", 2008.-352p. (rus).

[5] Ovi Nafas Aghaei Agh Ghamish, ZaychenkoYu.P.

[6] Financial analysis and forecasting of risk of bankruptcy of corporations in conditions of uncertainty. International Journal "Information Models and Analysis"/ Vol.1, №4.-2012.- pp. 336-348.

## AUTHOR'S PROFILE

Ovi Nafas Aghaei Agh Ghamish (Iran)- master of sciences., post-graduate student, National Polytechnic Institute 03056, Kiev-56, Ukraine. e-mail: ovinafas@yahoo.com field of scientific interests: artificial intelligence, fuzzy neural networks and their applications Author of 10 works

*Yuri Zaychenko – Professor, doctor of technical sciences, Institute for applied system analysis, NTUU "KPI", 03056, Kyiv, Peremogy pr. 37, Corpus 35; e-mail:baskervil@voliacable.com*

Major Fields of Scientific Research: Information systems, Fuzzy logic Decision-making theory, Computationalintelligence, author of 15 books and more than 100 scientific works.