Annals of the University of Craiova, Mathematics and Computer Science Series
Volume 46(2), 2019, Pages 372–380
ISSN: 1223-6934

Genetic algorithm/extreme learning machine paradigm for cancer detection

Mircea-Sebastian Serbanescu

Abstract. Biological systems inspire many machine learning systems. Two of these systems are genetic algorithms (GAs) and neural networks (NNs). A special type of NN is Extreme Learning Machine (ELM). ELM is a single hidden layer feedforward NN, in which the training step is done in just one step. Even if the hidden-output weights are analytically computed, studies have shown that the ELM still maintains the universal approximation capability. In the classical ELM the input-hidden weights are randomly generated. This paper deals with the evolutionary training of an ELM, by using a GA routine to set the input weights. The new hybrid has been applied on two real-world datasets concerning breast cancer detection. The results obtained show that the new technique is competitive to other state-of-the-art methods.

2010 Mathematics Subject Classification. Primary 60J05; Secondary 60J20.
Key words and phrases. neural networks, genetic algorithms, extreme learning machine, cancer detection.

1. Introduction

Since 1975, when Holland presented in his book a formal framework of genetic processes that can be used in different disciplines, there have been many efforts to combine evolutionary computation and neural networks (NNs). Mathematically speaking, the idea behind this merge is that we can use GAs to evolve NNs by setting up the weights, since finding the weights in a NN represent an optimization problem. There are several studies that have addressed this issue. For example, in [28] the authors trained a multi-layered feedforward NN using mutation and crossover operators in order to process passive sonar data from arrays of underwater acoustic receivers. The study, [36], emphasized mutation and trained a NN using GAs to resolve classic problems such as XOR, 424-encoder and the two-bit adder. In the case of binary classification problems, Pendharkar et al. compared the GA/NN with the log maximum-likelihood gradient ascent and the root-mean-square error [30]. For further references please see also [31], [35] and [27].

Classical and novel NNs have been used in medical diagnosis for the last three decades: in breast cancer detection and recurrence [4], [13], [5], [16], [25], cardiovascular diseases [29], colon cancer [32], Alzheimer's disease [8], liver fibrosis stadialization [14], [15], [3], [1], ovarian and lung cancer, [2], [6].

Received June 26, 2019. Accepted September 19, 2019.

The BC1 and BC2 databases were obtained from the University of Wisconsin Hospitals, Madison. Thanks go to Dr. William H. Wolberg, W. Nick Street and Olvi L. Mangasarian for providing the data.

Using medical imaging like magnetic resonance imaging, conventional imaging, and positron emission tomography one establishes breast cancer diagnosis. An accuracy that ranges between 80% and 90% has been reported in literature, [33], [12], [26]). NNs have been massively employed in cancer detection: Multi-layer perceptrons (MLP), radial-basis function (RBF), recurrent neural network, and support vector machine have been considered to statistically evaluate their potential contribution to predict recurrent events in breast cancer [23], [24]; a hybrid NN/ rough sets has been used in [17]; evolutionary programming and NNs detected breast cancer using radiographic features together with patient age, [11]. In [7] has been shown that NNs can predict the recurrence probability of breast cancer and also to determine which patients have good prognosis versus bad prognosis.

ELM represents a new NN algorithm that caused some discussions. Nevertheless, variants of the original algorithm have been proposed in literature and used in numerous medical applications, [37], [38], [34]. The paper is organized as follows: section 2 describes the design and implementation of the GA/ELM algorithm and section 3 presents the two databases on which the new hybrid has been applied on. In Section 4 we discuss the experimental results obtained and also the statistical comparison analysis to other three NNs models (classical ELM, MLP and RBF). The conclusions are presented in Section 5.

2. The GA/MLP model

2.1. ELM. Huang et al. introduced a new type of single hidden layer neural network, the ELM, [20], [21], [22]. The weights between the input and hidden layer are randomly initialized, whereas the weights between the hidden and output layer are computed analytically using the Moore-Penrose generalized inverse. The classical ELM algorithm is presented below:

ELM algorithm

Step 1: Randomly assign the input weights w_i.

Step 2: Compute the hidden layer output matrix M.

Step 3: Compute the output weight vector $h = M^{+}o$, where M is the hidden output matrix computed at step 2, M^{+} is the Moore-Penrose generalized inverse of M, and o is the output vector.

2.2. The GA model. GAs mimic the biological process of life, with all its stages from reproduction till death. A GA consists of: populations of chromosome, selection operator based on each chromosome's fitness, crossover operator in order to produce new offspring, and a mutation operator for the newly created offspring. In the initial run of the GA, we create a large population of random chromosomes. The chromosome's genes contain crucial information related to the potential candidates of the maximization or minimization problem. At each iteration the whole population of chromosomes is evaluated by a cost function in order to determine which are best chromosomes that need to be kept. After this step the crossover operator is used for mating of the chromosomes so that the population is replenished. Randomly the offspring resulted from the mating are mutated in order to get the search out of the local minima or maxima. The cost function evaluates again the new population, and so on the process is repeated until a given convergence criterion is met. The GA algorithm can be summarized as it follows:

GA algorithm
Step 1. Encode the data in a vector form and set the crossover and mutation probabilities.
Step 2. Chose the chromosome population.
Step 3. Compute the fitness function for each chromosome.
Step 4. Apply the selection, crossover and mutation operators.
Step 5. Replace the current population by the new one.
Step 6. Use the stopping criterion to stop the evolutionary process, [9], [18].

2.3. GA/ELM algorithm. In this subsection we shall present the GA/MLP hybrid model that we have designed and implemented for breast cancer detection. The model has two decision classes: A and B, A being malignant tumor and B being a benign tumor. Medical professionals made the initial classification. The new model contains two components:
a) a GA model that computes the best initial weights for the ELM,
b) an ELM.

The ELM consist of a number n of input units (the predictive attributes in each dataset), a hidden layer with 9 hidden units and an output layer with two units, one for each decision class A and B. Taking into account that the model works only a two-class decision problem we have transformed the corresponding class label y_j using the "$one - hot - encoding$" rule for categorical data: y_1
/ $(0, 1)$ and y_2 $(1, 0)$. The output was computrd using the winner-takes-all rule, which implies that the highest activation unit gives the class. The hyperbolic tangent, $f(u) = 1.7159 \cdot tanh(2u/3)$, was chosen as non-linear activation function for each hidden neuron, instead of the sigmoid function, due to the fact the hyperbolic tangent's output is not zero-centered. The weight vector is represented through a chromosome that has a number of genes that equals the number of neurons from the input layer multiplied by the number of neurons from the hidden layer. We chose an initial random population of 100 weight vectors - chromosomes, each weight being between -1 and 1. The cost function was computed as the classification accuracy obtained after the run of the algorithm on the training dataset. High accuracy implies high fitness. At each iteration we evaluate all the 100 existing chromosomes, and only the best 40 of them are kept for reproduction and mutation. By using the crossover operator we obtain 40 new offspring. As crossover operators we have used:

- Total (whole) arithmetic recombination operator (AX): two offspring are produced using the following formulas:

$$offspring_i^1 = \alpha \cdot chromosome_i^1 + (1 - \alpha) \cdot chromosome_i^2$$
$$offspring_i^2 = \alpha \cdot chromosome_i^2 + (1 - \alpha) \cdot chromosome_i^1$$

- Blend crossover ($BLX - \alpha$): two offspring are (uniformly) randomly generated from the interval $[chromosome_{min} - I, chromosome_{max} + I]$, where:

$$chromosome_{\min} = \min\{chromosome_i^1, chromosome_i^2\},$$
$$chromosome_{\max} = \max\{chromosome_i^1, chromosome_i^2\},$$
$$I_\alpha = chromosome_{max} - chromosome_{min}.$$

- Wrights heuristic crossover (WHX): assuming that $chromosome_i^1$s fitness is better than $chromosome_i^2$s fitness, two offspring are produced using the following formula:
  $offspring_i^k = u \cdot (chromosome_i^1 - chromosome_i^2) + chromosome_i^1$, where k = 1, 2.
- Linear BGA crossover (BGAX): assuming that $chromosome_i^1$s fitness is better than $chromosome_i^2$s fitness, then two offspring are produced using the following formula:

$$offspring_i^k = chromosome_i^1 \pm r_i \cdot \gamma \cdot \Lambda,$$

  where all the chromosomes are generated between the interval $[a_i, b_i]$, $\gamma = \sum_{k=0}^{no\_genes} \alpha_k \cdot 2^{-k}$, and $\Lambda = \frac{(chromosome_i^2 - chromosome_i^1)}{||chromosome_i^2 - chromosome_i^1||}$
- Uniform crossover (UX), working as follows. Two offsprings are created following this scheme: the value of each gene in the offspring is determined by the random choice of the values of this gene in the parents:
  $h_i^k = \begin{cases} chromosome_i^1, u = 0 \\ chromosome_i^2, u = 1 \end{cases}$.

For further details regarding these crossover operators please see [19]. The mutation probability was considered 0.3. Practically, for each chromosome's gene we have generated a random number between 0 and 1. If the number was smaller than the mutation probability then the gene was mutated. The mutation process consists of two steps: 1) generate a number between 0 and 1, if the number is smaller than 0.5 then we make a subtraction, otherwise we make an addition. Using the chromosome error, which is problem dependent [5] we compute the new value of the gene. The formula for the chromosome's error is presented below:
$chromosome\_error = \frac{(100 - chromosome\_accuracy)}{100}$

3. Materials. The data.

The GA/ELM algorithm has been applied on two publicly available breast cancer datasets. In what follows we shall briefly present the data.
The Wisconsin Prognostic Breast Cancer -BC1 dataset contains of 683 cases with two decision classes: benign 444 (65%) instances and malign 239 (35%) instances. The database has nine ordinal (categorical) attributes (UCI Machine Learning repository: `http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/`)).
The Wisconsin Prognostic Breast Cancer -BC2, contains of 569 cases, with two decision classes: benign 357 (62.74%) instances and malign 212 (37.25%) instances. The database has thirty numerical attributes.

We have chosen these two datasets to see how the novel algorithm performs on different types of data: categorical and numerical. Since both ELM and GA have a stochastic nature, we used as testing method the 10-fold cross-validation. Thus, both the training and testing accuracies have been computed 10 times, each time leaving out one of the sub-samples to be used as the test samples for the cross-validation process. Both the new model as well as the standard NN models have been run 100 times, and then averaged to give the 10-fold estimate of the classification accuracy, since we need to be sure that the algorithm cross-validates well.

Table 1. Comparison of classification performances depending on crossover operator.

	Crossover	BC1	BC2
Average/SD training performance (%)	AX	92.21 / 3.07	95.47 / 3.02
	BLX	91.38 / 7.42	91/05 / 4.58
	WHX	89.72 / 1.14	91.76 / 2.10
	BGAX	79.48 / 2.18	90.77 / 1.01
	UX	92.11 / 1.20	93.78 / 2.30
Average/SD testing performance (%)	AX	91.63 / 3.13	92.92 / 3.49
	BLX	91.18/ 7.91	92.92 / 3.49
	WHX	87.66 / 4.17	91.16 / 3.74
	BGAX	70.87 / 2.60	91.23 / 1.92
	UX	90.09 / 2.53	82.80 / 2.66

Table 2. Confusion matrix BC1 database.

	Predicted classes	
Observed classes	Malign (+)	Benign (-)
Malign (+)	213	21
Benign (-)	35	414

Table 3. Confusion matrix BC2 database.

	Predicted classes	
Observed classes	Malign (+)	Benign (-)
Malign (+)	84	15
Benign (-)	84	273

4. Results

Considering the fact that we have used multiple crossover operators, the first natural step was to compare the performances of the hybrid algorithm when using each type of operator. Accordingly, we have run the model 106 times with each operator on each of the two datasets. The results of this comparison in terms of mean and standard deviation (SD) averaged over the 106 computer runs are presented in table 1. Due to the fact that we have a large number of runs, we can assume that the distribution of data is nearly Gaussian.

From table 1 we can depict that:

- On BC1 dataset, the AX, BLX and UX operators have the same training performance approximately 92%, while on the BC2 dataset, AX performs better.
- There is no significant difference between the training and testing performances, thus we can draw the conclusion that the hybrid cross-validates well.
- The standard deviation values are small, thus the hybrid also has a balanced behavior.

After this brief statistical analysis we have chosen for this study the AX crossover operator. Please keep in mind that the classification accuracy of the GA/ELM hybrid is in accordance with classical medical imaging techniques. Using the AX crossover we have further deepened the analysis and computed four important classification parameters: sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV). The confusion matrices results are displayed in Table 2 and 3.

Table 4. Sensitivity/specificity and PPV/NPV parameters.

	WPBC1	WPBC2
Sensitivity	85.89	70.11
Specificity	95.17	94.79
PPV	92.20	92.92
NPV	91.80	76.47

Table 5. Models classification performances (averaged over 106 computer runs).

	Average testing performance (%)	
Classifiers	BC1	BC2
GA/ELM	91.63	92.92
MLP	95.52	81.39
RBF	89.36	87.42
ELM	90.21	91.80

The corresponding sensitivity/specificity, PPV and NPV, obtained in a complete cross-validation cycle, are displayed in Table 4. Recall that the sensitivity measures the proportion of true positives that are identified as such, whereas specificity measures the proportion of false negatives that are identified as such. PPV is the proportion of cases with positive test results that are labeled correctly, whereas NPV is the proportion of cases with negative test results that labeled correctly.

We can see from table 4 that the sensitivity ranges from 70% to 85% and the specificity form 94% to 95%, in contrast with the corresponding sensitivity. All the values are in accordance with the medical imagining sensitivity and specificity reported in literature [12], [10], [26].

We have continued our statistical analysis with the comparison of the model's performance to the performances of three classic NNs. The benchmarking rule was the same for all the comparisons, 106 different computer runs in a complete cross-validation cycle for each algorithm. The results are presented in table 5.

From the above table we can depict that:

- the novel model outperforms the other classifiers on the BC2 dataset, and on the BC1 dataset is surpassed only by the MLP,
- the GA/ELM has a better accuracy than the parent ELM,
- RBF performs the poorest on both datasets.

We have seen how the new model performs compared to other ML algorithms in terms of accuracy (table 5). Next, we shall statistically assess the testing performances of the four classifiers used for comparison. We have used the Shapiro-Wilk's W test to verify the normality of each data set. Unfortunately, the normality was not met for any data set (the W statistics ranged form 0.56 to 0.94, having the corresponding p-level = 0.000). The Levene's test was used for testing the variances, and the results were once again unsatisfactory, the F statistics value ranging form 287.40 to 301.56, having the corresponding p-level = 0.000. Due to the results obtained after these two tests, we decided that we cannot perform the t-test for independent variables, and applied the Mann-Whitney U test instead. Table 6 presents the results obtained by the Mann-Whitney U test:

As we can see in table 6, there are significant differences in means between all models, which imply that each one of the algorithms has a unique way in handling the dataset. We have also tested the differences between two population proportions using the z-test statistic.

Table 6. Comparing testing performances (Mann-Whitney U test).

Database	GA/ELM vs. MLP (z-value / p-level)	GA/ELM vs. RBF (z-value / p-level)	GA/ELM vs. ELM (z-value / p-level)
BC1	12.31 / 0.000	12.24 / 0.000	12.29 / 0.000
BC2	12.21 / 0.000	12.24 / 0.000	12.32 / 0.000

Table 7. Comparing testing performances (two-sided z-test).

Database	GA/ELM vs. MLP	GA/ELM vs. RBF	GA/ELM vs. ELM
BC1	0.239	0.612	0.006
BC2	0.009	0.129	0.008

From table 7 we see that there is no significant difference between the average proportions of correctly classified cases for the new model and the all the other ML algorithms, no matter of which dataset has been used.

As an overall conclusion we can state that the novel algorithm can compete with classical NN methods, and in some cases even outperformed them. For sure, this assumption does not imply that the GA/ELM will surpass the other methods in any case, the results being dependent on the dataset used.

5. Conclusions

In the last years, artificial intelligent methods have become more and more important in medical diagnosis, due to their high speed and low cost. In this paper we have studied the effectiveness of a novel model that has a GA component for finding the weights between the input and hidden layer, and borrows the analytical one step training from the classical ELM. The model had been tested on two real world breast cancer datasets. To assess the models performance we have compared it to other three classical NNs. The model outperformed in some cases the other algorithms, and in some cases had a similar performance with the best of them. Future work will include the use of the other crossover operators and other activation functions. We would also like to extend the study to multiple decision classes.

References

[1] S. Belciug, F. Gorunescu, Error-correction learning for artifcial neural networks using the Bayesian paradigm. Application to automated medical diagnosis, *Journal of Biomedical Informatics* **52** (2014), 329–337.

[2] S. Belciug, F. Gorunescu, Learning a single-hidden layer feedforward neural network using rank correlation-based strategy with application to high dimenional gene expression and proteomic spectra datasets in cancer detection, *Journal of Biomedical Informatics* **83** (2018), 159–166.

[3] S. Belciug, M. Lupsor, R. Badea, Feature selection approach for non-invasive evaluation of liver fibrosis, *Annals of the University of Craiova, Mathematics and Computer Science Series* **35** (2018), 15–20.

[4] S. Belciug, F. Gorunescu, M. Gorunescu and A. Salem. Assessing performances of unsupervised and supervised neural networks in breast cancer detection, *2010 The 7th International Conference on Informatics and Systems (INFOS)*, Cairo, Egypt, March 28-30, 2010, IEEE Press, (2010), 1-8.

[5] S. Belciug and F. Gorunescu, A hybrid neural network/genetic algorithm system applied to the breast cancer detection and reccurence, *Expert Systems* **30** (2013), no. 3 , 243–254.

[6] R. Berglund and S. Belciug, Improving Extreme Learning Machine Performance using Ant Colony Optimization Feature Selection. Application to automated medical diagnosis, *Annals of*

the University of Craiova, Mathematics and Computer Science, Science Series **45** (2018), no. 1, 151–155.

[7] C.L. Chi, W.N. Street and W.H. Wolberg, Application of artificial neural network-based survival analysis on two breast cancer datasets, *AMIA Annu. Symp. Proc.* **11** (2007), 130-134.

[8] F. Copped, E. Grossi, M. Buscema and L. Migliore, Application of artificial neural networks to investigate one-carbon metabolism in Alzheimer's disease and healthy matched individuals, *PLoS One* **8** (2013), no. 8, e74012.

[9] A.E. Eien and J.E. Smith, *Introduction to Evolutionary Computing*, Springer-Verlag, 2003.

[10] H. Elsamaloty, M.S. Elzawawi, S. Mohhamad and N. Herial, Increasing accuracy of detection of breast cancer with 3-T MRI, *Am. J. Roentgenol.* **192** (2009) , no. 4, 1142–1148.

[11] D.B. Fogel, E.C. Wasson, E.M. Boughton and V.W. Porto, A step toward computer-assisted mammography using evolutionary programming and neural networks, *Cancer Lett.* **119** (1997), no. 1, 93–97.

[12] F.J. Gilbert, S.M. Astley, M.G.C. Gillan, O.F. Agbaje, M.G. Wallis, J. James, C.R.M. Boggis and S.W. Duffy, Single Reading with Computer-Aided Detection for Screening Mammography, *N. Engl. J. Med.*, **359** (2008), no. 16, 1675–1684.

[13] F. Gorunescu and S. Belciug, Boosting backpropagation algorithm by stimulus-sampling: Application in computer-aided medical diagnosis, *Journal of Biomedical Informatics* **63** (2016), 74–81.

[14] F. Gorunescu, S. Belciug, M. Gorunescu and R. Badea, Intelligent decision-making for liver fibrosis stadialization based on tandem feature selection and evolutionary-driven neural network, *Expert Syst. Appl.* **39** (2012), 12824–12833.

[15] F. Gorunescu and S. Belciug, Evolutionary strategy to develop learning-based decision systems. Application to breast cancer and liver fbrosis stadialization, *J. Biomed. Inform.* **49** (2014), 112–118.

[16] F. Gorunescu, M. Gorunescu, E. El-Darzi and S. Gorunescu, A statistical framework for evaluation neural networks to predict reccurent events in breast cancer, *International Journal of General Systems* **39** (2010), no. 5, 471–488.

[17] M. Gorunescu, F. Gorunescu and K. Revett, Investigating a Breast Cancer Dataset Using a Combined Approach: Probabilistic Neural Networks and Rough Sets, *Proceedings of the 3rd ACM International Conference on Intelligent Computing and Information Systems - ICICIS07*, 2007, Police Press, 246-249.

[18] R.L. Haupt and S.E. Haupt, *Practical Genetic Algorithms (Second Edition)*, John Wiley & Sons, 2004.

[19] F. Herrera, M. Lazano and A.M. Sanchez, A taxonomy for the Crossover Operator for Real-Coded Genetic Algorithms: An Experimental Study, *International Journal of Intelligent Systems* **18** (2003), 309-338.

[20] G.B. Huang, Q.C. Zhu and C.K. Chee-Kheong Siew, Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks, *Proc. Intl. Joint Conf. Neur. Net.*, 2006, 985-990.

[21] G.B. Huang, X. Ding and H. Zhou, Optimization Method Based Extreme Learning Machine for Classification, *Neurocomp.* **74** (2010), 155–163.

[22] G.B. Huang, H. Zhou, X. Ding and R. Zhang, Extreme Learning Machine for Regression and Multiclass Classification, *IEEE Trans. Systems, Man, and Cybernetics* **42** (2012), no. 2, 513–529.

[23] E.D. Ibeyli, Implementing automated diagnostic systems for breast cancer detection, *Expert Systems with Applications: An International Journal* **33** (2007), no. 4, 1054–1062.

[24] E.D. Ibeyli, Comparison of different classification algorithms in clinical decision-making, *Expert Systems* **24** (2007), no. 1, 17–31.

[25] A.A. Kalteh, P. Zarbakhsh, M. Jirabadi and J. Addeh, A research about breast cancer detection using different neural networks and K-MICA algorithm, *J. Cancer Res. Ther.* **9** (2013), no. 3, 456–466.

[26] J.H. Lee, E.L. Rosen and D.A. Mankoff, The Role of Radiotracer Imaging in the Diagnosis and Management of Patients with Breast Cancer: Part 1Overview, Detection, and Staging, *Journal of Nuclear Medicine* **50** (2009), no. 4, 569–581.

[27] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1998.

[28] D. Montana and L. Davis, Training feedforward networksusing genetic algorithms, *Proceedings of the 11th International Joint Conference on AI* , Detroit, MI, August 20-25, 1989, 762-767.

[29] K. Nakajima, S. Matsuo, H. Wakabayashi, K. Yokoyama, H. Bunko, K. Okuda, S. Kinuya, K. Nystrm and L. Edenbrandt, Diagnostic Performance of Artificial Neural Network for Detecting Ischemia in Myocardial Perfusion Imaging, *Circ. J.* **79** (2015), no. 7, 1549–1556.

[30] P.C. Pendharkar, A comparison of gradient ascent, gradientdescent and genetic-algorithm-based artificial neural networksfor the binary classification problem, *Expert Systems* **24** (2007), 65–86.

[31] J.D. Schaffer, D. Whitley and L. Eshelman, *Combination of genetic algorithms and neural networks: the state of the art, Combination of Genetic Algorithms and Neural Networks*, IEEE Computer Society, 1992.

[32] L. Spelt, J. Nilsson, R. Andersson and B. Andersson. Artificial neural networks - a method for prediction of survival following liver resection for colorectal cancer metastases, *Eur. J. Surg. Oncol.* **39** (2013), no. 6, 648–654.

[33] D. Vranjesevic, J.E. Filmont, J. Meta, D.H. Silverman, M.E. Phelps, J. Rao, P.E. Valk and J. Czernin, Whole-Body 18F-FDG PET and Conventional Imaging for Predicting Outcome in Previously Treated Breast Cancer Patients, *Journal of Nuclear Medicine* **43** (2002), no. 3, 325–329.

[34] G.-G. Wang, M. Lu, T.-Q. Dong and X.-J. Zhao. Self-adaptive extreme learning machine, *Neural Computing and Applications* **27** (2016), no. 2, 291–303.

[35] D. Whitley, Genetic Algorithms in Engineering and Compute Science. In *J. Periaux and G. Winter (eds) The rise of modern science*, 3rd edition, John Wiley & Sons Ltd, UK (1995), 230–257.

[36] D. Whitley and T. Hanson, Optimizing neural networks using faster more accurate genetic search, *Proceedings of the 3rd International Conference on Genetic Algorithms*, San Francisco, USA, 1989 (J.D. Schaffer, Eds.), Morgan Kaufmann, 391-396.

[37] L. Zhang and D. Zhang, . Evolutionary Cost-Sensitive Extreme Learning Machine, *IEEE Transactions on Neural Networks and Learning Systems* **PP** (2016), no. 99, 1–16.

[38] T.-P. Zhao, Parsimonious kernel extreme learning machine in primal via Cholesky factorization, *Neural Networks* **80** (2016), no. C, 95–109.

(Mircea-Sebastian Serbanescu) Department of Medical Informatics and Biostatistics, University of Medicine and Pharmacy of Craiova, 2-4 Petru Rares Street, Craiova, 200349, Romania

E-mail address: mircea_serbanescu@yahoo.com