8. A Neuro-Classification Model for Socio-Technical Systems

Iulian NĂSTAC[*]
Angelica BACIVAROV[**]
Adrian COSTEA[***]

Abstract

This paper presents an original classifier model based on an artificial neural network (ANN) architecture that is able to learn a specific human behavior and can be used in different socio-economic systems. After a training process, the system can identify and classify a human subject using a list of parameters. The model can be further used to analyze and build a safe socio-technical system (STS). A new technique is applied to find an optimal architecture of the neural network. The system shows a good accuracy of the classifications even for a relatively small amount of training data. Starting from a previous result on adaptive forecasting, the model is enhanced by using the retraining technique for an enlarged data set.

Keywords: artificial neural network, training process, classification, socio-technical system

JEL Classification: A13, C45, C89, C90, Z13

1. Introduction

The performance classification problem concerns many active players: from producers to consumers, from educators to students, from investors to decision makers, from creditors to auditors, they are all interested to find where one institution is situated against its competitors, what are its strengths and weaknesses, and how the decision process can be influenced so that poor performance is avoided. Usually, the classification problem literature emphasizes binary classification, also known as two-group discriminant analysis problem, which is a simpler case of the classification problem. In the case of binary classifications everything is seen in black and white (e.g., a model which implements a binary classifier would just show a bankruptcy or a non-bankruptcy situation giving no detailed information about the real problems of the

[*] *Associate Professor, Polytechnic University of Bucharest, nastac@ieee.org (corresponding author), e-mail: idnastac@yahoo.com.*
[**] *Professor, Polytechnic University of Bucharest.*
[***] *Associate Professor, Bucharest Academy of Economic Studies.*

companies). Greater information would be obtained out of the classification models if one particular sector would be divided into more than two performance classes (and it would be easier to analyze the companies/ institutions placed in these classes).

The question of performance classification has been tackled in the literature for nearly 40 years. A useful taxonomy of classification models is presented by Pendharkar (2002). First, *statistical techniques* have been deployed on different systems: univariate statistics for prediction of failures (introduced by Beaver (1966)), multivariate analysis in Altman (1968), linear discriminant analysis (LDA) introduced by Fisher (1936) who first applied it to Anderson's iris data set (Anderson, 1935), multivariate discriminant analysis (MDA – Edmister, 1972); Jones, 1987), and probit and logit models – (Hamer, 1983; Zavgren, 1985; Rudolpher *et al*., 1999). The next step in solving the classification problem was the establishment of the *induction techniques*. Some of the most popular techniques of this kind are: recursive partitioning algorithm (RPA) in Frydman *et al*., (1985), CART (Breiman *et al*., 1984), and ID3-C4.5-C5.0 (Quinlan, 1993a; Quinlan, 1993b). The SOM classification (Kohonen, 1997) is based on the self-organizing map (SOM) model, which was introduced by Kohonen in early '80s and is an unsupervised learning technique that creates a two-dimensional topological map from *n*-dimensional input data. This topological map preserves the neighborhood relations, and therefore similar input vectors have close positions on the map. A series of complex applications such as financial benchmarking (Eklund *et al*., 2003) and socio-economic development (Collan *et al*., 2007) use SOM as a visualization tool.

Many authors (Davis, 1991; Michalewicz, 1992; Fogel *et al*., 1995, 1998; Bhattacharyya and Pendharkar, 1998; Pendharkar and Rodger, 2004; etc.) studied the applications of artificial intelligence models (neural networks, genetic algorithms, machine learning, hybrid solutions, etc.) in classification problems.

Zupan *et al*. (1998) proposed a classification technique (HINT) which is based on function decomposition for the transformation of input feature space. The idea is to separate the input space in two less complex disjoint feature spaces that recombined yield the original input feature space. The original input feature space can be reduced if one of the two disjoint feature spaces has redundant features. Zupan *et al*. (1998) used as case studies two well-known machine-learning problems (monk1, monk2) and a housing-loan allocation problem. They compared their system (HINT) with Quinlan's C4.5 decision tree algorithm in terms of prediction accuracy, finding out that for all the above problems the system based on function decomposition yielded significantly better results.

Classification data mining has two different aims: the uncovering of hidden relationships and patterns in the data and the construction of usable models (Zupan *et al*., 2001). One type of these models is represented by classification models or models for predicting the relative positions of newly observed cases against what is already known (Costea and Năstac, 2005).

In this paper, we will concentrate on STS. Socio-technical systems are systems that include technical systems but also operational processes and people who use and interact with the technical system. STS are governed by organizational policies and rules. More specifically, we present an original classifier model based on artificial neural network (ANN) architecture that is able to learn a specific human behavior. It is well known that the human behavior can have a great influence over financial decisions (Becker, 1976; Shiller, 1999; Sanfey, 2003). This kind of classification that we describe in this paper can be easily used in combination with a model of a financial predictive system (see section 4, Experimental Results) presented in the paper of Năstac, Dobrescu, and Pelinescu (2007). A socio-technical approach is considered in order to obtain the data set for an ANN model. The aim is to design an accurate classifier by using two strategies. The first aims to rebuild the ANN architecture according to a new data set, while the second one uses an adaptive forecasting procedure.

2. Artificial Neural Network Classifiers

This study introduces Artificial Neural Networks (ANN) to help solving the multi-class classification problems. Moreover, the study investigates a new empirical method to determine the proper ANN architecture.

The main advantages of neural approaches for classification over the traditional ones are: ANNs are free of any distributional assumptions, can deal with inter-correlated data, and they provide a mapping function from the input to the outputs without any *a priori* knowledge about the function form since they are universal approximators (Hornik, Stinchcombe, and White, 1989). A comprehensive study on ANN for failure prediction is provided by O'Leary (1998). Here the author investigates fifteen related papers for a number of characteristics: what data were used, what types of ANN models, what software, what kind of network architecture, etc. Costea and Năstac (2005) analyze the implications of three different factors (preprocessing method, data distribution and training mechanism) on the classification performance of neural networks.

Regarding ANN learning aspect, various comparative studies, on different problems, were initiated in literature in order to establish the optimal training algorithm (Demuth and Beale, 2001). As a general conclusion, it is difficult to know which training algorithm will provide the best (fastest) result for a given problem. A smart choice depends on many parameters of the ANN involved, the data set, the error goal, and whether the network is used for pattern recognition (classification) or function approximation. Statistically speaking, it seems that numerical optimization techniques present numerous advantages. Analyzing the algorithms that fall into this class, we observed that the Scaled Conjugate Gradient (SCG) algorithm (Moller, 1993) performs well over a wide variety of problems, including the one presented in this paper. Even if SCG is not the fastest algorithm (as Levenberg-Marquardt in some situations), the great advantage is that this technique works very efficiently for networks with a large number of weights. The SCG is something of a compromise; it does not require large computational memory, and yet it still has a good convergence and is very robust. Furthermore, we always apply the early stopping method (***validation stop***) during the training process, in order to avoid the over-fitting phenomenon. And it is well known that for early stopping, one must be careful not to use an algorithm that converges too rapidly. The SCG is well suited for the validation stop method.

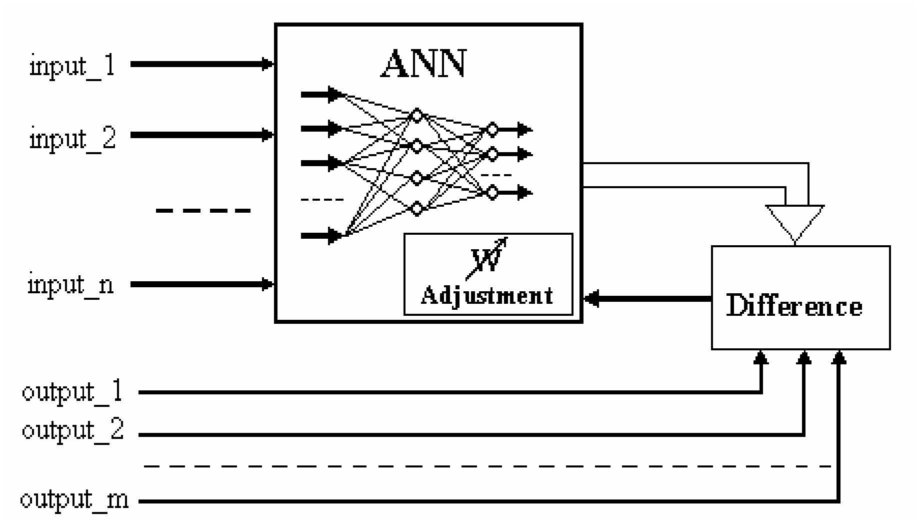
3. ANN Model

The structural design of the ANN is a crucial factor that determines its performance (Hagan *et. al*., 1996). A suitable choice for the global architecture of the model is not a trivial task, if one wants to make a good classification.

We start to consider a system with ***n*** inputs and ***m*** outputs. In Figure 1 we present our general idea of training a feed-forward ANN such that the latter becomes a classifier.

Figure 1

ANN Model for classification. Training Process



Each ANN target-output corresponds to the one of ***m*** classes, and it is activated by the proper values of the corresponding inputs that are associated to its class. Therefore, the system tries to match the current values of the output, by properly adjusting a function of the inputs (Figure 1).

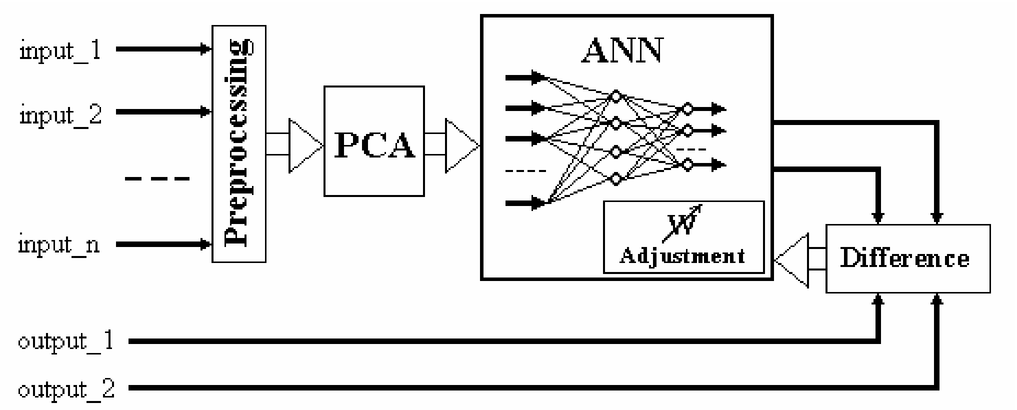
The goal of our research was to find a mathematical model describing the relationship between 36 input variables and two outputs that model a classifier, which split the input data into two classes. The inputs were provided from a complex questionnaire that was filled out by a various number of students or graduates. The first class consists of the persons with a background up to the second year university level, while the second class includes people with stronger university background (more than two university years or graduates). Each input has no direct connection with the university background and reveals the opinion of the subject concerning various areas. By analyzing one question out of 36 it is nearly impossible to establish the education level of the subject. Moreover, the separation point between these two classes is somehow fuzzy since it is hard to make a clear difference among the students' way of thinking. The classifier must be robust enough and able to identify the difference between a student from the second university year versus one from the third year. The model can be further extended to other similar applications that concern the human behavior (or human level of knowledge) in specific applications.

We apply Principal Component Analysis (PCA) (Jackson 1991) to reduce the dimensionality of the input space and to un-correlate the inputs. Before applying PCA, we preprocessed the inputs by applying the normalization. Data preprocessing prepares the raw data for the forecasting model and turns it into a format that will be easier and more effectively processed. Data preprocessing is an essential step of the knowledge discovery process in the real world applications, and it greatly improves the network's ability to capture valuable information, if it is correctly carried out (Hagan *et al.*, 1996, Basheer *et al.*, 2000).

Therefore, by taking into consideration these assumptions concerning the specific application of the classifier, the system becomes more complex (Figure 2).

Figure 2

The model of a socio-technical system classifier



In Figure 2 we have only two outputs that correspond to the previously mentioned classes. We use feed forward ANNs with two hidden layers in order to achieve a good classification function, based on our preliminary research, where we have obtained better results in case of two hidden layers than in case of one, however maintaining a similar ratio (approx. 2/1, and greater than 1/1) between the number of the training samples and the total number of the weights.

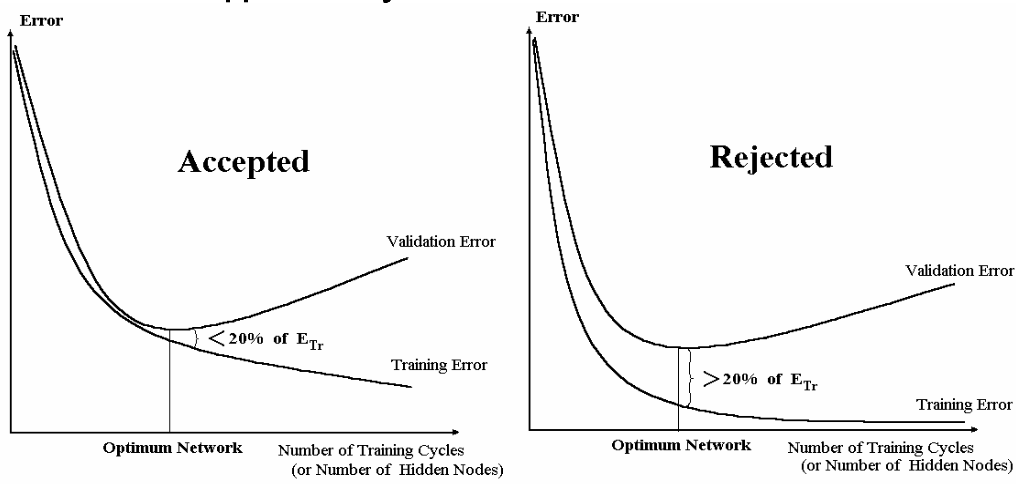
As learning technique we employed the Scale Conjugate Gradient (SCG) algorithm (Moller, 1993) for the ANN training purpose. In order to avoid the overfitting phenomenon, we apply the early stopping method (validation stop) during the training process. As a splitting criterion, we randomly choose approximately 85% of the data (*V* samples) for training set, and the rest for validation. Furthermore, we imposed the supplementary condition:

$$E_{val} \leq \frac{6}{5} \cdot E_{tr} \tag{1}$$

to avoid a large difference (see Figure 3) between the error of the training set (E_{tr}) and the error of the validation set (E_{val}). This way, the overfitting phenomenon on the test set will be considerably reduced. In our approach the validation set acts at the same time as a kind of test set, even though there is a real and separate test set of *T* different input-output pairs (where T<<V).

Figure 3

A supplementary condition over the validation set



We search for the proper number of hidden neurons for each hidden layer (N_{h1} and N_{h2}). Each of the training sessions started with the weights initialized for small, uniformly distributed values (Hagan *et al.*, 1996, Năstac and Matei, 2003). We tested several ANN architectures, with N_{h1} and N_{h2}, taking values somehow in the vicinity of the geometric mean (Basheer and Hajmeer, 2000) of the neighboring layers, and trying to observe the following rules:

$$\sqrt{N_i \cdot N_{h2}} + q \geq N_{h1} \geq \sqrt{N_i \cdot N_{h2}} - q \qquad (2)$$

$$\sqrt{N_{h1} \cdot N_o} + q \geq N_{h2} \geq \sqrt{N_{h1} \cdot N_o} - q \qquad (3)$$

Above, N_i is the number of inputs after PCA block, N_o is the number of outputs, and q is a small positive integer (ex. q=3). In fact we did not follow exactly the previous two relations and independently vary the number of neurons, from each hidden layer, between 8 and 11, in order to avoid a large number of cases and, consequently, to reduce the time and computational effort. Each architecture was tested five times, with random initial settings of the weights and different training/validation sets. We chose the best model with respect to the smallest error between the desired and the simulated outputs. This error (E_{tot}) was calculated for *V* data that included both training and validation sets. By using this way we selected the optimum network from a total of 4×4×5=80 different instances.

4. Experimental Results

In the first experiment, the number of samples used for the training process was V=147 (that includes both effective training and validation samples). As we previously mentioned, we used in our model 36 input variables and only two outputs, which split the input data into two classes. The inputs had only integer values in the range of 0 to 10, while the outputs took one of the following two cases: (0, 1) or (1, 0). For instance, one of the questions ask for the relevance of some specific factors (salary, work conditions, work schedule, relevance of the activity, promotion opportunity) that can concern a person who filled the questionnaire, by assuming the idea (or simulating) that is looking for a new job. Each factor implies a separate input in the model under condition that the sum of these factors, belong to a particular question, have to be 10.

A number of T=13 samples were used for the test set. These data were selected using a random process from a total set of 160 samples (that includes 68 samples for the class 1 and 92 for the class 2). During the network selection process the first hidden layer (N_{h1}) varied from 10 to 15 neurons and the second one (N_{h2}) from 2 to 5 neurons. Finally, we chose a network with N_{h1}=13 and N_{h2}=3. The PCA transformation matrix reduces the input spaces from 36 to 25 dimensions by retaining only those components which contribute more than a specified fraction of 0.001 (this way we lose only 0.1% of initial information and un-correlate the inputs). After the training process, the success rate of the test set was P=0.769.

The second experiment has an increased volume of data. The number of samples used for the training process was V=164, while the number of T=16 samples were used for the test set. These data were selected using a random process from a total set of 180 samples (that includes 68 samples for class 1 and 112 for class 2). After the ANN selection process a network with N_{h1}=11 and N_{h2}=2 was chosen. The PCA transformation matrix reduces the input spaces from 36 to 25 dimensions as in the previous experiment. After the training process, the success rate for the test set was P=0.75.

We can analyze these results in Table 1. In both experiments, we find good classifiers on the test set, even if the amounts of data were quite poor. The data used in the second experiments consist of the entire data from the first one plus twenty new samples that belong to the class 2. The results of the model will be improved when the raw data will be larger than the ones used in this preliminary approach. With more data during the training process, the model will be able to capture more details (patterns) that separate these two classes.

Table 1

Experimental results

Experiment	The total number of samples	The distribution of samples		ANN architecture	Success rate on test set P
		Class 1	Class 2		
Experiment 1	160	68	92	25 : 13 : 3 : 2	0.769
Experiment 2	180	68	112	25 : 11 : 2 : 2	0.75
Experiment 2r (starting from Experiment 1 with retraining)	180	68	112	25 : 13 : 3 : 2	0.783

Even if the total number of samples is greater in the second experiment we notice a better success rate in the first experiment. A good explanation for this result is that we have a more balanced distribution of samples for the first experiment (see Table 1). Both experiments (1 and 2) started from scratch by searching the best suited architecture following the way that we previously described at the beginning of this section. This is the reason why the ANN architecture slightly differs for these two experiments.

As an alternative strategy, we enhanced the result of the second experiment by using a specific retraining technique described in the paper of Năstac, Dobrescu and Pelinescu (2007) for a financial application. We redesigned experiment 2 in a different manner and we called it Experiment 2r (where **r** comes from *retraining*). This time we did not start the second experiment from scratch and we reused the architecture of the ANN that we obtained with the first experiment. We adapt the network to the new set of data by using the previous information in a special way described in the previous paper. The advantage of the retraining procedure is that some relevant aspects are preserved ("remembered") from the previous training phase. We notice a better result (success rate on test set) in the Experiment 2**r**.

The classification accuracy of an ANN system based on the described model can be continuously improved, at times, in a practical application by using the adaptive retraining procedure (Năstac, 2004; Năstac and Matei, 2003; Năstac and Cristea, 2005, Năstac, Dobrescu and Pelinescu, 2007), even for a non-stationary environment, which implies that the distribution of the data changes over time. This procedure allows: to establish an optimum ANN architecture, and, then, to refine its performance by using a kind of "remembering" (scaling) process without structural changes. There is a mechanism for extracting practical information directly from the weights of a reference ANN that was already trained in a preliminary phase. The retraining procedure reduces the reference network weights (and biases) by a *scaling factor* γ, $0<\gamma<1$. The reduced weights are further used as the initial weights of a new training sequence, with the expectation of a better accuracy.

Conclusions

Our preliminary results look very promising. In both experiments we find good classifiers for the test set, even if the amounts of data were quite poor. The results of the model will be improved when the raw data will be larger than the ones used in this preliminary approach. With more data during the training process, the model will be able to capture more details (patterns) that separate these two classes. We show that this kind of classification that we present here can be easily used in combination with a financial predictive system presented in the paper of Năstac, Dobrescu, and Pelinescu (2007). Starting from a previous result on adaptive forecasting, the model is enhanced by using the retraining technique for an enlarged data set. At this point the research can be further continued by including in a financial forecasting model a new parameter that results from a human behavior classification.

In the case of binary classifications everything is seen in black and white (e. g., a model which implements a binary classifier would just show a bankruptcy or a non-bankruptcy situation giving no detailed information about the real problems of the companies). Greater information would be obtained out of the classification models if one particular sector would be divided into more than two performance classes (and it would be easier to analyze the companies/institutions placed in these classes). The model can be extended to other similar applications that concern the human behavior (or human level of knowledge) in specific applications of socio-technical systems or financial domain. Some of these systems (banks, nuclear power plants, chemical plants, etc.) involve a high level of security, which also depend on their personnel's level of knowledge.

Current research targets the implementation of an adaptive system, which will be periodically retrained, in order to continuously learn the latest evolution of the real classification process. In Dobrescu (2006) it is mentioned that the forecast information can be useful for macroeconomic policies. It is easy to advance that some macroeconomic policies (or classes of policies) can somehow be predicted using a set of data (specific statistic questionnaires) and then, additionally include them in a feedback process in order to forecast other financial/economic parameters.

Acknowledgements

The CNCSIS Research Project No 630/2007 (Resilient Socio-Technical Systems) partially supported this work. The authors would like to thank to Gabriel Petrica for his effort to build the initial data base and implement the online internet questionnaire.

References

Basheer, I.A. and Hajmeer, M. (2000), "Artificial neural networks: fundamentals, computing, design, and application", *Journal of Microbiological Methods*, Elsevier Science, 43: 3-31.

Becker, G.S. (1976), *The Economic Approach to Human Behavior*, the University of Chicago Press, 1976.

Collan, M., Eklund, T., and Back, B. (2007), "Using the Self-Organizing Map to Visualize and Explore Socio-Economic Development", *EBS Review*, 22(1): 6-15.

Costea A., and Năstac, I. (2005), "Assessing the Predictive Performance of ANN Classifiers Based on Different Data Preprocessing Methods", *Internat. Journal of Intelligent Sys. Acc. Fin. Mgmt.* vol. 13, issue 4 (December 2005), pp. 217-250, DOI: 10.1002/isaf. 269, John Wiley & Sons, Wiley InterScience.

Demuth, H. and Beale, M. (2001), *Neural Network Toolbox*, The MathWorks, Inc., Natick.

Dobrescu, E. (2006), *Macromodels of the Romanian Market Economy*, Editura Economică, Bucharest.

Eklund, T., B. Back, H. Vanharanta, A. Visa (2003), "Using the Self-Organizing Map as a Visualization Tool in Financial Benchmarking", *Information Visualization*, 2(3): 171-181.

Hagan, M.T., Demuth, H.B., and Beale, M. (1996), *Neural Networks Design*, MA: PWS Publishing, Boston.

Hornik, K., Stinchcombe, M. and White H. (1989), “Multilayer feedforward networks are universal approximators”, *Neural Networks*, 2: 359-366.

Jackson, J.E. (1991), *A user guide to principal components*, John Wiley, New York.

Moller, M.F. (1993), “A scaled conjugate gradient algorithm for fast supervised learning”, *Neural Networks*, 6: 525-533.

Năstac, I., Dobrescu, E. and Pelinescu, E. (2007), “Neuro-Adaptive Model for Financial Forecasting”, *Romanian Journal of Economic Forecasting*, Vol. 4, 8(3): 19-41.

Năstac, I., (2004), “An Adaptive Retraining Technique to Predict the Critical Process Variables”, *TUCS Technical Report*, No. 616, June 2004, Turku, Finland (http://www.tucs.fi/research/series/serie.php?type=techreport&year =2004).

Năstac, I., and Costea, A. (2004), “A Retraining Neural Network Technique for Glass Manufacturing Data Forecasting”, *Proceedings of IJCNN 2004*, Vol. 4, IEEE Print, Budapest, pp. 2753-2758.

Năstac, I., and Cristea, P. (2005), “Neuro-Adaptive Forecasting for Nonstationary Sequences”, *Proceedings of IEEE-SOFA 2005*, pp. 179-186.

Năstac, I., and Matei, R. (2003), “Fast retraining of artificial neural networks”, in Guoyin Wang *et al*. (Eds.), *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, Springer-Verlag, in the series of Lecture Notes in Artificial Intelligence (LNAI 2639), pp. 458-462.

O’Leary, D.E. (1998), “Using Neural Networks to Predict Corporate Failure”, *International Journal of Intelligent Systems in Accounting, Finance & Management* **7**, pp. 187-197.

Pendharkar, P.C. (2002), “A computational study on the performance of artificial neural networks under changing structural design and data distribution”, *European Journal of Operational Research* 138 (2002) 155-177.

Sanfey, A.G., Rilling, J.K., Aronson, J.A., Nystrom, L.E., Cohen, J.D. (2003), “The Neural Basis of Economic Decision-Making in the Ultimatum Game”, *Science*, June 2003, Vol. 300. No. 5626, pp. 1755 – 1758.

Shiller R.J. (1999), “Human behavior and the efficiency of the financial system”, Chapter 20 in *Handbook of Macroeconomics*, Elsevier 1999, vol. 1, Part C, pp 1305-1340.

Zupan, B., Demšar, J., Kattan, M.W., Ohori, M., Graefen, M., Bohanec, M., Beck, J.R. (2001), Orange and Decisions-At-Hand: “Bridging predictive data mining and decision support”, *IDDM-2001: ECML/PKDD-2001 Workshop Integrating Aspects of Data Mining, Decision Support and Meta-Learning* (eds. Giraud-Carrier, C., Lavrač, N., Moyle, S., Kavšek, B.), Freiburg, pp. 151-162.