

# Fuzzy and Neuro-Symbolic Approaches to Assessment of Bank Loan Applicants

Ioannis Hatzilygeroudis<sup>a</sup>, Jim Prentzas<sup>b</sup>

<sup>a</sup> University of Patras, School of Engineering  
Department of Computer Engineering & Informatics  
26500 Patras, Greece  
ihatz@ceid.upatras.gr

<sup>b</sup> Democritus University of Thrace, School of Education Sciences  
Department of Education Sciences in Pre-School Age, Laboratory of Informatics  
68100 Nea Chili, Alexandroupolis, Greece  
dprentza@psed.duth.gr

**Abstract.** In this paper, we present the design, implementation and evaluation of intelligent methods that assess bank loan applications. Assessment concerns the ability/possibility of satisfactorily dealing with loan demands. Different loan programs from different banks may be proposed according to the applicant's characteristics. For each loan program, corresponding attributes (e.g. interest, amount of money that can be loaned) are also calculated. For these tasks, two separate intelligent systems have been developed and evaluated: a fuzzy expert system and a neuro-symbolic expert system. The former employs fuzzy rules based on knowledge elicited from experts. The latter is based on neurules, a type of neuro-symbolic rules that combine a symbolic (production rules) and a connectionist (adaline unit) representation. Neurules were produced from available patterns. Evaluation showed that performance of both systems is close although their knowledge bases were derived from different types of source knowledge.

## 1 Introduction

An important task of every bank involves the assessment of applications for bank loans. Such an assessment is important due to the involved risks for both clients (individuals or corporations) and banks. In recent financial crisis, banks suffered losses from a steady increase of customers' defaults on loans [12]. So, banks should avoid approving loans for applicants that eventually may not comply with the involved terms. It is significant to approve loans that satisfy all critical requirements and will be able to afford corresponding demands. To this end, various parameters related to applicant needs should be considered. Furthermore, different loan programs from different banks should also be taken into account, to be able to propose a loan program best tailored to the specific applicant's status. Computer-based systems for evaluating loan applicants and returning the most appropriate loan would be useful

since valuable assessment time would be spared and potential risks could be reduced. Banking authorities encourage developing models to better quantify financial risks [9].

Due to the complexity/significance of the assessment process, intelligent assessment systems have been used to reduce the cost of the process and the risks of bad loans, to save time and effort and generally to enhance credit decisions [1]. Such systems are mainly based on neural networks (e.g. [2]), but approaches such as genetic algorithms (e.g. [1]) and support vector machines (e.g. [12]) and other methods (e.g. [9]) have been applied too. There are some requirements in designing such a system. First, the system needs to include loan programs offered by different banks, to be able to return the most suitable one(s) for the applicant. Second, experience of banking staff specialized in loans is useful in order to outline loan attributes, applicant attributes, assessment criteria and the stages of the assessment process. Third, available cases from past loan applications are required to design and test the system.

In this paper, we present the design, implementation and evaluation of two intelligent systems that assess bank loan applications: a fuzzy expert system and a neuro-symbolic expert system. To construct those systems different types of source knowledge were exploited. The fuzzy expert system is based on rules that represent expert knowledge regarding the assessment process. The neuro-symbolic expert system is based on neurules, a type of hybrid rules integrating a symbolic (production rules) and a connectionist representation (adaline unit) [4]. Neurules exhibit advantages compared to pure symbolic rules such as, improved inference performance [4], ability to reach conclusions from unknown inputs and construct knowledge bases from alternative sources (i.e. symbolic rules or empirical data) [4], [5]. Neurules were produced from available cases (past loan applications). Evaluation of the fuzzy expert system encompassing rule-based expert knowledge and the neurule-based expert system encompassing empirical knowledge showed that their performance was close.

The rest of the paper is organized as follows. Section 2 introduces the domain knowledge involved and the stages of the assessment process. Section 3 discusses development issues of the fuzzy expert system. Section 4 briefly presents neurules and discusses development issues of the neurule-based expert system. Section 5 presents evaluation results for both systems. Finally, Section 6 concludes.

## **2 Domain Knowledge Modeling**

In this section we discuss issues involving the primary loan and applicant attributes modeled in the systems as well as the basic stages of the inference process. The corresponding knowledge was derived from experts.

### **2.1 Modeled Loan Attributes**

Each loan involves a number of attributes that need to be taken into account during inference. A basic attribute is the type of loan. Various types of loans exist. The two main types involve loans addressed to individuals and loans addressed to corporations.

Each main type of loan is further discerned to different categories. Loans addressed to individuals are discerned to personal, consumer and housing loans. Loans addressed to corporations are discerned to capital, fixed installations and other types of loans. The type of loan affects other attributes such as the amount of money that can be approved for loaning, the amount of installments and the interest.

Moreover, according to the type of loan, different requirements and applicant characteristics are taken into account. For instance, in the case of personal and consumption loans, a main applicant attribute taken into account is the net annual income. The maximum amount of money that can be approved for loaning is up to 70% of the annual income subtracting the amounts of existing loans in any bank. For example, an applicant with annual income €20,000 may borrow up to €13,000. In case there is a pending loan of €5,000, then he/she may borrow up to €8,000.

Applications for housing loans are thoroughly examined since the involved amount of money is usually large and the risks for banks are high. Various applicant attributes need to be considered such as property status and net annual income. For instance, the net annual income should be at least €15,000 and annual installments should not exceed 40% of the net annual income.

So, loan attributes such as the following are considered: *type of loan, the reason for applying, supporting documents, name of bank, type of interest* (i.e. fixed, floating), *commencement and termination of loan payment, amount of money loaned, loan expenses, way of payment* (e.g. monthly installments).

## 2.2 Modeled Applicant Attributes

To assess an applicant's ability to deal satisfactorily with loan demands, various applicant parameters are considered. The most significant attributes are the following:

- *Net annual income.* Expenses, obligations (e.g. installments) are excluded.
- *Financial and property status.* Possession of property is considered important (or even obligatory) for certain types of loans. This involves available bank accounts, bonds, stocks, real estate property, etc.
- *Personal attributes.* Personal attributes such as age, number of depending children, trade are considered important. Trade is considered a parameter of the applicant's social status. The number of depending children corresponds to obligations. Banks usually do not loan money to persons younger than twenty and older than seventy years old due to high risks (from the perspective of banks).
- *Warrantor.* A primary parameter for the overall assessment of an applicant is also the warrantor, who accepts and signs the bank's terms. In case the client cannot comply with obligations concerning the loan, the warrantor will undertake all corresponding responsibilities.

### 2.3 Inference Process Stages

The inference process involves four main stages outlined in the following. Table 1 summarizes the outputs produced in each stage.

In the first stage, basic inputs are given to the inference process concerning the requested loan. More specifically, the type of loan and the reason for loan application are given by responding to relevant system queries. Both inputs are stored as facts in the systems and are used to retrieve relevant loan programs from the Loan Programs Base. The approach involves loan programs from four different banks. The retrieved loan programs are used in subsequent stages.

**Table 1.** Summary of outputs for the four stages of the inference process

<b>Stage</b>	<b>Outputs</b>	
<i>Stage 1</i>	<i>Retrieves relevant loan programs</i> from the Loan Programs Base according to basic inputs (e.g. type of loan and reason for loan application)	
<i>Stage 2</i>	<i>Applicant assessment</i> <i>Warrantor assessment</i>	<b><i>Overall assessment</i></b>
<i>Stage 3</i>	<i>Restrictions</i> involving the funding of a loan are taken into account	
<i>Stage 4</i>	<i>Relevant loan programs are returned.</i> For each loan program return: interest, approved amount of money, installment and loan payment period.	

**Table 2.** Summary of variables involved in applicant/warrantor assessment (second stage)

<b>Variable</b>	<b><i>Applicant assessment</i></b>	<b><i>Warrantor assessment</i></b>
Net annual income (bad, fair, good)	×	×
Overall financial status (bad, fair, good)	×	×
Number of depending children (few, fair, many)	×	
Age (young, normal, old)	×	×
Social status (bad, fair, good)		×

The second stage involves an overall assessment of the applicant. This stage consists of three tasks: (a) applicant assessment, (b) warrantor assessment and (c) overall applicant assessment. The third task takes as input the results of the other two tasks. Such a process is applied since an overall applicant assessment is based on assessment of both applicant and warrantor. Rule-based inference performs all tasks. The variables involved in applicant and warrantor assessment are summarized in Table 2 (where '×' means 'depends on').

Applicant assessment is considered an intermediate variable. Its evaluation is based on the values of input variables such as net annual income, overall financial status, number of depending children and age. Variable 'net annual income' can take three values: bad, fair and good. Variable 'overall financial status' can take three values: bad, fair and good. Variable 'number of depending children' can take three values: few children (corresponding to 0-2 children), fair number of children (corresponding

to 3-5 children) and many children (corresponding to at least six children). Finally, variable 'age' can take three values: young, normal and old. Based on the values of these variables, the value of the intermediate variable 'applicant assessment' is evaluated. This variable takes three values: bad, fair and good. The design of systems has also taken into consideration that variables 'net annual income' and 'overall financial status' are more important in performing applicant assessment compared to the other two input variables (i.e. 'number of depending children' and 'age').

Warrantor assessment is also considered an intermediate variable. Such an assessment is significant in approving an application for a loan. The role of a warrantor is considered important in case of loans involving high risks. A warrantor is acceptable in case certain criteria (mainly financial) are satisfied. Evaluation of intermediate variable 'warrantor assessment' is based on warrantor attributes (input variables) such as net annual income, overall financial status, social status and age. Variables 'net annual income', 'overall financial status' and 'age' have similar representations as in the case of the applicant. Variable 'social status' depends on two parameters: monthly income and trade. This variable can take three values: bad, fair and good. Based on the values of the aforementioned variables, the value of the intermediate variable 'warrantor assessment' is evaluated. This variable takes three values: bad, fair and good.

The values of the intermediate variables ('applicant assessment' and 'warrantor assessment') are used to evaluate the value of the variable 'overall applicant assessment'. This variable takes five values: very bad, bad, fair, good and very good.

In the third stage, restrictions involving the funding of a loan are taken into account. As mentioned in a previous section, in case of personal and consumption loans, the maximum amount of money that can be approved for loaning is up to 70% of the annual income subtracting the amounts of existing loans in any bank. In case of home loans, annual installments should not exceed 40% of the annual income. Furthermore, applicant obligations (e.g. existing loans) are given as input.

In the fourth stage, results of all previous stages are taken into account. The maximum amount of money that can be approved for loaning is calculated. In case it is assessed that not all of the desired amount of money can be loaned to the applicant, the maximum possible amount of money that can be loaned is calculated. Relevant loan programs are produced as output. For each relevant loan program, all corresponding attributes are returned: interest, approved amount of money, installment and loan payment period.

### **3 The Fuzzy Expert System**

Domain knowledge is characterized by inaccuracy since several terms do not have a clear-cut interpretation. Fuzzy logic makes it possible to define inexact domain entities via fuzzy sets. One of the reasons is that fuzzy logic provides capabilities for approximate reasoning, which is reasoning with inaccurate (or fuzzy) variables and values, expressed as linguistic terms [11]. All variables involved in the second stage of the inference process are represented as fuzzy variables.

The developed fuzzy expert system has the typical structure of such systems as shown in Fig. 1. The fact base contains facts given as inputs or produced during inference. The rule base of the expert system contains fuzzy rules. A fuzzy rule includes one or more fuzzy variables. Definition of each fuzzy variable consists of definitions of its values. Each fuzzy value is represented by a fuzzy set, a range of crisp (i.e. non-linguistic) values with different degrees of membership to the set. The degrees are specified via a membership function. Fuzzy values and corresponding membership functions have been determined with the aid of the expert. We used triangles and trapezoids to represent membership functions. The system also includes the loan programs base containing loan programs from different banks.

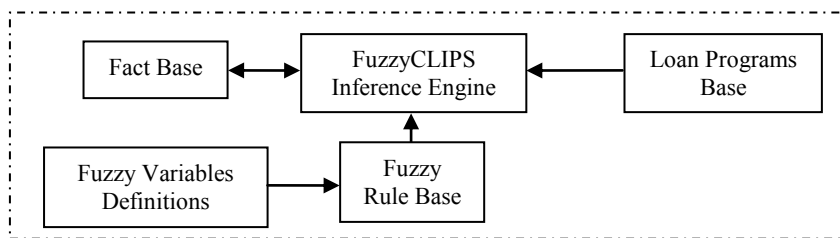


Fig. 1. Architecture of the fuzzy expert system

Reasoning in such a system includes three stages: fuzzification, inference, defuzzification. In fuzzification, the crisp input values (from the working memory) are converted to membership degrees (fuzzy values). In the inference stage, the MIN method is used for the combination of a rule's conditions, to produce the membership value of the conclusion, and the MAX method is used to combine the conclusions of the rules. In defuzzification, the centroid method is used to convert a fuzzy output to a crisp value, where applicable.

The system has been implemented in the FuzzyCLIPS expert system shell [13], an extension to CLIPS that represents fuzziness. Finally, about 55 fuzzy rules have been constructed. These rules were constructed with the aid of experts from four different banks specialized in loan programs. An example fuzzy rule is the following:

*"If net\_annual\_income is bad and number\_of\_dependent\_children is few and age is young and overall\_financial\_status is bad then applicant\_assessment is bad"*

#### 4 The Neurule-Based Expert System

Neurules are a type of hybrid rules integrating symbolic rules with neurocomputing giving pre-eminence to the symbolic component. Neurocomputing is used within the symbolic framework to improve the inference performance of symbolic rules [4]. In contrast to other hybrid approaches (e.g. [3]), the constructed knowledge base retains the modularity of production rules, since it consists of autonomous units (neurules), and also retains their naturalness in a great degree, since neurules look much like symbolic rules. The inference mechanism is a tightly integrated process resulting in

more efficient inferences than those of symbolic rules [4] and other hybrid approaches [8]. Explanations in the form of if-then rules can be produced [6].

The form of a rule is depicted in Fig.2a. Each condition  $C_i$  is assigned a number  $sf_i$ , called its *significance factor*. Moreover, each rule itself is assigned a number  $sf_0$ , called its *bias factor*. Internally, each rule is considered as an adaline unit (Fig.2b). The *inputs*  $C_i$  ( $i=1, \dots, n$ ) of the unit are the *conditions* of the rule. The weights of the unit are the significance factors of the rule and its bias is the bias factor of the neurule. Each input takes a value from the following set of discrete values: [1 (true), -1 (false), 0 (unknown)]. This gives the opportunity to easily distinguish between the falsity and the absence of a condition in contrast to symbolic rules. The *output*  $D$ , which represents the *conclusion* (decision) of the rule, is calculated via the formulas:

$$D = f(\mathbf{a}), \quad \mathbf{a} = sf_0 + \sum_{i=1}^n sf_i C_i$$

$$f(\mathbf{a}) = \begin{cases} 1 & \text{if } \mathbf{a} \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

where  $\mathbf{a}$  is the *activation value* and  $f(x)$  the *activation function*, a threshold function. Hence, the output can take one of two values ('-1', '1') representing failure and success of the rule respectively. The general syntax of a condition  $C_i$  and the conclusion  $D$  is:

<condition> ::= <variable> <l-predicate> <value>  
 <conclusion> ::= <variable> <r-predicate> <value>

where <variable> denotes a *variable*, that is a symbol representing a concept in the domain, e.g. 'net annual income', 'age' etc, in a banking domain. <l-predicate> denotes a symbolic or a numeric predicate. The symbolic predicates are {is, isnot} whereas the numeric predicates are {<, >, =}. <r-predicate> can only be a symbolic predicate. <value> denotes a value. It can be a *symbol* or a *number*. The significance factor of a condition represents the significance (weight) of the condition in drawing the conclusion(s). Table 3 presents an example neurule, from a banking domain.

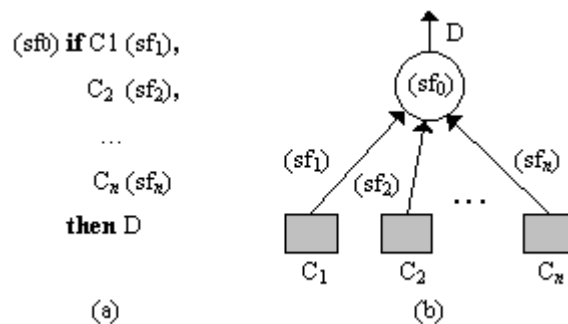


Fig. 2. (a) Form of a neurule (b) a neurule as an adaline unit

Variables are discerned to input, intermediate or output ones. An input variable takes values from the user (input data), whereas intermediate or output variables take

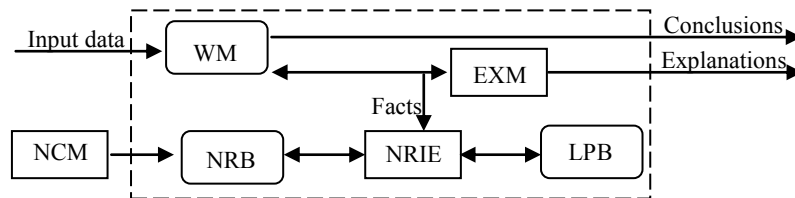
values through inference since they represent intermediate and final conclusions respectively. We distinguish between intermediate and output neurules. An intermediate neurule is a neurule having at least one intermediate variable in its conditions and intermediate variables in its conclusions. An output neurule is one having an output variable in its conclusions.

**Table 3.** An example neurule

N1 (-7.7) <b>if</b> social_status is good (6.6), net_annual_income is bad (3.4), age is normal (3.4), property_status is bad (3.3), property_status is fair (2.7) <b>then</b> warrantor_assessment is fair
--

Neurules can be constructed either from symbolic rules thus exploiting existing symbolic rule bases [4] or from empirical data (i.e. training patterns) [5]. In each process, an adaline unit is initially assigned to each intermediate and final conclusion and the corresponding training set is determined. Each unit is individually trained via the Least Mean Square (LMS) algorithm (e.g. [3]). When the training set is inseparable, more than one neurule having the same conclusion are produced.

In Fig. 3, the architecture of the neurule-based expert system is presented. The runtime system (in the dashed shape) consists of the following modules: the *working memory (WM)*, the *neurule-based inference engine (NRIE)*, the *explanation mechanism (EXM)*, the *neurule base (NRB)* and the *loan programs base (LPB)*. The neurule base contains neurules. These neurules are produced off-line from available empirical data concerning past loan cases (see Section 5). The construction process is performed by the *neurule construction mechanism (NCM)* [5].



**Fig. 3.** Architecture of the neurule-based expert system

## 5 Evaluation of the Systems

To evaluate the fuzzy expert system (FES) and the neurule-based expert system (NBES), we used 100 past loan cases mainly deriving from the Bank of Greece and the Bank of Cyprus. 30% of those cases were randomly chosen and used to test both systems. Random choice of test cases was performed in a way that an equal number of test cases corresponded to each of the five values of overall applicant assessment (i.e.



very bad, bad, fair, good, very good). The rest 70% of the available cases were used to construct the neurules contained in NBES. Thus, the knowledge base contents of the systems were derived from different source types (i.e. expert rules and cases).

Evaluation results for applicant assessment, warrantor assessment and overall applicant assessment are presented in Tables 4, 5 and 6 respectively. In each table, we present separate results for each one of the involved classes as well as average results for all classes. As mentioned in Section 2, applicant and warrantor assessment involve three classes whereas overall applicant assessment involves five classes. We use ‘accuracy’ accompanied by ‘specificity’ and ‘sensitivity’ as evaluation metrics:

$$accuracy = (a + d)/(a + b + c + d), \text{ sensitivity} = a/(a + b), \text{ specificity} = d/(c + d)$$

where  $a$  is the number of positive cases correctly classified,  $b$  is the number of positive cases that are misclassified,  $d$  is the number of negative cases correctly classified and  $c$  is the number of negative cases that are misclassified. By ‘positive’ we mean that a case belongs to the corresponding assessment class and by ‘negative’ that it doesn’t. Results show that performance of both systems is comparable considering all metrics. In overall applicant assessment, NBES performed slightly better in terms of accuracy and much better as far as sensitivity is concerned.

**Table 4.** Evaluation results for applicant assessment

Applicant Assessment	Accuracy		Sensitivity		Specificity	
	<i>FES</i>	<i>NBES</i>	<i>FES</i>	<i>NBES</i>	<i>FES</i>	<i>NBES</i>
<i>Bad</i>	0.90	0.83	0.80	0.87	0.95	0.81
<i>Fair</i>	0.73	0.76	0.60	0.60	0.80	0.85
<i>Good</i>	0.86	0.86	0.80	0.83	0.85	0.88
<b><i>Average</i></b>	<b>0.83</b>	<b>0.82</b>	<b>0.73</b>	<b>0.77</b>	<b>0.87</b>	<b>0.85</b>

**Table 5.** Evaluation results for warrantor assessment

Warrantor Assessment	Accuracy		Sensitivity		Specificity	
	<i>FES</i>	<i>NBES</i>	<i>FES</i>	<i>NBES</i>	<i>FES</i>	<i>NBES</i>
<i>Bad</i>	0.93	0.93	0.80	0.80	1.00	1.00
<i>Fair</i>	0.76	0.76	0.70	0.80	0.80	0.75
<i>Good</i>	0.83	0.83	0.80	0.70	0.85	0.90
<b><i>Average</i></b>	<b>0.84</b>	<b>0.84</b>	<b>0.77</b>	<b>0.77</b>	<b>0.88</b>	<b>0.88</b>

**Table 6.** Evaluation results for overall applicant assessment

Overall Applicant Assessment	Accuracy		Sensitivity		Specificity	
	<i>FES</i>	<i>NBES</i>	<i>FES</i>	<i>NBES</i>	<i>FES</i>	<i>NBES</i>
<i>Very bad</i>	0.93	0.90	0.66	0.83	1.00	0.90
<i>Bad</i>	0.96	0.93	0.83	0.83	1.00	0.95
<i>Fair</i>	0.86	0.90	0.50	0.83	0.95	0.91
<i>Good</i>	0.80	0.90	0.83	0.50	0.79	1.00
<i>Very good</i>	0.96	0.96	0.50	1.00	0.95	0.95
<b><i>Average</i></b>	<b>0.90</b>	<b>0.92</b>	<b>0.66</b>	<b>0.80</b>	<b>0.94</b>	<b>0.94</b>

## 6 Conclusions

In this paper, we present the design, implementation and evaluation of two intelligent systems for assessing bank loan applicants. Loan programs from different banks are taken into consideration. The intelligent systems involve a fuzzy expert system and a neuro-symbolic expert system constructed from expert rules and available cases respectively. The knowledge base of the neuro-symbolic expert system contains neurules, a type of hybrid rules integrating symbolic rules with neurocomputing.

Evaluation results for both systems are comparable, despite the different types of knowledge sources. In certain aspects, one system performs better than the other. This means that both types of knowledge sources can be exploited in producing outputs. Based on the results, an integrated (or hybrid) approach could be developed. In this perspective, there are two directions for future research. One involves development of an hybrid system involving both systems as separate cooperating modules. The other direction involves development of an integrated system the exploits both types of knowledge sources. Research (e.g. [10]) has shown that synergies from using both (rule-based) domain theory and empirical data may result in effective systems. An approach as in [7] combining neurules and cases could be investigated.

## References

1. Abdou, H.A.: Genetic Programming for Credit Scoring: The Case of Egyptian Public Sector Banks. *Expert Systems with Applications* 36 (2009) 11402-11417.
2. Eletter, S.F., Yaseen, S.G., Elrefae, G.A.: Neuro-Based Artificial Intelligence Model for Loan Decisions. *American Journal of Economics and Business Administration* 2 (2010) 27-34.
3. Gallant, S.I.: *Neural Network Learning and Expert Systems*. MIT Press (1993).
4. Hatzilygeroudis, I., Prentzas, J.: Neurules: Improving the Performance of Symbolic Rules. *International Journal on AI Tools* 9 (2000) 113-130.
5. Hatzilygeroudis, I., Prentzas, J.: Constructing Modular Hybrid Rule Bases for Expert Systems. *International Journal on AI Tools* 10 (2001) 87-105.
6. Hatzilygeroudis, I., Prentzas, J.: An Efficient Hybrid Rule-Based Inference Engine with Explanation Capability. *Proceedings of the 14<sup>th</sup> International FLAIRS Conference*. AAAI Press (2001) 227-231.
7. Hatzilygeroudis, I., Prentzas, J.: Integrating (Rules, Neural Networks) and Cases for Knowledge Representation and Reasoning in Expert Systems. *Expert Systems with Applications* 27 (2004) 63-75.
8. Hatzilygeroudis, I., Prentzas, J.: Neurules: Integrated Rule-Based Learning and Inference. *IEEE Transactions on Knowledge and Data Engineering* 22 (2010) 1549-1562.
9. Min, J.H., Lee, Y.-C.: A Practical Approach to Credit Scoring. *Expert Systems with Applications* 35 (2008) 1762-1770.
10. Prentzas, J., Hatzilygeroudis, I.: Categorizing Approaches Combining Rule-Based and Case-Based Reasoning. *Expert Systems* 24 (2007) 97-122.
11. Ross, T.J.: *Fuzzy Logic with Engineering Applications*. John Wiley & Sons (2010).
12. Zhou, L., Lai K.K., Yu, L.: Least Squares Support Vector Machines Ensemble Models for Credit Scoring. *Expert Systems with Applications* 37 (2010) 127-133.
13. [http://awesom.eu/~cygal/archives/2010/04/22/fuzzyclips\\_downloads/index.html](http://awesom.eu/~cygal/archives/2010/04/22/fuzzyclips_downloads/index.html)