Social Validation of Solutions in the Context of Online Communities

Lydia Nahla Driff, Lamia Berkani, Ahmed Guessoum, Abdellah Bendjahel

Social Validation of Solutions in the Con text of Online Communities

An Expertise-Based Learning Approach

Lydia Nahla Driff, Lamia Berkani, Ahmed Guessoum and Abdellah Bendjahel

Artificial Intelligence Laboratory (LRIA), Department of Computer Science, USTHB
driff.nahla@gmail.com, l_berkani@hotmail.com,
{lberkani, aguessoum}@usthb.dz

Abstract. Online Communities are considered as a new organizational structure that allows individuals and groups of persons to collaborate and share their knowledge and experiences. These members need technological support in order to facilitate their learning activities (e.g. during a problem solving process).We address in this paper the problem of social validation, our aim being to support members of Online Communities of Learners to validate the proposed solutions. Our approach is based on the members' evaluations: we apply three machine learning techniques, namely a Genetic Algorithm, Artificial Neural Networks and the Naïve Bayes approach. The main objective is to determine a validity rating of a given solution. A preliminary experimentation of our approach within a Community of Learners whose main objective is to collaboratively learn the Java language shows that Neural Networks represent the most suitable approach in this context.

Keywords: Learning Community; Social Validation; Expertise-Based Learning; Machine Learning.

1 Introduction

Today, with the great development of Information and Communication Technologies, a wide diversity of social learning frameworks have been promoted, including Online Learning Communities (OLCs) and social networks. The notion of OLC has been defined in different ways, exploring mainly the social aspects of collaborative learning (Laister and Kober, 2013) and the research and theory concerned with social support for learning (Swan and Shea, 2005).

One of the most important challenges of such communities is to enhance the knowledge exchange and sharing among the different members. With the increasing number of interactions, members collectively produce new knowledge in various formats (documents, solutions to problems, etc.) that will subsequently be published to the whole community (Le Boulch, 2009).The production of this knowledge is increasingly developed by members of the community who have different levels of expertise (experts, novices, etc.) and this highlights the need for validation of the new

knowledge before it is stored and published to the rest of the community so as to ensure its reliability.

Validation is the expression of a judgment on a concept or whatever needs to be assessed after study/observation. This judgment can be favorable or not. Social validation of a concept is a collective action that aims at the evaluation of this concept based on various judgments and opinions expressed by different people from the field (Herr and Anderson, 2008), on the basis of statistical analysis, or approved opinions, experiences, etc. Its main objective is the assessment whether the concept is good or not and this can be represented by a degree of validity which is the percentage of conformity of the concept.

We focus in this work on the social validation of the proposed solutions within an OLC. We aim to support members in this process, providing them with a tool that will help them to "automatically" validate newly proposed solutions. We address the need to ensure the credibility of the validation process and we propose an expertise-based learning approach, by reusing past experiences (i.e. previous validations made by supervisors).

The review of the literature about social validation shows that little work has focused on this aspect in an educational setting. We especially mention the work proposed by Cabana et al. (2010) about the social validation on collective annotations where the authors addressed the problem of scalability (i.e. a resource which is more and more annotated is less and less exploitable by individuals). The authors proposed a way to socially validate collective annotations with respect to the social theory of information. Berkani et al. (2013) proposed a social validation of learning objects based on two features: (1) the members' assessments, formalized semantically, and (2) an expertise-based learning approach, applying a machine learning technique. The authors used neural networks because of their proven efficiency in many domains such as complex problem solving.

The remainder of this paper is structured as follows. In Section 2, we present the context of the study. Our contribution is presented in Section 3, where we give a detailed description of the parameters that are related to the evaluation of a solution. Then, we present the application of the aforementioned Machine Learning techniques to any given solution using the defined parameters. The experimental results are presented in Section 4 where a discussion of the strengths and weaknesses of each technique is presented. The conclusion and perspectives are stated in Section 5.

2 Context of the study

In our study, we consider a learning community related to the domain of higher education. The members of this community target the learning of the JAVA programming language and its different concepts. The main objective of this community is to improve the learners' skills and his acquisition of new knowledge.

We assume that the community includes members with different skill levels (beginners, advanced, experts, etc.) and the Java language includes various concepts

(classes, abstraction, and so on), including the handling of tools (Java web GUI, JVM, etc.).

We have found out that the discussion forum and/or the Frequently Asked Questions are services that can be considered as the most used by the community members. Indeed, these often use these services in their interactions and information exchange.

However, one of the problems encountered by the community members is the validation of the proposed solutions; several questions can be asked at this point: Is a given solution correct? If yes, to what extent is it accurate? And, more importantly, who has validated it? Was it validated by a set of members, by a single expert, or within some other setting? This is why we focus in this work on the social validation of a solution, and we try to automate this process in order to support community members in their learning activities.

Automation is very cost-effective when it comes to time saving. It helps avoid several phases and replaces the manual work done by the supervisor (or teacher) whose role is to validate the solutions that are proposed by the community members (or learners). On the other hand, the social validation can give some precision about the obtained results by considering distinct opinions that are based on different criteria.

In the next section, we present our approach for automating the process of social validation, bearing in mind that we consider the validation of one solution at a time.

3 Contribution

In our work, we have tried to automate the validation process using Machine Learning (ML) techniques. The choice for these techniques has been dictated by the fact that they allow to take into account the rich experience (knowledge repository) in terms of validations of solutions throughout the lifetime of a community. As such, they directly take into account the existing experience, which makes them very different from conventional algorithmic methods where an exact and accurate understanding of the factors that are taken into account in any validation of a solution is required.

We start by the modelling of the solution validation problem in terms of some specific parameters that need to be represented. Our goal is to define an evaluation in a unique way so as to be able to manipulate it in the (automatic) training phase.

In this section, we present these parameters as well as the process followed by each of the considered ML techniques.

3.1 Parameters of an Evaluation

As explained above, the prediction of a degree of validity for a given solution is collective, based on the various members' assessments of the solution. To this end, we have defined the parameters that we consider as being the most important and significant ones to identify, characterize, and implement each assessment (see Fig. 1). Thus we believe that the evaluation by a given member M_i concerns the level of the evaluator, the assigned score, the confidence, the evaluation context, the skill level and the success. We now explain each of these parameters.

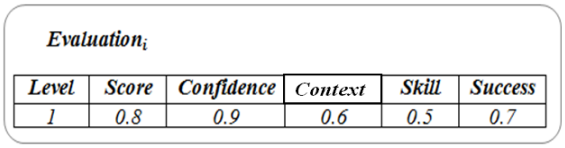

Fig. 1. Parameters of an evaluation

1. ***Level of the evaluator:*** Each assessment corresponds to a single member. This member has a certain amount of knowledge and expertise which are measured by the parameter "Level". In our case, a member is either a Professor, a PhD student, a Master's student, a student preparing a Bachelor's degree or a member with basic knowledge.
   The evaluator's Level is predefined in this user's profile at the time of his registration into the community. The value of this "level" is a coefficient that depends on the significance of the evaluator's level with respect to all the existing levels in the community.

$$Coefficient(Level) = Score/N \quad (1)$$

   where: N is the number of levels

Table 1. Computation of the level of the evaluator

Level	Coefficient
Professor	5/5=1
PhD	4/5=0.8
Master's	3/5=0.6
Bachelor's	2/5=0.4
Basic	1/5=0.2

2. ***Score*:** For each evaluation, a score will be given by the evaluating member. This score represents the member's opinion of the solution being evaluated: the evaluator may assign a high score if the solution is good or very good, and an average score if the solution is not quite correct, or even a low score if the solution is judged incorrect.

3. ***Confidence:*** The score given by an evaluating member reflects his opinion. A percentage of confidence is assigned by the evaluator himself to each score he gives: if he is sure of his score, he will give a high degree of confidence; otherwise, the degree of confidence will be lower.

4. ***Evaluation context:*** In evaluating a solution, the member does his assessment of the solution based on a given source. This source could be a book, a document, an article, etc. The importance of the sources is different based on each one's credibility. We thus assign a weight to each source to indicate its reliability. The evaluation context can take several formats: tested results, research outcome, a similar problem, an approved opinion, or a new problem.

5. ***Skill level:*** A profile is associated with each member of the community. This profile is mainly used to retrieve information about the expertise of a member according to what he/she described as areas of expertise with respect to all the areas identified in the community by the input parameter "degree of expertise". As such, for a given problem in a specific domain, a member will have a certain level of expertise that we call "skill level". More precisely, this level represents the quality of a member in relation to his expertise in a specific area. This will allow us to get an idea about the credibility of his evaluation.

   Three cases can be distinguished:

   - Either the domain belongs to the member's skills set (high rate)
   - Or the domain belongs to the member's centre of interest (average rate)
   - Or the member has no knowledge at all about the domain (low rate)

   In order to calculate the skills level of a member with respect to the domain of the solution under evaluation, we need to calculate the similarity between all of the member's areas of expertise and the domain of this solution. To this end, we have defined a taxonomy that encompasses the existing domains in a given community and have added rules that cover all the possible cases for the relative positions of two domains in the taxonomy. We summarize these rules in the following table:

Table 2. Similarity rules

	Description	**Similarity**
Rule 1	D is the same as D'	Sim = 1
Rule 2	D (direct or indirect) parent of D'	Sim = 1
Rule 3	D (direct/indirect) son of D'	$\text{Sim} = \frac{\lvert \text{weight}(D) - \text{weight}(D') \rvert}{\Delta \text{Lev}}$
Rule 4	D and D' are independent	$\text{Sim} = \begin{cases} \text{weight}(D_c) \text{ if D and D}' \text{ at the same level} \\ \frac{\text{weight}(D_c)}{\Delta \text{Lev}} \text{ otherwise} \end{cases}$

where:

D is the problem domain;
D' is one of the member's domains of expertise
Dc is the closest (parent) domain common to D and D'
ΔLev is the difference between the levels of D and D'

The skill level of a member is calculated in relation to all his areas of expertise using an ontology that uses the mentioned degrees of expertise (see Fig.2). These degrees reflect the coverage of subdomains (son nodes) from the related domain (parent node) in terms of knowledge. For example, a member who has knowledge in the field IGraphic, *a-fortiori* covers the subdomains Swing and AWT.

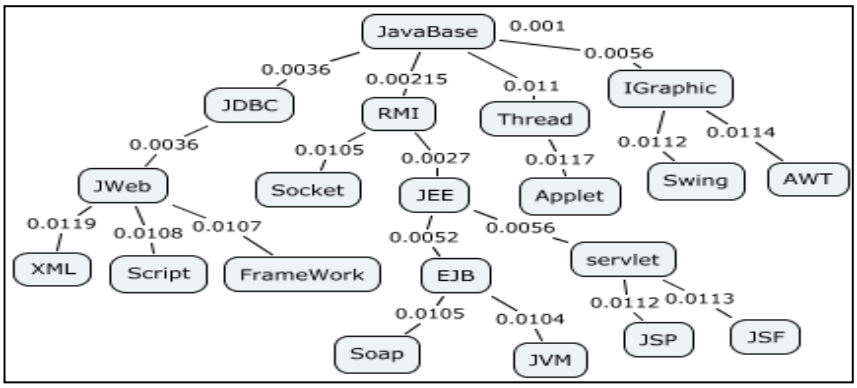

Fig. 2. Example of a taxonomy for the Java domain

This skill level is calculated according to the following steps:
Calculate the similarity between each domain of expertise D_i and the desired/searched-for domain D:

$$Similarity(D, Di) \tag{2}$$

Associate the similarity of each domain with the member's degree of expertise on D_i

$$Similarity(D, Di) * Expertise(M, Di) \tag{3}$$

Find the domain that corresponds to the value that maximizes the obtained values using the expression:

$$k = argmax_i(Similarity(D, Di) * Expertise(M, Di)) \tag{4}$$

Calculate the *quantum* which represents the amount of acquired skills in other domains to be added to the global competence of the member. The aim is not to neglect this additional skill.

$$Quantum= \{\{i = 1 - N\} \text{ and } i \neq k, (Similarity(D, Di) * Expertise(M, Di) / 10*(N - 1) \} \tag{5}$$

We point out that the number 10 is a factor that we can manipulate to increase or decrease the quantum with which we will adjust the additional competency provided by other domains rather than that giving the maximum of similarity.
Calculate the final competency of M with respect to D given that this value must be at least equal to a maximum value and does not exceed the value 1.

$$Competency\ (M, D) = min[1, ((S_k * P_k) * (1 + quantum))] \tag{6}$$

where:

S_k is the similarity between the domain D_k which maximizes the competence of M and the domain of the solution

P_K is the member's expertise
D is the domain of the solution

6. The degree of success: An active member may be considered as a trusted source because of his correct assessments in two cases: (1) he generally evaluates positively solutions that at the end of the validation process obtain a high degree of validity; and (2) he generally evaluates negatively solutions that at the end of the validation process are assigned a low degree of validity.
Accordingly, we assign to him a degree of success which represents the distance of one of his evaluations to the final validity of the solution according to the score he gave it. The following is the formula we propose to calculate the degree of success:

$$Success(Si) = 1 - |Score - validity(Si)| \quad (7)$$

A member's success score is calculated with respect to the relative success of all the solutions he has evaluated:

$$Success = \frac{1}{N}\sum_{i=1}^{N}(Success(Si)) \quad (8)$$

3.2 Use of machine learning techniques

Machine Learning (ML) techniques are powerful in terms of their flexibility and ease of extraction of hidden relations that exist within data of the various applications they could be used for. We have decided to use ML in our problem of social validation of solutions; the intuition is to have automated learning from past experience of users' evaluations and the experts' assessments of the quality of these evaluations. We use the representation of an evaluation as presented above and apply different ML techniques on data given in this representation.

3.2.1 Modelling of machine learning methods.

We are mainly interested in three methods: Genetic Algorithms (GA) (See (Goldberg, 1989), (Holland, 1992) and (Mitchell, 1996)); Neural Networks (NN) (See (Muller and Reinhardt, 1994) and (Fausett, 1994)); and the Naïve Bayes Approach (Mitchell, 1997), as presented in the following sub-sections:

3.*2.1.1 Genetic Algorithms*: Genetic algorithms (GA) are often used for optimization. In our case, we have used a similar approach to Data Mining guided by the GA to highlight useful information for solving our problem. The approach proceeds as follows in the learning phase:

- Consider an initial population as a set of evaluations, carried out on different solutions, and represented using the six aforementioned parameters.
- Conduct a series of crossings and mutations by randomly changing the parameter values.
- Keep the final population which corresponds to the final validity predicted by a supervisor. The fitness function used is the following:

$$Fitness = \frac{1}{N+1}\sum_{i=1}^{N}(c_i * x_i) \quad (9)$$

where x_i is the value of the attribute and c_i its coefficient.

- Encode the population obtained after the previous step using a binary encoding.
- Apply an algorithm for mining association rules such as the Close algorithm (Pasquier et al., 1999; Pei et al, 2000).

Gradually enrich the rule base with new rules. These new rules will be added to cater for newly encountered cases that are not covered by the already generated rule base.

In our context, an Association Rule (Han et al., 2006; Sarawagi et al, 2000) is an implication of the form X → Y where X is a conjunction of Attribute-Value pairs of the form "$Attribute_i$ = $Value_j$" and Y is a pair Attribute-Value of the same form which represents the degree of validity of an evaluation. Example:

$$If\ \big((Score = 0{,}8)\ and\ (Confidence = 0{,}6) and\ (Success = 0{,}9)\big)\ then\ Validity = 0{,}8 \quad (10)$$

The CLOSE algorithm is used for the extraction of informative association rules based on the informative content of the database (Pasquier et al., 1999; Pei et al, 2000).

3.2.1.2 *Neural Networks*: Artificial Neural Networks (ANNs) have been designed to mimic information flow in the human brain. Neural networks are efficient for complex problem solving, especially for pattern matching, classification, and optimization problems. In our case, we have used this method to predict a degree of validity of a given evaluation.

After designing several models of ANNs, we have tried various learning and activation functions, varying the number of neurons in each case. We have selected the NN architecture as follows for as good a learning phase as possible:

- Design a multilayer perceptron containing: six inputs, twenty neurons on the hidden layer and one neuron on the output layer.
- Feed in input into the network each 6-value input describing an evaluation of a solution.
- Give as output the validity score given by the supervisor for the given input.
- Train the network until the best learning is obtained (trying various architectures).
- Once a good learning has been achieved, simulate the network.

3.2.1.3 Naive Bayes Approach: The Naive Bayes Approach is a probabilistic approach based on conditional probability calculations. It is called Naïve due to the assumption it makes of independence of the various events (attributes) it considers. In spite of this, this assumption has not prevented them from providing an efficient and often good approach. In our case, we have considered a set of evaluations on various

solutions, represented using the six parameters. We present below the steps followed for the calculation of the probabilities:

- Calculate the probability of the validity value Vi :

$$P(V_i) = \frac{Frequency(value_i)}{N} \quad (11)$$

where N is the number of evaluations considered for the learning.

- Calculate the conditional probability that an attribute takes a value $value_i$, given that V_i is the value of the validity:

$$P(value_i/V_i) = \frac{Frequency(value_i)}{P(V_i)} \quad (12)$$

- Calculate the conditional probabilities that the validity takes different possible values V_i bearing in mind that the attributes have some given values.

$$P(Vi/attribute_i = value_1, attribute_j = value_2, \ldots, attribute_n = value_m) \quad (13)$$

- Consider the validity V_i corresponding to the maximal probability of the set obtained as the validity of the evaluation.

3.2.2 Application of the machine learning techniques:

In the case of a new solution to be validated, the steps to be followed are as follows:

- Represent all the evaluations of the solution using the six attributes.
- Predict a degree of validity for each one of the evaluations for the three methods.
- In the case of the GA: associate with each evaluation the most suitable rule, and then generate rules based on the validity rates for each evaluation.
- In the case of the ANN approach: simulate each evaluation by the ANN assigning a validity rate for each evaluation.
- In the case of the Naïve Bayes approach: calculate the probabilities of the evaluations and associate each evaluation with a validity rate.
- Remove the incorrect values from the set of validity rates.
- Apply a credibility formula to calculate the final validity:

$$FinalValidity = \sum_{i=1}^{N} (Credibility_i * Validity_i) \quad (14)$$

4 Implementation and tests

4.1 Description of the testing phase

In order to test our approach, we have developed an online community platform for Java learning, including the different functionalities related to our approach. Furthermore, we have developed a prototype to automatically generate a large number of

solutions and their evaluations. This has allowed us to create the database and hence to carry out our learning process. In addition, we have used the community platform, where members can add new problems, propose new solutions or evaluate some existing ones. Then we have represented all the obtained evaluations according to the six parameters, as proposed in section III. Finally, we have applied the selected ML techniques.

4.2 Discussion of the findings

We have implemented the three ML techniques to predict the degree of validity of a given solution. After some tests and experimentations, we obtained the results shown in the following figures:

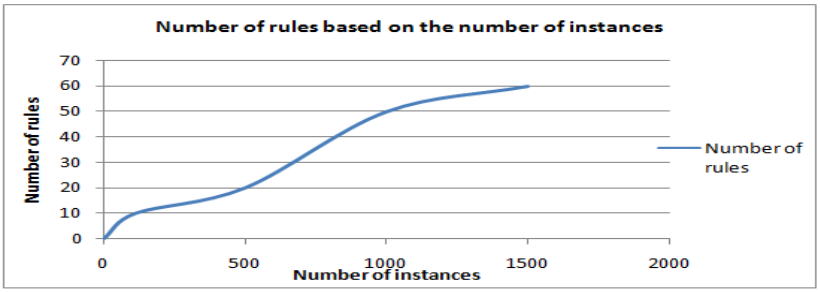

Fig. 3. Increasing number of the association rules (GA approach)

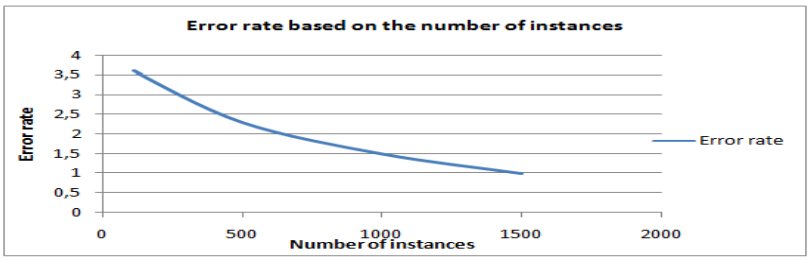

Fig. 4. Error rate of the ANN method

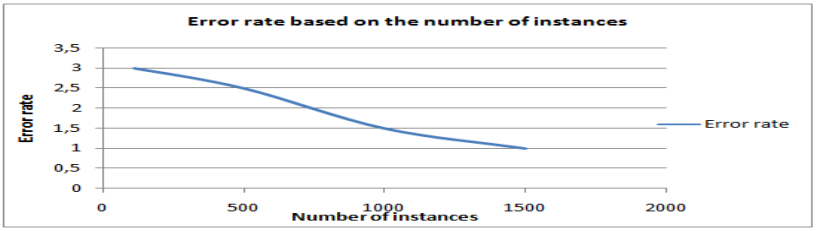

Fig. 5. Error rate of the Naïve Bayes approach.

- We can deduce from the application of the GA approach that more and more new solutions are validated, more rules can be generated and the accuracy of the rules application increases improving the degree of validity (see Fig. 3).
- The application of ANNs allows us to conclude that the error rate decreases with the increasing number of examples that are used as input during the ANN learning phase (see Fig. 4).
- Finally, the application of the Naïve Bayes approach allows us to deduce that with the increasing number of examples, the error rate decreases, which implies that the number of correct predictions increases, and hence the results become more accurate (see Fig. 5).

On the other hand, an analysis of the results we have obtained has allowed us to compare the three ML methods on the basis of four criteria (see Table 3).

- *Criterion 1 – Data redundancy:* if data appears frequently, then the learning outcomes are improved.
- *Criterion 2 – Number of instances:* when the number of data instances used in the learning increases, the learning is better guided and gives more accurate results.
- *Criterion 3 – Appropriateness of results:* the learning is considered good if it frequently gives accurate results with low error rates.
- *Criterion 4 –New cases: l*earning is considered good if it can handle well and robustly a situation where a new case arises (a case which was not seen during the learning phase).

Table 3. Comparison of the three ML methods.

	GA	**ANN**	**NBA**
Criterion 1	Yes	No	Yes
Criterion 2	Yes	Yes	Yes
Criterion 3	Somewhat	Strongly	Always
Criterion 4	Frequently	Somewhat	Never

According to these results and analysis, we conclude that the neural networks have given the best performance compared to the two other approaches. In order to improve the obtained results, it would be very interesting to combine these approaches in different ways and to compare the performance of the different algorithms.

5 Conclusion and perspectives

We are interested in this work in the problem of social validation of solutions proposed in the context of a learning community. We have considered the evaluations carried out on already proposed solutions and modeled the problem according to several criteria. An automatic validation process was proposed using three machine learning techniques: genetic algorithms, neural networks and the Naive Bayes Approach. An experimental study of the developed prototype has been conducted. The results show that neural networks have given the best performance.

As future work we envisage to make further tests on a real community of learners and collect as much data as possible to enrich the learning. We envisage also to check the possibility of combining some of the different learning techniques and to generalize the process of social validation of more than one proposed solutions to the same problem.

References

1. Berkani, L., Driff, L.N., Guessoum, A. Social Validation of Learning Objects in Online Communities of Practice Using Semantic and Machine Learning Techniques. Modeling Approaches and Algorithms for Advanced Computer Applications, pp. 237-247, (2013)
2. Cabanac, G., Chevalier, M., Chrisment, C., Julien, C.: Social validation of collective annotations: Definition and experiment. Journal of the American Society for Information Science and Technology, vol. 61. No. 2, pp. 271-287, (2010).
3. Fausett, L.: Fundamentals of Neural Networks: Architectures, Algorithms, and Applications, Prentice Hall, ISBN: 0133341860, (1994)
4. Goldberg, D.: Genetic Algorithms in Search, Optimization and Machine Learning. Reading, MA: Addison-Wesley Professional, ISBN 978-0201157673, (1989)
5. Herr, K., Anderson, L.G.: The Action Research Dissertation: A Guide for Students and Faculty. Thousand Oaks,.ISBN 0-7619-2991-6, (2008).
6. Holland, J.: Adaptation in Natural and Artificial Systems. MA: MIT Press, (1992).
7. Laister, J., Kober, S.: Technikum Joanneum Social Aspects of Collaborative Learning in Virtual Learning Environments. http://comma.doc.ic.ac.uk/inverse/papers/patras/
8. Le Boulch, D. Bouyssou, D., Grundstein, M.: Towards a redefinition of the relationships between information systems development and individual cognition. In Information Technologies in Environmental Engineering, Springer-Verlag Heidelberg, (2009)
9. Mitchell, T.: Machine Learnin", McGraw Hill,. ISBN 007042807, (1997).
10. Mitchell, M.: An Introduction to Genetic Algorithms. MA: MIT Press, (1996).
11. Muller, B., Reinhardt, J.: Neural Networks, Springer Verlag, (1991).
12. Pasquier, N. Bastide, Y. Taouil, R., Lakhal, L.: Discovering frequent closed itemsets for association rules. In 7th International Conference on Database Theory, January (1999).
13. Pei, J. Han, J., Mao, R.: Closet: An effcient algorithm for mining frequent closed itemsets. In SIGMOD Int. Workshop on Data Mining and Knowledge Discovery, (2000).
14. Swan, K., and Shea, P.: The development of virtual learning communities. In. S. R. Hiltz & R. Goldman, Asynchronous Learning Networks: The Research Frontier. New York: Hampton Press, pp. 239-260, (2005).
15. Han, .J, Kamber, .M, Data Mining: Concepts and Techniques, 2ed. Morgan Kaufmann Publishers, March 2006. ISBN 1-55860-901-6. (2006)
16. Sarawagi, S.,Thomas, S., Agrawal, R.: Integrating Association Rule Mining with Databases: Alternatives and Implications. Data Mining and Knowledge Discovery Journal, pp4 (2/3), (2000).