

An Educational Data Mining Model for Predicting Student Performance in Programming Course

A.F.ElGamal
Department of CS
Mansoura University
Egypt

ABSTRACT

This paper presents an educational data mining model for predicting student performance in programming courses. Identifying variables that predict student programming performance may help educators. These variables are influenced by various factors. The study engages factors like students' mathematical background, programming aptitude, problem solving skills, gender, prior experience, high school mathematics grade, locality, previous computer programming experience, and e learning usage. The proposed model includes three phases; data preprocessing, attribute selection and rule extraction algorithm.

Keywords

Data Mining, Student Performance, Programming Course, Rule Extraction.

1. INTRODUCTION

The term KDD refers to the overall process of knowledge Discovery in Databases. Data mining is a particular step in this process, involving the application of specific algorithms for extracting patterns (models) from data. The additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, ensures that useful knowledge is derived from the data [1]. Data mining offers capabilities to process data of different types (e.g., qualitative and quantitative) and originating at different sources. None of the traditional approaches is able to cover such a wide spectrum of data diversity [2].

The purpose of a data mining effort is normally either to create a descriptive model or a predictive model. A descriptive model presents the main features of the data set. The most characteristic descriptive modeling task is clustering. The purpose of a predictive model is to allow the data miner to predict an unknown (often future) value of a specific variable, the target variable. The predictive model is created from given known values of variables, possibly including previous values of the target variable [3]. Data mining becomes one of the most important tools used for solving most of today's problems that are related to different sectors of our life. The wide range of applications from business tasks (credit risk analysis) to scientific tasks (prediction of mutagenicity of organic compounds) have led to the development of a huge variety of learning methods and algorithms for rule extraction and

prediction. The general tasks of classification, regression, clustering, or deviation analysis have a large number of solutions such as neural networks, decision tree learners, rule learners or Bayesian networks [4]. P. Andreeva, M. Dimitrova and P. Radeva presented a comparison of different learning models and a practical guideline how to select the most suited algorithm for a specific application and some empirical criteria for describing and evaluating learning methods are given. Different data mining systems (free software and demo versions) are explored, based on specific techniques for classification, clustering and decision-making. Three case studies for medical data sets are presented and potential benefits of the proposed methodology for diagnosis learning are suggested [5]. Rule extraction can be used to discover relationships between data and possibly to improve the generalization of solution. A. S. Kuttilyl presented some issues concerning rule extraction techniques: Granularity: The level of detailed hypotheses and evidence that the system can provide with each of its output decisions. Transparency: How well the decisions or conclusions can be explained. Modifiability: Ability of extracted rules to be updated when the corresponding trained with different data sets. Robustness: Measure of how insensitive the method to corruptions in the training data or initial domain knowledge [6].

In the most of studies researchers use Artificial Neural Network (ANN) and Support Vector Machine (SVM) for classification and then extract rules from this system [7-11]. Other studies use Group Method of Data Handling (GMDH) to extract rules [12, 13].

2. EDUCATIONAL DATA MINING

Educational data mining is an interesting research area which extracts useful, previously unknown patterns from educational database for better understanding, improved educational performance and assessment of the student learning process [14]. It is concerned with developing methods for exploring the unique types of data that come from educational environment which include students' results repository. Figure 1 shows the application of data mining in educational systems. C. Romero, S. Ventura introduced a survey from 1995 to 2005 for the application of data mining to traditional educational systems, particular web-based courses, well-known learning content management systems, and adaptive and intelligent web-based educational systems.

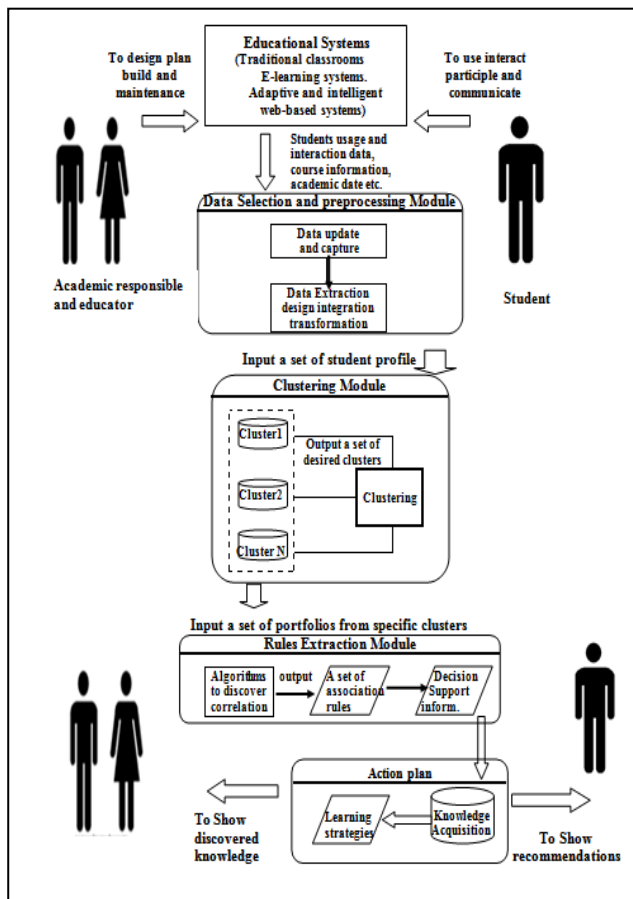


Fig 1: Application of data mining in educational systems

Each of these systems has different data source and objectives for knowledge discovering. After preprocessing the available data in each case, data mining techniques can be applied: statistics and visualization; clustering, classification and outlier detection; association rule mining and pattern mining; and text mining [15]. E. Chandra presented data mining in education environment that identifies students' failure patterns using mining techniques; the traditional Apriori algorithm was implemented to generate the hidden patterns from the students' failed courses dataset [16]. An approach to classify students in order to predict their final year grade based on the features extracted from logged data in an educational web-based system was reported in [17]. The relationship between students' university entrance examination results and their success was studied using cluster analysis and k-means algorithm techniques in [18]. S.Charles, L.Arockiam, and V.Kumar investigated the association between learning behavior and programming skills. Clustering is used to determine the similarity in the students' dataset. Each cluster reveals the identity based on its learning behavior of the student. Multilayer Perception (MLP) technique classifies the learning behavior of students and their programming skills based on learning by example [19].

3. Predictors of Programming Performance

Most students believe that computer programming is difficult. High failure rates were reported among in introductory programming courses poses an important challenge for freshmen students of many programs. These courses require students to devote a sustained effort during the whole course

and a failure to do so may students taking programming courses [20-22].

Although several factors that affect learning to program have been identified over the years, we are still far from a full understanding of why some students learn to program easily and quickly while others flounder. The most frequently mentioned factor is previous computer programming experience. Self-efficacy for programming is influenced by previous programming experience, and student self-efficacy increases substantially during an introductory programming course. Furthermore, students' mental models of programming influence their self-efficacy, and both the mental model and self-efficacy have a direct effect on overall success in an introductory course. [23]. Other factors that may affect course success have been less well investigated such as relationship of mathematics or science background to computer programming success. A relationship between student learning styles and learning to program has been found [24]. There is a body of research on the student's mental model of programming in relation to success in specific programming tasks [25]. In summary, there is a substantial literature on factors affecting student performance [26].

Predicting student performance in a particular course, or even on assessments within a course, is a difficult but useful undertaking. Given such predictions, a professor can help focus student effort on potential problem areas for particular students given their performance in previous courses.

Previous studies have identified a number of predictors. Wilfred W.F. Lau, Allan H.K. Yuen examined the effect of a combination of predictors (gender, learning styles, mental models, prior composite academic ability, and medium of instruction) on programming performance [27].

A.T. Chamillard reported the use of statistical analysis techniques to build predictive models. While many of the generated models did not have sufficient predictive power to be useful, the stronger models and other observations from the analysis provide useful insight into the relationships between the various courses. The models presented use only previous course grades as predictor variables [28]

Sally Fincher et al. presented a multi-national, multi-institutional study that investigated introductory programming courses. Report presents a study of possible influencing factors that is distinctive in a number of ways. First, it is both multi-institutional and multi-national, with participants from eleven institutions in three countries. . The study was based on four different diagnostic tasks: a spatial visualization task (a standard paper folding test); a behavioral task used to assess the ability to design and sketch a simple map; a second behavioral task used to assess the ability to articulate a search strategy; and an attitudinal task focusing on approaches to learning and studying (a standard study process questionnaire) [29].

3.1 Programming Aptitude Test

Many attempts have been made to design tests that can reliably predict whether or not students are adequately prepared for CS1 courses, and a number of studies have attempted to determine specific testable attributes that correlate highly with success in CS1 [30].

Programming aptitude testing has coexisted with the programming profession and education for decades. Probably the best known early test used to predict programming aptitude was the IBM Programmer Aptitude Test (PAT) [31]. Many

researches investigated the relationship between programming skill acquisition and various measures of individual differences including: 1) mathematical background; 2) problem solving abilities; and 3) cognitive skills [32-38]. In this study we formulate a test contains 20 questions and tasks from tasks represent the Huoman Programming Aptitude Test, Dehnadi's assessment instrument, problem solving questions, and mathematically oriented programming skills test. Table 1 contains sample of these tasks.

Table 1. Sample of programming Aptitude Test' tasks

Huoman Programming Aptitude Test	Dehnadi's Assessment Instrument	problem solving questions	Math. Oriented Prog. Skills Test
Your job is to deduce which of the given h numbers is the smallest. How many comparisons you have to make if you have: a) 3 numbers b) 6 numbers c) n numbers (n>=1)	Ali, Nour, and Body each have a favorite food: donuts, egg, and fish, in no practical order. List all of the ways in which you can choose two people and foods. For example: Ali, Nour; daunts, egg.	Solve the following: black sheep = dag kip white dog = tin bud black cow = dag stam white sheep = ?	An ATM machine has 20 and 50 coins. Find The mathematical function for finding m, n as a function of x, such that x=50m+20n.

3.2 Collecting Students Data

The test was applied to students from computer science department at Mansoura University over three years. The obtained results and students' data about other variables are collected. These variables or attributes contained: gender, prior experience (student's score in the previous academic year), high school mathematics grade, locality, previous computer programming experience, and e learning usage where the students' programming course has an e learning form which is produced under supervision of National E-Learning Center, Egypt. The e learning course activated over three academic years. The collected data are analyzed by data mining techniques and predicting the results as illustrated in the following section.

4. Knowledge Extraction for Student Performance Prediction

The Students attributes' data is used to extract knowledge. The automated knowledge extraction model developed through three phases: data preprocessing, attribute selection, and rule extraction. The students' data includes 10 predictive attributes and one target attribute. The predictive attributes are Student ID, High school mathematics grade, mathematical background, problem solving, programming aptitude, prior experience, previous computer programming experience, gender, locality, and e learning usage. The target attribute is

the Grade (student performance in programming course). Sample of students' dataset is shown in table 2.

4.1 Data Preprocessing

The values of attributes in real world databases may take different shapes, normal values, unique value, single value, almost one value, missing values, continuous values and multi values. The attributes which have normal values contain valuable information and are important in knowledge acquisition. The unique value attributes have a different value for every single record or nearly every record. These attributes identify each record exactly and do not have predictive value. The one value attributes (unary-valued) do not contain information that helps to distinguish between the different records. Therefore they should be ignored for data mining purposes. If a given database includes continuous attributes (real values) then the search based on all possible conjunction values for extracting rules yields to a burdensome computation and consumes much time. This problem can be solved by using a fuzzification process which leads to the reduction of

search space. The fuzzy subset of the universe of discourse U

is described by a membership function $\mu_v(V) : U \rightarrow [0,$

1], which represents the degree to which $\mu \in U$ and belongs to the set V. A fuzzy linguistic variable, V, is an attribute whose domain contains linguistic values, which are labeled for the fuzzy subsets [39]. Therefore, the continuous attributes can be transformed into linguistic terms such as; Short (S), Medium (M) and Long (L). A non-overlapping rectangular membership functions may be used and the bounds of each linguistic term can be determined by using the smooth histogram of real values [40]. The preprocessing stage performs fuzzification process for High school mathematics grade, mathematical background, problem solving, and programming aptitude into nominal values.

4.2 Attribute Selection

Preprocessing stage of dataset constructs a new universal attribute set, U_{att_new} ($U_{att_new} \subseteq U_{att}$). This set is now suitable for the extraction of the most relevant attributes, U_{att_most} , ($U_{att_most} \subseteq U_{att_new}$). The most relevant attributes are determined according to an evaluation function (E_s), presented in [41].

$$E_s = \frac{n * C_{AT}}{\sqrt{n + n(n-1) * C_{AA}}}$$

Where;

$$C_{AT} = \frac{K * \sum_{a_i \in A} P(a_i)^2}{(1 - \sum_{t_j \in T} P(t_j)^2) * \sum_{t_j \in T} P(t_j)^2}$$

$$k = \left[\sum_{t_j \in T} P(t_j) * (1 - P(t_j)) \right] - \sum_{a_i \in A} \left[\frac{P(a_i)^2}{\sum_{t_j \in T} P(a_i)^2} * \sum_{t_j \in T} P(t_j | a_i) * (1 - P(t_j | a_i)) \right]$$

Where;

t_j : All possible values of target attribute T.

a_i : All possible values of attribute A.

n : Number of attributes in test set.

C_{AA} : Average of attribute - attribute correlations.

C_{AT} : Average of attribute -target correlations.

$P(a_i)$: Probability that an attribute has value a_i .

$P(t_j|a_i)$: Conditional probability that target has value t_j when an attribute has a value a_i .

The attribute selection process determines the most relevant attribute set as: {High school mathematics grade, mathematical background, programming aptitude, prior experience, previous computer programming experience, locality, and e learning usage}, which have the largest evaluation function value, Es.

4.3 Rule Extraction

Decision trees are a classic method of inductive inference that is still very popular. They are not only easy to implement and use for classification and regression tasks, but also good predictive performance, computational efficiency [42]. The construction of a decision tree is based on splitting internal nodes recursively. The selection of split attributes on internal nodes is extremely important during the construction process and determines to a large extent the final structure of the decision tree. Many efforts have been made on this aspect and a set of split criteria, such as the Gini index, the information gain and the Chi-Square test, are available. Entropy theory is adopted to select the appropriate split attribute by the well-known C4.5 algorithm. Let N be the size of the dataset D and N_j the number of samples in class j . Assuming that there are K class labels, the entropy theory states that the average amount of information needed to classify a sample is as follows:

$$Inf\alpha(D) = \sum_{j=1}^k \frac{N_j}{N} \log_2 \left(\frac{N_j}{N} \right)$$

When the dataset D is split into several subsets $D_1, D_2 \dots D_n$ according to the outcomes of attributes X , the information gain is defined as:

$$Gain(X, D) = Info(D) - \sum_{j=1}^k \frac{N_j}{N} Info(D_j)$$

Where N_j is the number of samples in subset D_j . ID3 favors all attributes with the largest gain. C4.5 applies Gain ratio, instead of Gain, as following:

$$Gain - ratio(X, D) = Gain(X, D) / \left(- \sum_{i=1}^n \left(\frac{N_i}{N} \right) \right)$$

C4.5 greedily partitions nodes until a trivial value of the Gain ratio is achieved. A prune procedure is then performed in order to avoid generating a complex tree that over fits the data [43]. A part of Decision tree for the students' dataset is shown in figure 2.

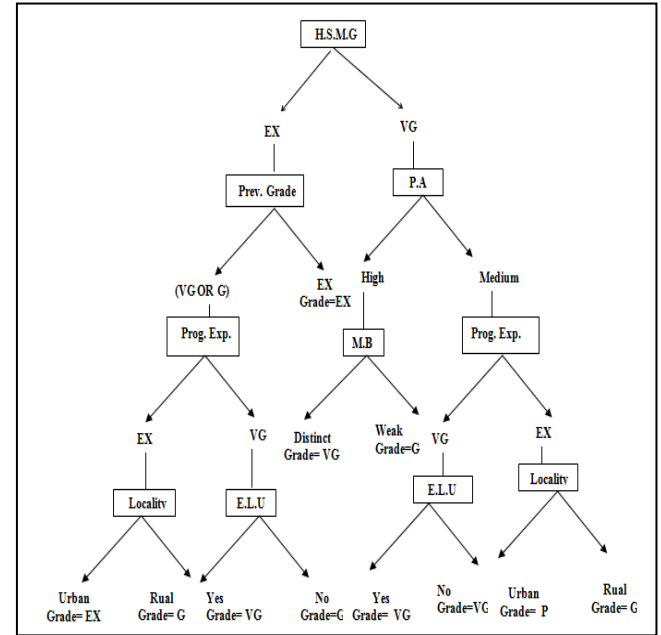


Fig 2: A Part of Decision Tree

The final stage is the rule extraction algorithm. The proposed algorithm divides a given dataset into N predictive attributes and one target attribute (class). Each attribute is denoted by A_i ($i = 1, 2 \dots N$). The different possible values, m_i , of the attribute A_i is V_{ij} ($j = 1, 2 \dots m_i$). The target attribute has K classes. Each class is denoted by class k ($k = 1, 2 \dots K$). The level of search depth is labeled as level L ($1 \leq L \leq N-1$). The steps of the proposed algorithm are:

```

For k 1 to K                                     (number of classes)
  For i 1 to N                                     (number of attributes)
    For j 1 to  $m_i$                                (number of values in each attribute)
      L = 1                                         (supporting level)
      S = {  $V_{ij}$  }                                (Test Set)
      Do until the end of all attribute
        Do while L < N-1
          P = P ( S | Class k )
          IF P = 1
            THEN
              { Get rule from values of S and level L
                Drop the values used from attributes
                Check conjunction of all remaining
                  values in set S }
              L = L + 1
            End Do
          End Do
        Next j
      Next i
    Next k
  Begin Refinement of the extracted rules

```

```
{
    Neglect rules which have supporting level less than
        specific threshold value
    Remove redundant rules
    Summarize rules
    Defuzzify the continues attribute values
}
```

End

Sample of the extracted rules is given in the following:

R1: If High School Mathematics Grade = VG and P.A. = Medium --> then target attribute (Grade) = G

R2: If High School Mathematics Grade = VG and previous grade = G --> then target attribute (Grade) = G

R3: If High School Mathematics Grade = VG and Locality = rural --> then target attribute (Grade) = VG

R4: If previous grade = VG and M.B = Distinct --> then target attribute (Grade) = G

R5: If Programming Exp = G and e-learn usage = yes and Locality = urban --> then target attribute (Grade) = VG

The proposed algorithm is efficient for extracting accurate and comprehensible set of rules from the students' dataset for the following reasons; (i) it extracts only rules in which the antecedent(s) satisfy 100 % of the conditional probability within certain class, consequently the output rules are accurate. (ii) It controls the number of antecedent(s) in each extracted rule according to stopping criterion. The number of antecedent

doesn't exceed N-1. Therefore the extracted rules are comprehensible for the user. (iii) The proposed algorithm extracts only a set of rules which satisfy a specific threshold supporting level (number of records in the dataset covered by the extracted rule). The high supporting level reduces the number of output rules and increases their quality.

5. Conclusion

The ability to predict an individual's potential to learn programming concepts is important for many reasons. Identifying variables that predict student programming performance may help educators and employers select potential students and employees. Curriculum committees can use prediction results to guide changes to the curriculum and evaluation of the effects of those changes. Data mining can be used to discover novel relationships that may exist in the data and possibly to improve the generalization of solution. This study investigates the potential of data mining for enhancing the effectiveness of academic planners and level advisers in higher institutions of learning. Students' data in the department of computer science from Mansoura University are collected over three academic years and data mining algorithm is applied to extract rules predicting students' performance in programming course. A set of accurate and comprehensible rules is obtained. The machine learning algorithm extracts only the rules which have the antecedent(s) satisfy 100 % of the conditional probability within certain class. The extracted rules confirm the importance of variables such as High School Mathematics Grade and programming aptitude. Educators should take these variables into account in the qualifying exam to join computer science departments at universities.

Table 2. Sample of students' dataset

ID	High School Mathematics Grade	M.B.	P.S	P.A.	Prior experience	Programming Exp.	Gender	Locality	E-learn usage	Student Grade
1	44.5	2	4	6	VG	Ex	F	rural	yes	Ex
2	46.5	2	2	5	VG	VG	F	urban	yes	G
3	42.5	3	2	1	p	p	M	urban	no	VG

6. REFERENCES

- [1] Sushmita Mitra, Senior Member., Sankar K. Pal, Fellow, and Pabitra Mitra, "Data Mining in Soft Computing Framework: A Survey", IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 13, NO. 1, JANUARY 2002.
- [2] A. Kusiak, "Data Mining and Decision Making", in B.V. Dasarathy (Ed.), Proceedings of the SPIE Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology IV, Vol. 4730, SPIE, Orlando, FL, April 2002, pp. 155-165.
- [3] Löfström, T. & Johansson, U., 2005. Predicting the Benefit of Rule Extraction: A Novel Component in Data Mining. Human IT, 7(3), pp. 78-108.
- [4] Murat Koklu_, Humar Kahramanli and Novruz Allahverdi, "A New Approach To Classification Rule Extraction Problem by the Real Value Coding", International Journal of Innovative Computing, Information and Control Volume 8, Number 9, September 2012
- [5] P. Andreeva, M. Dimitrova and P. Radeva, "Learning Models and Algorithms for Rule Extraction and Prediction", www.cvc.uab.es/~petia/maya%20saer3.pdf
- [6] A. S. Kuttiyil, "Survey of Rule Extraction Methods", Master Thesis, Wayne State University, Detroit, Michigan, 2004.
- [7] L. Ozbakir, A. Baykasoglu, S. Kulluk and H. Yap_c_, TACO-miner, "An ant colony based algorithm for rule

- extraction from trained neural networks", *Expert Systems with Applications*, vol.36, pp.12295-12305, 2009.
- [8] L. E. Zarate, S. M. Dias and M. A. J. Song, FCANN," A new approach for extraction and representation of knowledge from ANN trained via formal concept analysis", *Neurocomputing*, vol.71, no.13-15, pp.2670-2684, 2008.
- [9] R. Nayak, "Generating rules with predicates, terms and variables from the pruned neural networks", *Neural Networks*, vol.22, no.4, pp.405-414, 2009.
- [10] A. E. Elal_, R. Haque and M. E. Elalami," Extracting rules from trained neural network using GA for managing E-business", *Applied Soft Computing*, vol.4, no.1, pp.65-77, 2004.
- [11] M. A. H. Farquad, V. Ravi and S. B. Raju, "Support vector regression based hybrid rule extraction methods for forecasting", *Expert Systems with Applications*, vol.37, no.8, pp.5577-5589, 2010.
- [12] KOJI FUJIMOTO* and SAMPEI NAKABAYASHI, "Applying GMDH algorithm to extract rules from examples", *Systems Analysis Modelling Simulation* Vol. 43, No. 10, October 2003, pp. 1311-1319.
- [13] Nekuri Naveen, V. Ravi, and C. Raghavendra Rao, "data Mining via Rules Extracted from GMDH: an application to predict churn in bank credit cards, KES 2010, part I, LNAI6276, pp. 80-89, 2010.
- [14] A.Y.K. Chan, K.O. Chow, and K.S. Cheung. "Online Course Refinement through Association Rule Mining" *Journal of Educational Technology Systems* Volume 36, Number 4 / 2007-2008, pp 433 – 44, 2008.
- [15] C. Romero , S. Ventura, "Educational data mining: A survey from 1995 to 2005", *Expert Systems with Applications* 33 (2007) 135–146.
- [16]E. Chandra,"Knowledge Mining from Student Data", *European Journal of Scientific Research*, ISSN 1450-216X, Vol.47 No.1 (2010), pp.156-163.
- [17] B.Minaei-Bidgoli, D. A. Kashy, G. Kortemeyer and, W. F. Punch."Predicting student performance: an application of data mining methods with the educational web-based system LON-CAPA" In *Proceedings of ASEE/IEEE Frontiers in Education Conference*, Boulder, CO: IEEE, 2003.
- [18] F. Castro, A. Vellido, A. Nebot, and F. Mugica. "Applying Data Mining Techniques to e-Learning Problems". *Evolution of Teaching and Learning Paradigms in Intelligent Environment* ISBN: 10.1007/978-3-540-71974-8_8 Volume 62, pp 183-221. Springer Berlin Heidelberg, 2007.
- [19] S.Charles, L.Arockiam, and V.Kumar, "Deriving Association between learning behavior and programming skills", *Computer Networks and Information Technologies Communications in Computer and Information Science*, Volume 142, 2011, pp 96-103.
- [20] Luis de-la-Fuente-Valentín, Abelardo Pardo, Carlos Delgado Kloos, "Addressing drop-out and sustained effort issues with large practical groups using an automated delivery and assessment system", *Computers & Education* 61 (2013) 33–42.
- [21] Douglas A. Kranch, "Teaching the novice programmers: A study of instructional sequences and perception", *Educ. Inf. Technol.*, Springer, 2012.
- [22] Jens Bennedsen, Michael E. Caspersen, "Failure rates in introductory programming", *ACM SIGCSE Bulletin* , Volume 39 Issue 2, June 2007, Pages 32-36
- [23] Susan Wiedenbeck, Deborah LaBelle, and Vennila N.R. Kain,"Factors Affecting Course Outcomes in Introductory Programming' 16th Workshop of the Psychology of Programming Interest Group. Carlow, Ireland, April 2004.
- [24] Lynda Thomas, Mark Ratcliffe, John Woodbury and Emma Jarman, "Learning Styles and Performance in the Introductory Programming Sequence", *Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, Vol. 34, No. 1. (March 2002), pp. 33-37.
- [25] Richard Bornat, Saeed dehnadi, and , " Mental models, consistency and programming aptitude", *Proceedings of the tenth conference on Australasian computing education - Volume 78,2008, Pages 53-61.*
- [26] Leo F. Denton and Dawn McKinney , " Affective Factors and Student Achievement: A Quantitative and Qualitative Study", 34th ASEE/IEEE Frontiers in Education Conference, 2004, Session T1G, pp.6-11.
- [27] Wilfred W.F. Lau, Allan H.K, " Modelling programming performance: Beyond the influence of learner characteristics", *Computers & Education* 57 (2011) 1202–1213.
- [28] A.T. Chamillard, "Using Student Performance Predictions in a Computer Science Curriculum" *ITiCSE'06*, June 26–28, 2006, Bologna, Italy.
- [29] Sally Fincher et al., "Programmed to succeed?: A multi-national, multi-institutional study of introductory programming courses", *Technical Report No. 1-05*, University of Kent, April 2005.
- [30] Carl Farrell, " PREDICTING (AND CREATING) SUCCESS IN CS1", *Issues in Information Systems*, Volume VII, No. 1, 2006, pp.259-263.
- [31] Markku Tukiainen and Eero Mönkkönen, " Programming aptitude testing as a prediction of learning to program", 14th Workshop of the Psychology of Programming Interest Group, Brunel University, June 2002.
- [32] Valerie J. Shute, "Who is likely to acquire programming skills?", *Educational computing research*, Vol. 7(1), 1991, pp.1-24.
- [33] Carl Farrell ,PREDICTING (AND CREATING) SUCCESS IN CS1, *Issues in Information Systems*, Volume VII, No. 1, 2006, pp.259-263.
- [34] Saeed Dehnadi, "A Cognitive Study of Learning to Program in Introductory Programming Courses", thesis submitted to Middlesex University, May 2009.
- [35] Y.B.-D. Kolikant, S. Pollack, "Improving mathematically oriented programming skills in computer science studies" *fie*, vol. 1, pp.T1G3-8, 32nd Annual Frontiers in Education (FIE'02), 2002.
- [36] Gerald E. Evans and Mark G. Simkin, "What best predicts computer proficiency?", *Communications of the*

- ACM, November 3989 Volume 32 Number 11, pp.1322-1327.
- [37] Doane, William E. J," Predicting student performance in introductory computer programming courses", State University of New York at Albany, ProQuest Dissertations and Theses, 2008.
- [38] Stanley TenEyck Schuyler,"Using Problematising Ability to Predict Student performance In A First Course In Computer Programming", Robert Morris University, Copyright © Stanley TenEyck Schuyler 2008.
- [39] Vassilios Petridis, Vassilis G. Kaburlasos, " Clustering and Classification in Structured Data Domains Using Fuzzy Lattice Neurocomputing (FLN)", IEEE Transactions on Knowledge and Data Engineering, March/April, Vol. 13, No. 2, 2001.
- [40] J.H.Wang, Wen-Jeng Liu and Lian-Da Lin, " Histogram-Based Fuzzy Filter For Image Restoration", IEEE Trans On Systems, Man, and Cybernetics, Vol.32, No.2, PP. 230-238, Apr.2002.
- [41] R.M.Hogarth. " Methods For Aggregating Opinions", In H.Jungermann And G.De Zeeuw, Editors, Decision Making And Change In Human Affairs. D. Reidel Publishing, Dordrecht-Holland, 1977.
- [42] Sebastian Nowozin, "Improved Information Gain Estimates for Decision Tree Induction", Appearing in Proceedings of the 29 th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012.
- [43] T Warren Liao and Evangelos Triantaphyllou, Recent advances in data mining of enterprise data algorithms and applications, Series on Computers and Operations Research, Vol 6, 2007.