

Intelligent System for Improving Student Performance in Programming Language via Error Detection and Analysis

A. E. E. ElAlfi

Dept. of Computer Science
Faculty of Specific Education
Mansoura, Egypt

H. A. A. Abase

Dept. of Educational and Psychological Sciences
Faculty of Specific Education
Mansoura, Egypt

S. A. Wasef

Dept. of Educational and Psychological Sciences
Faculty of Specific Education
Mansoura, Egypt

Ash. R. Abdel Rahman

Dept. of Educational and Psychological Sciences
Faculty of Specific Education
Mansoura, Egypt

Abstract -The development of an intelligent system to detect student' errors in the programming language and to improve his/her programming style is a difficult task; this is due to the difficulty of finding an effective method for teaching course that can be suitable for all students. This paper presents an intelligent system to improve student performance in java programming language; the system is composed of e-course, proposed student model and an advisory system. The e-course is designed through the collecting of common programming errors via human expert, literature review, student diagnostic test and course analysis. A proposed student model is designed through literature review that includes the components of the student model. An advisory system is developed based on integrated knowledge resources; these resources are the extracted rules via decision tree, human experts and student tests. The artificial neural network is used for student prediction to assist student in learning the programming language course.

Keywords- *Error Detection, Student Model, Decision Tree, Intelligent System, Artificial Neural Network.*

i. INTRODUCTION

Teaching computer programming languages within the university context has proven to be a difficult and demanding task. Computer science educators have shown growing concern over the difficulties with which computer programmers learn programming principles [1].

Programming is a fundamental part of the computer science curriculum, but one which is often problematic. It seems to be difficult to find an effective method of teaching that is suitable for all students [2].

Programming learning has a complex and cognitively challenging nature [3].

It is important to study programming errors since the study can lead to get a better understanding of problem-solving strategies and will highlight the difficult aspects of programming and programming instruction [4].

Analyses of programming errors can help improving the teaching of novice programming in various ways [5].

The main aim of this paper is to design an intelligent system to predicate student errors and overcomes the presence of these errors.

This paper is organized as follows:-

First explain the errors detection in the programming language.

Then discuss of the student model which is the important element within the electronic program for the treatment of common errors.

The second phase of this study is focused on knowledge acquisition from decision tree and rules pruning.

On the other hand, described the advisory system which predict of current and previous history for student from artificial neural network.

Finally the discussion of the students results who studied using the electronic program with the development of a model student and the students who studied in the traditional way.

ii. THE PROPOSED FRAMEWORK

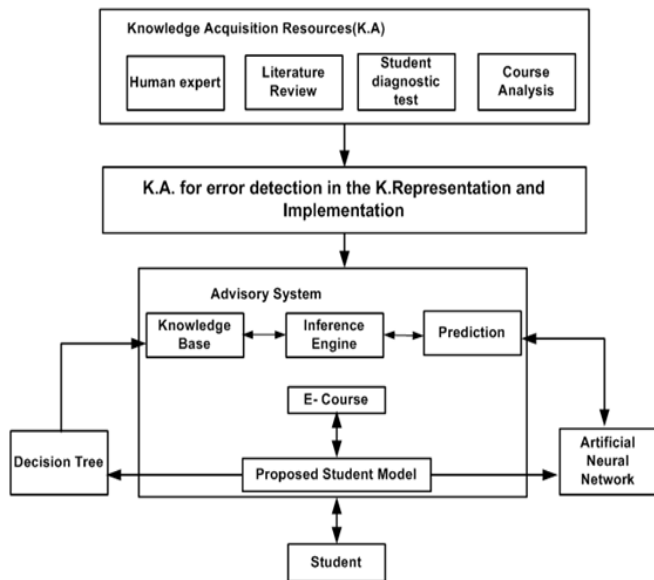


Figure.1 Framework for Student Assessment in the Programming Language Error Detection & Correction

The overall system framework which is developed in this paper is shown in figure1. The system is based on three main steps

- The first step concerns with the knowledge acquisition resources, these resources depend on the human experts, literature review, student diagnostic test and course analysis.
- The second step is the domain knowledge representation and implementation. The output of this step is the knowledge base for error detection in the programming language.
- The third step deals with building the e-course which includes the integration of a proposed student model, the knowledge base and an ANN for student prediction.

a. Knowledge Acquisition for Error Detection

There are different resources for error detection in the programming language under the study. These resources are domain expert, literature review, student diagnostic tests and course analysis.

The common errors detected are explained in the following section:

b. Common Programming Errors

The main types of programming errors can be categorized as syntax, semantic and logical errors [5].

Syntax errors refer to the spelling mistakes, punctuation and order of words in the program. Generally missing some symbols such as a semicolon produces this type of error [6].

Semantic errors appear due to misuse of programming concepts. It is generated when the meaning of the code is not consistent with the language, for instance using a non-static global variable inside the static method [4].

Logical errors or "Lexical errors" constitute the third class of error. This occurs when a token is unrecognized [2].

Logical errors occur when the program does not solve the problem that the programmer meant for it to solve [5].

Vin diagram that represents these errors is shown in figure2.

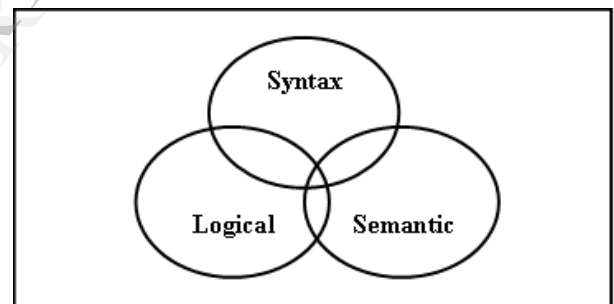


Figure.2 Relationship between three types of errors programming

Samples of these errors can be explained as shown in table1.

TABLE1. Examples of errors in Java language

Error Type	Errors	Error Correction
Syntax Errors	Public class main{ Public static void main(string [] args) { system.out.println("welcome")	public class main{ public static void main(String [] args) { System.out.println("welcome"); } }
Semantic Errors	int sum=5; double sum=5.9; int void=5;	double sum=5.9; int void=5;
Logical Errors	int n=1; switch (n) { case 1: System.out.println("the number is 1"); case 2: System.out.println("the number is 2"); default: System.out.println("the number is not 1 or 2"); }	int n=1; switch (n) { case 1: System.out.println("the number is 1"); break; case 2: System.out.println("the number is 2"); break; default: System.out.println("the number is not 1 or 2"); }
Syntax &Semantic Errors	public class Main { public static void main(String () args) { n=22 System.out.println("n= "+ n)	public class Main { public static void main(String () args) { int n=22; System.out.println("n= "+ n); }}
Syntax &Logical Errors	for (int i=1, i<=10, i++) if i==5; break; System.out.println("Looping");	for (int i=1; i<=10; i++) { if (i==5) break; System.out.println("Looping"); }
Semantic &Logical Errors	string counter=2; do { System.out.println(counter); counter+=2; } while(counter>2);	int counter=2; do { System.out.println(counter); counter+=2; } while(counter<20);
Syntax &Semantic & Logical Errors	int a, b, c; product =1; c=4; a=0; b=c/a; while (c<=5) product *=c; ++c; System.out.println(product); System.out.println(b);	int a, b, c; int product =1; c=4; a=2; b=c/a; while (c<=5){ product *=c; ++c; System.out.println(product); System.out.println(b); }

TABLE 2. List of compiler messages in Java program

Syntax errors	Forgot semicolon at the end of statement.
	Identifier expected (Can't find symbol).
	Missing parentheses in while statement loops.
	Mismatched brackets/parentheses.
	Misspelling else.
	Placing semicolon after the condition in an if structure.
	Invalid method declaration.
Syntax errors	Using keywords as an identifier.
	Unused variables.
	Using increment or decrement operator in wrong way.
	Not an expression statement.
	Possible loss of precision.
	Operator...can't be applied to.
	Variable ...might not have been initialized.
Logical errors	Inconvertible types.
	Class not found
	Using commas instead of the two required semicolons in for header
	Forgetting a break statement when are is needed in a switch structure
	Division by zero

Table2 shows some compiler messages that appear during student training and exercises.

iii. THE MAIN STUDENT MODELS

The student model is an abstract representation that an intelligent tutoring system (ITS) or a Web-based education system (WBES) depicts about specific attributes of an individual [7].

A computer educational system, as an ITS or WBES, that holds a student model pursues to automatically adapt the selection of lectures, the type of content, the way of evaluation, and many more items and tasks to the particular profile of each student [8].

Usually, the student model encompasses attributes stemmed from specific domains, such as: acquired knowledge about the teaching domain, background, misconceptions, behavior, goals and interests of the individual [7].

To state what aspects of the student model affected a particular teaching action. It should equally be possible we believe that the goal of modeling the student at the level that an excellent teacher can do is an unnecessarily difficult goal. Computer-based

teachers are fundamentally different from human teachers [9].

The student model is also used as the basis for corrective feedback, feedback on performance is given in real-time and the performance data is also captured. Course instructors are also kept informed of the progress of students [10].

a. Student Model Components

One of many instructional approaches that support inquiry-based science is the 5E model, which includes five specific components: engage, explore, explain, elaborate, and evaluate [11].

Engagement: Is to make connection between past learning experiences and present learning situations. The role of each student in this level is, encourage students to compare their ideas with those of others and make connection between the teacher and student.

b. The Proposed Student Model

The goal of this step is to develop a student model using the 5E models by integrating some of the basic components of a model student in educational programs such as e-mail, previous experience, net beans, internet, goals, subjects (syntax errors, semantic errors, logical errors), external materials, activities and testes where this elements were merged in the proposed student model in this paper within the 5E models, as the following table:

TABLE 3. Student Models Adaptation

5E Models	Student attributes	No
Engagement	Via E-mail, Previous Experience	2
Exploration	Net beans, Internet	2
Explanation	Goals, Subjects (Syntax errors, Semantic errors, Logical errors)	2
Elaboration	External materials	1
Evaluation	Activities, Testes	2
Total		9

iv. KNOWLEDGE ACQUISITION

Knowledge is the bottleneck of any intelligent system. Knowledge is acquired via experts and machine learning tool. Decision tree is one of the popular any easy tool for acquiring knowledge from existing databases. Decision tree is a way to discover knowledge from data that come from educational domain [14].

a. Knowledge Acquisition via Decision Tree

One of the main advantages of the decision tree is that it can be interpreted as a set of rules.

These rules are generated by traversing the tree starting from the root node till we reach some decision at a leaf. These rules also give a clear analytical view of the system under investigation [15].

Decision tree is one of the most popular models, because it is simple and easy to understand. It is flow- chart like tree structure, where each internal node denotes test on an attribute, each branch represents an outcome of the test, and leaf node represent class [16].

Given a data set of attributes together with its classes, a decision tree produces sequences of rules that can be used to recognize the classes for decision making.

This paper aims to find the performance of decision tree algorithms as well as the influence of qualitative factors in student's performance.

Entropy is the measure for the amount of information that is contained about a source of message [17].

It ranges from 0 to 1. When entropy is equal 1 means dataset is homogenous [16].

Entropy is calculated by formula:

$$Entropy(S) = \sum_{i=1}^c -P_i \log_2 P_i \quad (1)$$

Where, P_i is the probability of S belonging to class i.

Information gain measures the expected reduction in entropy. ID3 algorithm calculates the gain of all attributes, and selects the one with highest gain [16].

The gain is calculating by the following formula:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (2)$$

Where A is the set of all possible values for attribute A, and S_v is the subset of S for which the attribute A has value v. The attribute with highest information gain among the attributes is located as the root node in the decision tree [14].

The knowledge represented by decision tree can be extracted and represented in the form of IF-THEN rules [17].

The decision variable has one of the two possible values *Yes* (students who have Excellent Level), and *No* (students who have Low Level).

Sample of a dataset obtained from a student's model database is shown in table4.

Sample of the decision tree output rules are explained as follows:

- 1- IF Syntax error =Very High and Goals = Very High THEN Decision =Yes.
- 2- IF Syntax error =Very High and Goals = Low THEN Decision =No.
- 3- IF Semantic Errors = Very High and Syntax error =High THEN Decision =Yes.
- 4- IF Semantic Errors = Low and Syntax error =High THEN Decision =No.
- 5- IF Logical Errors = Very High and Syntax error =Medium THEN Decision =Yes.
- 6- IF Logical Errors = Medium and Syntax error =Medium THEN Decision =No.
- 7- IF Subjects = Very High and Syntax error =Very High THEN Decision =Yes.
- 8- IF Subjects = Medium and Syntax error =Very High THEN Decision =Yes.
- 9- IF Subjects = Low and Syntax error =Very High THEN Decision =No.

- 10- IF Net beans =Very High and Syntax error = Very High THEN Decision =Yes.
- 11- IF Net beans =Low and Syntax error = Very High THEN Decision =No.
- 12- IF External Materials =Medium and Syntax error =High THEN Decision =No.
- 13- IF External Materials =Low and Syntax error =High THEN Decision =No.
- 14- IF Syntax error =Medium and Internet = Very High THEN Decision =Yes.
- 15- IF Syntax error =Medium and E-mail = Very High THEN Decision =Yes.
- 16- IF Syntax error =Medium and E-mail = Low THEN Decision =No.

b. Rules Pruning

The quality of a rule can be improved by a pruning step. This step removes the irrelevant terms from the rule antecedent.

As a result, pruning can improve both the predictive accuracy and the comprehensibility of the rule [18].

Sample of rules after pruning can be explained as follow;

- 1- IF Semantic Errors = Medium or Low and Syntax error =High THEN Decision =No.
 - 2- IF Logical Errors = Medium or Low and Syntax error =Medium THEN Decision =No.
 - 3- IF Goals = Low and Syntax error =Very High THEN Decision =No.
 - 4- IF Subjects = Low and Syntax error =Very High THEN Decision =No.
 - 5- IF Net beans =Low and Syntax error = Very High THEN Decision =No.
 - 6- IF External Materials =Medium or Low and Syntax error =High THEN Decision =No.
 - 7- IF Syntax error =Medium and Internet = Medium or Low THEN Decision =No.
- IF Syntax error =Medium and E-mail = High or Medium or Low THEN Decision =No.

TABLE 4. Sample of student model features vectors

Id	Student attributes (VH very high- H high- M medium- L low)											Student Evaluation (Targets)			
	syntax	semantic	logical	time	goals	subjects	Activities	External material	Net beans	E-mail	internet	Y=VH N for others	Y=H N for others	Y=M N for others	Y=L N for others
1	H	VH	VH	H	L	H	H	M	H	M	L	N	N	N	Y
2	VH	VH	M	M	L	M	M	M	M	VH	M	N	N	N	Y
3	H	M	H	H	M	L	VH	H	VH	VH	L	N	N	Y	N
4	VH	VH	M	H	H	H	M	H	M	H	H	N	Y	N	N
5	VH	VH	VH	VH	VH	VH	VH	VH	VH	H	H	Y	N	N	N
6	VH	M	H	L	M	VH	VH	L	VH	M	VH	N	N	Y	N
7	VH	VH	M	H	VH	M	H	VH	H	L	H	Y	N	N	N
8	M	M	H	M	H	M	M	VH	M	H	L	N	Y	N	N
9	VH	H	M	M	M	M	M	H	M	L	L	N	N	Y	N
10	VH	L	H	M	L	H	H	M	H	H	VH	N	N	N	Y
11	VH	VH	H	L	VH	H	H	H	H	M	VH	Y	N	N	N
12	VH	L	VH	H	H	H	H	M	H	VH	L	N	Y	N	N
13	VH	VH	M	VH	H	H	M	VH	M	H	M	N	Y	N	N
14	L	VH	H	H	H	M	L	VH	L	M	L	N	Y	N	N
15	H	L	VH	M	M	L	M	H	M	M	M	N	N	Y	N
16	VH	VH	L	L	M	H	M	H	M	M	M	N	N	Y	N
17	VH	H	H	L	VH	VH	H	M	H	L	L	Y	N	N	N

v. ADVISORY SYSTEM

Advisory system provides the advices and assists for solving problems that are normally solved by human experts. They can be classified as a type of expert system [19].

These systems are constructed by eliciting knowledge from human experts and coding it into a form that can be used by a computer in the evaluation of alternative solutions to problems within that domain of expertise [20].

a. Student performance via ANN prediction

Artificial Neural Network (ANN) is mathematical representations inspired by the functioning of the human brain [21].

Although ANN can generally predict better than decision trees for pattern classification problems, ANN are often regarded as black boxes since their predictions cannot be explained clearly like those of decision trees [22].

Learning algorithm is a technique to identify a model that best fits the relationship between the attribute set and class label of the input data.

The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before [23].

A training set consisting of records whose class labels are known must be provided. The training set is used to build a classification model, which is subsequently applied to the test set, which consists of records with unknown class labels [24].

The aim is to train the network to achieve a balance between the ability to respond correctly to the input

datasets that are used for training and the ability to provide good responses to the input that were similar [21].

The ANN structure is shown in figure3.

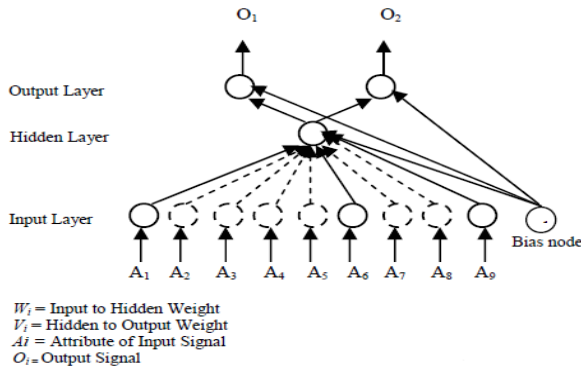


Figure. 3 Artificial Neural Network structure

The new student record is defined as a query input the trained ANN. Consequently an advice can be given to the student to assist his/her performance in the e-course.

b. The Advisory System Description

An advisory system for errors detection and correction in java is shown in figure4. It contains examples of the three types of errors; syntax, semantic and logical errors. The purpose of this system; is to detect errors and assist student to overcome them.

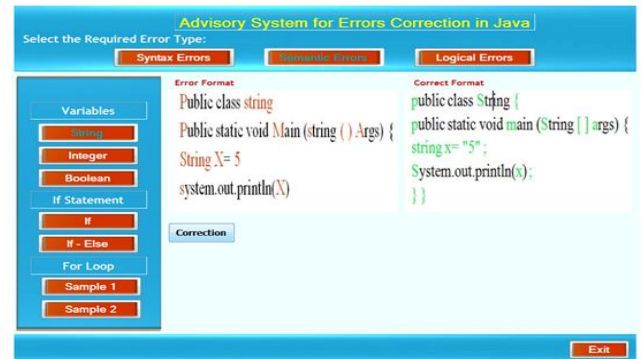


Figure.4 Advisory system for errors correction in java

vi. APPLICATION & RESULTS

The experimental study in the electronic program was consisted of two groups; experimental and control group. Each group includes 40 students, a pre and post tests were made. The results were analyzed using t-test to assure the efficiency of the suggested program.

The statistical results were obtained by running t-test with the SPSS software package.

Tables 5 and 6 compare the students' achievement before and after they used the suggested electronic program. The average scores received by the experimental group (10.5250) are apparently better than the average of the control group (7.2500).

The effect size columns in tables 5 and 6 show that:-
 The total effect value of the experimental group is 98.2%.

The total effect value of the control group is 86.1%.
 Table 7 shows that the mean value of the experimental group is 37.825 and the mean value of the control group is 22.4 at significance level 0.001.

Consequently; the experimental group is higher than the control group. This proved the effectiveness of the proposed system.

Table 5. Significance of differences between the mean of the experimental group in the pre and post tests

Type of error	Test	N	Mean	Standard deviation	t	df	Significance	Effect size
Syntax errors	Pre test	40	2.8750	1.89720	31.385	39	0.001	96.1 %
	Post test	40	14.0500	1.76795				
Semantic Errors	Pre test	40	3.5500	2.09945	21.657	39	0.001	92.3 %
	Post test	40	11.1750	1.27877				
Logical Errors	Pre test	40	4.1000	2.46826	19.514	39	0.001	90.7 %
	Post test	40	12.6000	2.16972				
Total	Pre test	40	10.5250	3.80275	46.644	39	0.001	98.2 %
	Post test	40		1.69293				

Table 6. Significance of differences between the mean of the control group in the pre and post tests

Type of error	Test	N	Mean	Standard deviation	t	df	Significance	Effect size
Syntax errors	Pre test	40	2.8250	1.93334	9.375	39	0.001	69.2 %
	Post test	40	6.6750	1.93996				
Semantic Errors	Pre test	40	3.4250	2.06171	7.987	39	0.001	62 %
	Post test	40	7.3500	2.14297				
Logical Errors	Pre test	40	4.0000	2.42846	8.647	39	0.001	65.7 %
	Post test	40	8.3750	2.24964				
Total	Pre test	40	7.2500	4.65061	15.561	39	0.001	86.1 %
	Post test	40	7.2500	3.66410				

Table 7. Significance of differences between the mean scores of the control and experimental groups in the posttest

Type of error	Group	N	Mean	Standard deviation	t	df	Significance	Effect size
Syntax errors	Control group	40	6.6750	1.93996	17.771	78	0.001	80.1 %
	Experimental group	40	14.0500	1.76795				
Semantic Errors	Control group	40	7.3500	2.14297	9.694	78	0.001	54.6 %
	Experimental group	40	11.1750	1.27877				
Logical Errors	Control group	40	8.3750	2.24964	8.550	78	0.001	48.3 %
	Experimental group	40	12.6000	2.16972				
Total	Control group	40	22.4000	3.66410	24.170	78	0.001	88.2 %
	Experimental group	40	37.8250	1.69293				

vii. CONCLUSIONS AND FUTURE WORK

A framework for student assessment in the programming language error detection and correction is presented. The knowledge acquisition recourses used are domain expert, literature review, student diagnostic test and course analysis.

The e- course presented includes the integration of the proposed student model, the knowledge base and the advisory system.

The proposed intelligent system (IS) has proven that the student performance is increased. This is mainly due to the knowledge base and the advisory system which are included in the IS. They contain the human experience in addition to the current and previous history of the student.

The future work is to utilize video proceeding and image recognition for more attractive e-course.

REFERENCES

- [1] Ioana Tuugalei Chan Mow: "Conceptualization of an effective learning environment for programming through an analysis of the underlying research issues in teaching programming", International Conference on Internet Computing, Informatics in E-Business and applied Computing, 27 – 28 July, 2012.
- [2] Marzieh Ahmadzadeh, Dave Elliman and Colin Higgins: "An Analysis of Patterns of Debugging Among Novice Computer Science Students", ITiCSE'05, June 27–29, 2005.
- [3] Anthony Robins, Janet Rountree and Nathan Rountree: "Learning and Teaching Programming: A Review and Discussion", Computer Science Education, VOL.13, NO.2, PP.137–172, 2003.
- [4] Rex P. Bringula, Geecee Maybelline A. Manabat and Miguel Angelo A. Tolentino: " Predictors of Errors of Novice Java Programmers", World Journal of Education, Available at: www.sciedu.ca/wje, Vol. 2, No. 1; February 2012.
- [5] Ioana Tuugalei Chan Mow, " Analyses of Student Programming Errors In Java Programming Courses", Journal of Emerging Trends in Computing and Information Sciences, VOL. 3, NO. 5, May 2012.
- [6] Maria Hristova, Ananya Misra, Megan Rutter, and Rebecca Mercuri, Ph.D: " Identifying and Correcting Java Programming Errors for Introductory Computer Science Students", SIGCSE '03, February 19-23, 2003.
- [7] Alejandro Peña Ayala: " Student model based on psychological models", World Conference on Educational Sciences 2009.
- [8] Judy Kay: " Learner Knowthy self: Student Models to Give Learner Control and Responsibility", Available at: judy@cs.su.oz.au, AUSTRALIA, 2006.
- [9] Ashraf A. Kassim, Sabbir Ahmed Kazi and SURENDRA Ranganath: " A Web-based Intelligent Learning Environment for Digital Systems", Int. J. Engng Ed. Vol. 20, No. 1, pp. 13–23, 2004.
- [10] Dongming Xu, Huaqing Wang and Kaile Su: "Intelligent Student Profiling with Fuzzy Models", Proceedings of the 35th Hawaii International Conference on System Sciences, 2002.
- [11] MEGHANN A. CAMPBELL: "The Effects of the 5E Learning Cycle Model on Students Understanding of Force and Motion Concepts", B.S. Millersville University, 2000.
- [12] Rodger W. Bybee, et al.: "The BSCS 5E Instructional Model: Origins, Effectiveness, and Applications", BSCS Available at www.bsccs.org, July, 2006.
- [13] Porandokht Fazelian, Abdolrarim Naveh ebrahim and Saeed Soraghi: "The effect of 5E instructional design model on learning and retention of sciences for middle class students", Procedia Social and Behavioral Sciences 5, 140–143, 2010.
- [14] Abeer Badr El Din Ahmed, Ibrahim Sayed Elaraby: " Data Mining: A prediction for Student's Performance Using Classification Method", World Journal of Computer Application and Technology 2, 43-47, 2014.

- [15] Abdul Fattah Mashat, Mohammed M. Fouad, Philip S. Yu, Tarek F. Gharib: "A Decision Tree Classification Model for University Admission System", International Journal of Advanced Computer Science and Applications, Vol. 3, No. 10, 2012.
- [16] Sweta Rai, Priyanka Saini and Ajit Kumar Jain: " Model for Prediction of Dropout Student Using ID3 Decision Tree Algorithm", International Journal of Advanced Research in Computer Science & Technology, Vol. 2 Issue 1 Ver 2 Jan-March 2014.
- [17] T. Miranda Lakshmi, A. Martin, R. Mumtaj Begum and Dr. V. Prasanna Venkatesan: "An Analysis on Performance of Decision Tree Algorithms using Student's Qualitative Data", I.J. Modern Education and Computer Science, 5, 18-27, 2013.
- [18] S. M. Kamruzzaman¹ and Md. Monirul Islam²: "An Algorithm to Extract Rules from Artificial Neural Networks for Medical Diagnosis Problems", International Journal of Information Technology, Vol. 12 No. 8, 2006.
- [19] Brandon A. Beemer, Dawn G. Gregg: "Advisory System to Support Decision making", International Hand Books Information System, pp. 511-527, 2008.
- [20] A. E. E. ElAlfi, M. E. ElAlami: "Intelligent Advisory System for Supporting University Managers in Law", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 3, No. 1, 2009.
- [21] Bart Baesens, Rudy Setiono, Christophe Mues and Jan Vanthienen: " Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation", Management Science INFORMS, Vol. 49, No. 3, pp. 312-329, March 2003.
- [22] Allen Chan and Alex Freitas: "A New Classification-Rule Pruning Procedure for an Ant Colony Algorithm", E. Talbi et al. (Eds.), pp. 25 - 36, 2006.
- [23] Amit Choudhary and Rahul Rishi: "Improving the Character Recognition Efficiency of Feed forward Bp Neural Network", International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 1, Feb 2011.
- [24] Bart Baesens, et al. : " Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation", Management Science © 2003 INFORMS, Vol. 49, No. 3, pp. 312-329, March 2003.