Available online at www.scientiaresearchlibrary.com

Scientia Research Library **ISSN 2348-0424**
USA CODEN: JETRB4

Journal of Engineering And Technology Research, 2018, 6 (2):1-12

(*http://www.scientiaresearchlibrary.com/arhcive.php*)

Development and Implementation of an Intelligent Tutoring System Using Bayesian Network in Teaching C# Programming Language

Juliet V. Menor

Pangasinan State University, San Carlos Campus
Roxas Blvd., San Carlos City Pangasinan 2420, Philippines
Corresponding Author : psujulietmenor@gmail.com

ABSTRACT

The utilization of computer and educational technologies paved to the implementation of an intelligent tutoring system to proactively assist the student in their learning process, especially once a learning difficulty needs to be remediated. This study aims to develop an intelligent tutoring system that dynamically identifies the learning difficulty then employs Bayesian network to help students. Student's responses to the programming questions or exercises were stored and collected as images running screenshots and converted into numerical weight as an input to the Bayesian. Control and an experimental group composed of 100 students were used in the study. Pre-test and post-test results were analyzed using standard deviation and statistical correlation. The results of mean scores and standard deviation show that there is a significant difference between the control and experimental group. Correlation results show no significant relation. Data from diagnostic and post-assessment is a highly significant difference based on academic performance, skill acquisition and problem solving. The experimental group performs better, an indication that the intelligent tutoring system with remediation is better than the control group. Based on the results, an intelligent tutoring system helps the students while doing the actual programming by employing the Bayesian network.

Keywords: Bayesian Network, Artificial, Intelligent, Programming Language, Academic Performance

---

INTRODUCTION

The utilization of computer and information technologies and intelligent tutoring systems is becoming more famous and widely spread throughout the world. It is an indication that anyone could learn at any place and anytime. There are teaching questions in some tutoring systems that students' responses cannot understand in time. Thus, it is vital to develop an intelligent tutoring system (ITS) with the end objective of giving learning and support services to students for better understanding and learning. intended to give individual feedback and support to students who are working on problems.

The difficulties in programming evolved from issues such as IDEs are troublesome, misunderstanding of memory operations, abstract nature without a proper foundation in programming, and misconceptions between a class and an object. However, there are some other issues students had underestimated and described as not at all difficult in their learning of C# programming. The students, potential learning difficulties encountered in learning in C# programming is the issue concerning Class-Object and the relationship between them.

An identified potential problem of student self-assessment about learning C# programming cannot understand the operation inside the computer when it executes a program such as lacks knowledge about memory operations. Besides, insufficient proper helping tools and reference materials; and difficulties in reading someone else code, testing and debugging of applications, and detecting of logic errors. With all these motivations, the researcher developed a system and came up with an Intelligent Tutoring System in C# Programming Language using Bayesian network.

OBJECTIVES OF THE STUDY

The objectives of this study are (1) Identify the learning difficulties: (2) Design and develop Intelligent Tutoring System using Bayesian network; and (3) Significant difference between control group and experimental group in terms of academic performance, skill acquisition and problem solving.

MATERIALS AND METHOD

The method and step presented will guide the researcher in completing the input, process, and output phases for developing of an Intelligent Tutoring System using Bayesian network. To assess the effectiveness of the system to be developed, and testing plan devised. These will guide the researcher in evaluating the functionality of the Intelligent Tutoring System in its final phase.

Data

To test the effectiveness of the course tool, which is an Intelligent Tutoring System (ITS), an experimental application process conducted at Pangasinan State University, San Carlos Campus. In the context of the application process, students from online or web-based programs of the Pangasinan State University taken an active part within the related groups chosen for a typical experimental approach. In this sense, some remarkable points about the experimental application process expressed.

In the experimental application process, 50 BSIT students composed of an experimental group, whereas another group of 50 BSIT students formed the control group. Thus, a total of 100 students taken in the preliminary application process. The distribution of the respondents shown in Table 1.

Table 1. Distribution of the Respondents

Group	No. of Students	Age Distribution	Program
Experimental	Male: 22 50	17-20, 20-23	BSIT
	Female: 28	17-20, 20-23	
Control	Male: 25 50	17-20, 20-23	BSIT
	Female; 25	17-20, 20-23	

On the other hand, the control group took traditional sessions formed by theoretical and applied to teach approaches provided by the teacher (courses related to C# programming language performed via classroom discussions). The whole process continued along with a term in which the "C# Programming" course given to improve students' knowledge and ability levels about the computer programming approach and the related C# programming language.

Methodologies are needed to determine the viability of the developed system. These methodologies should measure the development and implementation of an Intelligent Tutoring System (ITS) using Bayesian network in terms of learning difficulty, design an intelligent tutoring system using Bayesian network, and significant difference between the control group and experimental group. The following statements detailed methodologies used for specific objectives.

Identify the learning difficulty

The student probability of errors is conditional through an expected set of Bayesian reasoning using the related evidence used in this study and described in the following formula:

$$\boldsymbol{P(E)} = \frac{\textit{Expected number of times it } \quad s}{\textit{correct expected number of times it } \quad s \textit{ true} + \textit{Expected number of times it's false}}$$

$P(E) = 0$

$$P(E) = \frac{n(E)}{n(E) + n(F)}$$

and

$$P(F) = 1 - P(E)$$

where:

$P(E) - probability\ of\ difficulties$
$n(E) - no.\ of\ difficulties$
$n(F) - no.\ of\ no\ difficulties$

The equation shows above the probability of errors is equal to the expected number of times its right divide by the expected number of times it's the real plus expected number of times it's false.

Develop and design the System Architecture ITS

The system architecture composed of ITS components, as shown in (Figure 1), namely: (a) interface; (b) Student module; (c) Tutor module; and (d) Knowledge module. The internal structure of these modules is independent of the tutor specific content, thus equal for all the ITS implemented using this approach. The Interface is the part of the Intelligent Tutoring System (ITS), where the variables defined in user Inference kept. This module retains user information from the interaction with the tutor. The content subjects taught held in the knowledge domain structured as a knowledge map and a set of topics. These are the unique module in a working ITS which is dependent on the contents, producing the desired separation between the domain application dependent part and the generic part. User assessment performed in the evaluation, which combines information from the knowledge module with user data (from the user interface) in a Bayesian, inferring the user state of knowledge. The tutor module also supported by the said algorithm.

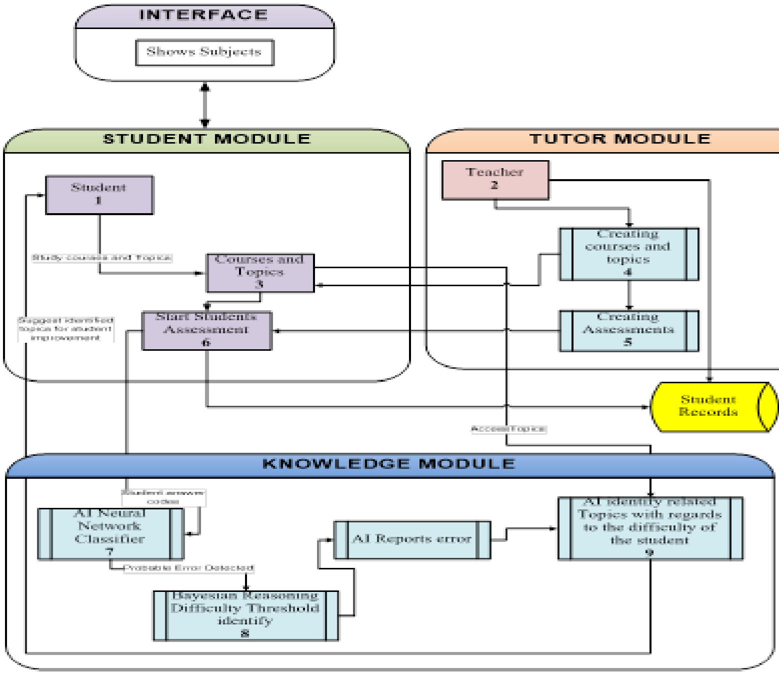

Figure 1. System Architecture

The result of the Intelligent Tutoring System (ITS) reasoning process is the page composed and revealed in the evaluation module. It collects the topic and several questions to present, the tutoring suggestions, and the most reasonable step to be taken. This architecture serves to isolate the different parts of the system, defining their functionalities, and specifying the implementation directions.

Interface

The interface is essential it designed in a manner that attracts students' interest and should encourage students' long-term retention of the concepts through the vivid animations of the user interface. Thus, the layout of the user interface given much consideration here — a learning style questionnaire conducted before the design of the ITS model.

Student Module

Before a new student can use the C# intelligent tutoring system, he/she must register first. The student information contains such as first name, last name, username and password, student name. Moreover, the current score, overall score, level difficulty completed for every lesson, and rating during each session. The current rating represents student scores for the current level. The total score represents student scores for all levels.

Tutor Module

Tutor module also called tutoring module, works as a coordinator that controls the functionality of the C# programming language intelligent tutoring System. Through this model, a student can answer questions generated in every difficulty level of each lesson. If the student gets more than a 75 percent mark, he/she can move to the second difficulty level. Otherwise, if he/she got a score between 50 percent and less than 75 percent then he/she repeats to the assessment of the same difficulty level; however, if he/she gets a mark less than 50 percent, he/she will be taken back to lesson to study it well then come back to try the assessment again.

Knowledge Module

This component is sometimes called the knowledge module of other ITS architectures. It contains the material to be taught for the students. In our current tutoring system, the content consists of the following topics in C# programming based on the syllabus.

Difference between pre-test and post-test

This study adopted a pre-test and post-test control group experimental design. The design of this study presented in Table 2 and 3. In determining the efficiency of ITS this study involves dependent variables which include two different combinations of delivery systems, online ITS, and face-face lecture. The dependent variable is the students' assessment in C# programming. The researcher conducted an evaluation last January 23 to 24, 2020. On the 23rd day of January, the researcher administers the 50 BSIT 3rd-year students represented as the control group, whereas, on the 24th day of January, another 50 BSIT 3rd year students for the experimental group both groups randomly assigned and those who enrolled the said course.

RESULTS AND DISCUSSION

Learning Difficulty

C# is a complex language, and it can be challenging to learn Object-Oriented Programming when a student first starts. There are so many rules and standards which bound to make mistakes but learning from the mistakes and avoiding these common ones help become a better coder and make the programs much more efficient. Based on the encountered student's difficulties, the following difficulties namely, keyboard mistype, an undeclared variable, operator misuse, data type misuse, runtime error, and unhandled exception. Shows on the figures below the percentage of student difficulties.

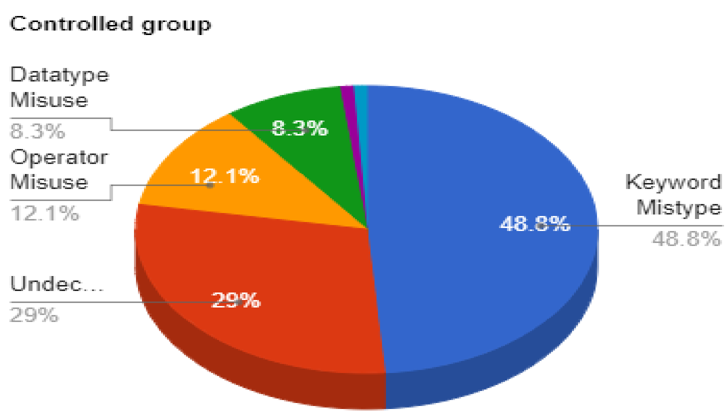

Figure 2. Control Group with Difficulty and Percentage

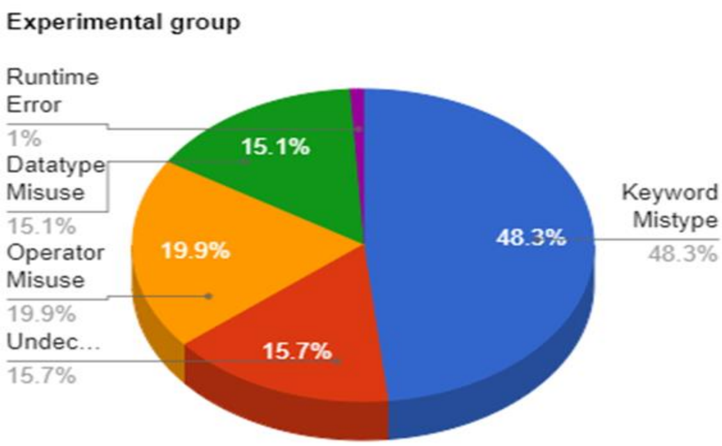

Figure 3. Experimental Group with Difficulty and Percentage

Figure 2 revealed the percentage of students for control and experimental group encountered difficulties during the assessments. In control group the keyword mistype has 48.8% students identified the difficulty. In contrast, the undeclared variable had 29% students, whereas an operator misuse had 12.1% students, although the data type misuse has 8.3% students. However, the unhandled exception has 0, and to end runtime error has 0 of student encountered the difficulty.

Figure 3 shows the keyword mistype has 48.3% students identified the difficulty. In contrast, the undeclared variable had 15.7% students, whereas an operator misuse had 19.9% students, although the data type misuse has 15.1% students. However, the unhandled exception has 1%, and to end runtime error has 0% of student encountered the difficulty. The results showed above are the assessment representation of pie chart for control and experimental group of students who acquired the assessment.

ITS using Bayesian Network

ALGORITHM OF BAYESIAN FUNCTION

```
// Threshold is the initialized value to be compared to the resulting probability of the Bayesian reasoning
// Encoded Errors are the probable real-time errors that the compiler identified
// Lines with No Error are the number of lines that didn't contain code that possibly didn't have mistakes
//Parse Identify is the array of possible errors that are related to the detected compiler errors which are the encoded Errors

public function
studentHintBool($threshold,$encodedErrors,$linesWithNoError,$parseIdentify)
{
//COMPUTING THE PROBABILITY OF THE STUDENT DIFFICULTY
  $probability=0;
  $errorsArray=explode("|",$encodedErrors); //Decode array of detected errors
  $errorCounter=0; //Error counter
    $x=0;
    for($x=0;$x<count($errorsArray);$x++) //Loop through all the compiler errors detected
        {
        if(in_array($errorsArray[$x], $parseIdentify)//Compare the detected compiler errors to the possible
        related errors that the AI has in its classifier
          {
          $errorCounter++; //Increment counter if it is within the identified errors that are in the datasets that the AI detected
          }
       }
   $probability=count($errorsArray)/($errorCounter+$linesWithNoError);//Compute probability of
   difficulty
        if($probability>$threshold) //Check if the result probability is above the difficulty threshold
          {
          return "true";  //Student is probably experiencing difficulty thus trigger assistance
          }
        else
          {
```

Figure 8. Bayesian Network Algorithm

As gleaned from Figure 8, the pseudo-codes comprised of Bayesian function specifically designed to generate a message from artificial intelligence formulated. The purpose of an algorithm is to detect difficulties of students with the presence of AI, suggest and guide the student to overcome the learning difficulties. For example, if the student takes an assessment exam once overcome the problem, the compiler errors to the related errors. The AI has its classifier, such as run time error, syntax error, and logic error classified the error found. At the same time, the neural network algorithm returned the results of error, which is the Bayesian function.

In terms of Academic Performance

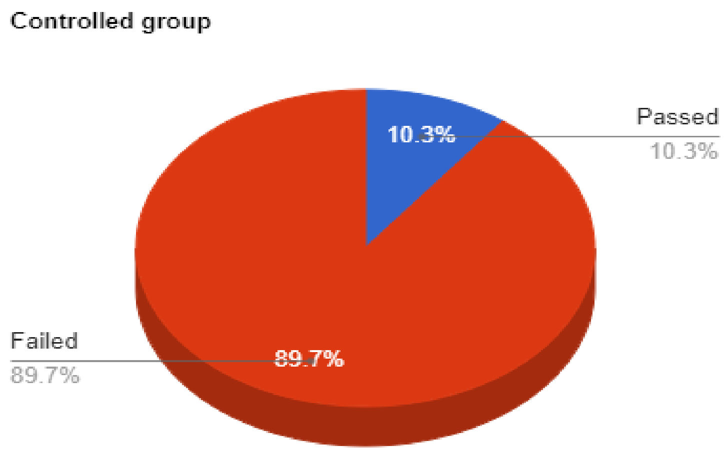

Figure 9. Academic percentage performance of the control group

Figure 9 shows the pre and post-test, which were administered to the students using the performance test. The figure above revealed 10.3% passed and 89.7% failed was extracted from the system based on the percentage performance of the students. The statistical calculations made on the percentage of students in line with the average.

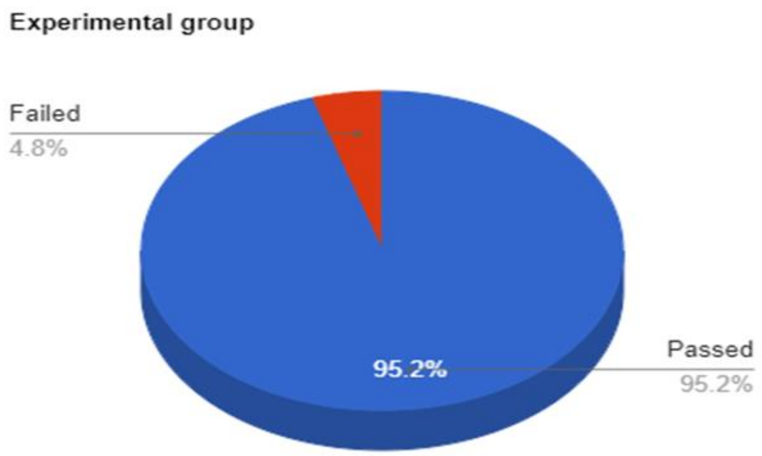

Figure 10. Academic percentage performance of the control group

Skills Acquisition

Table 2a and 2b illustrated the summary of students before diagnostic exams, and after assessment exams of the 100 BSIT students who took up the reviews, the results of their skill acquisition during the diagnostic and assessment procedure were observed. The evaluation of both groups was considering that the experimental group has 9 percent errors. Likewise, the control group has 13 percent errors, which is higher than the experimental group. Hence, the results have a significant difference between the two groups the statistical treatment shown in Table 4-7.

Table 2a. Before the Diagnostic Exam and After Assessment Exam

No. of Students	Keyword Mistype	Undeclared Variable	Operator Misuse	Datatype Misuse	Runtime Error	Unhandled Exception	Total No. of Errors	No. of Students	Keyword Mistype	Undeclared Variable	Operator Misuse	Datatype Misuse	Runtime Error	Unhandled Exception	Total No. of Errors
	Diagnostic Exam								Assessment Exam						
1	16	14	8	6	1	1	46	51	11	6	6	5	0	0	28
2	13	10	3	7	0	0	33	52	5	5	3	0	0	0	13
3	16	9	0	5	0	0	30	53	15	6	0	4	0	0	25
4	14	15	3	0	0	0	32	54	6	6	4	3	0	0	19
5	21	9	7	7	0	0	44	55	8	1	3	4	0	0	16
6	19	3	5	0	0	0	27	56	8	0	7	4	0	0	19
7	15	7	7	0	0	0	29	57	13	1	3	1	0	0	18
8	14	8	6	0	4	1	33	58	8	4	6	5	0	0	23
9	10	13	3	0	0	0	26	59	8	6	3	3	0	0	20
10	10	10	7	4	0	0	31	60	14	2	7	0	0	0	23
11	17	7	6	1	0	0	31	61	10	0	8	2	0	0	20
12	13	5	7	0	0	0	25	62	5	1	5	1	0	0	12
13	8	10	2	1	0	0	21	63	10	2	2	3	0	0	17
14	11	13	11	0	0	2	37	64	8	3	7	3	0	0	21
15	9	4	2	0	0	1	16	65	6	0	2	2	0	0	10
16	13	5	5	0	0	1	24	66	9	0	0	0	0	0	9
17	10	11	12	1	0	0	34	67	7	2	8	4	0	0	21
18	9	6	2	3	0	0	20	68	1	6	2	4	0	0	13
19	17	3	4	0	0	0	24	69	13	0	4	1	0	0	18
20	14	4	5	5	0	1	29	70	7	4	5	4	0	0	20
21	11	0	2	4	0	1	18	71	9	0	0	1	0	0	10
22	16	9	0	5	0	0	30	72	9	4	1	7	0	0	21
23	14	8	6	1	0	0	29	73	9	0	9	1	0	0	19
24	12	8	0	7	0	0	27	74	6	0	3	2	0	0	11
25	14	16	1	6	0	0	37	75	6	4	3	3	0	0	16
26	9	7	1	6	0	0	23	76	6	4	6	7	0	0	23
27	12	9	0	2	0	0	23	77	7	0	2	1	0	0	10
28	14	4	0	1	0	0	19	78	5	4	0	2	0	0	11
29	11	5	5	3	0	0	24	79	7	0	9	4	0	0	20
30	7	6	1	1	0	0	15	80	7	4	4	2	0	0	17
31	8	4	1	1	0	0	14	81	7	1	2	2	0	0	12
32	19	4	0	1	0	1	25	82	8	4	2	2	0	0	16
33	11	6	1	5	0	0	23	83	8	1	0	2	0	0	11
34	19	11	0	3	3	0	36	84	7	3	0	4	3	0	17
35	12	9	3	0	2	0	26	85	11	4	7	0	3	0	25
36	17	8	2	0	0	1	28	86	5	4	5	0	0	0	14
37	13	2	2	3	0	0	20	87	13	2	2	9	0	0	26
38	12	7	0	3	0	0	22	88	11	5	0	5	0	0	21

Table 2b. Before the Diagnostic Exam and After Assessment Exam

No. of Students	Keyword Mistype	Undeclared Variable	Operator Misuse	Datatype Misuse	Runtime Error	Unhandled Exception	Total No. of Errors	No. of Students	Keyword Mistype	Undeclared Variable	Operator Misuse	Datatype Misuse	Runtime Error	Unhandled Exception	Total No. of Errors
	Diagnostic Exam								Assessment Exam						
39	12	12	0	0	0	1	25	89	8	2	0	2	0	0	12
40	15	8	2	0	0	0	25	90	14	8	2	1	0	0	25
41	12	11	3	1	0	0	27	91	8	9	3	1	0	0	21
42	12	6	2	1	0	0	21	92	10	4	2	1	0	0	17

43	19	5	4	7	0	0	35	93	7	4	5	4	0	0	20
44	11	2	3	1	0	0	17	94	4	2	3	1	0	0	10
45	19	2	2	0	0	0	23	95	14	2	2	0	0	0	18
46	8	8	2	1	0	1	20	96	2	3	2	1	0	0	8
47	15	6	3	0	0	0	24	97	11	3	3	0	0	0	17
48	15	7	7	0	0	0	29	98	13	2	3	0	0	0	18
49	8	7	5	8	0	0	28	99	8	2	0	3	0	0	13
50	15	13	6	2	0	0	36	100	13	6	4	0	0	0	23
Percentage of Errors							13%	Percentage of Errors							9%

Problem Solving

Tables 3a and 3b, shows the summary of control and experimental assessment results of problem-solving and presented below the diagnostic assessment from the control group. The researcher was administered 50 BSIT students who took up the traditional. However, another 50 BSIT students from the experimental group which utilized the developed system. The comparison of the mean between the control and experimental groups revealed in Table 4-7.

Table 3a. Summary of Grades Control and Experimental Group

No. of Students	Diagnostic 1-5					No. of Students	Assessment 1-5				
1	75	75	70	75	75	51	80	80	75	100	100
2	75	75	70	75	75	52	90	80	75	100	100
3	75	75	70	75	75	53	90	80	75	100	90
4	70	75	70	75	75	54	90	90	75	100	90
5	70	75	75	75	75	55	90	90	75	100	90
6	70	75	75	75	75	56	100	90	80	90	100
7	70	70	75	75	70	57	100	90	80	90	100
8	70	70	75	70	70	58	100	80	80	90	100
9	70	70	75	70	70	59	100	90	90	90	100
10	70	70	75	70	70	60	90	90	90	100	90
11	75	55	75	70	55	61	90	80	100	100	90
12	75	70	55	75	55	62	100	80	75	100	90
13	70	70	55	75	55	63	100	90	80	100	80
14	70	70	55	75	55	64	90	90	80	90	80
15	70	55	75	75	75	65	90	90	80	90	80
16	70	55	75	55	75	66	90	80	75	90	80
17	70	55	75	55	75	67	100	90	75	90	75
18	70	90	75	55	80	68	100	80	90	80	75
19	70	70	70	55	80	69	90	90	90	80	75

Table 3b. Summary of Grades Control and Experimental Group

No. of Students	Diagnostic 1-5					No. of Students	Assessment 1-5				
20	70	70	70	75	80	70	90	80	90	80	75
21	70	70	70	75	80	71	90	90	100	75	100
22	70	55	70	75	75	72	80	100	100	75	90
23	70	55	70	70	70	73	80	100	90	75	90
24	70	70	55	70	75	74	100	90	90	80	90
25	55	70	75	70	70	75	80	90	90	75	70
26	70	70	70	70	75	76	80	100	90	80	100
27	55	70	75	75	70	77	80	100	100	90	80
28	55	70	70	75	80	78	80	90	100	90	80
29	55	75	75	55	80	79	100	90	100	90	80
30	70	75	80	55	80	80	100	100	90	100	75
31	80	75	80	55	70	81	90	100	90	100	80
32	75	75	80	55	70	82	90	80	90	90	75
33	75	70	90	80	75	83	100	90	90	90	80
34	75	70	70	80	75	84	100	90	100	80	90
35	75	70	70	80	75	85	90	80	100	90	90

36	75	70	70	80	75	86	90	90	90	80	90
37	75	80	70	80	70	87	90	90	90	90	90
38	70	80	75	80	70	88	100	90	90	80	100
39	70	80	75	80	70	89	100	80	90	75	80
40	70	80	70	75	55	90	100	100	100	75	80
41	70	70	75	75	55	91	90	100	100	75	80
42	70	70	70	75	55	92	90	100	100	100	75
43	55	70	75	70	55	93	90	100	90	80	80
44	55	55	70	70	70	94	100	80	90	80	75
45	55	80	75	70	70	95	100	80	80	80	75
46	55	80	70	55	75	96	90	80	90	75	80
47	55	80	75	55	70	97	100	90	90	100	75
48	55	55	55	55	75	98	100	90	80	100	80
49	55	55	55	55	70	99	90	90	80	90	90
50	55	55	55	55	80	100	90	75	75	90	90
Control Group						**Experimental Group**					

Statistical Treatment

Table 4 shows the mean scores and standard deviation of academic performance from the control and experimental group. The variables of the t-test analysis between the two (2) groups were completed. The significant difference between both groups obtained data and statistical calculations are shown in Table 6.

Table 4. Mean Score and Standard Deviation of Control and Experimental Group

Variable	Control Group Pre-test		Experimental Group Post Test	
	Mean	S.D.	Mean	S.D.
1. Academic Performance	3.95	0.86	7.70	0.64
2. Skill Acquisition	26.82	6.89	17.34	5.07
3. Problem Solving	69.76	4.31	88.52	3.18

Table 5depicted the result of the correlation between the control and the experimental group on significance at the .05 level. Furthermore, the degree of relationship is low. Hence, the test was reliable there is no significant relationship between the control and experimental group concluded.

Table 5. Correlation between the Control and Experimental Group

Variables	Correlations	Groups	Degree of Relationship
1. Academic Performance	Correlation	0.213	Low
	Sig.	0.137	
2. Skill Acquisition	Correlation	.464**	Moderate
	Sig.	0.001	
3. Problem Solving	Correlation	0.213	Low
	Sig.	0.137	

**. Correlation is significant at the 0.05 level (2-tailed).

Table 6 revealed the comparison of the results of the control group and the experimental group. The comparison and significance at the 0.05 level (2-tailed). Furthermore, the results indicated that there is a highly significant difference between the control group and the experimental group in terms of academic performance. The assessment was tested and proven using paired t-test statistical treatment. Shown below.

Table 6. Comparison of the Mean between the Control and Experimental Group

Variables	Groups	Mean	Mean Difference	T	Sig. (2-tailed)
1. Academic Performance	Control	3.952	-3.752	-27.76	0.000**
	Experimental	7.704			
2. Skill Acquisition	Control	26.820	9.480	10.50	0.000**
	Experimental	17.340			
3. Problem Solving	Control	69.760	-18.760	-27.76	0.000**
	Experimental	88.5200			

**. Significant at the 0.05 level (2-tailed).

CONCLUSION

Based on the findings, the following conclusions were drawn:

1. The system was able to identify the learning difficulty.
2. The system architecture of an Intelligent Tutoring System successfully modeled in the system prototype and Bayesian was effective accordingly in the Intelligent Tutoring System.
3. Results show that the system helps the students in their learning process and demonstrated from difference statistical and extracted results from the system, such as academic performance, skill acquisition and problem solving.

ACKNOWLEDGMENTS

The authors wish to thank the students of the Pangasinan State University for the success of the survey study as well as the constructive comments of the reviewers.

REFERENCE

[1] Delgado, A. J., Wardlow, L., McKnight, K., & O'Malley, K. (**2015**). Educational technology: A review of the integration, resources, and effectiveness of technology in K-12 classrooms. *Journal of Information Technology Education: Research,* 14, 397-416.

[2] Weragama, D., & Reyezs, J. (**2013**). The PHP Intelligent Tutoring System. In Artificial Intelligence in Education (pp. 583-586): Springer.

[3] Rivers, K., & Koedinger, K. R. (**2015**). Data-Driven Hint Generation in Vast Solution Spaces: a Self-Improving Python Programming Tutor. *International Journal of Artificial Intelligence in Education,* 1-28.

[4] Al-Hanjori, M. M., Shaath, M. Z., & Abu Naser, S. S. (**2017**). Learning computer networks using an intelligent tutoring system. International Journal of Advanced Research and Development (2), 1. system for students learning to program in C++. *Journal of Applied Sciences Research*, 5(1), 109-114.

[5] Bastami B. G, & Naser S. S. **(2017).** Design and Development of an Intelligent Tutoring ...

[6] Bashar G. H. Al-Bastami, Samy S. Abu Naser-Design ... Issue 10 / January **2017** 8798 Intelligent Tutoring.

[7] Chrysafiadi, K., & Virvou, M. (**2014**, July). KEM Cs: A set of student's characteristics for modeling in adaptive programming tutoring systems. In Information, Intelligence, Systems, and Applications, IISA 2014, The 5th International Conference on (pp. 106-110). IEEE.

[8] Lee, M. R., & Chen, T. T. (**2015**). Digital creativity: Research themes and framework. Computers in Human Behavior, 42, 12-19

[9] Malouf, A., Albatish, I., Mosa, M., & Abu Naser, S. S. (**2017**). Knowledge-Based System for Long-term Abdominal Pain (Stomach Pain) Diagnosis and Treatment. *International Journal of Engineering and Information Systems* (IJEAIS), 1(4), 71-88.

[10] Rivers, K., & Koedinger, K. R. (**2015**). Data-Driven Hint Generation in Vast Solution Spaces: a Self-Improving Python Programming Tutor. *International Journal of Artificial Intelligence in Education*, 1-28.

[11] Whip Q., Mendjoge N., and Sanghvi D., “Review of Knowledge Representation Techniques for Intelligent Tutoring System,” pp. 2508–2512, **2016**.