Open
Full Length Article

ORIGINAL ARTICLE

Fuzzy Logic-based Expert System for Assessing Programming Course Performance of E-Learning Students

Khubaib Khawar[1], Saima Munawar[2], Nasir Naveed[3]

[1, 2, 3] Department of Computer Science, Virtual University of Pakistan, Lahore

ABSTRACT

The collaboration between e-learning activities and those offering formative assessment and summative assessment has gained great attention in the educational community. These types of activities can be incorporated into online assessment tools. There is no doubt that all the big and small educational institutes use some kind of protocols and disciplines to assess the performance of regular and e-learning students conventionally or traditionally. This paper proposes and introduces an efficient, automatic and intelligent approach to calculate or assess the performance of e-learning students using an Expert System (ES) based on the Fuzzy Logic System (FLS) Artificial Intelligence Algorithm. The ES takes five parameters for formative assessment (FA) such as skype session attendance, internal course marks, and programming lab activity, and summative assessment (SA) includes mid-term and final-term marks. The result shows that the Expert System for Assessing Programming Course Performance (ESACP) evaluates the performance efficiently. The proposed model also has an impact on the educational institution's assessment system and focus on how to improve education standard with the help of assessing the students' performance using formative and summative analysis.

Keywords: Fuzzy Logic, Experts System, Artificial Intelligence, Machine Learning, Assessing Performance

Author`s Contribution
[1,2,3]Manuscript writing, Data Collection Data analysis, interpretation, Conception, synthesis, planning of research, and discussion

Address of Correspondence
Saima Munawar
Saima.munawar@vu.edu.pk

Article info.
Received: Sept 14, 2019
Accepted: May 23, 2020
Published: June 30,2020

Cite this article: *Khawar K, Munawar S, Naveed N.Fuzzy Logic based Expert System for Assessing Programming Course Performance of E-Learning Students. J. Inf. commun. technol. robot. appl.2020; 11(1):54-64*

Funding Source: *Nil*
Conflict of Interest: *Nil*

INTRODUCTION

The success or educational level of students in their academic years is assessed by their performance in the exams and other academic activities in the Universities. Keeping in view this fact that IT Industry is progressing quickly and there is a big gap in IT Industry and Educational Institutions, thus need of an automatic, intelligent and efficient system to calculate or assess the performance of the students of Computer Science or Computer Programming courses.

Formative assessment (FA) [1] is an assessment used in the classroom to attain information about students' progress. With this information, teachers can regulate and familiarize their teaching methodologies to improve learning and students are aware of their learning process [2]. The online tools can be used to carry out these learning assessment activities such as formative weekly quizzes or formative assessment exercises in labs [3, 4]. This type of assessment is differentiated from the interim assessment which is conducted several times during the instructional period to assess both the educational program and the on-going performance of students; and from summative assessment (SA) [5] which is performed at the end of

the instructional period to identify the student's learning level [6].

This paper discusses the process of development of such Expert System that was the sole purpose of this Project. It explains briefly about the methodologies and technologies that are used in the development of this ES. It will become the motivation to use this system to assess the overall performance of students of all courses in the educational institutes and universities.

The remainder of the paper is structured as follows. In Section 2, the background study is presented. In Section 3, the research methodology is presented. Section 4 delineates the results and discussion, while Section 5 concludes the paper and gives pointers to future work

LITERATURE REVIEW

The systematic review was conducted related to learning styles to adhere to the e-Learning student's problems. Amendment of e-learning content concerning several e-learning glitches is an open research issue in forward-facing of all of us. The authors discussed and reviewed the effectiveness of learning style and different classification methods in various e-learning snags from Elsevier, Springer, Wiley, PubMed resources [7]. The other reviewed work related to Feedback Provides for Students in E-Learning Systems. The authors selected papers from 2008 to 2019 based on IEEE Explore Digital Library and focused on personalized feedback [8]. The other authors also discussed challenges regarding work-in-progress through fuzzy sets that were explored as a measurement instrument [9].

In educational institutes, the traditional or conventional method is being used at a vast scale to calculate or assess the performance of students of a specific Computer Science (CS), Computer Programming (CP) or Software Engineering (SE) courses. In the traditional or conventional method, a teacher or instructor takes the final results of all academic activities of a student of a specific course and puts time in analyzing those results to come up with the linguistic value of the performance of that student in that specific course. If there are few students of one discipline that might be easy but if there are hundreds or thousands of students of more than one discipline that is almost impossible for one teacher or instructor to handle it. The author claims that it will require a lot of human effort and human resource to handle all of it manually and thus an automatic, intelligent and efficient system is required that can easily handle this situation [10]. The authors focused on computer science students specifically computer programming and suggested that it can enhance the academic Performance through explanatory Modeling [11].

The authors proposed an educational game that was FuzzEG and also claimed that it showed very encouraging results. It also determined the learner's knowledge level [12]. The authors focused on the fuzzy-based approach system that is more effective in a learning environment and it introduced intelligent e-learning system which was based on fuzzy logic for determining the student knowledge level and provide recommendation[13].

It also developed the system which uses fuzzy logic to assess the performance of calculation subjects of UNITEM Muadzam Shah Students, using MATLAB. They have focused on the performance value rather than what the usual methodology focuses. The system takes two inputs viz. Subject A and Subject B, and produces the output as a performance value. The research and developed system were used to assess the performance of twenty-two students, making the system successful [14]. Another proposed system to evaluate the activities of students in laboratory applications of engineering education. They proposed a new approach on students' grades, named as 'refinement processes, which use peer assessment, group assessment and personal assessment methods. Fuzzy helps more reliable decision making and evaluation of applications using a pre-determined list of criteria. This research and study show that fuzzy system can provide a better evaluation system than classical or traditional systems [15]. The authors also discussed the overall course outcome of Computer science courses using a fuzzy logic approach for evaluating and assessment [16]. Another paper discussed the learning modalities for Students' Interaction Activities with the implementation of Fuzzy

Logic and specifically discussed for learning computer programming [17]. The authors suggested that the combination of fuzzy and cognitive theories are more effective in adaptive e-assessment [18]. The authors also introduced a hybrid approach by the convolutional neural network (CNN) and a fuzzy system for facial expressions in an adaptive E-Learning environment. It has also monitored the learning performance of each learner[19].

FUZZY LOGIC: THEORETICAL REVIEW OF AN AI ALGORITHM

One of the classification methods is to use Fuzzy Logic Technique [20] to classify the subject [21]. The nature of FL is different from other methods. It is used when the answer to the classification question is more versatile than just having 'Yes' or 'No' or similar two variations. For example, 'How are you?' usually has two variations which are 'Good' and 'Not Good', but if the requirement is to keep it more versatile and to have more than two variations such as 'Very Bad', 'Bad', 'I don't know', 'Good', 'Very Good', then it becomes an application of FL. In Mathematics and Computer Science, the researchers have designed a special algorithm that follows the workflow to implement fuzzification and defuzzification, which are the core aspects of FL Methodology.

- **Defining Input and Output Variables (Crisp Values)**

The very first step in the FL Algorithm is to define the input variables that are going to be passed to the algorithm as input parameters. Example input variables are shown in Table 1.

Table 1. Example of Input Variables

Input Variables	Description
p	Input variable one
q	Input variable two
r	Input variable three

Next step is to define the output variable. Table 2 shows the example output variable.

Table 2: Example of Output Variable

Output Variable	Description
z	Output variable

Next step is to assign the numeric range (crisp values) to input and output variables. Table 3 shows the example range assigned to the defined variables.

Table 3. Example of Assigned Range to Defined Variables

Variables	Min Range Value	Max Range Value
P	1	10
Q	1	15
R	1	20
Z	1	10

- **Fuzzification**

Next step is to assign the linguistic words to the variables at the specific span of the range. Later, the triangular membership function is applied to each variable to get Fuzzification of the target variables [22]. Mathematically, triangular membership function is defined as below:

$$f(x,a,b,c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases}$$

It can also be defined in the other way as:

$$f(x,a,b,c) = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$$

Code Snippet I, II, III shows the Pseudo Code of assigning example linguistic words to input variables 'p', 'q' and 'r' respectively.

Code Snippet I: Example Pseudo Code of Assigning Linguistic Words to Input Variable One

```
1.  IF p is greater or equal to 1
    and p is smaller or equal to
    3.3
2.    THEN Set p to the string 'BAD'
3.      ELSE IF p is greater than 3.3
    and p is smaller than 6.6
4.          THEN Set p to the string
    'AVERAGE'
5.      ELSE
6.          Set p to the string
    'GOOD'
7.      END IF
```

Code Snippet II: Example Pseudo Code of Assigning Linguistic Words to Input Variable Two

```
1.    IF q is greater or equal to 1
   and q is smaller or equal to 5
2.       THEN Set q to the string
   'BAD'
3.    ELSE IF q is greater than 5
   and q is smaller than 10
4.       THEN Set q to the string
   'AVERAGE'
5.    ELSE
6.       Set q to the string 'GOOD'
7.    END IF
```

Code Snippet III: Example Pseudo Code of Assigning Linguistic Words to Input Variable Three

```
1. IF r is greater or equal to 1
   and r is smaller or equal to
   6.7
2.    THEN Set r to the string 'BAD'
3.    ELSE IF r is greater than 6.7
   and r is smaller than 13.4
4.       THEN Set r to the string
   'AVERAGE'
5.    ELSE
6.       Set r to the string 'GOOD'
7.    END IF
```

Now to fuzzify each variable, triangular membership function is applied.

- **Defuzzification**

The same triangular function is used for output variable 'z' as defuzzification [23]. Code Snippet IV shows the Pseudo Code of assigning example linguistic words to variable 'z'

Code Snippet IV: Example Pseudo Code of Assigning Linguistic Words to Output Variable

```
1. IF z is greater or equal to 1
   and z is smaller or equal to
   6.7
2.    THEN Set z to the string 'BAD'
3.    ELSE IF z is greater than 6.7
   and z is smaller than 13.4
4.       THEN Set z to the string
   'AVERAGE'
5.    ELSE
6.       Set z to the string 'GOOD'
7.    END IF
```

- **Defining the Fuzzy Inference Rules**

Fuzzy Inference Rules (FLS) calculates the output based on the rules known as Fuzzy Inference Rules. These rules are based on IF-THEN statements and define the output based on different variations of inputs [24]. The following formula is used to know the number of rules:

$$r = m^n$$

In the above formula, 'r' is the total number of rules, 'm' is the total input linguistic words, and 'n' is the total input variables. As per the current example, the number of rules is:

$$r = 3^3 = 27$$

This means, a total of 27 unique rules should be defined and the order does not matter thus rules can be defined in any order. The things to be noted here are, each rule must be unique and must not repeat. Table 4 shows the Fuzzy Inference Rules for the current example.

Table 4. Example of Fuzzy Inference Rules

#	p	q	R	Z
1	BAD	BAD	BAD	BAD
2	BAD	BAD	AVERAGE	BAD
3	BAD	BAD	GOOD	BAD
4	BAD	AVERAGE	BAD	BAD
5	BAD	AVERAGE	AVERAGE	AVERAGE
6	BAD	AVERAGE	GOOD	AVERAGE
7	BAD	GOOD	BAD	BAD
8	BAD	GOOD	AVERAGE	AVERAGE
9	BAD	GOOD	GOOD	AVERAGE
10	AVERAGE	BAD	BAD	BAD
11	AVERAGE	BAD	AVERAGE	AVERAGE
12	AVERAGE	BAD	GOOD	AVERAGE
13	AVERAGE	AVERAGE	BAD	AVERAGE
14	AVERAGE	AVERAGE	AVERAGE	AVERAGE
15	AVERAGE	AVERAGE	GOOD	AVERAGE
16	AVERAGE	GOOD	BAD	AVERAGE
17	AVERAGE	GOOD	AVERAGE	AVERAGE
18	AVERAGE	GOOD	GOOD	GOOD
19	GOOD	BAD	BAD	BAD
20	GOOD	BAD	AVERAGE	AVERAGE
21	GOOD	BAD	GOOD	AVERAGE
22	GOOD	AVERAGE	BAD	AVERAGE
23	GOOD	AVERAGE	AVERAGE	AVERAGE
24	GOOD	AVERAGE	GOOD	GOOD
25	GOOD	GOOD	BAD	AVERAGE
26	GOOD	GOOD	AVERAGE	GOOD
27	GOOD	GOOD	GOOD	GOOD

These rules were generated using a computer program written in Python 3.8. Code Snippet V shows the Python 3.8 code that generated the Fuzzy Inference Rules. The code can be customized according to the required generating rules.

Code Snippet V: Computer Program Written in Python Language to Generate Fuzzy Rules

```
1.     ling_words = ("bad",
   "average", "good")
2.
3.     def rule(i, *args):
4.         a = ""
5.
6.         for i in
   range(len(args)):
7.             b =
   ling_words[args[i]] + ' '
8.             a = a + b
9.
10.        if a.count("good") == 3
   or (a.count("good") == 2 and
   a.count("average") == 1):
11.            return a + 'good'
12.        elif a.count("bad") == 3
   or (a.count("bad") == 2 and
   a.count("average") == 1) or
   (a.count("bad") == 2 and
   a.count("good") == 1):
13.            return a + 'bad'
14.        elif a.count("average")
   >= 3 or (a.count("average") ==
   2 and a.count("good") == 1) or
   (a.count("average") == 2 and
   a.count("bad") == 1) or
   (a.count("good") == 2 and
   a.count("bad") == 1):
15.            return a + 'average'
16.        else:
17.            return a + 'average'
18.
19.    i = 0
20.    for p in range(0, 3):
21.        for q in range(0, 3):
22.            for r in range(0, 3):
23.                print   (i   +   1,
   rule(i, p, q,
```

Next, these Fuzzy Rules are passed to the system as an Inference Engine, which is then used to generate and defuzzify the output.

DJANGO: AN OVERVIEW OF THE MOST POWERFUL WEB-FRAMEWORK

Web Framework is a dedicated and special environment developed for the sake of making Web Development an easy task and brings versatility in the development options. There are many web frameworks which are being used for the Web Development or Web App Development. Some of the most famous Web Frameworks include MVC (C#), Flask (Python) and Django (Python). However, despite being popular (as per the organization's requirements), MVC and Flask still lack in a lot of factors. Flask can't be a good choice for big and heavy websites and web apps. On the other hand, undoubtedly, Django started as a Model-View-Controller (MVC), but Django's philosophy of batteries included means that experienced developers do not have to plan too much ahead on what kind of application infrastructure they need and instead just start developing web applications quickly [25].

Databases are stored as Models in Django. They are created as a Class in Python Language. On the other hand, server-side is managed as Views in Django. Views can be both Python's function-based and Python's Class-based. Django handles URLs very intelligently. In earlier versions of Django, RE was used in URL patterns but later the developers of Django added the functionality to make the handling of URL patterns easy for those who are unfamiliar with RE. Now, URL patterns can be handled easily with a path() function without using RE, but expert still recommends RE, as they provide more versatility and freedom [26]. Django handles Forms very efficiently as well. Again, it uses Python's Class definition to create forms. Forms can be both independent and Model-based, Model-based Forms are usually connected/associated with the specific Model (Database), and Model can be specified in the Class definition of Form as an attribute 'model', such as 'model = ModelName'. Model-based Forms are used when the intention is to update the data of the specific Database with the data entered by the user. On the other hand, normal Forms (non-Model-based Forms) are used for just getting information from the user to perform specific tasks. Last but not least

element of Django is Templates. Django uses Templates to format and engineer the Front-End of the web app or website. Templates intelligently use the View that is associated with it, to forward the Form or other calculation or data at Front-End using HTTP Response. According to some experts, if you do master the Django, you can control the Internet, the way you want'. Django also offers the Default User Model (Database) which has the entire common and basic User's functionality and it also offers default Login Page View, Logout Page View, Register Page View and Reset Password Page View. All of these Views have the complete functionality which is a sole requirement for a basic login system in any website or a web app. Django also provides the facility to extend these default Page Views and User Model or define the custom Views and a custom User Model [27].

The ESACP was planned to be efficient, intelligent and automatic. There are Artificial Intelligence techniques which play a big role when there is a need for intelligence and efficiency. Among many approaches, FLS is considered to be the best technique as it contains more relevancies to the subject and it produces completely accurate results. Hence, the developed ES takes five parameters as inputs viz. skype session attendance, internal course marks, programming lab activity, mid-term marks and final-term marks, and uses FLS AI Algorithm considering these parameters to intelligently and efficiently assess or calculate the performance of students of CS, CP or SE courses.

RESEARCH METHODOLOGY

As it was required to make ESACP as a web app, ESACP is built using Django version 3.0.3 and Python version 3.8.1. Table 5 shows the libraries with their versions which were used in a development environment while developing ESACP.

Table 5. Python Libraries used in the Development of ESACP

Python Libraries	Version
asgiref	3.2.3
cycler	0.10.0
Decorator	4.4.2
Django	3.0.3
django-crispy-forms	1.8.1
Kiwisolver	1.2.0
Matplotlib	3.2.1
network	2.4
Numpy	1.18.1
Pillow	7.0.0
Psutil	5.6.7
Pyparsing	2.4.7
python-dateutil	2.8.1
Pytz	2019.3
scikit-fuzzy	0.4.2
Scipy	1.4.1
Six	1.14.0
Sqlparse	0.3.0

The FLS AI Algorithm in ESACP is designed using the forward-chaining approach and Mamdani Technique with Triangular Membership Function.

- **Defining Input and Output Variables**

ESACP has 5 inputs viz. Skype Session Attendance, Internal Course Marks, Programming Lab Activity, Mid-Term Marks and Final-Term Marks. The ranges assigned to these inputs are just decided by a presumption. Table 6 shows the input variables of ESACP.

Table 6. Input Variables of ESACP

Input Variables	Description
p	Skype Session Attendance
q	Internal Course Marks
r	Programming Lab Activity
s	Mid-Term Marks
t	Final-Term Marks

ESACP has only one output that is Performance. The range assigned to it is also decided by a presumption. Table 7 shows the output variable of ESACP.

Table 7. Output Variable of ESACP

Input Variables	Description
Z	Performance

As said earlier that ranges assigned to all variables of ESACP are decided by a presumption. Table 8 shows the ranges assigned to the variables of ESACP.

Table 8. Assigned Range to Defined Variables of ESACP

Variables	Min Range Value	Max Range Value
p	0	20
q	0	40
r	0	25
s	0	45

t	0	90
z	0	10

Code Snippet VI shows the Python code that was used to assign ranges to the variables

Code Snippet VI: Computer Program Written in Python Language to Assign Ranges to Input and Output Variables

```
p = control.Antecedent(np.arange(0, 21, 1), 'p')
q = control.Antecedent(np.arange(0, 41, 1), 'q')
r = control.Antecedent(np.arange(0, 26, 1), 'r')
s = control.Antecedent(np.arange(0, 46, 1), 's')
t = control.Antecedent(np.arange(0, 91, 1), 't')
z = control.Consequent(np.arange(0, 11, 1), 'z')
```

The 'control' in the above code is imported from the Python's scikit-fuzzy (mostly known as skfuzzy) library and 'np' is an alias of Python's numpy library. The Antecedent method is used for input variables and Consequent method is used for the output variable.

- **Assigning Linguistic Words to the Defined Input Variables (Fuzzification)**

For ESACP, three linguistic words were decided to be assigned to input variables viz. 'Bad', 'Average', 'Good'. Triangular membership function was used to fuzzify the inputs. Code Snippet VII shows the Python code that was used to assign linguistic words to input variables and fuzzify them using triangular membership function.

Code Snippet VII: Computer Program Written in Python Language to Assign Linguistic Words to Input Variables and Fuzzify Them

```
p['bad'] = fuzzy.trimf(p.universe, [0, 4.5, 8])
p['average'] = fuzzy.trimf(p.universe, [7, 11.5, 15])
p['good'] = fuzzy.trimf(p.universe, [14, 17.5, 20])

q['bad'] = fuzzy.trimf(q.universe, [0, 6.5, 13])
q['average'] = fuzzy.trimf(q.universe, [12, 19.5, 26])
q['good'] = fuzzy.trimf(q.universe, [25, 32.5, 40])

r['bad'] = fuzzy.trimf(r.universe, [0, 4.5, 8])
r['average'] = fuzzy.trimf(r.universe, [7, 12.5, 16])
r['good'] = fuzzy.trimf(r.universe, [15, 20.5, 25])

s['bad'] = fuzzy.trimf(s.universe, [0, 7.5, 15])
s['average'] = fuzzy.trimf(s.universe, [14, 22.5, 30])
s['good'] = fuzzy.trimf(s.universe, [29, 37.5, 45])

t['bad'] = fuzzy.trimf(t.universe, [0, 15, 30])
t['average'] = fuzzy.trimf(t.universe, [29, 45, 60])
t['good'] = fuzzy.trimf(t.universe, [59, 75, 90])
```

The 'fuzzy' in the above code is an alias of 'skfuzzy' (Python's scikit-fuzzy library name), and trimf() method is an implementation of the triangular membership function. There are other membership functions as well such as, Trapezoidal Membership function and Sigmoid Membership function, but for this work Triangular Membership function is used.

- **Assigning Linguistic Words to the Defined Output Variable (Defuzzification)**

For ESACP, five linguistic words were decided to be assigned to output variable viz. 'Poor', 'Bad', 'Average', 'Good', 'Excellent'. Triangular membership function was also used for the output variable. Keep in view the fact that we have to use the same membership function for both input variables and the output variable. Code Snippet VIII shows the Python code that was used to assign linguistic words to input variables and fuzzify them using triangular membership function.

Code Snippet VIII: Computer Program Written in Python Language to Assign Linguistic Words to Output Variable and Defuzzify Them

```
1.     z['poor'] =
    fuzzy.trimf(z.universe, [0, 1,
    2])
2.     z['bad'] =
    fuzzy.trimf(z.universe, [1,
    2.5, 4])
3.     z['average'] =
    fuzzy.trimf(z.universe, [3,
    4.5, 6])
4.     z['good'] =
    fuzzy.trimf(z.universe, [5,
    6.5, 8])
5.     z['excellent'] =
    fuzzy.trimf(z.universe, [7,
    8.5, 10])
```

- **Defining Fuzzy Inference Rules**

As said earlier in 'Fuzzy Logic: A Theoretical Review of an AI Algorithm', FLS calculates the output based on rules which should be defined by the developer and these rules are known as Fuzzy Inference Rules. Total numbers of Fuzzy Rules for ESACP are given as:

$$r = m^n$$
$$r = 3^5 = 243$$

Fuzzy Rules for ESACP are generated using the same Python Code that is presented in Code Snippet V. Table 9 shows the first ten fuzzy rules from the ESACP FLS AI Algorithm.

Table 9. First Ten Fuzzy Inference Rules from ESACP FLS AI Algorithm

#	p	Q	r	s	t	z
1	Bad	Bad	Bad	Bad	Bad	Poor
2	Bad	Bad	Bad	Bad	Average	Poor
3	Bad	Bad	Bad	Bad	Good	Bad
4	Bad	Bad	Bad	Average	Bad	Poor
5	Bad	Bad	Bad	Average	Average	Bad
6	Bad	Bad	Bad	Average	Good	Bad
7	Bad	Bad	Bad	Good	Bad	Bad
8	Bad	Bad	Bad	Good	Average	Bad
9	Bad	Bad	Bad	Good	Good	Bad
10	Bad	Bad	Averag e	Bad	Bad	Poor

All of the 243 Fuzzy Rules were stored in a Knowledge Base text file whose extensions are '.kb'. The rules were then passed to an Inference Engine, which produced an output based on these rules. Code Snippet IX shows the Python code that was used to pass all of the Fuzzy Rules in Inference Engine and generate an output.

Code Snippet IX: Computer Program Written in Python Language to Process the Fuzzy Rules as an Inference Engine

```
1.     rls = []
2.
3.     with open(RULES_abs_path,
    "r") as fh:
4.         for line in fh:
5.             if line != '\n' and
    line != None:
6.                 temp =
    line.split(" ")
7.                 result =
    control.Rule(p[temp[0]] &
    q[temp[1]] & r[temp[2]] &
    s[temp[3]] & t[temp[4]],
    z[temp[5].rstrip()])
8.
    rls.append(result)
9.
10.    perf_calc =
    control.ControlSystem(rls)
```

EXPERIMENTAL RESULTS

Two experiments were performed to check the performance and results. Both experiments included manual and ESACP evaluation. The first manual

experiment is about a student named Alishba Ibrar. Table 10 shows the manual evaluation details of the performance of Alishba Ibrar in subject Introduction to Programming.

Table 10. Manual Evaluation Details of Student Alishba Ibrar		
Name	**Alishba Ibrar**	
Subject	**Introduction to Programming**	
Details	**Obtained/Attended**	**Total**
Skype Session Attendance	15	20
Internal Course Marks	20	40
Programming Lab Activity	22	25
Mid-Term Marks	38	45
Final-Term Marks	75	90
Performance	Good	

After manual evaluation, ESACP was used to evaluate the Alishba's performance on the same details. Figure. 1 shows the updating of Alishba's details of subject Introduction to Programming.

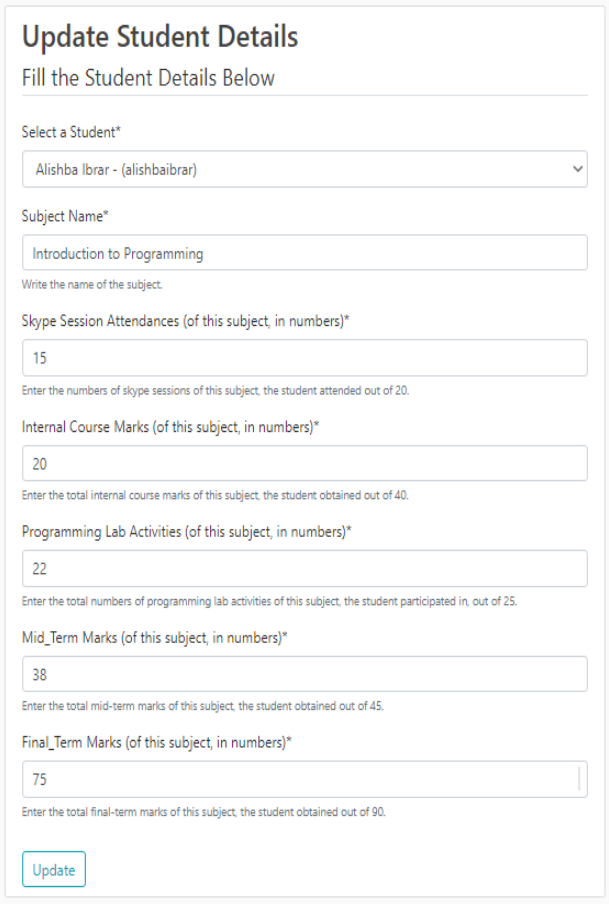

Figure. 1 Updating details of student Alishba Ibrar of subject Introduction to Programming

While the details were updated in the background, the FLS AI Algorithm was also working and stored the performance in the database. Next, ESACP evaluated performance was plotted in Figure 2. Figure. 2 shows Alishba's performance in subject Introduction to Programming.

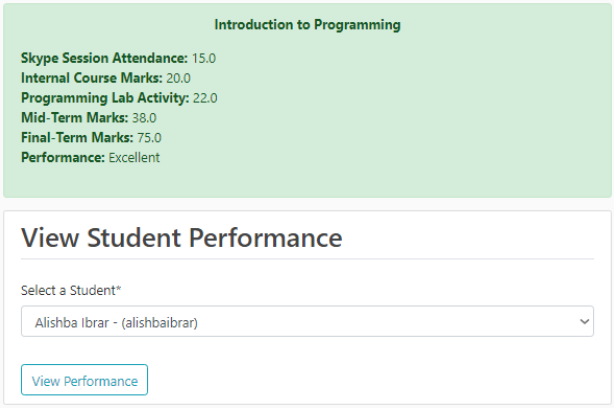

Figure 2. Calculated Performance of the student Alishba Ibrar of subject Introduction to Programming

The second manual experiment is about a student named Ahmer Baloch. Table 11 shows the manual evaluation results of the performance of Ahmer Baloch in subject Introduction to Programming.

Table 11. Manual Evaluation Details of Student Ahmer Baloch		
Name	**Ahmer Baloch**	
Subject	**Introduction to Programming**	
Details	**Obtained/Attended**	**Total**
Skype Session Attendance	10	20
Internal Course Marks	18	40
Programming Lab Activity	15	25
Mid-Term Marks	25	45
Final-Term Marks	60	90
Performance	Average	

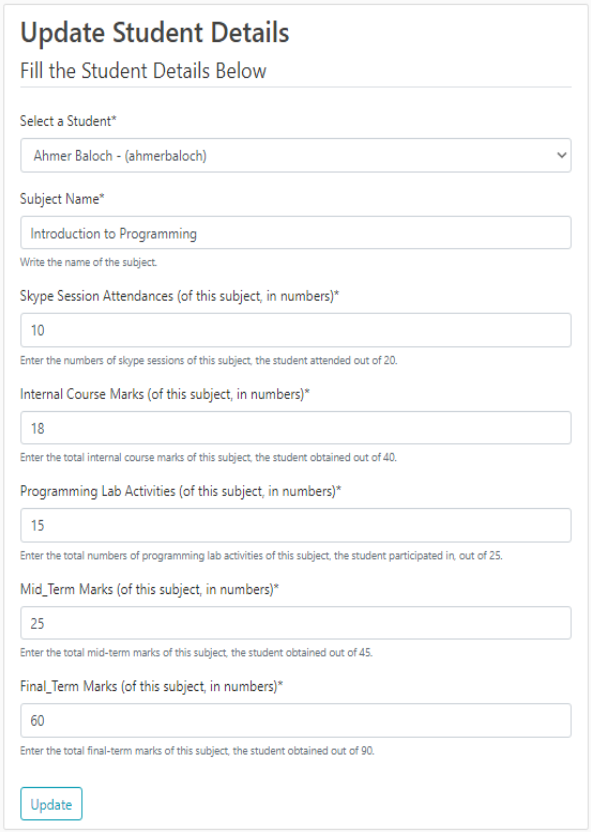

Figure 3. Updating details of student Ahmar Baloch of subject Introduction to Programming

Next, the performance of ESACP was evaluated. Figure. 4 shows Ahmer's performance in subject Introduction to Programming.

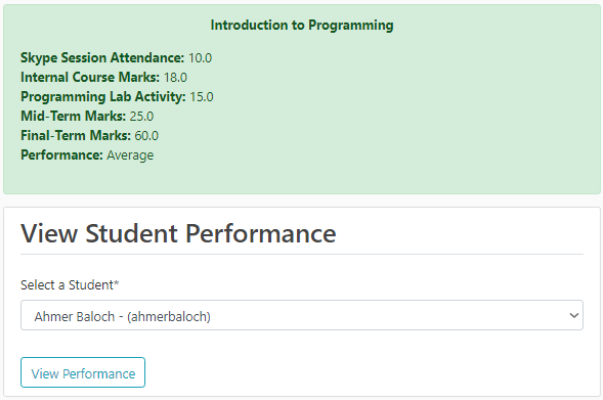

Figure 4. Calculated Performance of the student Ahmer Baloch of subject Introduction to Programming

Both of the experiments were a success and verify that ESACP evaluates the performance intelligently, efficiently and properly. The ESACP is acquired using URL 'https//www.institutewebsite.com/esacp', and at LocalHost, as 'https://localhost:8000/esacp'

CONCLUSION

The experimental study showed that ESACP evaluates the performance very effectively. It concludes that as a substitute of degenerative a human resource and time for evaluating the performance of thousands of students manually, it is better to utilize ESACP software that evaluates the performance of a large number of students. It will also be beneficial to preserve the time and energy of teachers and professors and can enhance the pedagogy in the future for student performance enhancement.

REFERENCES

1. P. Black and D. Wiliam, "Developing the theory of formative assessment," Educational Assessment, Evaluation and Accountability (formerly: Journal of Personnel Evaluation in Education), vol. 21, p. 5, 2009.
2. E. Panadero, H. Andrade, and S. Brookhart, "Fusing self-regulated learning and formative assessment: A roadmap of where we are, how we got here, and where we are going," The Australian Educational Researcher, vol. 45, pp. 13-31, 2018.
3. J. W. Gikandi, D. Morrow, and N. E. Davis, "Online formative assessment in higher education: A review of the literature," Computers & education, vol. 57, pp. 2333-2351, 2011.
4. I. J. Arnold, "Cheating at online formative tests: Does it pay off?," The Internet and Higher Education, vol. 29, pp. 98-106, 2016.
5. W. Harlen and M. James, "Assessment and learning: differences and relationships between formative and summative assessment," Assessment in Education: Principles, Policy & Practice, vol. 4, pp. 365-379, 1997.
6. F. SCASS, "Distinguishing formative assessment from other educational assessment labels," Retrieved on, vol. 3, p. 2014, 2012.
7. A. Khamparia and B. Pandey, "Association of learning styles with different e-learning problems: a systematic review and classification," Education and Information Technologies, pp. 1-29, 2019.
8. O. Shvets, K. Murtazin, and G. Piho, "Providing Feedback for Students in E-Learning Systems: a Literature Review, based on IEEE Explore Digital Library," in 2020 IEEE Global Engineering Education Conference (EDUCON), 2020, pp. 284-289.
9. H. Martin, "Measuring Qualitative Performance Criteria with Fuzzy Sets," in International Conference on Business Information Systems, 2019, pp. 417-423.

10. N. Meenakshi and N. Pankaj, "Application of Fuzzy Logic for Evaluation of Academic Performance of Students of Computer Application Course," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 3, pp. 260-7, 2015.
11. L. Mutanu and P. Machoka, "Enhancing computer students' academic performance through explanatory modeling," in Annual Conference of the Southern African Computer Lecturers' Association, 2019, pp. 227-243.
12. S. Papadimitriou, K. Chrysafiadi, and M. Virvou, "FuzzEG: Fuzzy logic for adaptive scenarios in an educational adventure game," Multimedia Tools and Applications, vol. 78, pp. 32023-32053, 2019.
13. R. Karthika, L. J. Deborah, and P. Vijayakumar, "Intelligent e-learning system based on fuzzy logic," Neural Computing and Applications, pp. 1-10, 2019.
14. I. B. Ishak, "Application of fuzzy logic to student performance in calculation subjects," in National Symposium & Exhibition on Business & Accounting, 2015.
15. Z. Yıldız and A. F. Baba, "Evaluation of student performance in laboratory applications using fuzzy decision support system model," in 2014 IEEE Global Engineering Education Conference (EDUCON), 2014, pp. 1023-1027.
16. M. A. M. Sahagun, "A Fuzzy Logic Approach for Course Outcomes-based Assessment," in 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), pp. 1-6.
17. C. Troussas, A. Krouska, and C. Sgouropoulou, "Dynamic Detection of Learning Modalities Using Fuzzy Logic in Students' Interaction Activities," in International Conference on Intelligent Tutoring Systems, 2020, pp. 205-213.
18. K. Chrysafiadi, C. Troussas, and M. Virvou, "Combination of fuzzy and cognitive theories for adaptive e-assessment," Expert Systems with Applications, p. 113614, 2020.
19. M. Megahed and A. Mohammed, "Modeling Adaptive E-Learning Environment using Facial Expressions and Fuzzy Logic," Expert Systems with Applications, p. 113460, 2020.
20. L. A. Zadeh, "Fuzzy logic," Computer, vol. 21, pp. 83-93, 1988.
21. M. A. Hogo, "Evaluation of e-learners behaviour using different fuzzy clustering models: a comparative study," arXiv preprint arXiv:1003.1499, 2010.
22. D. Sinha and E. R. Dougherty, "Fuzzification of set inclusion: theory and applications," Fuzzy sets and systems, vol. 55, pp. 15-42, 1993.
23. W. Van Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," Fuzzy sets and systems, vol. 108, pp. 159-178, 1999.
24. L. M. De Campos and S. Moral, "Learning rules for a fuzzy inference model," Fuzzy Sets and Systems, vol. 59, pp. 247-257, 1993.
25. W. S. Vincent, Django for Beginners: Build websites with Python and Django: Still River Press, 2020.
26. B. Lopatin, "Structuring standalone Django apps," in Django Standalone Apps, ed: Springer, 2020, pp. 11-15.
27. A. Shyam and N. Mukesh, "A Django Based Educational Resource Sharing Website: Shreic," Journal of Scientific Research, vol. 64, 2020.