A Conceptual Design of a Mamdani Fuzzy Rule-Based Personalized Feedback System for Programming Intelligent Tutoring

Ramlah Mailok[1*] , Supiyandi[2]

[1]Faculty Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, Perak, Malaysia

[2]Universitas Pembangunan Panca Budi Medan, Medan, Indonesia

[*]Corresponding Author

DOI: https://doi.org/10.47772/IJRISS.2026.100900423

Received: 21 September 2026; Accepted: 26 September 2026; Published: 13 October 2026

ABSTRACT

Personalized feedback is essential in programming education because students differ in prior knowledge, response accuracy, time spent on tasks and error patterns. However, conventional computer-based learning systems commonly provide uniform feedback that may not match individual learning needs. This paper presents the conceptual design and formalisation of a Personalized Feedback System (PFS) for programming intelligent tutoring using a Mamdani fuzzy rule-based system (FRBS). The proposed PFS integrates a student model, an expert model and a feedback generation model. The fuzzy inference mechanism uses three input variables such as question difficulty, response time and correctness to determine one of five feedback and scaffolding levels. Linguistic variables, membership functions, a fuzzy rule base, inference procedures and centroid defuzzification are specified to support adaptive feedback selection. The output feedback ranges form minimal confirmation and repeated attempts to error flagging, worked examples, conceptual explanations and knowledge of correct response. This paper is a conceptual design study; therefore, it does not claim empirical effectiveness. Its contribution is a transparent fuzzy-logic mechanism that can be implemented and subsequently evaluated through usability testing and learning-outcome experiments with programming learners.

Keyword: Personalized Feedback system, Fuzzy Logic-Mamdani Type FRBS, Programming

INTRODUCTION

Advances in artificial intelligence and educational technology have created new opportunities to support adaptive learning. The availability of web-based learning facilities allows for course delivery to a large number of students, addresses issues related to the distribution of notes or software, and makes learning more flexible (Maghsudi et al., 2021). However, some challenges or barriers arise, such as students feeling isolated or receiving inadequate guidance. This occurs because communication from educators has increased, while individual student needs, particularly in terms of specific feedback, are often overlooked.

The intelligent tutoring system environment leads to a change in the roles of students and educators, emphasizing a student-centered approach (Kerimbayev, et al., 2023). This means that students need to know how to actively acquire knowledge to meet their learning targets or goals. Educators only serve as facilitators, helping and guiding them to achieve their learning objectives. Given the broad role of educators to monitoring and understanding each problem faced by students while guiding and assisting them. These also providing individualized guidance to large cohorts can be challenging for educator. Therefore, based on existing technological advancements available for developing most educational software, researchers believe that using artificial intelligence (AI) approaches can help enhance the role of educators in assisting students in addressing their problems, particularly in providing specific feedback to each student.

Although many digital learning systems are available, their capacity to provide adaptive and individualized feedback remains uneven. One major issue faced is the less effective interactive environment for all students (Zulaikha, et al., 2021). Currently, studies on student modeling have been conducted to ensure it meets the needs of every individual student (fulfilling the needs of all students). An example of a system that uses an interactive environment is SMITHTOWN (Shute et al., 2007; Narciss & Huth, 2004), which employs a simple student model and basic expert systems. However, developing a dynamic student model presents a significant challenge for researchers because it is too complex and difficult within computer-based learning environments.

Although many educational software or applications have been developed to support learning whether interactive or static like SMITHTOWN, WebCT, Blackboard, CentraOne, etc, but in research or in practice, the integration of dynamic feedback into complex learning settings, such as digital game-based programming learning, remains limited (Mao et al, 2024). Many systems emphasize content delivery, while offering limited adaptation of feedback to individual learner characteristics (Hattie & Timperley, 2007; Bangert-Drowns et al., 1991). Therefore, this proposed learning software will differ in its emphasis by focusing on diverse feedback while integrating fuzzy logic-Mamdani type FRBS (Fatemeh & Masoud, 2019; Zadeh, 1994) as an engine for generating that feedback.

Existing educational feedback system vary in the data used to model learners, the mechanisms used to generate feedback and the extend to which feedback is adaptive. Conventional learning management systems commonly provide automated feedback based on correctness of completion, whereas intelligent tutoring systems may apply explicit pedagogical rules to guide learners. More recent learning analytics and artificial intelligence approaches use historical and behavioral learner data to generate predictions or PFS. The PFS is intended as a interpretable fuzzy rule-based approach that combines question difficulty, response time and answer correctness to determine feedback intensity (Fatemeh & Masoud, 2019).

Table 1: Comparison of feedback approaches relevant to the proposed PFS

Approach	Typical adaptation variables	Feedback mechanism	Strength	Typical limitation or consideration
Conventional LMS	Correct/ incorrect response, quiz score, completion status	Static confirmation, automated score report or predefined message	Easy to deploy because feedback can be configured using standard assessment settings	Usually provides limited personalization because the feedback commonly does not incorporate multiple learner-performance indicator, such as response time, task difficulty or error patterns (Maghsudi et al., 2021; Hattie & Timperley, 2007)
Crisp rule-based ITS	Learner state, response rules, domain knowledge and predefined conditions	Predefined hints, prompts, solution steps or instructional actions	Transparent logic because educators and designers can inspect and revise explicit IF-THEN rules	Crip categories may make abrupt decisions at category boundaries; fuzzy logic is useful when learner states or performance indicators are imprecise or overlapping (Karaci, 2019; Zadeh, 1994)
Data-driven AI feedback	Historical learner data, behavioral traces, assessment performance, clickstream data	Performance prediction, personalized recommendation, risk alerts or adaptive feedback.	Can identify complex learner patterns and support personalized feedback using	Requires sufficient, appropriate-quality data and careful attention to transparency, interpretability, fairness and

	and learning patterns		multiple sources of learner data (Hooda et al., 2022)	ethical use of learner data (Zulfiqar, 2026; Papakostas et al., 2023)
Proposed PFS	Question difficulty, response time and answer correctness	Mamdani fuzzy rules infer five feedback generator then select confirmation retry opportunity, hints, error flagging, worked examples, conceptual explanation or knowledge correct response	Transparent and gradual adaptation because overlapping membership functions can represent uncertainty between learner performance categories (Karaci, 2019; Zadeh, 1994)	Requires software implementation, expert validation of rules and membership-function parameters and empirical evaluation with programming learners before claims about effectiveness can be made (Karaci, 2019; Zadeh, 1994)

The proposed PFS does not claim to replace data-driven AI approaches. Instead, it is designed for contexts in which transparency and educator control are important. Unlike static feedback, the PFS considers multiple learner and task indicators. Unlike crisp rule-based approaches, it uses overlapping membership functions to accommodate uncertainty at category boundaries. Unlike data-driven approaches, the proposed system does not require a large historical dataset for its initial rule specification, however its membership functions and rule base must be validated and calibrated through expert review and learner data during future implementation (Maghsudi et al., 2021; Papakostas et al., 2023; Karaci, 2019; Zadeh, 1994)

Research Design and Scope

This study adopts a conceptual design and development approach to formulate a Personalized Feedback System (PFS) for programming intelligent tutoring. The study does not involve a participant experiment, classroom intervention, usability evaluation or statistical validation of learning outcomes. Therefore, no claims are made regarding the effectiveness of the proposed system.

The design process consisted of four (4) stages. First, relevant literature on formative feedback, scaffolding, intelligent tutoring systems and fuzzy rule-based systems was reviewed to identify feedback characteristics relevant to programming learning. Second, the core PFS components were specified, namely the student model, expert model, and feedback-generation model. Third, the fuzzy inference mechanism was formalized by defining input and output variables, linguistic terms, membership functions, a rule base, inference operator and a defuzzification method. Finally, representative learner scenarios were used to illustrate how the proposed system selects an appropriate feedback level.

The scope of the proposed system is formative feedback for programming exercises. The current design considers question difficulty, answer correctness, and response time as primary inputs. Other learners' characteristics, such as prior knowledge, misconception type, code quality and learning motivation are recognized as relevant but are proposed as extensions for future implementation.

PFS Framework

Figure 1 presents the illustrates the architecture of the proposed PFS. The student model captures learner interaction data, including response correctness, response time, prior performance and error information. The expert model stores the question bank, pedagogical feedback templated, question metadata. Membership-function parameters and fuzzy rules. These inputs are processed by the Mamdani fuzzy inference engine to determine a feedback intensity level. The feedback generation module then uses this level together with answer correctness and error information to deliver an appropriate feedback message and scaffolding action.

This architecture aims to enhance personalized learning experiences by providing tailored feedback that addresses individual student needs through a structured and adaptive approach.

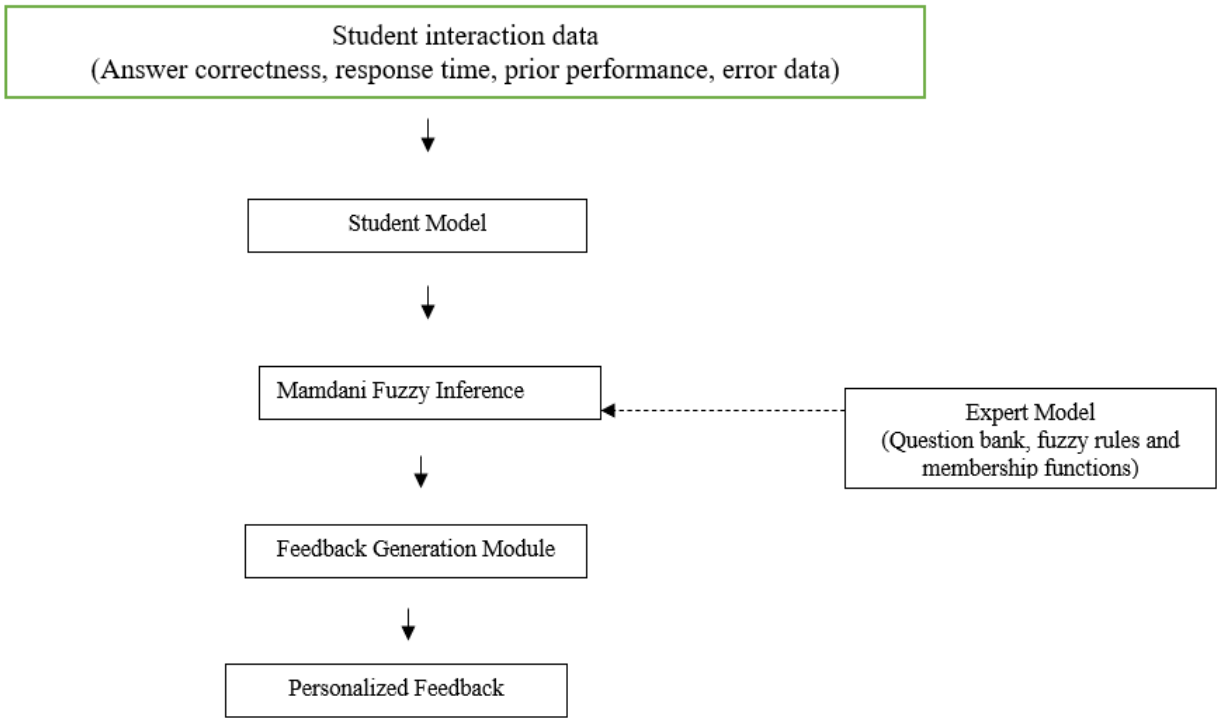


Figure 1: The architecture of the PFS

Personalized Feedback Based on Fuzzy Logic- Mamdani Type FRBS

The PFS is a constructivist learning environment where students are actively guided, and scaffolding is provided. In this system, emphasis is placed on two aspects of feedback: the type of feedback given (referring to the information contained in the feedback) and the aspect of scaffolding (the assistance provided).

Feedback types

The types of feedback used in this study are based on frameworks established by [9, 10, 11, 12]. There are five types of feedback considered in the study, as shown in Table 2. The feedback provided depends on prior knowledge, learning objectives, question difficulty, answer categories, and the time allocated to answer questions.

Explanation of Feedback Types

i. Knowledge of Response (KR): This type provides confirmation for each question item without additional information.

ii. Answer Until Correct (AUC): Students are allowed to repeatedly answer the same question until they provide the correct response.

iii. Knowledge of Correct Response (KCR): Feedback includes correct answers for each question item, enhancing learning by indicating what was right.

iv. Error Flagging: This feedback highlights where mistakes were made, helping students identify specific errors in their responses.

v. Elaborative: This type focuses on key concepts or examples related to the questions, providing deeper insights into the material.

Table 2: Feedback types used in PFS

Type of Feedback	Action			Notes
	Confirmation	Answer Given	Response Analysis	
Knowledge of Response (KR)	Yes	No	No	Confirmation for each item questions
Answer Until Correct (AUC)	Yes	No	No	Repetition of answering for the same question
Knowledge of Correct Response (KCR)	Yes	Yes	No	Answers will be provided for each item questions
Error Flagging	Yes	No	Yes	Indicates the position of errors made
Elaborative	Yes	Yes	No	Focus on keyword or relevant examples

The PFS is designed to offer personalized feedback that varies according to individual student needs based on their prior knowledge, learning objectives, question difficulty, answer categories, and response time. By utilizing fuzzy logic as an engine for decision-making, the system can effectively adapt feedback to enhance learning.

Feedback and scaffolding levels

Table 3: Feedback action and scaffolding based on student's answer

Fuzzy output	Feedback Action	
	Incorrect response	Correct response
F1	• Allow students to attempt to answer (AUC – Answer Until Correct). • If the answer is correct, move to a higher difficulty level of question (new question); if not, move to F2.	• Confirmation (Simply informs that the provided answer is correct). • Move to a higher difficulty level.
F2	• Show example answers that are similar to the answered question – Elaborative example and error flagging. Allow students to attempt to answer (AUC). • If the answer is correct, move to a higher difficulty level of question; if not, move to F3.	• Confirmation (Simply informs that the provided answer is correct). • Move to a higher difficulty level of question.
F3	• Explain the concepts needed to answer the question – Elaborative concepts and error flagging. Allow students to attempt to answer (AUC). • If the answer is correct, remain at the same difficulty level of question until the next question can be answered correctly; if not, move to F4.	• Confirmation and explanation of the correct answer. • Move to a higher difficulty level of question.
F4	• Explain the answer that should be obtained – KCR (Knowledge of Correct Response) and error flagging. • Move to a lower difficulty level of question or remain at the same level (F4).	• Confirmation and explanation of the correct answer. • Move to a higher difficulty level of question.
F5	• Explain the answer that should be obtained – KCR and error flagging. • Move to a lower difficulty level of question or remain at the same level.	• Confirmation and explanation of the concept of the correct answer. • Remain at the same position until able to answer the next question correctly. • If the answer is correct then completed, if not then move to next question (the same level)

While Level 1 (F1), the feedback provides minimal information for proficient students with high mastery, level 5(F5) assumes the question is difficulty, requiring more detailed support such as concepts, examples or

knowledge of Correct Response (KCR) to reinforce learning. Upon answering correctly, the students proceed to a question at the same or lower difficulty level.

Fuzzy rules for personalized feedback

The formation of fuzzy output in the PFS, as shown in Table 4, is based on the variables of question difficulty, answer categories, and the time taken to answer the questions. After a student answers a question, they require feedback determined based on these three variables. Table 4 shows the overall relationship in the form of fuzzy rules. These fuzzy rules are closely related to how an expert makes decisions in varying situations. For example, when an expert grades answers for questions of easy difficulty, they typically provide only confirmation feedback. If a student does not answer correctly, the expert will give them another chance and guide them on how to solve that question. Meanwhile, Figure 2 illustrates the fuzzy membership function for PFS, which functions similarly to other membership functions and will be explained in subsequent processes (fuzzy inference system process).

Table 4: Fuzzy rules for Personalized feedback

Response Time (T)	**Question difficulty (D)**	**Fuzzy output**	
		Incorrect	**Correct**
Minimum	Easy	F2	F1
	Medium	F2	F1
	Complex	F3	F2
Optimal	Easy	F2	F1
	Medium	F3	F2
	Complex	F4	F2
Maximum	Easy	F3	F2
	Medium	F4	F3
	Complex	F5	F3

Example IF-THEN rules

Rule 1: IF response time is Minimum AND question difficulty is Easy AND answer correctness is Correct, THEN feedback intensity is F1.

Rule 2: IF response time is Minimum AND question difficulty is Easy AND answer correctness is Incorrect, THEN feedback intensity is F2.

Rule 10: IF response time is Optimal AND question difficulty is Medium AND answer correctness is Correct, THEN feedback intensity is F2.

Rule 18: IF response time is Maximum AND question difficulty is Complex AND answer correctness is Incorrect, THEN feedback intensity is F5.

The full rule base contains 18 rules, representing all combinations of three response time categories, three question difficulty categories and two answer correctness categories.

Fuzzy Inference Mechanism

The proposed PFS uses a Mamdani fuzzy inference system to determine the appropriate feedback level for a student after each programming exercise attempt. The system receives crisp input values from the learning platform and transforms them into fuzzy linguistic values during fuzzification. The inference system contains three input variables and one output variable.

The input variables are:

- Question difficulty (D) represented on a scale from 0 to 100, where 0 represents a very easy question and 100 represents a highly complex question.
- Response time (T) calculated as the ratio between the student's actual completion time and the expected completion time for the item.
- Answer Category (C) represented as a binary value, where 0 indicates an incorrect response and 1 indicates a correct response.

The fuzzy inference system produces one output variable, namely feedback intensity F, represented on scale from 0 to 100 and classified into five linguistic level: F1, F2, F3, F4 and F5. The student's answer correctness is retained as an input variable and is subsequently used by the feedback generator to select the appropriate feedback content and scaffolding action for inferred intensity level.

Table 5: Define the Membership Functions

Variable	Linguistic term	Membership function
Question difficulty, D	Easy	Trapezoidal: trapm f(D;0,0,25,45)
Question difficulty, D	Medium	Triangular: trim f(D;30,50,70)
Question difficulty, D	Complex	Trapezoidal: trapm f(D;55,75,100,100)
Response time, T	Minimum	Trapezoidal: trapm f(T;0,0,0.5,0.8)
Response time, T	Optimal	Triangular: trim f(T;0.6,1.0,1.4)
Response time, T	Maximum	Trapezoidal: trapm f(T;1.2,1.5,2.0,2.0)
Answer Category, C	Incorrect	1, when C = 0; otherwise 0
Answer Category, C	Correct	1, when C = 1; otherwise 0
Feedback level, F	F1	Triangular: trim f(F;0,0,25)
Feedback level, F	F2	Triangular: trim f(F;15,25,40)
Feedback level, F	F3	Triangular: trim f(F;35,50,65)
Feedback level, F	F4	Triangular: trim f(F;60,75,85)
Feedback level, F	F5	Triangular: trim f(F;75,100,100)

Answer correctness is modelled as a binary or crisp input rather than a graded fuzzy variable in the present design. Therefor, an incorrect response has full membership in the Incorrect set, whereas a correct response has full membership in the Correct set. Future versions of the PFS may extend this variable to include partially correct responses, syntax errors, logical errors and misconception categories.

The triangular membership functions

$$\mu_{\text{tri}}(x; a, b, c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a < x \le b \\ \dfrac{c-x}{c-b}, & b < x < c \\ 0, & x \ge c \end{cases}$$

The trapezoidal membership functions

$$\mu_{\text{trap}}(x: a, b, c, d) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a < x \le b \\ 1, & b < x \le c \\ \dfrac{d-x}{d-c}, & c < x < d \\ 0, & x \ge d \end{cases}$$

Example, for the medium difficulty function trimf(D; 30, 50, 70)

$$\mu\text{Medium}(D) = \begin{cases} 0, & D \leq 30 \\ \dfrac{D-30}{50-30}, & 30 < D \leq 50 \\ \dfrac{70-D}{70-50}, & 50 < D < 70 \\ 0, & D \geq 70 \end{cases}$$

In the proposed design, triangular membership functions are used for intermediate categories because they are simple to interpret and implement. Trapezoidal membership functions are used for extreme categories, where a range of values is assumed to present clearly easy/complex questions or clearly short/long response times. The membership-function parameters should subsequently be calibrated using expert judgement and empirical learner data.

Fuzzification

Crisp values obtained from the learning platform are converted into degrees of membership between 0 and 1. For example, difficulty level of question, D score of 60 may have partial membership in both the medium and Complex categories. This allows the system to model uncertainty at category boundaries rather than forcing each learner response into a single fixed category.

Rule Evaluation

The rule base contains 18 rules derived from the combinations of three (3) response time categories, three (3) question difficulty categories and two answer categories. Each rule follows in the general form,

IF Time is T_i AND Difficulty is D_j AND Answer is C_k THEN Feedback is F_l

The logical AND operator is implemented using minimum operator. Thus, the firing strength of a rule is

$$\alpha_r = \min[\mu_{T_i}(T), \mu_{D_j}(D), \mu_{C_k}(C)]$$

Where

α_r is the activation strength of rule r.

$\mu_{T_i}(T)$is the membership degree for response time T in linguistic set T_i.

$\mu_{D_j}(D)$ is the membership degree for question difficulty D in linguistic set D_j.

$\mu_{C_k}(C)$ is the membership degree for answer correctness C linguistic set C_k .

Aggregation and defuzzification

The outputs of all activated rules are aggregated using the maximum operator. The aggregated fuzzy output is converted into a single crisp feedback level value using centroid defuzzification,

$$F^* = \frac{\int f\, \mu_F(f) df}{\int \mu_F(f) df}$$

The crisp output value F^* mapped to the feedback-intensity category with the highest membership degree or equivalently to the nearest feedback level centroid.

Work example of Fuzzy Feedback Generation

To illustrate the operation of the proposed PFS, consider a learner answering a programming question on conditional statements. The question has difficulty score of D = 60 on a 0-100 scale. The expected completion time is 120 seconds, while the learner takes 162 seconds.

Therefore, the response-time ratio is

$$T = \frac{162}{120} = 1.35$$

The learner submits an incorrect answer; therefore, answer correctness is presented as C= 0. The crisp input vector is consequently (D, T, C) = (60, 1,35, 0).

Step 1: Fuzzification

Based on your existing Table 5 membership functions:

Question difficulty

- Medium: trim f(D; 30,50,70)
- Complex: trapm f(D; 55, 75, 100, 100)

Response-time ratio

- Optimal: trim f(T;0.6, 1.0, 1.4)
- Maximum: trapm f((T; 1.2, 1.5, 2.0, 2.0)

Answer correctness

- Incorrect: C = 0

The degree of membership for Medium question difficulty is

$$\mu_{Medium}(60) = \frac{70-60}{70-60} = \frac{10}{20} = 0.50$$

The degree of membership for Complex question difficulty is

$$\mu_{Complex}(60) = \frac{60-55}{75-55} = \frac{5}{20} = 0.25$$

The degree of membership for Optimal response time is

$$\mu_{Optimal}(1.35) = \frac{1.40-1.35}{1.40-1.00} = \frac{0.05}{0.40} = 0.125$$

The degree of membership for Maximum response time is

$$\mu_{\text{Optimal}}(1.35) = \frac{1.35-1.20}{1.50-1.20} = \frac{0.15}{0.30} = 0.50$$

Because the learner's answer is incorrect

$\mu_{\text{Incorrect}}(0) = 1.00$

Thus, the learner's response belongs partially to the Medium and Complex difficulty sets, and partially to the Optimal and Maximum response-time sets. This demonstrates why fuzzy logic is appropriate; the learnes's state is not forced into single rigid category.

Step 2: Activated rules and min operator

According to rule base

Activated rule	Antecedents	Consequent	Firing strength
R9	Optimal **AND** Medium **AND** Incorrect	F3	min(0.125,0.50,1.00)=0.125
R11	Optimal **AND** Complex **AND** Incorrect	F4	min(0.125,0.25,1.00)=0.125
R15	Maximum **AND** Medium **AND** Incorrect	F4	min(0.50,0.50,1.00)=0.50
R17	Maximum **AND** Complex **AND** Incorrect	F5	min(0.50,0.25,1.00)=0.25

Four rules are activated. Rule R15 has the highest firing strength, $\alpha_{15} = 0.50$ and supports feedback, level F4. Rule R17 is also activated at 0.25 and support F5. Therefore, the aggregated fuzzy output represents a feedback requirement located primarily in the F4 region with partial support for F3 and F5.

Step 3: Aggregation and F*

After aggregation, centroid defuzzification produces a crisp feedback-intensity score F*=72.4. Given that F4 has the strongest activated output set, the crisp result is expected to lie when or nearest to the F4 region. The system therefore maps F* to feedback intensity F4.

Step 4: Feedback mapping

Because the learner's response is incorrect and the inferred feedback intensity is F4, the feedback-generation module provides error flagging, a concise explanation of the relevant conditional-statement concept, the correct solution strategy, and a similar programming exercise at the same or lower difficulty level. For example

"Your condition does not correctly test whether the input value is greater than or equal to 50. Review the use of the comparison operator ≥. A correct conditional statement is shown below.

```
if (mark >= 50)
        System.out.println ("Pass")
else
        System.out.println("Fail")
```

Now attempt a similar problem before proceeding to the next question."

This example demonstrates the complete transformation from learner interaction data to a personalised feedback action.Conclude the example

Because the learner's response is incorrect and the inferred feedback intensity is F4, the feedback-generation

module provides error flagging, a concise explanation of the relevant conditional-statement concept, the correct solution strategy, and a similar programming exercise at the same or lower difficulty level.

Figure 2, adjacent feedback levels overlap. This overlap enables a student to receive a feedback level that reflects gradual changes in performance rather than an abrupt transition between categories. For example, a

student whose performance lies between the F3 and F4 regions may activate both feedback levels, with the final level determined after rule aggregation and centroid defuzzification.

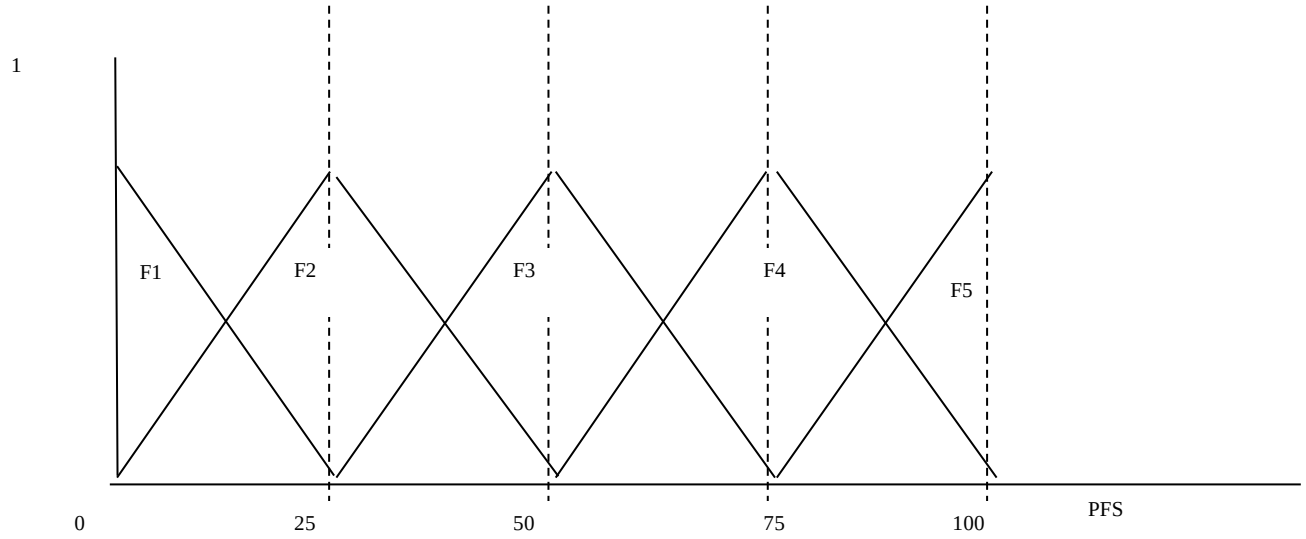


Figure 2: Output membership function

Fuzzy Inference System (Fuzzification Process, Fuzzy Reasoning, And Defuzzification) In Feedback Generation

Figure 3 illustrates the processes implemented in a fuzzy inference system, which includes the fuzzification process, fuzzy reasoning, and defuzzification. This depicts the structure of how information flows from left to right. In a fuzzy logic system, the simultaneous evaluation (parallel) of rules is an important aspect. It is the process of matching any rules that are most appropriate.

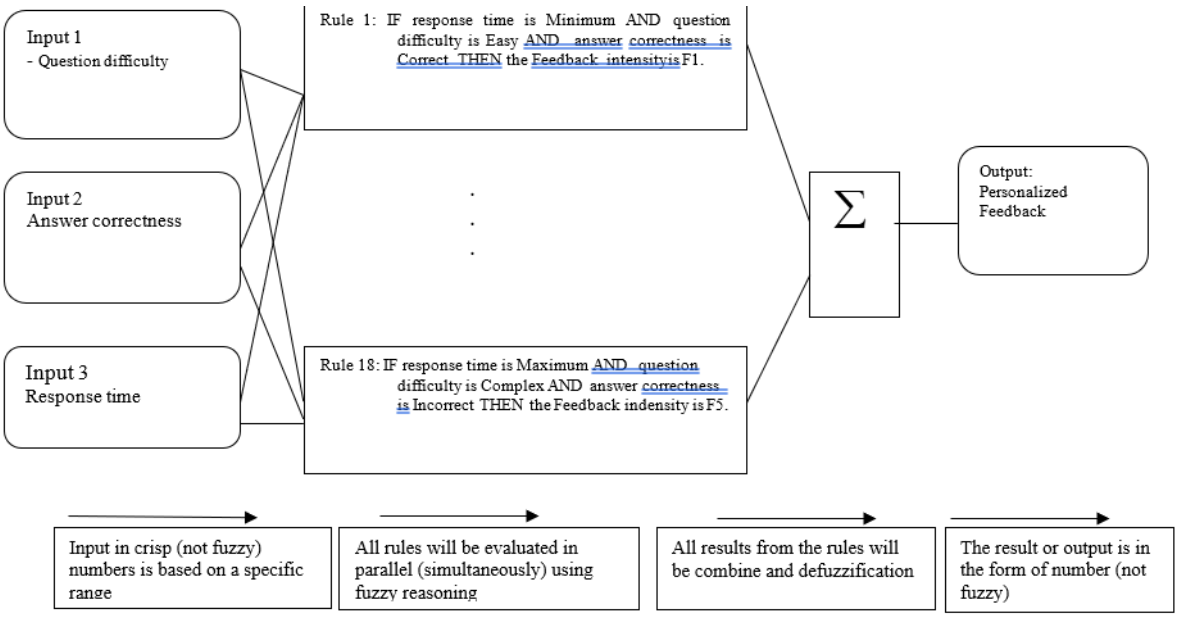


Figure 3: Mamdani fuzzy inference process for feedback generation

LIMITATION AND FUTURE WORK

This study is limited to the conceptual design and formal specification of a fuzzy rule-based personalized feedback mechanism. No software prototype, learner data, usability test or learning outcome experiment was

conducted. The proposed membership-function parameters and fuzzy rules are initial conceptual values and require validation by programming educators, instructional-design expert and fuzzy-system specialists.

Future research should implement the PFS in web-based programming environment and evaluate it with programming learners. A suitable evaluation could compare static feedback, non-adaptive feedback and fuzzy-based personalized feedback. Relevant measures may include programming achievement, response time, number of attempts, error-correction performance, perceived usefulness, usability, learner satisfaction and cognitive load.

CONCLUSION

This paper presented a conceptual design of a Personalized Feedback System (PFS) for programming intelligent tutoring using a Mamdani fuzzy rule-based system. The proposed PFS integrates a student model, an expert model, a fuzzy inference engine and a feedback generation module. The fuzzy mechanism uses question difficulty, response time and answer correctness to infer one of five feedback intensity levels from F1 to F5.

The main contribution of this study is the formal specification of an interpretable feedback selection mechanism, including input and output variables, membership functions, a fuzzy rule base, minimum-maximum inference operations and centroid defuzzification. The feedback generator subsequently maps the inferred intensity level and answer correctness to appropriate feedback actions, such as confirmation, an additional attempt, hints, error flagging, worked examples, conceptual explanation and knowledge of the correct response.

This study is limited to conceptual design and does not provide empirical evidence of usability, learner acceptance or learning effectiveness. Future research should implement the proposed PFS in programming learning environment, validate the membership-function parameters and fuzzy rules through expert review and compare the system with static feedback and non-adaptive feedback conditions. Such evaluation is necessary to determine whether the proposed approach improves programming performance, error correction, engagement or learner satisfaction.

REFERENCES

1. Bangert-Drowns, R. L., Kulik, C. C., Kulik, J. A., & Morgan, M. (1991). The instructional effects of feedback in test-like events. Review of Educational Research, 61(2), 213–238.
2. Fatemeh, A., & Mohammad Masoud, J. (2019). On the influence of using fuzzy extensions in linguistic fuzzy rule-based regression systems. Applied Soft Computing, 79. https://doi.org/10.1016/j.asoc.2019.03.047
3. Hattie, J., & Timperley, H. (2007). The power of feedback. Review of Educational Research, 77(1), 81–112.
4. Ikram, M., Hanefar, S. B. M., Saleem, S. M. U., & Zulfiqar, F. (2026) Artificial intelligence in education: a systematic review of personalized learning trends and future directions. Front. Educ. 11:1782626. DOI: 10.3389/feduc.2026.1782626
5. Karaci, A. (2019). Intelligent tutoring system model based on fuzzy logic and constraint-based student model. Neural Computing and Applications, 31(8), 3619–3628. https://doi.org/10.1007/s00521-017-3311-2
6. Hooda, M., & Rana, C., & Dahiya, O., & Rizwan, A., & Hossain, M. S. (2022). Artificial Intelligence for Assessment and Feedback to Enhance Student Success in Higher Education. Mathematical Problems in Engineering. 2022. 1-19. https://doi.org/10.1155/2022/5215722
7. Kerimbayev, N., Umirzakova, Z., Shadiev, R., & Jotsov, V. (2023). A student-centered approach using modern technologies in distance learning: A systematic review of the literature. Smart Learning Environments, 10, 61. DOI: 10.1186/s40561-023-00280-8

8. Lipnevich, A. A., & Panadero, E. (2021). A review of feedback models and theories: Descriptions, definitions, and conclusions. Frontiers in Education, 6. DOI: 10.3389/feduc.2021.720195 DOI: 10.3389/feduc.2021.720195
9. Maghsudi, S., Lan, A., Xu, J., & van der Schaar, M. (2021). Personalized education in the artificial intelligence era: What to expect next. IEEE Signal Processing Magazine, 38(3), 37–50. DOI: 10.1109/MSP.2021.3055032
10. Mao, P., Cai, Z., Wang, Z., Hao, X., Fan, X., & Sun, X. (2024). The effects of dynamic and static feedback under tasks with different difficulty levels in digital game-based learning. The Internet and Higher Education, 60. https://doi.org/10.1016/j.iheduc.2023.100923
11. Mason, J., & Bruning, R. (2001). Providing feedback in computer-based instruction: What the research tells us. Retrieved from http://dwb.unl.edu/Edit/MasonBruning.html
12. Narciss, S., & Huth, K. (2004). How to design informative tutoring feedback for multimedia learning. In H. M. Niegemann, D. Leutner, & R. Brünken (Eds.), Instructional design for multimedia learning (pp. 181–195).
13. Papakostas, C., Troussas, C., Krouska, A., & Sgouropoulou, C. (2023). PARSAT: Fuzzy logic for adaptive spatial ability training in an augmented reality system. Computer Science and Information Systems. 20. 1389-1417. https://doi.org/10.2298/CSIS230130043P
14. Shute, V. J. (2007). Focus on formative feedback. ETS Research Report Series (RR-07-11).
15. Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. Communications of the ACM, 37(3), 77–84.
16. Zulaikha, M. B., Azlin Norhaini, M., Khairul Azhar, J., & Bity Salwana, A. (2021). The effectiveness and challenges of online learning for secondary school students: A case study. Asian Journal of University Education, 17(3). https://doi.org/10.24191/ajue.v17i3.14514