

Enhancing Learning Outcomes in VR/AR and Robotics Programming Courses Using Real-Time Adaptive Feedback

Sunny Nanade¹, Debasis Dash², Aftab Haider Rizvi³,

^{1,2}MPSTME, SVKM's NMIMS Deemed to be University, Mumbai, India(0000-0001-7098-1084/0000-0002-8101-8696)

³VPSM, DY Patil Deemed to be University, Navi Mumbai, India(0000-0003-2525-2135)

¹sunny.nanade@nmims.edu, ²dashdebasis76@gmail.com, ³aftabhaiderizvi@yahoo.co.in

Abstract— The educational process of teaching programming skills to students in Virtual Reality (VR), Augmented Reality (AR), and Robotics domains creates major obstacles for engineering programs which teach through projects with many students enrolled. Students face challenges when they receive slow, generic feedback during their code review sessions, which prevents them from learning through multiple attempts and understanding programming concepts better. The educational program of this research uses real-time adaptive feedback, which operates as a teaching tool for AR/VR and robotics programming courses to boost student achievement and enable teachers to instruct more students. The intervention received its first implementation during the academic year 2024–25 when three undergraduate engineering courses at the university started teaching their content to 111 students who were enrolled in these courses. Students receive adaptive feedback through their regular programming work because instructors can provide them feedback which matches their course learning objectives and helps them learn. Researchers used a mixed-method evaluation approach, which combined standard coding rubrics with learning analytics and student perception data to measure academic achievement, student involvement, and their belief in their own abilities. The research findings show that students needed 34.1% less time to debug their programs, while their code quality scores improved by 11.3%, and their confidence and competence levels showed substantial progress. The system enabled instructors to reduce their grading duties by 62 per cent while they kept their teaching standards at the same level. The study demonstrates adaptive feedback systems help engineering education programs which teach programming in emerging technology fields through outcome-based methods that work for large student groups and focus on student needs.

Keywords— Adaptive Feedback; Engineering Education; Programming Pedagogy; Virtual Reality; Augmented Reality; Robotics; Learning Analytics; Outcome-Based Education.

I. INTRODUCTION

Every engineering field now requires students to achieve programming skills as their essential educational target because new fields like virtual reality, augmented reality, and robotics demand students to develop advanced mental abilities and technical expertise. Students must develop their algorithmic reasoning abilities through abstract thinking, understand spatial concepts, handle real-time system operations, work with sensors, and conduct multiple testing cycles as part of their academic disciplines. Students must achieve two essential skills for their learning experience, which involves application-based challenges: producing error-free code while mastering complex system operations and advanced debugging methods. The worldwide network of engineering educational institutions continues to face obstacles when teaching advanced programming material to their students who come from various backgrounds and possess different learning abilities. The educational approach of traditional teaching depends on teachers who conduct code evaluations and provide feedback after students have submitted their work. The educational value of this feedback exists, but its ability to handle large numbers of students and provide quick responses becomes problematic because AR/VR and robotics courses need students to perform multiple project iterations for their skill development. Students face obstacles when they receive delayed or generic feedback because this type of feedback prevents them from performing reflection, and their ability to rectify their wrong beliefs and improve their work steadily becomes limited.

Engineering educators now use automated assessment systems, which operate through technology to provide additional

Sunny Nanade

MPSTME, SVKM's NMIMS Deemed to be University, Mumbai, India
sunny.nanade@nmims.edu

evaluation methods that support their teaching feedback. Automated programming assessment systems and intelligent tutoring systems from previous research have shown promise for creating faster and more consistent evaluation processes. Most academic studies about programming education focused on beginner students who learnt text-based coding instead of exploring immersive courses that combined hardware and robotics with unique teaching methods. Although automated programming assessment systems have developed progress, there exists minimal research about their application for providing immediate feedback to students who take immersive robotics engineering courses.

The current research field lacks evidence demonstrating how adaptive feedback systems assist students in achieving learning goals, building their self-assurance, and enabling teachers to reach more students in AR/VR and robotics programming courses conducted in actual classroom environments. The study addresses this research gap by directly assessing a live adaptive feedback system used in various undergraduate engineering courses focused on VR, AR, and robotics. The study investigates how students achieve better results through complete feedback delivery during their programming work, which also boosts their motivation and enables teachers to spend less time with students. The researchers establish adaptive feedback as a teaching tool that functions within a student-centred educational approach and focuses on outcome-based methods for teaching programming in engineering education.

Research Objectives

The primary objective of this research is to:

Examine the effectiveness of real-time adaptive feedback in improving programming-related learning outcomes in VR, AR, and robotics courses.

Evaluate changes in student performance indicators, including code quality and debugging efficiency, within immersive and robotics-based learning environments.

Assess the influence of adaptive feedback on student engagement, confidence, and self-efficacy in learning emerging programming technologies.

Assess the extent to which adaptive feedback can reduce instructors' grading workloads while maintaining pedagogical quality and instructional rigour.

II. LITERATURE REVIEW

The section provides an evaluation of existing studies which examine automated programming assessment systems and machine learning applications for programming education while focusing on the missing elements that affect immersive and robotics-based engineering courses. The review presents these educational elements as an interrelated system which

demonstrates both unaddressed teaching problems and potential for new engineering learning methods.

Automated Programming Assessment in Engineering Education

The field of engineering and computer science education has studied automated systems for evaluating programming assignments since the 1980s, when researchers focused on checking program functionality and correct syntax use. The first systems focused on automated grading because they needed to handle rising student numbers while making teaching easier for staff who worked with beginner programming classes. The evaluation methods have been developed to assess code quality through multiple factors, including performance, code clarity, coding conventions, and software maintenance requirements, in order to achieve educational objectives that extend beyond basic correctness.

Automated assessment systems today let students create better versions of their work because these systems provide students with feedback during their learning process. Research reveals that students who receive quick feedback will build better conceptual knowledge while their mental workload decreases and they gain more autonomy in their learning process. The research field continues to focus on traditional programming environments which use written code to solve simple problems that generate predictable results.

The current assessment frameworks do not solve the educational challenges which programmers face when developing content for AR, VR, and robotics systems. The three domains require systems to operate in actual time while they must connect to physical sensors or virtual environments and experience unpredictable system behaviour, which makes output-based evaluation difficult. Students must develop spatial reasoning, temporal thinking, and system integration skills that exceed the capabilities of automated grading systems when they perform debugging tasks in immersive and robotics environments.

The current research field contains studies which show automated assessment systems fail to perform in complex environments, but researchers have not yet developed assessment methods which work for immersive and robotics programming courses. The present methods fail to show how automated feedback systems can create meaningful learning experiences for students who need to grasp these new subjects.

Machine Learning-Enabled Feedback in Programming Education

Research on intelligent educational systems has undergone major transformations because machine learning technology now drives automated tutoring systems together with adaptive feedback systems and learning analytics tools. The educational

field of programming uses machine learning to classify errors, detect code similarities, identify plagiarism, and predict student performance. Large language models which learn from source code and natural language pairs now demonstrate their ability to create code summaries and find bugs and produce automated hints.

The current feedback systems perform better than rule-based evaluations because they deliver learning assistance, which adapts to each student's needs. Research shows machine learning systems which provide feedback help students become more engaged, while faculty members need to intervene less for repetitive teaching activities that mostly occur in courses with many students. Research into programming education focuses on students who learn basic programming concepts through standardised languages that run in standard development environments; the research mostly studies students at basic or intermediate levels of programming knowledge.

The application of machine learning systems for feedback delivery to AR and VR environments and robotics programming operates with restricted boundaries because researchers have not yet proven its effectiveness through extensive studies. The current body of research focuses more on algorithmic performance and model accuracy instead of studying how educational approaches impact learning outcomes. Research studies handle feedback systems as technical elements, but they fail to connect these systems with course learning objectives and assessment rubrics and outcome-based education frameworks.

The gap between these two fields becomes most apparent when students work with immersive technology and robotics systems, as they encounter different types of errors that arise from environmental setup, live system operation, and the interaction of combined system components. Students will face challenges understanding their mistakes because the feedback systems lack proper domain-specific error detection functionality, which results in their growing frustration and diminished self-assurance. Machine learning provides powerful tools for adaptive feedback, yet researchers have not studied its full potential for teaching engineering students who work in new fields.

Pedagogical Distinctiveness of AR/VR and Robotics Programming

The programming fields of AR and VR and robotics require students to learn differently from conventional programming classes because their instructional methods and student learning activities operate on distinct principles. The courses adopt a project-based approach that requires students to work through multiple iterations while gaining practical experience; pupils must also combine their programming skills with design thinking, spatial reasoning, and system-level problem-solving abilities. Learning outcomes require students to develop their

abilities beyond writing correct code because they must understand debugging techniques and make fast decisions and work together as a team.

The feedback system needs to deliver its messages quickly while providing specific direction because it functions as a learning tool which helps students develop their skills during their current educational activities. The standard method of delayed feedback, which appears in traditional programming courses, fails to work well with immersive learning environments because students need instant feedback when they develop and test their work. The field of engineering education research contains insufficient studies that prove the effectiveness of feedback systems that work with these teaching methods.

Research findings from different studies show that automated assessment systems and machine learning feedback systems have developed separately from programming education, but their combination for robotics and immersive engineering courses remains unexplored. The existing research does not investigate how adaptive feedback systems impact student learning outcomes, confidence in learning, and the ability to manage educational content at various levels.

Research Gap and Positioning of the Present Study

Research shows automated feedback systems that use machine learning technology can help programming education, but scientists have yet to understand how these systems work in AR, VR, and robotics training environments. The field lacks actual research, which studies real-time adaptive feedback as a teaching method that operates in real engineering classrooms with immersive learning experiences. The research field contains only a few studies which evaluate both student learning results and teacher work responsibilities through a single outcome-based evaluation system.

The current study solves this problem by testing a real-time adaptive feedback system which operates in undergraduate courses that teach AR and VR and robotics programming. The study shows how adaptive feedback functions in engineering education through its effects on student learning and classroom involvement and teacher time efficiency, which supports research about programming education methods that help students learn technology at scale.

III. METHODOLOGY

The research follows a design-based research framework that combines different research techniques to study how real-time adaptive feedback systems in AR, VR, and robotics programming classes help students learn better. The research approach focuses on showing how the educational strategy works in classrooms to enhance student understanding and make teaching methods accessible for more educators instead of showing off the system's technical complexity.

Pedagogical Design of the Adaptive Feedback Intervention

The adaptive feedback system helps students learn in programming courses by giving them support while they work on projects that need several rounds of development and fixing errors. The system provides students with immediate feedback during their regular programming work, which shows them specific guidance that supports their course objectives.

The feedback system follows four educational stages which support its operational framework:

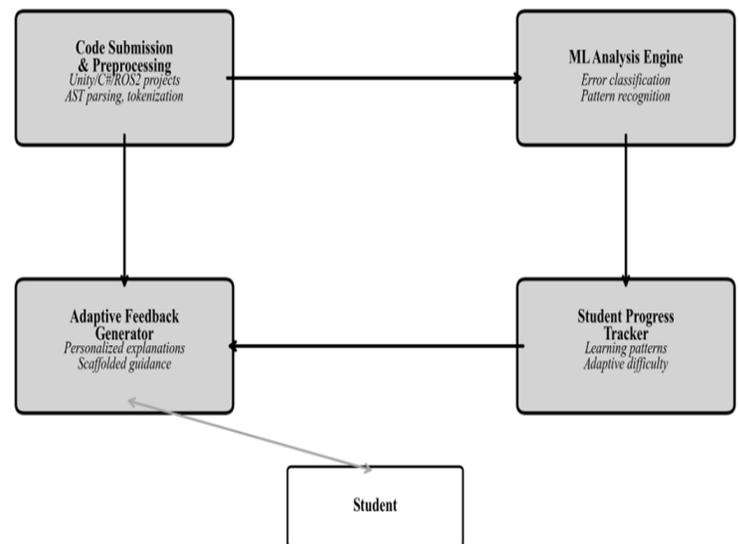
1. Students need to turn in their programming work as part of their standard academic requirements.
2. The system detects frequent student mistakes and incorrect beliefs through its automated error detection process.
3. The system provides students with specific formative feedback which includes explanations and corrective hints and recommended actions to take next, and
4. The system tracks student development, which allows it to modify its support system for students throughout their learning journey.

The system functions to support teacher decisions by handling basic student mistakes so instructors can dedicate their time to teaching essential concepts and design principles. The system provided teachers with summary data from student feedback, which allowed them to identify common student mistakes for creating focused classroom activities.

Real-Time Adaptive Feedback Process

The system creates instant feedback which students receive when they turn in their programming assignments through their work environment. Students in AR and VR courses needed to submit their work through Unity C# projects, but robotics students had to submit their assignments through ROS2-based implementations. The feedback system supports students during their development process because it provides assistance with error interpretation and logic improvement and debugging technique enhancement, which happens before assessment.

A learner progress tracker maintained a record of recurring error categories and resolution patterns at the individual level. The system changes its feedback detail level based on student work progress because it gives detailed support to students who make the same mistakes but provides brief suggestions for material they have already mastered. The method adjusts to student needs through formative assessment and self-regulated learning principles, which help students develop their abilities throughout the semester. Fig. 1 illustrates the pedagogically orientated workflow of the real-time adaptive feedback interventions used in AR, VR, and robotic programming courses.



Source: Authors

Fig. 1. Real-Time Adaptive Feedback Architecture (IVRAR, SMR, MDRIA)

Feedback Classification Approach: Pedagogical Rationale

The system receives feedback messages through a classification system which assigns each message to its proper feedback category for timely delivery of useful information about instructional content. The academic environment requires a model which teaches students through direct methods while keeping maintenance duties simple and delivering quick answers to their questions. The team tested various models capable of handling large data volumes but ultimately deployed a system that delivered accurate results using a model easily understood by both students and teachers, thereby enhancing feedback comprehension. The design choice supports educational methods because classrooms need basic systems which provide clear information instead of complex systems that offer only small technical improvements. Appendix A shows the results of the qualitative evaluation, which compares different feedback classification systems that were considered during system development.

Study Design and Data Collection

The study used a quasi-experimental design that operated through standard educational settings to assess how real-time adaptive feedback systems affect student learning results. The intervention programme took place during the academic year 2024–25, when three engineering programming courses for undergraduate students received their content. The research method allowed learning outcome analysis through typical classroom environments because it maintained the established curriculum and assessment methods.

The researchers collected data through multiple quantitative and qualitative tools, which enabled them to study various aspects of student learning achievement. The assessment process relied on standardised coding rubrics that served as

course evaluation tools, while the research team developed instruments to study programming skill development and debugging speed between the pre- and post-intervention phases. The research team collected qualitative information through student surveys, which asked students about their engagement levels, their confidence, and the usefulness of feedback during learning activities.

The research team evaluated instructional scalability through student performance results and by monitoring the amount of time instructors needed to complete their work. The researchers analysed changes in grades and feedback by measuring the time instructors used to grade programming assignments before and after the adaptive feedback system became available. The approach follows principles of engineering education research that aim to achieve two main objectives: effective student learning and practical teaching methods.

Participants and Educational Context

The research included undergraduate engineering students who took three different courses, which focused on hands-on programming and robotic systems. The courses were chosen because their development process demands multiple iterations, and students must perform continuous error correction while merging programming concepts with running systems. Table I summarises the course context and participants' demographics.

TABLE I
COURSE DETAILS AND PARTICIPANT DEMOGRAPHICS

COURSE	PROGRAMME	SEMESTER	STUDENTS (N)	PRIMARY TECH STACK
INTRODUCTION TO VIRTUAL REALITY AND AUGMENTED REALITY (IVRAR)	OTHER	SEM V	60	UNITY WITH C#
SIMULATED MIXED REALITY (SMR)	B. TECH MECHATRONICS	SEM VI	20	UNITY WITH C#
MODERN DAY ROBOTICS AND INDUSTRIAL APPLICATION (MDRIA)	B. TECH CSBS	SEM III	31	ROS2

Source: Authors

The students in every course needed to create working systems that they would start with and improve on throughout the academic year because the courses followed a project-based learning model. Programming tasks were embedded within practical assignments and laboratory activities rather than isolated exercises. The assessment system received adaptive feedback interventions that functioned without altering any course learning goals, material content, or evaluation systems. The study achieved ecological validity through integration,

which enabled research to take place in authentic engineering educational environments.

Validity and Limitations

The researchers' findings require readers to understand specific constraints that affect their ability to analyse the study's results. The research team selected their study groups through standardised academic enrolment instead of using random selection methods. The research method produces authentic environmental results, but it restricts the ability to discover cause-and-effect connections. The program started its operations at a particular school, which used its current curriculum to teach students about AR, VR, and robotic programming. The research results need particular changes before they become applicable for every programming and engineering field because they stem from a specific context.

The way teachers taught students likely affected the learning results which students achieved. The adaptive feedback system operated independently from teaching activities, yet instructors needed to lead laboratory work and project evaluation sessions. The research subjects interacted with each other, which probably led to the observed results, so scientists need to think about this when they examine their results. The study delivers important actual data about how adaptive feedback, which works in real time assists students in their learning process and helps them stay involved while making it easier for teachers to teach engineering through robotics and immersive educational methods.

IV. RESULTS

This section presents the results of the adaptive feedback intervention across three undergraduate engineering programming courses. Findings are organised around learning outcomes central to engineering education—code quality, debugging efficiency, and student self-efficacy—followed by an education-focused interpretation of each result.

Improvements in Code Quality

The assessment of code quality involved a combined evaluation system that checked for correct operation, optimised performance, easy maintenance, and well-documented code, all in accordance with engineering education outcome-based assessment methods. The three courses showed clear progress in their results after the adaptive feedback system was implemented, according to Fig. 3 and Table III.

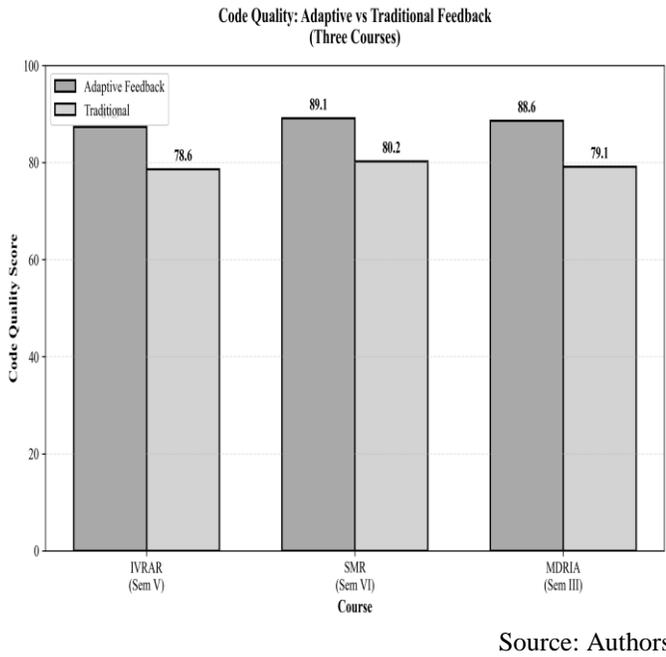


Fig. 2. Code Quality Comparison Across Three Courses

TABLE III
CODE QUALITY METRICS BY COURSE

COURSE	ADAPTIVE FEEDBACK (POST)	BASELINE (PRE)	DIFFERENCE
IVRAR (SEM V)	87.3	78.6	+8.7
SMR (SEM VI)	89.1	80.2	+8.9
MDRIA (SEM III)	88.6	79.1	+9.5

Source: Authors

Across courses, mean code quality scores increased from 79.0 (pre-intervention) to 88.0 (post-intervention), representing an overall improvement of 11.3%. Course-wise gains ranged from +8.7 to +9.5 points, indicating consistent benefits across immersive and robotics-based programming contexts. From an engineering education perspective, this improvement indicates that timely, formative feedback supports deeper engagement with programming practices beyond functional correctness. Students demonstrated greater attention to code structure, readability, and documentation—competencies essential for professional engineering practice but often underemphasised in traditional assessment models.

Reduction in Debugging Time

Debugging efficiency was examined using the average time between successive code submissions as a practical proxy for debugging effort. As illustrated in Fig. 4 and Table IV, substantial reductions in debugging time were observed across all three courses.

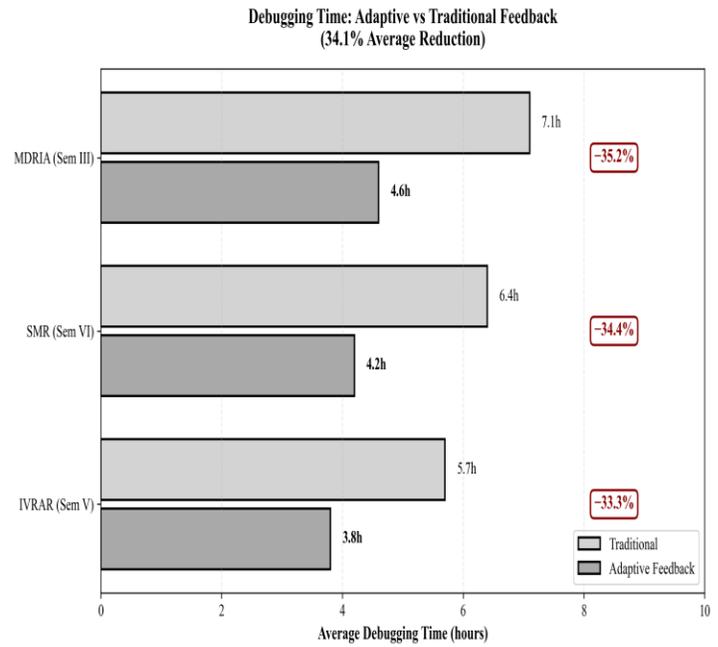


Fig. 3. Debugging Time Reduction (34.1% average)

TABLE IV
AVERAGE TIME BETWEEN SUBMISSIONS (DEBUGGING TIME PROXY)

COURSE	ADAPTIVE FEEDBACK (POST, HRS)	BASELINE (PRE, HRS)	REDUCTION (%)
IVRAR (SEM V)	3.8	5.7	33.3
SMR (SEM VI)	4.2	6.4	34.4
MDRIA (SEM III)	4.6	7.1	35.2
AVERAGE	4.1	6.2	34.1

SOURCE: AUTHORS

The time span between submissions showed a reduction from 6.2 hours before the intervention to 4.1 hours after the intervention, which represents a 34.1% decrease. The intervention resulted in course grade decreases ranging from 33.3% to 35.2% across different programming environments and student groups. The decrease shows that students in engineering education now understand how to analyse mistakes while developing skills to identify problems and find solutions for themselves. The speed of debugging cycles shows how well students learn through their work experience because they need less time to solve basic problems, which enables them to focus on complex design work and problem-solving activities.

Enhancement of Programming Self-Efficacy

Student self-efficacy in programming was measured using a standardised 7-point Likert scale administered before and after the intervention. The results, summarised in Table V, show consistent increases in perceived competence across all courses.

TABLE V
PROGRAMMING SELF-EFFICACY CHANGES (SCALE 1-7)

COURSE	PRE-	POST-	GAIN
	INTERVENTION	INTERVENTION	
IVRAR	3.7 (SD=1.3)	5.0 (SD=1.2)	+1.3
SMR	3.8 (SD=1.1)	5.2 (SD=1.0)	+1.4
MDRIA	3.9 (SD=1.2)	5.1 (SD=1.1)	+1.2

Source: Authors

The average self-efficacy ratings showed an upward trend between 1.2 and 1.4 points after the intervention, which resulted in all courses recording scores above 5.0. The students developed higher confidence levels when they learned to build immersive interactions in Unity (C#) and operate robotic systems with ROS2. The educational field of engineering shows that adaptive feedback systems create two important benefits for students who work on challenging engineering assignments. People who want to stay involved with new technology need to build their self-confidence because they face steep learning challenges, which would otherwise make them give up trying new things.

Summary of Learning Outcomes

The research findings show that the adaptive feedback system produced beneficial results, which enhanced various aspects of student learning development. Students who received real-time adaptive feedback in immersive robotics programming courses showed better code quality and shorter debugging times and developed strong programming self-efficacy, which demonstrates the educational value of this approach. Engineering education benefits from these results because they demonstrate how adaptive feedback systems enable outcome-based learning and student-focused education and scalable programming instruction for new technology fields.

V. DISCUSSIONS

The research investigated the educational results, which stem from an adaptive feedback system that operates in real time during undergraduate courses about AR and VR and robotics programming. The research indicates that students achieve better learning results when they receive prompt feedback, which teachers deliver as personalised guidance throughout their complex, project-based programming. The combination of code quality improvements with better debugging results and increased programming confidence shows that adaptive feedback enables students to learn better through both their thinking abilities and their emotional responses in new engineering fields.

The debugging process experiences its most severe time reduction when students work with AR/VR and robotics systems because these systems produce nonlinear errors which stem from their spatial organization and real-time operations and complex system integration. Students who receive real-time adaptive feedback develop better error interpretation skills while gaining greater confidence in implementing their correction methods. Students discovered solutions to common problems on their own, which enabled teachers to dedicate more time to teaching complex concepts and solving design-based problems.

The improvement in code quality shows that adaptive feedback helps students develop their engineering skills, which include maintenance, documentation, and organisational problem-solving methods. The three competencies are essential for engineers to practise their professions, as outcome-based education frameworks emphasise these areas; however, students struggle to master them in final tests. The student's self-efficacy growth shows that adaptive feedback methods help students manage their programming task anxiety when they learn new technological skills, which they find challenging.

A. Implications for Engineering Educators

From an instructional perspective, the findings emphasise the possible use of adaptive feedback to augment teaching effectiveness in large and heterogeneous classrooms. The delivery of quick personal feedback becomes a major obstacle which engineering educators must face when they teach programming-focused courses. The study shows how the educational approach lets teachers automate their review process, yet they keep their vital teaching role intact.

Teachers can direct their attention to student guidance and design review and concept explanation through the system, which identifies student mistakes as they occur during their work. The educational approach supports learning through direct feedback, which students obtain during their learning activities instead of receiving their feedback after completing assessments. The educational practice of adaptive feedback provides teachers who teach new technology courses with a useful tool which helps them control their teaching duties while they keep their students' academic standards at the correct level.

B. Implications for Curriculum Design and Outcome-Based Education

The research findings provide essential details which will shape educational programs that follow Outcome-Based Education (OBE) principles. The learning results achieved by students validate the programme's outcomes, which require students to analyse problems, create solutions, and maintain their learning development throughout their lives. Students receive ongoing assessments from adaptive feedback, which allows them to make continuous improvements to their work until they reach the specified learning goals.

The intervention offers a technology-based system which helps students reach their course goals through evidence-based methods that satisfy NBA accreditation standards and Washington Accord requirements. The adaptive feedback system's learning analytics produce data, which accreditation frameworks require for outcome documentation, continuous improvement, and curriculum evaluation through data analysis.

C. Scalability and Institutional Adoption

Three separate courses demonstrated the expansion of adaptive feedback systems to serve various emerging technology fields, achieving the intervention's goals. The educational difficulties examined in this research related to AR, VR, and robotics

programming also occur in Internet of Things (IoT) education, autonomous systems instruction, data science training, and computer vision teaching. The research findings demonstrate that adaptive feedback systems can operate with current educational programs because they do not need major modifications, which makes them suitable for schools that want to enhance their programming education through expandable solutions. Adaptive feedback systems benefit the institutional level by enabling better teaching and learning results, allowing faculty members to use their time more effectively, and increasing student satisfaction with their educational experience. The advantages provide engineering schools with solutions which help them handle their expanding student body and students who come from different backgrounds.

D. Limitations and Directions for Future Research

The research presents valuable insights, but it faces multiple restrictions that affect its overall quality. The study took place during one academic year while examining only several course groups at a single university, which creates challenges for applying results to different settings. The study design follows a quasi-experimental approach with non-randomised groups, which creates obstacles to proving direct causal connections. The researchers investigated short-term learning outcomes, but they did not study how students would keep their skills and apply them to different situations after the course ended. Research should build on this work by conducting multi-institutional studies, which should also include longitudinal research and engineering field comparisons. The research base for adaptive feedback in engineering education will become stronger through studies that examine its effects on design thinking, teamwork, and professional readiness development.

REFERENCES

- Wing, J. M. (2006). "Computational thinking." *Communications of the ACM*, 49(3), 33-35.
- Robins, A., Rountree, J., & Rountree, N. (2003). "Learning and teaching programming: A review and discussion." *Computer Science Education*, 13(2), 137-172.
- Edwards, S. H., & Perez-Quinones, M. A. (2008). "Web-CAT: Automatically grading programming assignments." *ACM SIGCSE Bulletin*, 40(3), 328-328.
- Hollingsworth, J. (1960). "Automatic graders for programming classes." *Communications of the ACM*, 3(10), 528-529.
- Ala-Mutka, K. M. (2005). "A survey of automated assessment approaches for programming assignments." *Computer Science Education*, 15(2), 83-102.
- Feng, Z., et al. (2020). "CodeBERT: A pre-trained model for programming and natural languages." *Findings of EMNLP*, 1536-1547.

APPENDIX

A: Comparison of Feedback Classification Approaches Considered

CONCLUSION

The research analysed how real-time adaptive feedback functions as an educational tool that students in undergraduate AR, VR, and robotics programming courses experience. The solution addresses three major obstacles that engineering students face in their projects by providing customised feedback during their standard programming exercises. The study results show that all courses achieved ongoing progress with their code quality, while they needed fewer hours for debugging, and their students developed stronger programming confidence.

Teachers need to follow adaptive feedback methods because research proves they are most effective for teaching outcome-based programming in student-centred learning environments. Students who worked on maintainability and documentation and debugging tasks showed better understanding of professional engineering methods than when they focused exclusively on making their programs work. The system, which reduced instructor grading workload, maintained teaching quality through its ability to boost educational productivity while protecting teachers from their traditional roles as learning mentors and student development facilitators.

The engineering field experiences constant transformation because AR, VR, and robotics technologies force educational institutions to create sustainable training systems, which proves their effectiveness in teaching specialised programming abilities. The study evaluates an educational model which enables teachers to add adaptive feedback to their current teaching materials while following outcome-based education requirements and accreditation standards. Researchers need to continue this study through multi-institutional and longitudinal research, which will test adaptive feedback performance with various emerging technology systems and determine its effects on student ability retention and professional development success.

Approach	Latency	Interpretability	Maintenance	Notes
Rule-based (regex + static rules)	Very low	High	Medium	Fast, but brittle for novel errors
Logistic regression (deployed)	Low	High	Low	Stable, fast, supports template routing
Linear SVM	Low-Medium	Medium	Medium	Strong margin-based classifier; harder to explain