ISSN 2522-6398 Vol. 9, Issue 5: 181-186, DOI: 10.25236/FER.2026.090525

Research on a Personalized Teaching Evaluation Framework for Data Structures in Smart Teaching Environments

Xin Wang[1], Juxiang Wang[2,*]

[1]School of Economics and Management, Anhui Jianzhu University, Hefei, 230601, China
[2]School of Mathematics and Physics, Anhui Jianzhu University, Hefei, 230601, China
**Corresponding author: wjxahjzdx@163.com*

Abstract: *Smart teaching environments provide online learning records, classroom interaction, assignments, quizzes, laboratory code, and learning feedback for course evaluation. However, end-to-end audio-video inference requires high computing power, bandwidth, annotation, privacy protection, and governance capacity, making it difficult to sustain in ordinary classrooms. This paper proposes a lightweight personalized teaching evaluation framework for the Data Structures course. The framework transforms multi-source process records into structured pedagogical indicators and builds a closed loop of data collection, learning diagnosis, personalized intervention, and effect feedback through indicator integration, student profiling, risk identification, and resource recommendation. The study argues that rule-guided, interpretable evaluation can lower the application threshold of smart teaching and provide an operable path for improving Data Structures teaching practice.*

Keywords: *Smart teaching; Personalized teaching evaluation; Learning analytics; Student profile; Data structures*

1. Introduction

Data Structures is a foundational course for computer-related majors. It involves abstract concepts, logical reasoning, closely connected knowledge points, and programming practice[1]. Traditional evaluation based mainly on final examinations and staged tests often fails to locate students' difficulties in modules such as linear lists, stacks and queues, trees, graphs, searching, and sorting in time. With the development of smart classrooms and online learning platforms, course teaching is moving toward online-offline integration[2]. Platforms such as Xuexitong can record attendance, responses, discussions, submissions, resource access, and quiz scores, while laboratory platforms can record code submissions, errors, and test case pass rates[3].

In practice, however, data that are available are not necessarily usable. The learning process of Data Structures includes conceptual understanding, algorithm tracing, code implementation, debugging, and integrated application. A single score cannot show whether a student's problem lies in abstract structure understanding, algorithm process tracing, programming implementation, or knowledge transfer. Meanwhile, direct analysis of classroom audio, video, or facial expressions through end-to-end deep learning is costly and difficult to govern. A more realistic path is to convert low-cost process records from different teaching scenarios into structured indicators with clear teaching meaning.

This paper proposes a personalized teaching evaluation framework for Data Structures in smart teaching environments. It focuses on operational indicator definitions, interpretable diagnosis for small classes, course-specific student profiles, and data governance boundaries designed together with the framework.

2. Theoretical Foundations and Problem Analysis

Smart teaching uses information technology to support learning resources, process tracking, teaching activities, and evaluation[4]. Learning analytics collects and interprets learner data to optimize learning processes. Personalized teaching further requires that content, resources, and feedback adapt to students' current levels and developmental needs; timely feedback and the zone of proximal development provide theoretical support for such adaptive intervention[5-6].

ISSN 2522-6398 Vol. 9, Issue 5: 181-186, DOI: 10.25236/FER.2026.090525

For Data Structures, evaluation should not stop at an overall score. The course involves conceptual understanding, algorithm tracing, code implementation, debugging, and complexity analysis. Existing evaluation faces delayed summative feedback, scattered process data, and insufficiently interpretable results. Therefore, evaluation needs to shift from single-score judgment to structured process diagnosis, student profiling, and continuous intervention[7].

3. Multi-Source Data Collection and Indicator System

3.1 Data Sources

Smart teaching data for Data Structures include five categories: classroom session statistics, teaching materials and learning resources, classroom interaction, assignments and quizzes, and student feedback with teacher observation. Classroom audio-video data are not used for facial expression, face recognition, individual gaze tracking, or attention recognition by default. If smart classroom devices are available, they only provide session-level indicators such as lecture duration, questioning frequency, number of student speeches, and activity rhythm. Other records include resource visits, video replays, in-class responses, assignment submissions, code errors, test case pass rates, self-reported difficulties, and teaching reflections; among them, code submissions and automated feedback records are especially useful for analyzing programming difficulties[8].

3.2 Design of the Indicator System

For ease of pedagogical interpretation, this paper transforms process records from different teaching scenarios into six categories of structured indicators. For ease of pedagogical interpretation, this paper transforms process records from different teaching scenarios into six categories of structured indicators, as shown in Table 1.

Table 1: Indicator System for Personalized Evaluation in the Data Structures Course

Indicator category	Main data sources	Example indicators	Pedagogical meaning
Learning participation indicators	Attendance, classroom interaction, discussion forums	Attendance rate, number of interactions, number of speeches, in-class response participation rate	Reflect participation in classroom and online learning activities
Knowledge mastery indicators	In-class quizzes, assignments, staged tests	Knowledge-point accuracy, item-type scores, conceptual question error rate, comprehensive question scores	Reflect students' mastery of core course knowledge points
Learning process indicators	Xuexitong, resource platforms, code platforms	Video viewing duration, number of replays, assignment submission delay, number of code resubmissions	Reflect learning pace, engagement, and persistence of difficulties
Practical ability indicators	Programming assignments, laboratory reports, project tasks	Code pass rate, number of debugging attempts, completeness of algorithm implementation, complexity analysis score	Reflect students' ability to transform theoretical knowledge into program implementation
Classroom state indicators	Session-level smart classroom statistics, classroom interaction	Teacher questioning frequency, student response time, number of speeches, classroom activity transitions	Reflect classroom organization and interaction rhythm; learning state is inferred only from behavioral proxies
Learning affect indicators	Questionnaires, interviews, learning feedback	Learning interest, self-efficacy, learning pressure, self-assessed difficulties	Reflect students' learning motivation and emotional state

The above indicators do not need to be collected completely at the initial stage. A "low-cost first,

ISSN 2522-6398 Vol. 9, Issue 5: 181-186, DOI: 10.25236/FER.2026.090525

gradual expansion" strategy can be adopted: Xuexitong records, assignments, quizzes, laboratory code, and classroom interaction are integrated first; audio-video data, if used, are limited to aggregated session statistics and are not used for expression or face-based attention recognition.

3.3 Data Processing and Diagnostic Methods

Process records from different platforms are scattered, temporally inconsistent, and heterogeneous. Students, knowledge points, and teaching weeks should therefore be used as basic indexes; missing values, outliers, and duplicate records should be cleaned; and indicators should be standardized or graded.

This paper does not advocate stacking complex models in small-class scenarios. A Data Structures class usually contains only 30 to 60 students, and one semester provides limited labeled records. Complex models easily overfit and are difficult to explain. Therefore, the diagnostic strategy is "interpretable rules as the main approach and lightweight models as supplements."

Interpretable rules are used for risk identification and resource matching. For example, low in-class accuracy with low assignment scores indicates weak knowledge mastery; repeated late submission with low code pass rates indicates engagement risk; frequent compilation errors with failed tests indicate code implementation difficulty. Each rule corresponds to micro-lessons, question-answering, or tiered exercises.

Threshold determination should avoid both pure teacher judgment and blind quantification. In a class of 30 to 60 students, the 25th, 50th, and 75th percentiles are statistically unstable and may vary across semesters. Therefore, percentiles from the same course and knowledge point in previous cohorts are used only as initial anchors, preferably pooled across recent cohorts when available. Before the semester begins, the teaching research group reviews the thresholds, records the sample size and adjustment basis, and forms a threshold list. In the middle of the semester, warning cases judged as false positives or false negatives are reviewed, and thresholds may be adjusted proportionally, but no more than twice within a semester. In this way, percentile thresholds serve as transparent starting values rather than fixed governance standards.

On top of rules, a small number of lightweight models can be added as supplements, such as shallow decision trees or logistic regression models. Model outputs serve only as reference evidence and do not override rule-based results. Feature importance should be printed for review. If a model judgment conflicts with rules, the rule-based result prevails and teachers conduct manual review.

3.4 Operational Definitions of Key Indicators

To improve repeatability, this paper defines several key indicators. Knowledge-point accuracy is calculated after each item is tagged with one to three knowledge points; unanswered items caused by absence are treated as missing rather than wrong. Code resubmissions count only valid compilable submissions and merge identical or accidental repeated submissions; suspected plagiarism is separately marked for manual review. Video replay counts non-sequential returns to the same segment and reflects active review rather than misoperation. Algorithm tracing scores preset intermediate steps in traversal, shortest path, or sorting questions, while complexity analysis is scored separately from code pass rate. Learning participation is divided into levels based on pooled historical distributions or reviewed thresholds, reducing sensitivity to short-term fluctuation.

4. Personalized Teaching Evaluation Framework

The evaluation framework proposed in this paper consists of six parts: the data layer, feature layer, rule-guided inference layer, diagnosis layer, intervention layer, and feedback layer.

4.1 Data Layer and Feature Layer

The data layer collects records from smart classrooms, Xuexitong, assignment platforms, quiz systems, and student feedback, and links them to students, knowledge points, and teaching activities. For classroom audio-video data, minimum necessity is followed: only statistical indicators required for teaching diagnosis are extracted, and long-term storage of raw sensitive data is reduced[9]. The feature layer then transforms records into interpretable indicators such as knowledge-point accuracy, video replay counts, code resubmissions, test case pass rates, and session-level classroom state indicators.

ISSN 2522-6398 Vol. 9, Issue 5: 181-186, DOI: 10.25236/FER.2026.090525

4.2 Rule-Guided Inference Layer

The inference layer evaluates knowledge mastery, risk level, and resource matching based on structured features. Its core is rule-guided diagnosis supplemented by lightweight models. Outputs are not final judgments of students' ability, but auxiliary evidence for teacher review, especially for high-risk students.

4.3 Diagnosis, Intervention, and Feedback Layers

The diagnosis layer forms individual student profiles and class profiles, describing knowledge mastery, participation, learning pace, practical ability, affective state, classroom interaction rhythm, common errors, and teaching difficulties. The intervention layer recommends micro-lessons, graphical explanations, tracing exercises, code templates, debugging guidance, tiered laboratories, or extension tasks according to diagnostic results. The feedback layer compares performance before and after intervention and supports continuous optimization of indicators, recommendation rules, and teaching design[5].

4.4 Data Governance and Ethical Boundaries

Smart teaching evaluation involves sensitive information such as student identity, learning behavior, classroom audio-video records, and feedback opinions. Governance rules must therefore be designed together with the framework itself[10]. First, students should receive an informed consent form in the first week, including data categories, purposes, retention periods, access permissions, and withdrawal methods. Students may refuse classroom audio-video collection or withdraw from profile analysis without affecting grades or learning rights.

Second, minimum collection should be followed. Classroom audio-video data do not support facial recognition or expression recognition by default. Only session-level indicators, such as teacher lecture duration, number of student speeches, questioning frequency, and interaction rhythm, are extracted. Raw videos are deleted within the semester following the course, and only aggregated structured indicators are retained. Platform behavior data are used only for this course and are not merged with non-teaching scenarios.

Third, raw data are stored on on-campus servers and de-identified through hashed student IDs and teaching class numbers. Course instructors can view profiles of their own students; teaching research group leaders can view de-identified class-level data; students can view and correct their own profiles; administrators can only obtain statistical results. Evaluation results are used for learning support, resource recommendation, and teaching improvement, not for scholarships, rankings, or public comparison.

5. Application of Personalized Evaluation in the Data Structures Course

5.1 Course Knowledge Modules and Evaluation Focuses

Fine-grained evaluation objects can be constructed according to the knowledge modules of the Data Structures course. Fine-grained evaluation objects can be constructed according to the knowledge modules of the Data Structures course, as shown in Table 2.

Table 2: Knowledge Modules and Evaluation Focuses of the Data Structures Course

Knowledge module	Learning difficulties	Observable data	Diagnostic focus
Linear lists	Conversion between sequential storage and linked storage; insertion and deletion operations	In-class questions, classroom response time, programming assignments, code error records	Determine whether students understand the difference between logical structures and storage structures
Stacks and queues	Recursion, expression evaluation, queue applications	Classroom responses, interaction accuracy, laboratory submissions, test case pass rates	Determine whether students can use abstract structures for problem solving
Trees and	Traversal processes,	Video replays, graphical	Determine whether

ISSN 2522-6398 Vol. 9, Issue 5: 181-186, DOI: 10.25236/FER.2026.090525

Knowledge module	Learning difficulties	Observable data	Diagnostic focus
binary trees	recursive thinking, Huffman trees	question scores, number of programming resubmissions, self-reported recursion difficulty	students understand hierarchical structures and recursive processing
Graphs	Graph storage, traversal, shortest paths, minimum spanning trees	In-class quizzes, comprehensive question scores, discussion records, classroom interaction rhythm	Determine whether students can handle complex relational structures
Searching and sorting	Algorithmic processes, complexity analysis, stability comparison	Algorithm tracing questions, complexity questions, laboratory reports, self-assessed difficulty and confidence	Determine whether students can compare applicable scenarios of algorithms

Classroom state indicators and learning affect indicators are used as cross-module auxiliary evidence in this table. They help teachers judge whether a difficulty is related to interaction rhythm, learning pressure, or confidence, but they do not independently determine students' mastery levels.

5.2 Profile Construction and Intervention

In the "trees and binary trees" module, repeated replay of recursion videos, low traversal quiz accuracy, incorrect recursion termination conditions, frequent code resubmissions, and self-reported difficulty in writing recursive code together indicate that the main problem may be recursive process modeling rather than concept memorization. If session-level classroom statistics also show fewer student responses, longer response time, or lower interaction accuracy during the recursion segment, they can serve as auxiliary evidence rather than independent attention judgments. The system can then recommend recursive execution review, visual tracing, guided code completion, and complete programming tasks.

Tiered intervention can be organized by risk level. Low-risk students receive extension tasks; medium-risk students receive targeted micro-lessons and exercises; high-risk students receive teacher tutoring, learning plan reconstruction, peer support, and staged tracking. Class profiles can also support teaching adjustment. For example, if graph shortest-path errors concentrate on relaxation and path updates, and classroom state indicators show longer response time or fewer effective interactions during this segment, teachers can add dynamic demonstrations, group tracing, and guided questioning instead of only increasing exercises.

5.3 Differences from General LMSs and Learning Analytics Platforms

General LMSs can provide attendance, resource access, quiz scores, assignment submission, and learning dashboard functions. Some platforms also support item tags or learning outcome objects. However, their data boundaries are usually centered on online activity records, and they provide limited coverage of offline classroom interaction, laboratory debugging, teacher observation, and immediate student feedback.

The course-specific value of this framework lies in three aspects. First, data are organized around Data Structures knowledge points rather than platform activities, forming a student-knowledge-point mastery matrix. Second, programming errors, algorithm tracing, complexity analysis, and code resubmission are separately recorded, making it possible to distinguish conceptual, procedural, analytical, and implementation difficulties. Third, each diagnostic result is linked to a teaching action, such as micro-lessons, tracing exercises, debugging guidance, or extension tasks. Therefore, the framework goes beyond grade summaries and supports concrete teaching adjustment.

6. Implementation Evaluation and Optimization Path

One Data Structures class can be selected as the pilot class and one parallel class as the control class. The pilot class adopts structured process evaluation and personalized intervention, while the control class adopts conventional teaching methods. Weekly reports include knowledge-point mastery, at-risk students, common errors, and teaching suggestions. Effects can be evaluated through learning outcomes, learning processes, and subjective experience, including test scores, laboratory performance, participation,

ISSN 2522-6398 Vol. 9, Issue 5: 181-186, DOI: 10.25236/FER.2026.090525

submission timeliness, code pass rates, learning interest, and system usability. The pilot should gradually form a knowledge-point indicator library, weekly profile update mechanism, and resource recommendation library. The framework does not replace teacher judgment; it helps teachers identify risks earlier and allocate teaching resources more effectively.

7. Discussion

Compared with general learning analytics solutions, this framework integrates online platform data, classroom interaction, and laboratory practice into course evaluation semantics. It makes three trade-offs: interpretable rules are prioritized over complex models in small-class settings; key indicators such as code resubmission, video replay, algorithm tracing, and complexity analysis are operationally defined; and governance requirements such as informed consent, minimum collection, retention periods, access permissions, and use limitations are built into the framework.

This study remains a framework design and has not yet completed large-scale empirical testing. Thresholds and rules still need validation in real classes, especially because small-class percentile thresholds are unstable. Future research should compare pilot and parallel classes, examine the predictive effects of different indicators, and explore low-disturbance ways to collect affective data.

8. Conclusion

In response to the needs of personalized teaching evaluation in Data Structures, this paper proposes a lightweight framework integrating online, classroom, and laboratory process data. Its contributions are operational indicator definitions, rule-guided diagnosis for small classes, and profile and intervention designs aligned with the course knowledge structure. Future work will test threshold stability, rule-model combinations, and the effect of profile feedback in real teaching classes.

Acknowledgement

This work was supported by the Anhui Provincial Quality Engineering Program for Higher Education Institutions (2023jyxm0437, 2024jyxm0255, 2024dzxkc047); and the Key Teaching Research Project of Anhui Jianzhu University (2023jy04).

References

[1] Yan W M, Wu W M. Data Structures (C Language Edition)[M]. Beijing: Tsinghua University Press, 2007.
[2] Singh H, Miah S J. Smart education literature: A theoretical analysis[J]. Education and Information Technologies, 2020, 25(4): 3299-3328.
[3] Gu X Q, Zhang J L, Cai H Y. Learning analytics: The emerging data technology[J]. Journal of Distance Education, 2012(1): 18-25.
[4] Demir K A. Smart education framework[J]. Smart Learning Environments, 2021, 8(1): 29.
[5] Wisniewski B, Zierer K, Hattie J. The power of feedback revisited: A meta-analysis of educational feedback research[J]. Frontiers in Psychology, 2020, 10: 3087.
[6] Margolis A A. Zone of proximal development, scaffolding and teaching practice[J]. Cultural-Historical Psychology, 2020, 16(3): 15-26.
[7] Bernacki M L, Greene M J, Lobczowski N G. A systematic review of research on personalized learning: Personalized by whom, to what, how, and for what purpose(s)[J]. Educational Psychology Review, 2021, 33(4): 1675-1715.
[8] Marwan S, Akram B, Barnes T, Price T W. Adaptive immediate feedback for block-based programming: Design and evaluation[J]. IEEE Transactions on Learning Technologies, 2022, 15(4): 406-420.
[9] Tzimas D, Demetriadis S. Ethical issues in learning analytics: A review of the field[J]. Educational Technology Research and Development, 2021, 69(2): 1101-1133.
[10] Jones K M L, Asher A, Goben A, Perry M R, Salo D, Briney K A, Robertshaw M B. "We're being tracked at all times": Student perspectives of their privacy in relation to learning analytics in higher education[J]. Journal of the Association for Information Science and Technology, 2020, 71(9): 1044-1059.