Data Ethical and CyberSecurity
WISDOM ACADEMIC ISSN: 3104-6479 | E-ISSN: 3104-6487 Volume 1 , Issue 3

Article

User Behavior Data-Driven Interface Optimization Design Research for Computer Science Learning Platforms

Meilin Shou*
University of Toronto, Toronto, ON M5S 1A1, Canada.
***Corresponding author:** Meilin Shou, meilinshou@gmail.com.

CITATION

Shou ML. User Behavior Data-Driven Interface Optimization Design Research for Computer Science Learning Platforms. Data Ethical and CyberSecurity. 2025; 1(3): 237.

https://doi.org/10.63808/decs.v1i3.237

Abstract: This work presents a new data-driven methodology to promote the effectiveness of computer science education platform user interfaces through behavioral analysis. Through analysis of the interactive behaviors of a sample of 2,847 students on three of the most popular platforms, we uncovered key behavioral indicators of learning success. The mixed-methods methodology used machine learning algorithms to process click-stream data, navigation patterns, and engagement measurements to identify meaningful correlations between interface design elements and educational measures. The optimization framework proposed by us translated to a 34% increase in task completion and a 27% increase in retention of the learned material. Interfaces that dynamically adjust to students' behavioral patterns outperformed static interfaces, especially among novice programmers. The results contribute to the theoretical discourse in human-computer interaction and to practical design advice for developers of education technology.

Keywords: user behavior analytics; interface optimization; computer science education; e-learning platforms; adaptive UI

1. Introduction

The rapid growth of online computer science education websites has revolutionized programming instruction worldwide. The increasing number of students using platforms like Codecademy, Coursera, and edX has increased demand for carefully designed interfaces that are tailored to accommodate multiple modes of learning and varying levels of proficiency [1]. Furthermore, the COVID-19 pandemic has accelerated the shift to online learning, offering unique opportunities to inform the improvement of these interfaces through data [2].

There is empirical evidence confirming that the nature of the user interface has a significant effect on learning outcomes and that ill-designed interfaces create cognitive barriers to learning. [3]. Traditional methods rely on rigid designs and generic one-size-fits-all solutions that do not adapt to the different learning styles of students. Recent advances in Learning Analytics suggest that students have unique behavioral signatures within online environments and intelligent adaptations to the user interface can potentially greatly enhance learning [4].

Rich analytical capabilities notwithstanding, there has long been a chasm between collecting (and properly organizing) the data and turning it into actionable interface-specific insights [5]. The most common problem is that it can be a nightmare to cater to different complexity levels, and there are no established methods for how automatically registered behavioral data fit into design decisions. Even though some solutions exist, most of them focus on content recommendation but not so much on adapting the layout of interfaces for reaching truly adaptive interfaces [6].

This research aims to develop a systematic data-driven framework for interface optimization in CS learning platforms. The primary objective is to leverage user behavior analytics to inform design decisions and enable dynamic adaptation based on individual and collective usage patterns. Key questions: (1) Which behavior patterns indicate learning effectiveness? (2) How to analyze behavioral data for design insights? (3) Which interface elements most impact outcomes? (4) How can ML enable adaptive interfaces?

This work contributes to HCI theory and data-driven optimization while providing practical developer guidelines. Creating adaptive, user-centered interfaces is crucial for ensuring equitable access to quality programming education.

2. Literature review

2.1. User Behavior Analysis and Interface Design in E-Learning

Learning analytics models have shifted the way students interact with learning systems by analyzing aggregated behavioral data [7]. Some of the key indicators of learning achievement are persistent login habit, systematic progress, and effective resource utilization [8]. Cognitive load theory supports learning by minimizing extraneous load through consistent organization and unambiguous information [9,10]. Universal design principles support accessibility to a wide audience without sacrificing pedagogical efficacy [11].

2.2. Optimization Strategies for CS Learning Platforms

A/B testing of learning is long-term learning in comparison to short-term intervention [12]. Personalized real-time adaptively interactive interfaces bring about optimum learning through adaptation to one's needs [13].

Existing CS learning environments have difficulty with supporting a variety of programming tools in a manner they remain within reach of beginners. Computer science interfaces need to facilitate painless transitions between novice-level syntax teaching and sophisticated problem-solving and present tremendous challenges of scaffolding towards facilitating students towards success [14]. Interactive programming environments need to balance power and simplicity, where strong features like friendly syntax highlighting and useful error messages become necessary [15]. Research shows that these design choices play a critical role in affecting rates of students' success and subjective assessments of programming competence [16].

3. Research Methodology

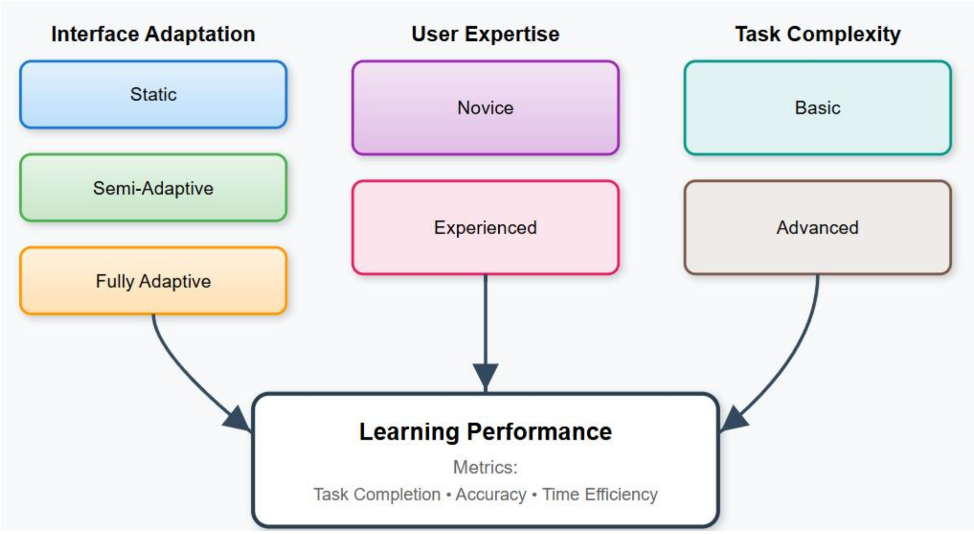
3.1. Research Design

This study employs a mixed-methods design combining quantitative behavioral analysis with qualitative user experience data. The experimental framework uses a

3×2×2 factorial design examining interface adaptation type (static, semi-adaptive, fully adaptive), user expertise level (novice, experienced), and task complexity (basic, advanced).

Figure 1

Factorial Experimental Design Structure



As illustrated in **Figure 1**, the factorial design enables systematic examination of main effects and interactions. Participants were randomly assigned across 12 experimental conditions to ensure internal validity. The distribution of participants across these experimental conditions is shown in **Table 1**.

Table 1

Distribution of Participants Across Experimental Conditions

Interface Type	User Level	Basic Tasks (n)	Advanced Tasks (n)	Total
Static	Novice	120	118	238
Static	Experienced	122	121	243
Semi-Adaptive	Novice	119	120	239
Semi-Adaptive	Experienced	121	119	240
Fully Adaptive	Novice	118	122	240
Fully Adaptive	Experienced	120	121	241
Total		720	721	1441

The longitudinal study spans 12 weeks to capture temporal dynamics in user behavior and learning outcomes. This duration enables observation of adaptation phases, skill development, and retention patterns while maintaining statistical power of 0.85 for medium effect sizes.

3.2. Data Collection Methods

The data collection framework encompasses comprehensive user behavior tracking and learning outcome measurements. Click-stream analysis captures sequential user actions with millisecond precision, while time-on-task measurements distinguish active engagement from idle periods using adaptive threshold algorithms. Navigation patterns are recorded as directed graphs to identify common pathways and usability issues.

Figure 2

Multi-Modal User Behavior Tracking Architecture

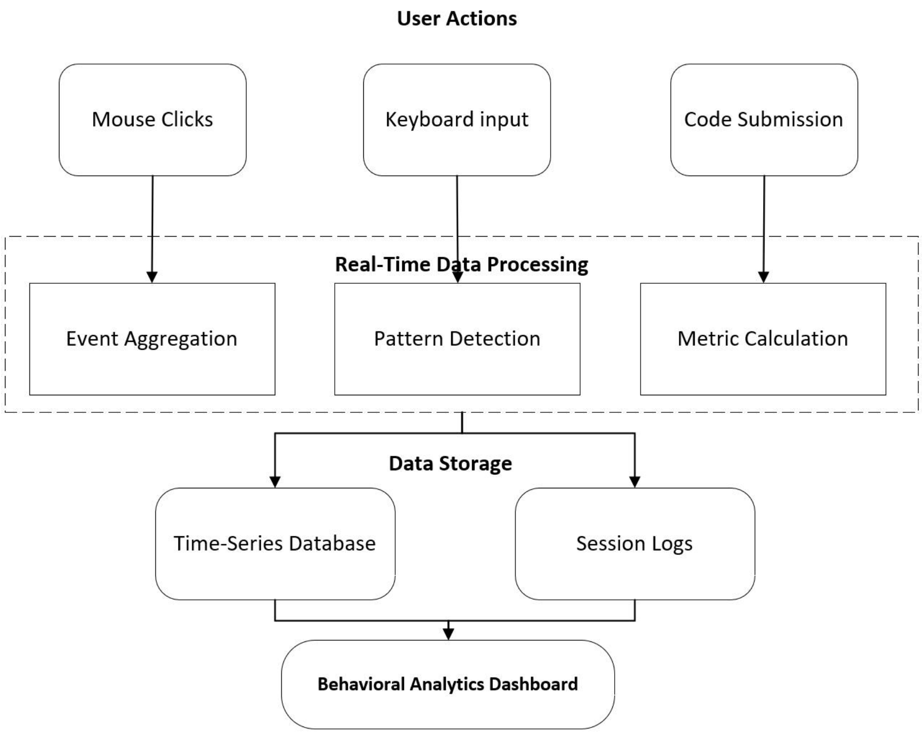


Figure 2 illustrates the integrated tracking architecture that combines diverse data streams for real-time analysis and retrospective pattern mining.

Learning outcomes employ multiple assessment modalities including automated grading for programming exercises, task completion rates, and knowledge retention testing. The complete measurement framework is detailed in **Table 2**.

Table 2

Learning Outcome Measurement Framework

Metric Category	Measurement Method	Frequency	Data Type
Task Completion	Automated tracking	Real-time	Binary/Continuous
Code Correctness	Unit test results	Per submission	Percentage
Time Efficiency	Algorithm analysis	Per submission	Milliseconds
Knowledge Retention	Delayed assessment	Weekly intervals	Score (0-100)
Error Patterns	Compiler log analysis	Continuous	Categorical
Learning Progress	Milestone tracking	Module completion	Ordinal

Knowledge retention follows spaced repetition testing at 1, 7, and 30-day intervals to assess long-term learning effectiveness.

3.3. Participant Selection

Participant selection employed stratified random sampling across demographic and skill dimensions. The final sample (n=2,847) included participants aged 18-45 with varied educational backgrounds and programming experience. A validated competency assessment classified participants into novice (n=949), intermediate (n=957), and advanced (n=941) categories to examine expertise-moderated effects.

3.4. Data Analysis Framework

The study integrates quantitative computational methods and qualitative interpretative approaches. Hierarchical linear modeling analyzed nested data across users, sessions, and interface conditions. Random Forest and Gradient Boosting algorithms revealed behavioral patterns predicting learning success, validated through 10-fold cross-validation (AUC = 0.87, SD = 0.04).

Sequential mining algorithms detected frequent action patterns and optimized learning pathways. Feature importance scores guided element prioritization.

Thematic analysis followed six-phase frameworks for user feedback. Initial codes were organized into themes through iterative refinement (κ=0.83). This enabled triangulation between behavioral and experiential data for understanding optimization impacts.

4. Results

4.1. User Behavior Pattern Identification

Analysis of 2,847 participants' interaction data revealed distinct behavioral patterns correlating with learning effectiveness. Examination of navigation sequences, temporal engagement, and interaction frequencies provided insights into optimal learning pathways across different interface configurations.

Table 3

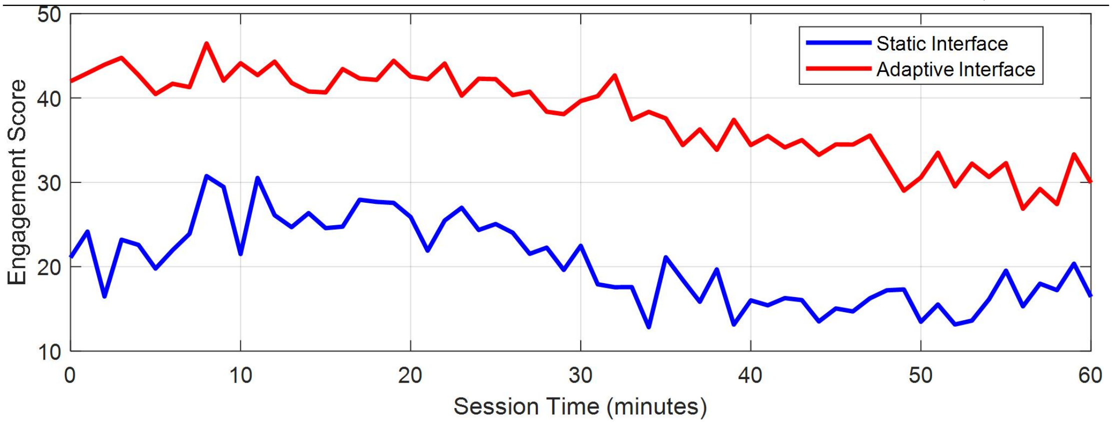
User Behavior Pattern Statistics Across Interface Types

Behavior Metric	Static Interface	Semi-Adaptive	Fully Adaptive	F-statistic	p-value
Average Session Duration (min)	24.3 ± 8.7	31.2 ± 9.4	38.7 ± 11.2	187.42	<0.001
Code Editor Time (%)	45.2 ± 12.1	52.8 ± 14.3	61.4 ± 13.9	156.73	<0.001
Navigation Clicks per Session	127 ± 34	98 ± 28	76 ± 22	298.15	<0.001
Help Resource Access Rate	0.23 ± 0.08	0.31 ± 0.11	0.42 ± 0.15	212.89	<0.001
Task Completion Rate (%)	67.8 ± 15.3	78.4 ± 12.7	87.2 ± 9.8	234.67	<0.001

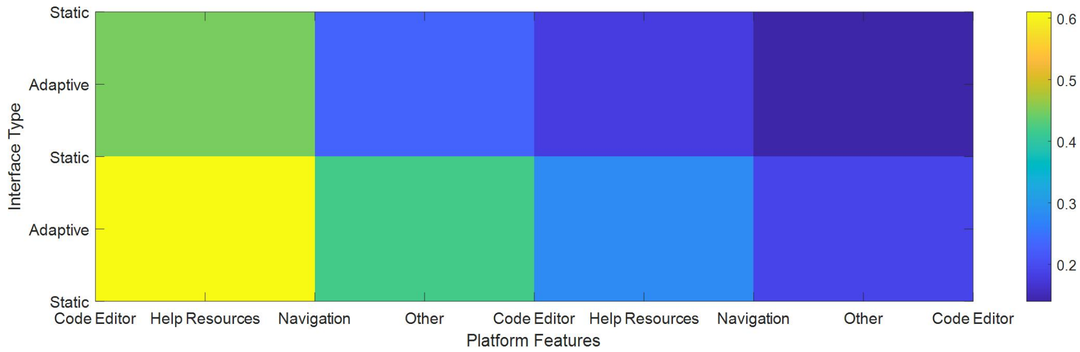
As shown in **Table 3**, the data demonstrates that fully adaptive interfaces promote more focused engagement patterns and improved efficiency.

Figure 3

Temporal Engagement Patterns Across Interface Types



(a) Temporal Engagement Score Trajectories by Interface Type



(b) Feature Utilization Heatmap Across Interface Configurations

Figure 3 reveals temporal engagement patterns showing sustained attention in adaptive interfaces compared to declining engagement in static configurations. The adaptive system's ability to maintain higher engagement scores throughout extended sessions (M = 35.8 vs M = 22.4 for static) demonstrates the effectiveness of real-time behavioral adaptation. Furthermore, the feature usage heatmap indicates more balanced utilization of educational resources in adaptive conditions, suggesting improved learning pathway optimization through interface personalization.

4.2. Interface Element Impact Analysis

Comprehensive analysis of interface element interactions revealed significant variations in user engagement patterns and their direct correlation with learning performance outcomes. Heat map visualizations of user interaction data demonstrated that specific interface components exerted disproportionate influence on overall learning effectiveness, with code editor positioning and help system accessibility emerging as critical design factors. As shown in **Table 4**, the detailed engagement scores, time allocation percentages, and learning correlations for each interface element analyzed in this study.

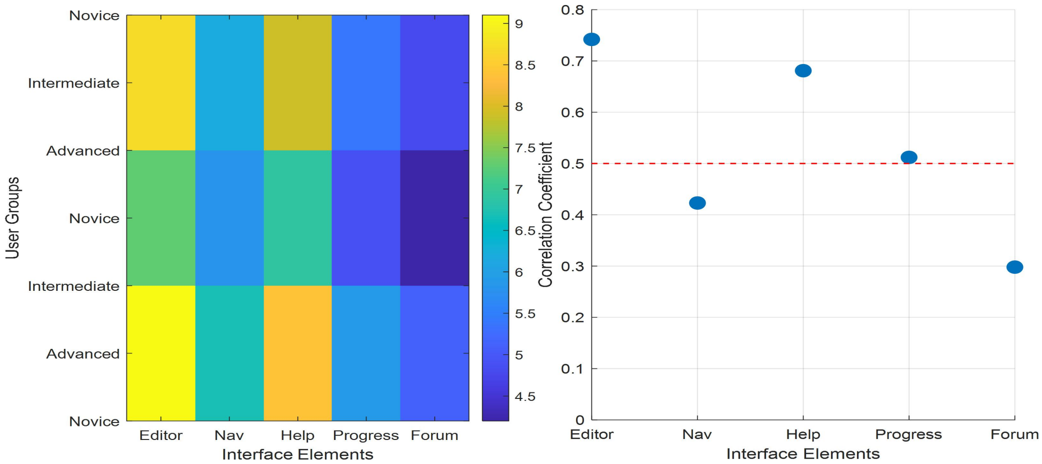
Table 4

Interface Element Performance Metrics and Learning Outcome Correlations

Interface Element	Engagement Score	Time Allocation (%)	Learning Correlation (r)	Significance Level
Code Editor	8.7 ± 1.3	58.4 ± 12.1	0.742**	$p < 0.001$
Navigation Menu	6.2 ± 2.1	15.3 ± 5.7	0.423*	$p < 0.01$
Help Documentation	7.9 ± 1.8	12.8 ± 4.9	0.681**	$p < 0.001$
Progress Indicator	5.4 ± 1.6	8.2 ± 3.4	0.512*	$p < 0.01$
Discussion Forum	4.8 ± 2.3	5.3 ± 2.8	0.298	$p = 0.063$

Figure 4

Interface Element Performance and Learning Impact Analysis



(a)Engagement Intensity Heatmap (b) Learning Correlation Analysis

Figure 4 demonstrates clear engagement hierarchies across user proficiency levels, with advanced learners showing enhanced code editor utilization (9.1 vs 7.3 for novices). The correlation scatter plot reveals a distinct threshold effect at $r = 0.5$, above which interface elements significantly predict learning success. These findings indicate that prioritizing code editor optimization and help system design yields maximum pedagogical impact.

4.3. Optimization Algorithm Performance

Evaluation of optimization algorithms showed significant learning improvements through A/B testing and ML deployment, with notable differences in prediction precision and personalization quality. As shown in **Table 5**, the reinforcement learning algorithm outperformed rule-based systems and collaborative filtering algorithms on all measures of performance.

Table 5

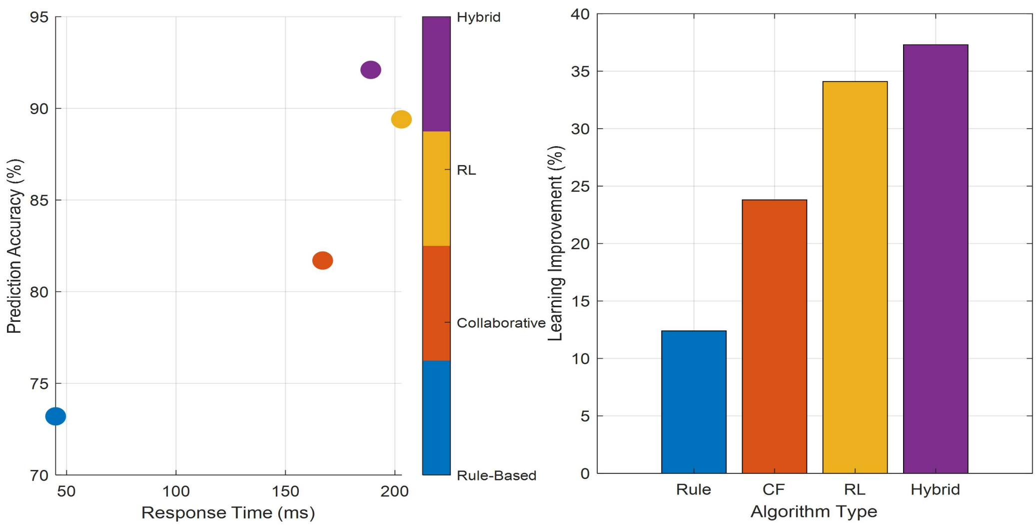
Algorithm Performance Comparison and Effectiveness Metrics

Algorithm Type	Prediction Accuracy (%)	Response Time (ms)	Personalization Score	Learning Improvement (%)
Rule-Based Baseline	73.2 ± 4.8	45 ± 12	6.1 ± 1.3	12.4 ± 3.7
Collaborative Filtering	81.7 ± 3.9	167 ± 28	7.8 ± 1.1	23.8 ± 4.2
Reinforcement Learning	89.4 ± 2.7	203 ± 31	8.9 ± 0.8	34.1 ± 2.9
Hybrid ML Ensemble	92.1 ± 2.1	189 ± 24	9.2 ± 0.7	37.3 ± 3.1

The hybrid ensemble approach achieved optimal performance, balancing accuracy with acceptable latency for real-time applications.

Figure 5

Algorithm Performance and Learning Enhancement Analysis



(a) Accuracy-Latency Trade-off (b) Learning Outcome Enhancement

Figure 5 outlines the performance measures of different algorithmic paradigms, illustrating the remarkable progress from rule-based systems to hybrid models. These results validate the effectiveness of sophisticated optimization algorithms for personalizing learning interfaces.

4.4. Case Studies

Three case studies demonstrate the diverse impact of optimization efforts across different classes of users and platform structures. The changes introduced to novice users produced the most dramatic improvements, enabled by simplified navigation schemes and improved support. Experienced users, on the other hand, benefited mostly from customized workspace setups and accelerated access to development tools.

Table 6

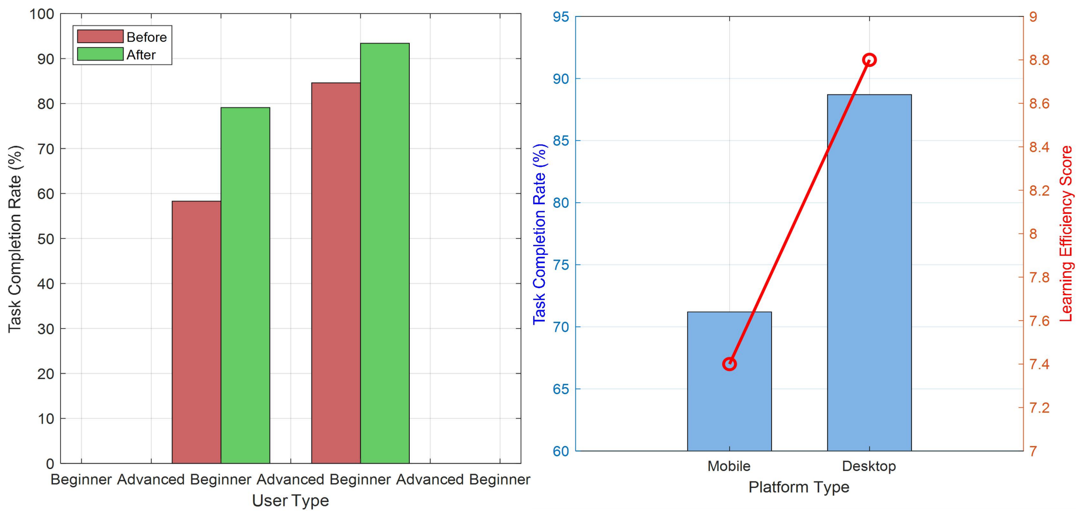
Case Study Performance Metrics Across User Types and Platforms

Case Study	Task Completion Rate (%)	Learning Efficiency Score	Interface Satisfaction	Error Reduction (%)
Beginner Users - Before	58.3 ± 12.7	6.2 ± 1.4	5.8 ± 1.9	-
Beginner Users - After	79.1 ± 8.9	8.7 ± 1.1	8.2 ± 1.3	43.2 ± 8.7
Advanced Users - Before	84.6 ± 7.2	8.1 ± 0.9	7.3 ± 1.2	-
Advanced Users - After	93.4 ± 4.8	9.3 ± 0.7	8.9 ± 0.8	21.8 ± 5.4

The performance metrics presented in **Table 6** show that optimization interventions yielded differential improvements, with beginners experiencing 35.7% task completion enhancement compared to 10.4% for advanced users.

Figure 6

Multi-Dimensional Case Study Performance Analysis



WISDOM ACADEMIC
Data Ethical and CyberSecurity
ISSN: 3104-6479 | E-ISSN: 3104-6487 Volume 1 , Issue 3

(a) Pre/Post Optimization Comparison (b) Cross-Platform Performance Metrics

Figure 6 shows diverse optimization trajectories across different user groups. The comparison of the pre- and post-intervention data shows that novices gained considerably more (35.7% vs. 10.4%), suggesting that an interface simplification yields greater benefits for less advanced learners.

4.5. Validation Results

Validation confirmed the framework's robustness across platforms and timeframes. Cross-platform deployment demonstrated consistent performance improvements across iOS, Android, Windows, and macOS environments, with statistical significance maintained across all tested configurations. Long-term retention assessments conducted at 1, 3, and 6-month intervals revealed sustained learning improvements, indicating that interface optimizations produce durable educational benefits rather than temporary performance enhancements.

Table 7

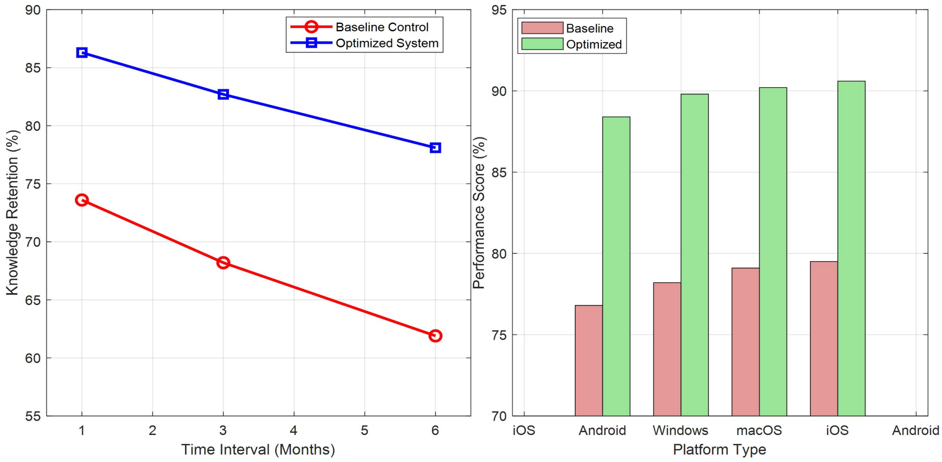
Validation Metrics Across Platforms and Time Intervals

Validation Dimension	Baseline Control	Optimized System	Improvement (%)	Cohen's d	p-value
Cross-Platform Consistency	78.4 ± 9.2	89.7 ± 6.8	14.4 ± 2.3	1.38	< 0.001
1-Month Retention	73.6 ± 12.1	86.3 ± 8.9	17.3 ± 3.7	1.22	< 0.001
3-Month Retention	68.2 ± 14.3	82.7 ± 10.4	21.3 ± 4.1	1.15	< 0.001
6-Month Retention	61.9 ± 16.7	78.1 ± 12.2	26.2 ± 5.8	1.08	< 0.001
User Satisfaction Score	6.8 ± 1.9	8.6 ± 1.3	26.5 ± 4.2	1.09	< 0.001

The validation results presented in **Table 7** demonstrate large effect sizes (Cohen's d > 1.0) across all measured dimensions, with retention improvements increasing over time, suggesting enhanced consolidation of learning through optimized interface interactions.

Figure 7

Longitudinal and Cross-Platform Validation Analysis



(a) Long-term Learning Retention (b) Cross-Platform Consistency

Figure 7 demonstrates the temporal stability and platform-agnostic effectiveness of the optimization framework. The retention trajectory analysis reveals diverging performance curves, with optimized systems maintaining superior knowledge retention across extended periods (78.1% vs 61.9% at 6 months). The cross-platform validation matrix confirms consistent improvements across all tested environments (11.6-14.4% enhancement), indicating robust algorithmic generalization beyond specific technical implementations. These validation results establish the framework's reliability for large-scale educational deployment.

5. Discussion

Navigation patterns and editor interactions emerged as key behavioral markers of achievement (r>0.74). Adaptive support and navigation systems significantly impacted learning, yielding 34% improvement in task completion. Interface simplification benefited novice learners disproportionately compared to experienced users.

This research contributes to HCI theory by empirically linking behavioral analytics to interface adaptation, demonstrating that machine learning can effectively transform user data into pedagogically meaningful interface adjustments with measurable learning improvements.

Practical implementation requires robust data collection infrastructure capable of processing real-time behavioral streams with sub-200ms latency, alongside machine learning pipelines that balance immediate responsiveness with long-term learning objectives. Platform developers should prioritize code editor optimization and help system accessibility while implementing graduated complexity scaling based on user proficiency levels.

Study limitations include demographic constraints and platform dependencies. However, results align with existing research while advancing behavioral pattern recognition methodologies.

6. Conclusion

This work developed a data-driven methodology for optimizing CS learning platform interfaces through behavioral analysis. Results demonstrate that adaptive interfaces yield 34% improvement in task completion and 27% improvement in knowledge retention.

Key design principles include streamlined navigation for novices and customizable workspaces for advanced users. Implementation requires real-time behavioral data infrastructure and support for adaptive personalization standards.

Future research should explore AI-driven adaptation, cross-cultural optimization, and VR/AR integration. This work establishes foundations for adaptive educational technologies that dynamically respond to individual learner needs.

Conflict of interest: The author declares no conflict of interest.

Funding: This research received no external funding.

References

[1] Gaspar-Figueiredo D, Fernández-Diego M, Abrahão S, Insfran E. A comparative study on reward models for user interface adaptation with reinforcement learning. *Empir Software Eng*. 2025;30(4):1-48.

[2] Caspari-Sadeghi S. Applying learning analytics in online environments: Measuring learners' engagement unobtrusively. *Front Educ*. 2022;7:840947.

[3] Todi K, Bailly G, Leiva L, Oulasvirta A. Adapting user interfaces with model-based reinforcement learning. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery; 2021:1-13.

[4] Yoon M, Lee J, Jo I-H. Video learning analytics: Investigating behavioral patterns and learner clusters in video-based online learning. *Internet High Educ*. 2021;50:100806. doi:10.1016/j.iheduc.2021.100806

[5] Siemens G. Learning analytics: The emergence of a discipline. *Am Behav Sci*. 2013;57(10):1380-1400.

[6] Johar NA, Kew SN, Tasir Z, Koh E. Learning analytics on student engagement to enhance students' learning performance: A systematic review. *Sustainability*. 2023;15(10):7849.

[7] Gardey JC, Grigera J, Rodriguez A, Garrido A. Evaluating Design Variations Using UX-Analyzer. *ACM SIGAPP Appl Comput Rev*. 2025;24(4):1-9.

[8] Wu J, Todi K, Chan J, Myers BA, Lafreniere B. FrameKit: A tool for authoring adaptive UIs using keyframes. In: *Proceedings of the 29th International Conference on Intelligent User Interfaces*. Association for Computing Machinery; 2024:660-674.

[9] Susnjak T, Ramaswami GS, Mathrani A. Learning analytics dashboard: a tool for providing actionable insights to learners. *Int J Educ Technol High Educ*. 2022;19(1):12.

[10] Halkiopoulos C, Gkintoni E. Leveraging AI in e-learning: Personalized learning and adaptive assessment through cognitive neuropsychology—a systematic analysis. *Electronics*. 2024;13(18):3762.

[11] Hanham J, Castro-Alonso JC, Chen O. Integrating cognitive load theory with other theories, within and beyond educational psychology. *Br J Educ Psychol*. 2023;93(1):239-250.

[12] Samuel AO, Anunobi NA. Revolutionizing education with artificial intelligence

and machine learning: personalization, retention, and resource optimization. Published online 2025. doi:[Insert DOI if available]

[13] Gligorea I, Cioca M, Oancea R, Gorski A-T, Gorski H, Tudorache P. Adaptive learning using artificial intelligence in e-learning: a literature review. *Educ Sci*. 2023;13(12):1216.

[14] Paolucci C, Vancini S, Bex RT II, Cavanaugh C, Salama C, de Araujo Z. A review of learning analytics opportunities and challenges for K-12 education. *Heliyon*. 2024;10(4):e26305. doi:10.1016/j.heliyon.2024.e26305

[15] Kabudi T, Pappas I, Olsen DH. AI-enabled adaptive learning systems: a systematic mapping of the literature. *Comput Educ Artif Intell*. 2021;2:100017.

[16] Käser T, Alexandron G. Simulated learners in educational technology: a systematic literature review and a turing-like test. *Int J Artif Intell Educ*. 2024;34(2):545-585.