information

MDPI

Article

Personalized Instructional Strategy Adaptation Using TOPSIS: A Multi-Criteria Decision-Making Approach for Adaptive Learning Systems

Christos Troussas *, Akrivi Krouska, Phivos Mylonas and Cleo Sgouropoulou

[1] Department of Informatics and Computer Engineering, University of West Attica; {ctrouss, akrouska, mylonasf, csgouro}@uniwa.gr
* Correspondence: ctrouss@uniwa.gr

Abstract: The growing number of educational technologies presents possibilities and challenges for personalized instruction. This paper presents a learner-centered decision support system for selecting adaptive instructional strategies, that embeds the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) in a real-time learning environment. The system uses multi-dimensional learner performance data, such as error rate, time-on-task, mastery level, and motivation, to dynamically analyze and recommend the best pedagogical intervention from a pool of strategies, which includes hints, code examples, reflection prompts, and targeted scaffolding. In developing the system, we chose to employ it in a one off postgraduate Java programming course, as it represents a defined cognitive load structure and samples a spectrum of learners. A robust evaluation was conducted with 100 students and an adaptive system compared to a static/no adaptive control condition. The adaptive system with TOPSIS yielded statistically higher learning outcomes (normalized gain g = 0.49), behavioral engagement (28.3% increase in tasks attempted), and learner satisfaction. 85.3% of the expert evaluators agreed with the system decisions compared to the lecturer's preferred teaching response towards the prescribed problems and behaviors. In comparison to rule-based approach, it was clear that the TOPSIS framework provided a more granular and effective adaptation. The findings validate the use of multi-criteria decision-making for real-time instructional support and underscore the transparency, flexibility, and educational potential of the proposed system across broader learning domains.

Keywords: Adaptive Learning Systems; Multi-Criteria Decision Making; MCDM; Educational Decision Support; TOPSIS; Learner-Centered Design

Academic Editor: Firstname Lastname

Received: date
Revised: date
Accepted: date
Published: date

Citation: To be added by editorial staff during production.

1. Introduction

The rise of educational software across multiple learning environments has fundamentally modified the manner in which knowledge is delivered, accessed, and measured. Educational software creates automated feedback, self-regulated learning, and interactions that can accommodate a vast array of learners and subjects [1]. As these systems continue to be used more in both formal and informal learning contexts, researchers and software developers are less concerned with whether learners can access these educational technologies and are more concerned with how they can be tailored to suit the changing needs, aims, and behaviors of individual learners [2].

Today's classrooms, either in-person or online, are comprised of ever-growing heterogeneous groups of learners [3]. Learners join the classroom with a variety of social and cultural backgrounds, previous knowledge, cognitive styles, motivations, and learning preferences. Thus, a universal approach to instructional design is often not sufficient. Personalization has now become a priority in educational systems design [4], which aims to tailor content and learner support strategies to the individual learner's profile and needs at a specific moment.

Personalization in educational contexts is not solely about content sequencing or recommendation [5]. It is also important to personalized adaptations in instructional strategy - choice of how to respond to a learner's action online, in real time [6]. For instance, should the system provide a hint, give an explicit code example, prompt to self-reflect, or advance the learner? The determination of the most helpful instructional strategy at the right moment may influence the learner's level of engagement, understanding, and retention of material.

There are a number of approaches that have proposed methods for making these types of adaptive instructional decisions [7]. There are rule-based systems, Bayesian networks, fuzzy logic, reinforcement learning, case-based reasoning, and others. Each methodology has its own advantages and drawbacks. However, one family of methods Multi-Criteria Decision Making (MCDM) methods are of particular interest to education because instructors are often attempting to balance many different factors related to the learner [8]. One MCDM method is the Technique for Order Preference by Similarity to the Ideal Solution (TOPSIS).

TOPSIS is a widely used MCDM method that was introduced by Hwang and Yoon in 1981 [9]. TOPSIS finds the best solution alternative from a finite set of possible option alternatives by comparing the distance of each alternative to an ideal solution (the best case) and an anti-ideal solution (the worst case). The learner options are assessed with various criteria, with weights assigned, and the option that is closest to the ideal solution and furthest from the anti-ideal situation is selected. The method is intuitive, computationally simple, and appropriate for real-time usage, which makes it a very good injective for learner-centered instructional systems [10]. We used TOPSIS for this work because it can accommodate multiple, often competing, metrics of learners (performance, effort, engagement, etc.) in a mathematically principled and interpretable manner.

This paper presents a decision support system for learner-centered instructional strategy adaptation in personalized learning. The system collects real-time performance data from students and applies a decision support model based on the TOPSIS algorithm to determine the most pedagogically relevant instructional strategy among a collection of strategies (i.e., contextual hints, examples with annotations, reflection prompts, and scaffolding activities). This research is novel in that we are layering real-time multi-criteria decision-making model into an adaptive instructional engine, allowing interventions based on learner state rather than content order. As a testbed for our research, we applied this framework to a Java programming learning environment, which provided a well-structured and cognitively demanding context to validate adaptive instructional strategies. Java was selected as the instructional domain due to its structured syntax, object-oriented paradigm, and its widespread use in programming education—factors that make it ideal for evaluating the effectiveness of real-time adaptive instructional interventions [11-12]. The work describes the architecture of the system, decision-making model, and implementation; we provide a goal assessment regarding the system with cognitive, behavioral, and usability performance. The model is an open, extensible and empirically-based method of personalization founded on MCDM theory that be applied to many educational contexts. Summarizing, the contribution of this work is the integration of the TOPSIS multi-criteria decision-making algorithm into a real-time adaptive instructional decision-support system for personalized learning interventions. Unlike traditional systems focusing primarily on content sequencing, this approach dynamically selects instructional strategies tailored specifically to individual learner states.

The remainder of the paper is structured as follows. Section 2 discusses related work; Section 3 outlines the system architecture. Section 4 details the TOPSIS-based decision framework. Section 5 presents the instructional strategy adaptation and implementation. Section 6 covers the system evaluation. Section 7 discusses the findings, and Section 8 provides the conclusions and future directions.

2. Related Work

Adaptive learning systems in computer science education have progressed tremendously in programming pedagogy [13-19]. Adaptive learning systems typically seek to individualize the educational experience by quickly acting upon data from the learners' direct experience. When the context is Java programming and other technical subjects, customization often relates mostly to content difficulty, recommendations, and pacing [20, 21]. Some intelligent tutoring systems, such as [22, 23], analyze the learners' submissions for context-aware feedback. Other systems base activity selection or explanations upon a learner profile of learning style or prior knowledge. Generally, while these systems may adapt what is presented, addressing how to pedagogically intervene through adaptive instructional strategies in consideration of learner states is less common.

To aid with this kind of instructional decision-making, a number of learner modeling methods have been implemented [24]. The most straightforward learner models are the Rule-based systems that have straightforward development and maintenance and achieve a high level of transparency; however, they lack scalability and flexibility [25-27]. The more sophisticated probabilistic learner models include Bayesian networks, which model the probabilistic relationships between different learner variables [28-30]; these also include fuzzy logic systems, which use fuzzy membership functions to model uncertainty in learner variable interpretation [31-33]. Then there are Reinforcement learning methods that optimize instructional policies through trial and error, and case-based reasoning systems define new learners based on similar previous learner profiles and the successful instructional interventions [34-36]. There are advantages to all of these models, but they all share an issue of fairly high complexity, data dependency, or low interpretability, especially as they are deployed in real-time systems, which require immediate action.

In recent years, the MCDM approaches, particularly the Technique for Order Preference by Similarity to the Ideal Solution (TOPSIS), have begun to gain more attention and use in educational research [10, 37-46]. TOPSIS has been used in a variety of contexts in education, including, for example, evaluating learning management systems, recommending learning resources, and matching course recommendations with learner objectives and goals. Each of these scenarios demonstrates the use of TOPSIS in situations requiring consideration of several, often competing, criteria. The TOPSIS method lends itself to this problem and provides an approach that has mathematical rigor and is also transparent for educators and learners to understand. However, use of TOPSIS has remained primarily in offline analysis or generating generalized recommendations, and has not been integrated into real-time adaptive systems [47].

Several prominent MCDM methods have been proposed in literature, such as Analytic Hierarchy Process (AHP), ELECTRE, PROMETHEE, and VIKOR [48-51]. AHP is powerful for criteria weighting but can become computationally intensive in real-time contexts [52]. ELECTRE and PROMETHEE are useful in handling qualitative preferences, yet their complexity makes rapid decision-making challenging [53]. VIKOR is effective in compromise ranking but can be sensitive to criteria weights [54]. We selected the TOPSIS method due to its simplicity, computational efficiency, clear interpretability, and effectiveness in handling conflicting criteria. These attributes make TOPSIS particularly well-suited for the real-time adaptive instructional scenarios central to our study.

In this paper, we demonstrated a new paradigm for using TOPSIS by applying it in a learner-centered pedagogical framework for Java programming. Previous studies had used TOPSIS to make static assessments or recommendations of content, while our system was able to assess interactions in real time and determine the ranking of potential pedagogies post-learning task. This is a shift from adaptation of content to adaptation of pedagogy, and in doing so, provided additional dimensions of personalization- that is the tactical adaptation of how support is provided rather than what support is provided. Our work fills an important niche in describing and operationalizing a multi-criteria pedagogical decision-making process that is interpretable and dynamic, within the constraints programming education presents.

3. System Architecture

The proposed learning system is a web-based personalized learning ecosystem that facilitates Java programming instruction through real-time, data-driven adaptations to teaching strategies. The learner interface was developed as a web-based environment using standard front-end technologies suitable for interactive educational applications. This design enables real-time feedback, responsiveness, and seamless integration with the adaptive decision engine. The design of the system is based on pedagogy and computational decision support, ensuring that all learners receive timely and effective interventions based on their real-time performance and behavioral traces. All of the decisions with instructional design are based on the steps in TOPSIS, which sequentially aggregates and prioritizes multiple learner-centered criteria to identify the best instructional strategy the learner could be provided with at any learning step. The system is grounded in three core functional components: Learner Performance Monitoring, Decision Support Engine, and Instructional Content Delivery. The three components work within a feedback loop enabling real-time personalization.

The Learner Performance Monitoring component is responsible for collecting and keeping up-to-date a range of data about the learner's performance while interacting with Java programming tasks. While engaged with other components of the system, the environment provides learners with an integrated code editor, a submission console and assessments modules. This component collects all sorts of overt behaviors and performance measures of the learner including the number of syntax and semantic errors, total time in completing tasks, hint request frequency, the number of compilations before submission, quiz scores, and so on. These raw data points are converted to normalized learner features to be used by the Decision Support Engine. In addition, the system captures a time series of the learning events so that future iterations will be able to use this time series data to make adaptations based on trends in behavior.

With the learner profile updated, the Decision Support Engine is activated. The Decision Support Engine then deploys the TOPSIS algorithm to evaluate a number of predetermined instructional strategies against the individual characteristics of the learner, through demonstrating selected criteria. Each strategy is scored based on its estimated effectiveness given the learners recent performance. The characteristics and criteria for scoring the strategies are of a cognitive dimension (e.g., concept acquisition, overall error rate) and behavioral dimension (e.g., amount of time on task, motivational proxies) and these each may have different weights of importance pedagogically.

The Instructional Content Delivery component implements the chosen strategy in an efficient, unobtrusive manner. Due to circumstance and level of engagement, the system will either provide an interactive hint, or draw attention to a part of a past submission, or provide a marked-up code example, or prompt for reflection on performance. The platform is designed to provide instructional strategies with minimal disruption to instruction, while adapting to user instruction seamlessly. The Instructional Content Delivery logs all instructional activity for subsequent analysis and use by the system and also for research in pedagogy.

The entire system was developed to use a data flow and interaction cycle in real time. Each learner action (e.g., submission of code, request for help, submission of quiz response) becomes a data acquisition point. The performance monitoring module takes in the information, processes it and encodes it for use by the decision support engine. The Decision Support Engine analyzes the learner state in near real-time using a TOPSIS evaluation process and determines the most appropriate response from an instructional context. This creates a continuous cycle that allows the system to be responsive to changes in learner engagement and performance throughout a session.

To further illustrate the process, let's focus on a learner struggling with nested loops in Java. She has made several attempts, spent a long period attempting the task, and asked for several hints. All are indications of cognitive overload and low task mastery. The Learner Performance Monitoring captures these metrics, and the Decision Support Engine evaluates various strategies. Among the options—providing another hint, suggesting a simpler exercise, or showing a complete worked example—the Decision Support Engine determines, based on weighted learner criteria, that the most appropriate action is to display an annotated example. The Instructional Content Delivery will present this example as her focus will be on the structure and logic of nested loops. Following a review of this, she will complete the task again, and then the adaptive cycle continues.

The architecture support responsive instruction, while retaining semblance of interpretability, accountability, and pedagogically sound practices. Whereas, a black-box model would be opaque to the learner, the use of TOPSIS allows the learner to see prompts, hints, and other strategies being considered, allowing the instructor and designers the benefit of participating in the decision, and to refine and improve the choice of strategies. This enables the system to be built out further, with dynamic weighing, diversely pooling strategies, or even linkage to richer learner models.

A high-level outline of the system architecture is shown in Fig. 1, which encapsulates the major processing stages and their relationships in the real-time personalization loop while highlighting the modular and extensible architecture of the platform.

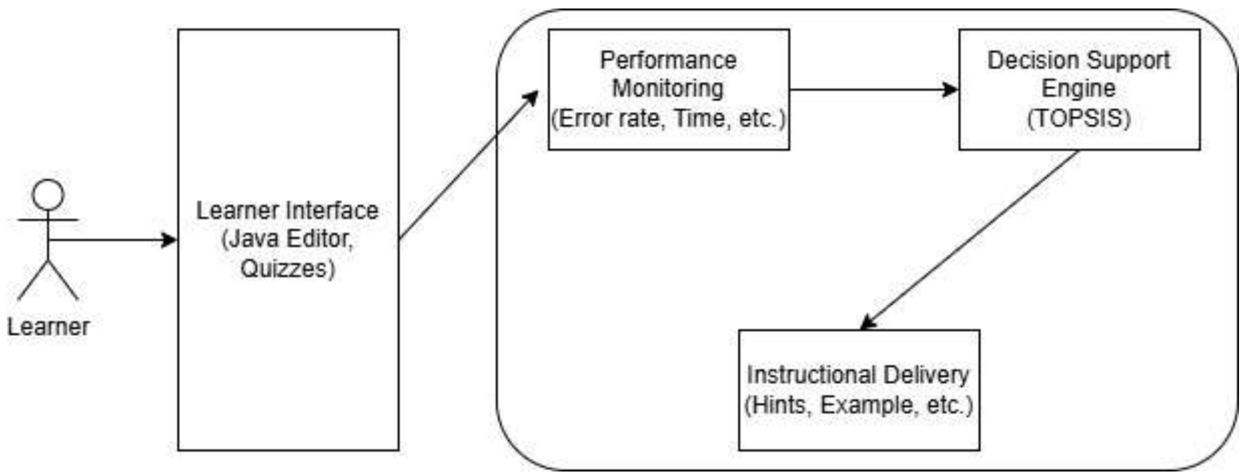

Figure 1. Workflow of the Decision Support Engine based on the TOPSIS method.

4. Multi-Criteria Decision Framework Using TOPSIS

At the heart of the instructional adaptation system is the Decision Support Engine, which utilizes the Technique for Order Preference by Similarity to the Ideal Solution (TOPSIS). Through the Decision Support Engine, real-time instructional personalization becomes possible in our emergent system, by choosing the most advantageous pedagogical action, given the multi-dimensional representation of the learner's cognitive and behavioral state. The procedures implemented by TOPSIS for educational decision making are dynamic, as opposed to traditional dynamic adaptation systems, which still mostly take the reactive approach of looking only at the learner's data to select a learning method. TOPSIS allows for the comparison of each instructional strategy as it respond to current data while drawing upon a structured reasoning approach based in education principles.

After every learner interaction, the system executes a comparison of a predetermined set of six instructional alternatives: (1) provide a contextual hint; (2) show an annotated code example; (3) assign an easier related task; (4) provide a reflection prompt; (5) permit the learner to proceed to the next concept; and (6) elicit a micro-quiz. These instructional alternatives were defined in collaboration with a group of programming educators and instructional designers. The instructional alternatives reflect different pedagogical intents, such as scaffolding, consolidation, remediation, motivation, and progression. This set was refined during iterative design sessions with educators who identified common instructional moves relevant to Java programming instruction.

The process of decision making is based on six pedagogically relevant criteria: Error Rate, Time on Task, Mastery, Motivation Score, Hint Usage Frequency (or the influence on learner independence or dependence), and Speed of Progress. These indicators were informed by literature review on learner analytics and validated through interviews with educators who experienced adaptive learning environments. Each criterion relates to a specific dimension of learner behavior and progress. For example, Error Rate indicates ongoing task performance; Time on Task indicates cognitive load; Mastery is inferred from performance on formative assessments; Motivation Score is related to persistence through tasks, voluntary resource access use and patterns of engagement with the platform; Hint Usage Frequency informs the indication of learner independence or dependency; and Speed of Progress indicates potential pacing and engagement. These six criteria are summarized in Table 1, which outlines their descriptions and pedagogical relevance within the adaptive decision-making process.

Table 1. Pedagogically Relevant Criteria Used in the Decision-Making Process.

Criterion	Description	Pedagogical Role
Error Rate	Frequency of syntax or logic errors during task completion	Indicates ongoing task performance
Time on Task	Total time spent working on a specific task	Reflects cognitive load or struggle
Mastery	Inferred from quiz scores and task outcomes	Represents understanding of core concepts
Motivation Score	Derived from platform engagement, retries, and voluntary actions	Suggests learner persistence and engagement
Hint Usage Frequency	Number of hints requested	Indicates learner independence or dependency
Speed of Progress	Rate of advancement through the learning sequence	Reflects learner pacing and consistency

To allow for comparison among these heterogeneous dimensions a min-max normalization was used to normalize all input values to a [0,1] scale. Once data was normalized, at each decision point a decision matrix was created that has a row for each instructional strategy and a column for the six criteria. Importantly, the entries of the decision matrix are not the learner's raw values but effect estimates of each strategy on each criterion given the learner's current stage of development. These estimates are based on expert-informed heuristics, pilot trial information, and directly observing previous uses of the system. For example, past patterns indicate that an easier task tends to improve perceived success and decrease frustration (lower error rate and time), and that a reflection prompt tends to improve motivation but may increase time on task.

Weight assignment is an integral part of TOPSIS; it defines how much each criterion contributes to the final ranking in the analysis. The weights used in this developmental study were W = [0.25, 0.15, 0.20, 0.15, 0.10, 0.15] and they were agreed upon using the Delphi method with five experts in computer science education, adaptive learning, and pedagogical experiences. Three iterative rounds for the experts to rank the criterion to what extent it supports student learning along with justifications for scores. The weight agreements in Criterion weights reflect the best identified priorities where Error Rate and Mastery (the right answer and understanding) were strongly regarded, followed by criterion relating to Effort, Motivation, and Pace indicators moderately weighted.

With the matrix fully populated and weighted, the system calculates both the ideal solution vector, made from the best values of each criterion (eg. error rate, mastery, speed) and the anti-ideal solution vector made from worst-case values. Each learning alternative for the particular educational context, was treated with an Euclidean distance formula, measuring calculated distance from the ideal and anti-ideal. The resultant closeness coefficient Ci was computed for each particular alternative:

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-}$$

where S_i^+ and S_i^- represent the distances to the ideal and anti-ideal solutions respectively. The strategy with the highest is selected for delivery.

To provide a tangible example, suppose we are considering three strategies—Hint, Code Example, and Micro-Quiz—with three simplified criteria. The normalized and weighted scores are shown in Table 2.

Table 2. Normalized and weighted decision matrix.

Strategy	Error Rate	Time on Task	Mastery	Weighted Score
Hint	0.8	0.6	0.3	0.54
Code Example	0.6	0.5	0.6	0.58
Quiz	0.3	0.4	0.9	0.60

Given ideal and anti-ideal vectors defined as [0.3, 0.4, 0.9] and [0.8, 0.6, 0.3], respectively, we calculate the Euclidean distances, which enables us to calculate the for each strategy. Notably, if the Quiz alternative has the highest closeness coefficient (Ci), it is chosen and implemented without delay. If two alternatives have almost identical values, the system employs a second rule (e.g., selection of the strategy that has not been used with the learner recently) to break ties and provide differentiated variability in pedagogy.

The benefit of this decision model lies in its flexibility, but also in its accountability. Instructors may retrace the decision path, examine the weightings, and follow the contributing factors towards a given adaptation. An additional benefit is that not only is the current model flexible, but it is also modular. With some restructuring, we would be able to add additional criteria (i.e., affective states from facial movement or typing patterns), or update mappings of strategies over time as we explore observable learner outcomes.

In conclusion, the TOPSIS-based Decision Support Engine provides an evidence-based, transparent, and scalable way of adapting instructional strategy in real-time. Providing a personalized pedagogical decision-making process built on multiple learner metrics, expert knowledge, and structured evaluation logic, it demonstrates a new level of personalization in pedagogical decision-making in the context of programming education. To visually summarize the full decision-making process, the overall workflow of the Decision Support Engine which is based on the TOPSIS method is shown in Fig. 2.

Figure 2. Workflow of the Decision Support Engine based on the TOPSIS method.

This quantitative scoring is then used to trigger the corresponding pedagogical action within the learning platform interface, as explained in Section 5.

5. Instructional Strategy Adaptation and System Implementation

The instructional strategies for instructional strategy adaptation implemented into the proposed learning system are a facilitated implementation of cognitive load theory principles relating to learner-centric responsiveness, scaffolding, and formative assessment. The strategies included in the learning system are designed for various types of learning difficulties, promote reflective thinking, developed motivation, and promote understanding of the appropriate concept. Additionally, each strategy serves a clear instructional goal. For instance, hints serve as a targeted form of scaffolding to assist learners who are experiencing temporary misunderstanding; whereas an annotated code example is used as scaffolding to assist with tacit understanding by learners who consistently fail to apply a programming construct. Moreover, the reflective prompts are intended to support learner metacognition, while micro-quizzes serve to support retention and provide evidence of residual misunderstanding.

The selection process is initiated by the system's perception of how the learner is transitioning through the learning state. The learner's state is expressed through a multi-dimensional profile from the constantly updated record of their patterns of interaction. Indicators like frequent compilation errors, excessive time spent on tasks, multiple hints, and no observable improvement indicate a potential bottleneck for learning. The data are represented by way of six pedagogically meaningful parameters, and these serve to structure the decisions of instructional options stemming from the principles of TOPSIS, one of the multi-criteria decision-making methods. Our specification of the probable effects for any instructional method – that is, the effect of each instructional method was defined in terms of the six pedagogical parameters - was obtained through some combination of heuristics informed by expert knowledge and pre-experimental data stemming from previous pilots with the system. For instance, it was noticed that learners who had been provided schema with annotated code examples in the previous months sporadically returned to the course learning with lower rates of errors and improved mastery, while those with reflection prompts tended to spend more time on assessment items than those who did not receive that instructional suggestion, but also displayed elevated engagement scores in post-task surveys. These patterns formed the basis of the values used in the estimates for impact for each of the instructional approaches in the decision matrix. It's important to know that the mappings were not arbitrary: we based them on accepted educational notions of learning behaviours and performance assessment, and the empirical evidence we witnessed in earlier versions of the platform.

The incorporation of the TOPSIS-based Decision Support Engine into the larger learning context was a complicated synchronization of the front-end interaction components with the back-end decision logic. The learning object was designed with a client-server architecture. The front-end interface of the learning context, designed with JavaScript and React, supports the interactive Java programming tasks in the learning object, provides personalized feedback to the learner, and gathers the learner's inputs. The back-end of the learning context, designed with Python, includes the TOPSIS engine, data normalization and scoring classifier routine, and the instructional response controller. The data viewing and processing components for the interactions are loosely coupled and communicate through the RESTful API, allowing data transfers to occur asynchronously and for each component to be updated in low-latency time. When a learner interacts with the learning context, an API call is created from the front-end interface. This call sends interaction movement data to the server, where the decision engine uses the data to perform the decision analysis of what is the optimal instructional strategy.

The adaptation mechanism must utilize a lightweight session-based learner model that is updated in real time to support the active adaptation process. The learner model doesn't try to predict the long term level of success but it does try to respond meaningfully to where the learner is currently at in their activity. Each time the learner submits work or requests help, the system retrieves performance metrics, and re-evaluates the instructional context to put the chosen strategy for a learner into the interface in a coherent manner. Once TOPSIS identifies the optimal instructional strategy, the system will instantiate and personalize the recommended strategy dynamically, by taking assets with associated metadata tags of pedagogical value from a library of pedagogical resources. Each resource wise asset (hint, code example, reasoning with the author... etc.) is indexed by topic, difficulty and a set of common misconceptions. Moreover, for example, if the student is working on a loop structure and the TOPSIS engine recommends 'hint' and recognizes repeated off-by-one errors, the system retrieves the hint “Consider whether your loop condition includes or excludes the endpoint value” from its database. The hint is contextually inserted within the learner's code editor so that only an adaptation to the UI is made, without requiring a page refresh or manual request. In a different case, if a code example is selected, the system produces an annotated example, in a highlighter, consistent with the learner's current concept (e.g., array iterating), formatted liberally with a small explanation box. Each pedagogical strategy is bracketed with the specific delivery situation: hints in-line, examples in expandable panels, reflection prompts in modals and so on—such that the envisioned interventions feel snugly situated in that workflow. With the immediacy of guttural nesting guided by the pedagogical rationale, adaptive support is not only algorithmically instantiated but also meaningfully experienced by the learner.

To maximize experimentation and development, the application was designed with modularity and extensibility in mind. The TOPSIS engine, data-processing pipeline, and instructional strategy modules are encapsulated as discrete services, updateable and replaceable without needing to rewrite the main application. Each decision and decision made by a learner is logged in a systematic way that will allow them to be analyzed and make continuous improvements to the adaptation logic. The system therefore demonstrates both theoretical rigor in its pedagogical underpinning and technical rigor in its implementation. It joins data-informed decision making with real-world instructional delivery and exemplifies that real-time personalization can be accomplished in programming education.

6. Evaluation

To thoroughly test the pedagogical effectiveness, behavioral incidence, and technical feasibility of the proposed TOPSIS-based adaptation system, we conducted a longitudinal mixed-method study with 100 postgraduate students in a conversion master's program in computer science at a Greek university. The cohort included students from non-STEM (Science, Technology, Engineering, and Mathematics) fields such as humanities, law, education, and psychology, representing a diverse range of learner profiles. Most participants had low baseline programming skills but highly variable motivation and prior experience—an ideal context in which to evaluate adaptive instructional technologies. The participants' ages ranged from 23 to 38 years old (M = 27.4, SD = 3.2), and the sample included 58 females and 42 males. This diversity in academic background, prior exposure to computing, and learner demographics provided a robust and realistic setting for assessing the system's capacity to personalize instruction effectively.

Participants were randomly assigned to a control group (n = 50), using a traditional static e-learning platform, and an experimental group (n = 50), using our adaptive platform and TOPSIS-based instructional strategic engine. Both groups covered the same four-week curriculum in Java fundamentals delivered by the same instructors. The only difference between the two was the personalization for the experimental group.

We assessed system effectiveness through a comprehensive evaluation framework combining quantitative and qualitative methods across four dimensions: (1) Learning outcomes, (2) Behavioral engagement, (3) Instructional strategy effectiveness, and (4) System responsiveness and usability.

6.1. Learning Outcomes

For the purpose of measuring conceptual development, we created a pre-test and post-test aligned with the course objectives and subsequently validated by expert review. In the pre-test/post-test, we ensured a near equal balance of 25 questions that addressed Java syntax, control structures, object-oriented design and code tracing. Our results are shown in Table 3. These data were given to us based on measure pre/post assessments that were rigorously scored from pre/post assessments and the average was taken for all students in each group. Normalized gain (g) was calculated with the Hake formula, with experimental students achieving an average normalized gain of 0.49, or Hake's gain, and the control group achieving an average normalized gain of 0.31. We calculated a two-tailed Welch's t-test (Welch = 1938) and determined statistical significance with $t(94.3) = 4.72$, $p < .0001$. We calculated an effect size following Cohen's d, where we arrived at an effect size of d = 0.82, which has a large educational impact based on established benchmarks.

Table 3. Comparison of pre/post test scores and normalized learning gains.

Group	Pre-Test Mean (%)	Post-Test Mean (%)	Normalized Gain (g)
Control (n=50)	47.8	63.1	0.31
Experimental (n=50)	48.2	72.0	0.49

6.2. Behavioral Engagement

We measured student engagement using rich interaction analytics, including system logs, timestamps for task completion, and utilization of various features. We combined these metrics to look at persistence, motivation, and depth of interactivity. As provided in Table 4, learners in the experimental condition engaged with the interventions almost 20% longer than those in the control condition on average (Experimental M = 38.1 minutes, SD = 6.2; Control M = 32.1 minutes, SD = 5.4). Furthermore, with respect to learning effort, the experimental group engaged with an average of 19.2 problems during a week, compared to 15.0 problems in the control group (28.3% increase). The definitions of "voluntary retries" (i.e., when learners attempt a problem again without being directed to do so) proved to be also revealing. The experimental condition maintained a rate of 63.5% for voluntary retries, while the control maintained a much lower rate of 41.0%. Voluntary retry behaviors demonstrate learner resiliency and intrinsic motivation (research that has looked at our earlier projects and freshman engineering students has provided various context about trying to convey resilience; but clearly we were impressed).

In addition to looking at frequency-based outcomes, we also gleaned qualitative data about some of the other use of interventions. One of the positive outcomes for us is that 87.6% of students in the experimental condition interacted with four or more types of instructional interventions (e.g., hints, annotated examples, reflective prompts, or micro-quizzes). This type of interaction provides us with some evidence of effective scaffolding and that students were using a personalized path through the content. Students used the adaptive elements that would change their behaviors based on performance activity to attempt many alternatives to use different types of strategies to address different types of learning needs. Together, these findings demonstrate that the adaptive system promoted more sustained, diverse, and reflective learner engagement compared to a static instructional environment.

Table 4. Behavioral engagement metrics comparing control and experimental.

Metric	Control Group	Experimental Group	% Difference
Avg. Session Time (min)	32.1	38.1	+18.7%
Problems Attempted	15.0	19.2	+28.0%
Voluntary Retry Rate (%)	41.0	63.5	+55.0%

6.3. Instructional Strategy Effectiveness

In order to evaluate the pedagogical quality, as well as learner perceived value of the instructional strategies selected by the TOPSIS-BP system, we followed a triangulation process comprised of an expert review, student review and statistical correlation. First, we conducted an expert validation with three skilled programming instructors having all taught programming to postgraduate learners for over seven years. We extracted a stratified random sample of 300 selections of instructional strategies made by the system over the duration of the study in order to capture learners of different profiles and contexts. Each expert independently reviewed each selection by looking at the learner's data snapshot in question, and the action made by the system. They rated each action on a three-point scale: 'appropriate', 'sub-optimal but acceptable', or 'not appropriate'. Across all 300 cases, the system's action was rated 'appropriate' in 85.3% of cases. Disagreement seemed to be due to interpretative differences related to escalating to a challenge (e.g., quiz) vs. providing support (e.g., example), deploying acceptable pedagogical variability, rather than outright failures.

Secondly, we embedded a real-time, strategic level satisfaction survey into the platform. After every adaptive instructional action (e.g., showing a hint or code example), learners rated the usefulness of the instructional action, using ratings on a five-point Likert scale. When we aggregated the ratings related to the strategy applications, 78.4% of those sampled were rated as 'very helpful' or 'helpful'. More in-depth qualitative data was obtained through written comments recorded in the open text boxes. Many learners mentioned that they found the annotated code examples and/or reflections prompt especially helpful for cementing their understanding of syntax structure and for building debugging skills. Learners frequently mentioned using the prompts helped them to stop, think and reframe their position—rather than simply moving on.

A correlation analysis between number of times an instructional strategy was used and individual student learning gains—measured using normalized test scores—revealed a moderate positive correlation ($r = 0.41$, $p < 0.01$). Higher engagement with adaptive strategies was positively correlated with more learning improvements—empirical support for the instructional value of such system-selected interventions.

In addition to Likert-scale ratings, learners provided open-ended feedback through free-text boxes embedded after each intervention. Common themes included appreciation for clarity, contextual relevance, and metacognitive prompting. For example, one learner noted, “The annotated code example helped me understand what I was doing wrong in a way that made sense to me.”. Another mentioned: “The reflection prompt made me stop and think instead of rushing through the task.”. These responses suggest that the system’s interventions not only guided performance but also encouraged deeper learning engagement. The qualitative feedback complements the quantitative findings, reinforcing the pedagogical value of the adaptive strategies.

6.4. Comparative Baseline: Rule-Based Engine

To evaluate the additional value of the decision mechanism based on TOPSIS, we undertook a comparative simulation study of the TOPSIS system against the rule-based system we implemented previously. During that implementation, the rule based engine used fixed thresholds such as an error rate > 60%, to trigger predefined instructional actions, without the clear prioritization of learners on various characteristics. For example, 93 learners' anonymized historical data from the previous year was utilized to simulate decision processes for both systems based on identical learner sets. The research team recalculated the mastery scores relative to the specific systems' instructional actions, and normalized them to provide a mastery improvement index. The results (Table 5) indicated that learners who received instructional support from the TOPSIS engine improved their mastering by 16.2% more, and completed exercises with a total of 24.7% fewer fixes (i.e., total mistakes). These results represent better performance for learners who were provided context-sensitive multi-criteria instruction compared to fixed thresholds. In addition to these parameters for effectiveness, feedback was gathered on usability using the

System Usability Scale (SUS), which was completed by 40 students (20 per condition) following a live demonstration of both interfaces. The students who used the TOPSIS-based system reported a SUS of 84.2 as compared to a SUS of 73.5 for the rule-based system. Overall, there was perceived higher levels of coherence, ease of use and perceived instructional quality from the players in the TOPSIS condition. On average, the TOPSIS system was rated 84.2 while the rule-based version was rated at 73.5.

Table 5. Comparison of post-test mastery improvement and SUS scores between the TOPSIS-based and rule-based systems.

Metric	Rule-based System	TOPSIS-based System
Mastery Improvement Index	1.00	1.162
SUS	73.5	84.2

6.5. System Responsiveness and Usability

The technical capacity of the system was assessed through a three-pronged approach using system logs, some backend analytics, and student feedback on system performance. During the body of the study, the platform logged elements of over 2,200 student sessions across a range of contexts of use. System response latency averaged 1.08s (sd = 0.14) based on a delay measure from the time the learner submitted their actions to the time learners saw the instructional strategy in the system. This latency adheres to the earlier definitions of real-time interaction in intelligent tutoring systems presented in the HCI literature.

System uptime was monitored through automated health tests every minute resulting in scheduled uptime and sessions for study. In total the platform averaged 99.8% system availability during the entire study length, with zero critical failure and two instances of temporary service degradation (each lasting less than 5 minutes).

To support the technical logs with user-based insights, we completed semi-structured usability interviews with 16 students in the experimental group. Students were intentionally selected based on their interaction diversity (i.e., frequent user vs. less frequent user). Overall feedback was extremely positive. Students mentioned repeatedly that adaptive interventions were unobtrusive, and provided praise for the seamless way feedback, hints and examples were integrated within the coding interface. Moreover, learners stated that the adaptive strategies "felt like a natural extension of the learning process" and "guided without displacing focus". Overall these findings suggest that the system both reliably works under realistic load conditions and meets usability standards for personalized instructional technology. To further structure the qualitative findings, Table 6 summarizes the most frequently mentioned usability themes during the interviews, along with illustrative student comments.

Table 6. Summary of Themes from Usability Interviews (n= 16).

Theme	Frequency (out of 16)	Representative Comment
Adaptive feedback was non-intrusive	12	"The hints and examples just appeared when I needed them–without breaking my flow."
Interface was intuitive and responsive	11	"I never had to reload or click around to figure out what to do next."
Feedback supported deeper understanding	10	"The examples made me realize what I misunderstood in the code."

Interventions felt natural and well-integrated	9	"It felt like a natural extension of the learning process."
Occasional uncertainty in navigation	3	"Sometimes I wasn't sure where to find the next step after completing a quiz."

In short, our multi-faceted evaluation demonstrates that the proposed system showed statistically important learning gains while maintain high levels of engagement, mapping well to the pedagogical expectations of engagement, and outperformed traditional adaptive methods. Together, these findings present strong evidence for the adoption of TOPSIS-focused personalized learning into actual programming education environments. The research was ethically approved by the university's institutional review board (IRB), and all participants provided informed consent prior to data collection.

7. Discussion

The evaluation results provide strong evidence for the utility of the TOPSIS-based instructional strategy adaptation framework in enhancing both cognitive and behavioral dimensions of learner engagement within a Java programming context. In this section, we interpret these findings through the lens of personalized learning, adaptive system design, pedagogical significance, and the scalability of the proposed approach.

From a pedagogical standpoint, the observed improvements in learning outcomes suggest that the system's adaptive strategies were well-aligned with learner needs and delivered support at critical points in the learning process. This alignment likely helped reduce cognitive overload during challenging tasks and offered timely scaffolding when learners were ready to engage with more complex material. Such timing is essential in promoting deep learning—particularly for novice programmers facing conceptual and syntactic difficulties.

The behavioral engagement patterns—reflected in increased time-on-task, problem attempts, and voluntary retries—point to the motivational benefits of real-time personalization. Learners did not passively consume instructional content but actively interacted with varied forms of support. This suggests that the system may have encouraged self-regulated learning behaviors, helping students to recognize and act upon their changing needs.

The consistency between expert evaluations, learner satisfaction, and system-selected strategies further reinforces the interpretability and instructional coherence of the TOPSIS framework. The system's decisions were rated pedagogically acceptable by instructors in the vast majority of cases, supporting the claim that the selected criteria and weights mirror expert instructional reasoning. Where discrepancies occurred, they reflected differences in teaching style rather than system misjudgment—highlighting the model's flexibility and contextual sensitivity.

Qualitative feedback from learners further corroborated the value of the adaptive interventions. Students emphasized that hints, prompts, and examples were not only helpful but also well-integrated into their learning experience. This supports the notion of the system functioning as a learning companion rather than a directive tutor—aligning with contemporary theories that prioritize metacognition, scaffolding, and learner autonomy.

The comparative results against a traditional rule-based adaptive engine underscore the added pedagogical value of using a multi-criteria decision-making method. Unlike fixed-threshold systems, the TOPSIS framework enabled nuanced, context-sensitive decisions that better reflected the complexity of individual learner states. This adaptability proved to be a key differentiator in both learning outcomes and perceived usability.

Additionally, the positive correlation between the use of adaptive strategies and learning gains—while not causal—suggests a dose-response relationship worth exploring further. This opens future research avenues in longitudinal modeling to understand how adaptive interventions shape learning trajectories over time.

In comparison to the existing literature on educational adaptive systems, the resulting contribution from this study contributes a more nuanced, and analytically supported model of personalization. Current work has typically concentrated on content sequencing (e.g., [50]), learner modeling approaches with Bayesian Knowledge Tracing ([26-28], [51]), or rule-based feedback models ([23-25]; [52]). While these earlier studies have made contributions in their particular contexts, the usability of these models is often compromised by low adaptability, a black-box to decide content sequence, or lack of instructor visibility. For instance, rule-based systems typically use rigid heuristics that do not adapt to the diversity or subtlety of learner behavior, while probabilistic approaches like BKT, though powerful in prediction, offer little interpretability to educators aiming to understand or modify adaptation logic.

Conversely, the TOPSIS-based system used in this study allows for multi-criteria reasoning similar to human instructional decision making. The ability to clearly define the trade-offs between performance indicators and rank actions of students provide instructional approaches that are both contextualized and pedagogically sound. Furthermore, the transparency of the TOPSIS framework permits administrators and designers of a system to explore decision pathways, adjust weights, and re-assess alternatives without the need to re-train a model. This interpretive quality - often lacking with deep learning or probabilistic frameworks - enhances its application in educational settings that require accountability and individualization.

In addition, the architecture of the system allows for scalability and extensibility. The modularization of the TOPSIS engine, the decoupled front-end/back-end design, and the transparent decision logs allows the system to be deployed, tracked, and refined when deployed in different subject domains and learning contexts. Somewhat beyond the need for a few adaptations, comparable models could be applied in math, engineering, and even writing intensive subjects where learners frequently experience the same cognitive difficulties. The ethical mechanisms introduced in the study, in the form of institutional ethics approval and informed consent, additionally contribute to the methodological integrity of the study, and its readiness for wider application. The use of semi-structured interviews, and collection of open feedback, also provides a human-centered evaluation of the system, beyond quantitative exam scores, in order to shed light on learner experience and instructional quality.

While many existing adaptive systems rely on static thresholds or probabilistic models with limited pedagogical transparency, the approach presented in this study bridges the gap between algorithmic precision and instructional interpretability. What distinguishes our system is not only its performance, but also its capacity to support meaningful pedagogical decisions through clear, adjustable criteria. Unlike systems that offer limited feedback on *why* a particular recommendation is made, our TOPSIS-based framework enables both educators and researchers to trace, understand, and revise the logic behind each adaptive intervention. This capacity positions the system not simply as a technical tool, but as a collaborative partner in instructional design—an important shift in the evolution of adaptive educational technologies. By embedding decision transparency and contextual flexibility into the core of the system, our work demonstrates how personalization can be both data-driven and pedagogically grounded.

To conclude, the evidence presented and discussed here supports the conclusion that the adaptive instructional system we adopted here is a strong pedagogically-grounded and technically-feasible solution for learner-centered programming education. The system operationalizes multi-criteria decision-making to enable decision-making through TOPSIS in real time, which reflects an innovative and effective way to provide intelligent support that is tailored to an individual learner. As adaptive learning continues to advance, the approach used in this study represents a powerful combination of precision, versatility, and transparency—qualities essential for the next generation of educational technologies.

8. Conclusions

This study introduced a learner-centric adaptive instructional system that supports Java programming education through the integration of the TOPSIS multi-criteria decision-making method. The system combines real-time learner data, expert-informed weighting of pedagogically relevant criteria, and targeted instructional strategies to deliver support that dynamically adapts to each learner's evolving needs. The approach was implemented and evaluated in a real-world educational setting with a diverse cohort of postgraduate students, resulting in statistically significant improvements in learning outcomes, engagement, and learner satisfaction. Compared to static instructional environments and traditional rule-based adaptation models, the TOPSIS-based framework demonstrated greater effectiveness, transparency, and responsiveness.

Importantly, the system's modular architecture and interpretable decision engine make it highly scalable across different subject domains. Its flexibility allows for easy adaptation to other learning contexts—such as mathematics, engineering, and writing-intensive courses—where personalized support and decision traceability are critical. The transparent criteria-based mechanism also facilitates customization by educators without requiring retraining or technical expertise. These features position the system as a pedagogically grounded and technically feasible solution for broader deployment in personalized learning environments. The results lend strong support to the value of multi-criteria instructional decision making as a central pillar for the next generation of adaptive educational technologies.

Although the system demonstrated promising results, certain limitations must be taken into account. First, the study was only done in one institutional context that yielded a fairly homogenous student sample in terms of academic level (all were postgraduate students) and course structure. Second, although we evaluated the data quantitatively and qualitatively, they were limited to short-term performance undertakings, and we did not measure long-term knowledge retention of learning or knowledge transfer situations. Third, although the expert validation procedure was rigorous, it was a small number of reviewers and could use broader consensus. Lastly, although the system was designed with extensibility in mind, its adaptation logic and criteria had been taken from programming education, and thus, this was not directly tested in relation to generalizability to other domains.

Going forward, we will modify these limitations in an expanded longitudinal study in multiple institutions within various learner populations for generalizability and to conduct transfer testing across domains. We will also implement long-term retention assessments, collect additional behavior and affective indicators, and include dynamic weight updating that will use the history of the learner. Moreover, continuous development will focus on allowing educators to configure and tune the decision model through an intuitive interface to enhance its use and empower instructors to adapt the system to context-specific pedagogical environments. These directions will contribute to advancing the utility, adaptability, and impact of intelligent instructional systems in diverse learning environments.

Author Contributions: Conceptualization, C.T. and A.K.; methodology, C.T. and A.K.; software, C.T. and A.K.; validation, C.T. and A.K.; formal analysis, C.T. and A.K.; investigation, C.T. and A.K.; resources, C.T. and A.K.; data curation, C.T. and A.K.; writing—original draft preparation, C.T., A.K. and P.M.; writing—review and editing, C.T., A.K. and P.M.; visualization, C.T. and A.K.; supervision, C.T. and A.K.; project administration, C.T., A.K., P.M. and C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval are not required for this study, as it exclusively involves the analysis of properly anonymized datasets obtained from past research studies through voluntary participation. This research does not pose a risk of harm to the subjects. All data are handled with the utmost confidentiality and in compliance with ethical standards.

Informed Consent Statement: Informed consent was obtained from all subjects at the time of original data collection.

Data Availability Statement: The data supporting the findings of this study are available upon request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jin, S.H.; Im, K.; Yoo, M.; et al. Supporting students' self-regulated learning in online learning using artificial intelligence applications. *Int. J. Educ. Technol. High Educ.* **2023**, *20*, 37. https://doi.org/10.1186/s41239-023-00406-5.
2. Yaseen, H.; Mohammad, A.S.; Ashal, N.; Abusaimeh, H.; Ali, A.; Sharabati, A.-A.A. The Impact of Adaptive Learning Technologies, Personalized Feedback, and Interactive AI Tools on Student Engagement: The Moderating Role of Digital Literacy. *Sustainability* **2025**, *17*, 1133. https://doi.org/10.3390/su17031133
3. Gabaldón-Estevan, D. Heterogeneity versus Homogeneity in Schools: A Study of the Educational Value of Classroom Interaction. *Educ. Sci.* **2020**, *10*, 335. https://doi.org/10.3390/educsci10110335
4. Chrysafiadi, K.; Troussas, C.; Virvou, M. Combination of fuzzy and cognitive theories for adaptive e-assessment. *Expert Syst. Appl.* **2020**, *161*, 113614. https://doi.org/10.1016/j.eswa.2020.113614.
5. Gunawardena, M.; Bishop, P.; Aviruppola, K. Personalized learning: The simple, the complicated, the complex and the chaotic. *Teach. Teacher Educ.* **2024**, *139*, 104429. https://doi.org/10.1016/j.tate.2023.104429.
6. Graf, A. (2023). Exploring the Role of Personalization in Adaptive Learning Environments. *International Journal Software Engineering and Computer Science (IJSECS)*, *3*(2), 50–56. https://doi.org/10.35870/ijsecs.v3i2.1200
7. Rincon-Flores, E.G.; Castano, L.; Guerrero Solis, S.L.; et al. Improving the learning-teaching process through adaptive learning strategy. *Smart Learn. Environ.* **2024**, *11*, 27. https://doi.org/10.1186/s40561-024-00314-9.
8. Alshamsi, A.M.; El-Kassabi, H.; Serhani, M.A.; et al. A multi-criteria decision-making (MCDM) approach for data-driven distance learning recommendations. *Educ. Inf. Technol.* **2023**, *28*, 10421–10458. https://doi.org/10.1007/s10639-023-11589-9.
9. Hwang, C.L.; Yoon, K. *Multiple Attribute Decision Making, Methods and Applications*; Lecture Notes in Economics and Mathematical Systems, Vol. 186; Springer-Verlag: New York, 1981.
10. Dharmawan, D. Application of TOPSIS Method to Design a Decision Support System in Assessing Teachers' Performance. *J. Inform. Teknol.* **2024**, *6*, 64–69. https://doi.org/10.60083/jidt.v6i1.474.
11. Efan; Krismadinata; Jama, J.; Mulya, R. A Systematic Literature Review of Teaching and Learning on Object-Oriented Programming Course. *Int. J. Inf. Educ. Technol.* **2023**, *13*, 302–312.
12. Abbasi, S.; Kazi, H.; Kazi, A.W.; Khowaja, K.; Baloch, A. Gauge Object Oriented Programming in Student's Learning Performance, Normalized Learning Gains and Perceived Motivation with Serious Games. *Information* **2021**, *12*, 101. https://doi.org/10.3390/info12030101

13. Reunanen, T.; Nieminen, N. Artificial Intelligence as a Catalyst: A Case Study on Adaptive Learning in Programming Education. In *Human Factors, Business Management and Society*; Salminen, V., Ed.; AHFE 2024 International Conference; AHFE Open Access, Vol. 135; AHFE International: USA, 2024. https://doi.org/10.54941/ahfe1004957.
14. Katona, J.; Katonane Gyonyoru, K.I. Integrating AI-based adaptive learning into the flipped classroom model to enhance engagement and learning outcomes. *Comput. Educ. Artif. Intell.* **2025**, *8*, 100392. https://doi.org/10.1016/j.caeai.2025.100392.
15. Hsiao-Chi, L.; Hsiu-Sen, C. Learning Performance in Adaptive Learning Systems: A Case Study of Web Programming Learning Recommendations. *Front. Psychol.* **2022**, *13*. https://doi.org/10.3389/fpsyg.2022.770637.
16. Strielkowski, W.; Grebennikova, V.; Lisovskiy, A.; Rakhimova, G.; Vasileva, T. AI-driven adaptive learning for sustainable educational transformation. *Sustainable Dev.* **2025**, *33*, 1921–1947. https://doi.org/10.1002/sd.3221.
17. Gaitantzi, A.; Kazanidis, I. The Role of Artificial Intelligence in Computer Science Education: A Systematic Review with a Focus on Database Instruction. *Appl. Sci.* **2025**, *15*, 3960. https://doi.org/10.3390/app15073960
18. Thompson, K.; Farr, A.C.; Saunders, T.; Winter, G. The Role of Adaptive Learning Technologies and Conditional Learning. In *Technology-Enhanced Learning and the Virtual University*; Sankey, M.D., Huijser, H., Fitzgerald, R., Eds.; University Development and Administration; Springer: Singapore, 2023. https://doi.org/10.1007/978-981-19-9438-8_26-1.
19. Alshammari, M.T. An Investigation into ChatGPT-Enhanced Adaptive E-learning Systems. *TEM J.* **2025**, *14*, 503–510. https://doi.org/10.18421/TEM141-45.
20. Anindyaputri, N.A.; Yuana, R.A.; Hatta, P. Enhancing Students' Ability in Learning Process of Programming Language using Adaptive Learning Systems: A Literature Review. *Open Eng.* **2020**, *10*, 820–829. https://doi.org/10.1515/eng-2020-0092.
21. Troussas, C.; Papakostas, C.; Krouska, A.; Mylonas, P.; Sgouropoulou, C. Personalized Feedback Enhanced by Natural Language Processing in Intelligent Tutoring Systems. In *Augmented Intelligence and Intelligent Tutoring Systems. ITS 2023*; Frasson, C., Mylonas, P., Troussas, C., Eds.; Lecture Notes in Computer Science, Vol. 13891; Springer: Cham, Switzerland, 2023; pp. 58. https://doi.org/10.1007/978-3-031-32883-1_58.
22. Kerimbayev, N.; Adamova, K.; Jotsov, V.; Shadiev, R.; Umirzakova, Z.; Nurymova, A. Organization of Feedback in the Intelligent Learning Systems. In *Proceedings of the 2024 IEEE 12th International Conference on Intelligent Systems (IS)*, Varna, Bulgaria, 2024; pp. 1–7. https://doi.org/10.1109/IS61756.2024.10705178.
23. As'ad, M. Intelligent Tutoring Systems, Generative Artificial Intelligence (AI), and Healthcare Agents: A Proof of Concept and Dual-Layer Approach. *Cureus* **2024**, *16*, e69710. https://doi.org/10.7759/cureus.69710.
24. Abyaa, A., Khalidi Idrissi, M. & Bennani, S. Learner modelling: systematic review of the literature from the last 5 years. *Education Tech Research Dev* **67**, 1105–1143 (2019). https://doi.org/10.1007/s11423-018-09644-1
25. Rai, S.S.; Nikam, A.V. The Role of Rule-Based Teaching and Learning in Knowledge-Based System for Management Education. *Int. J. Creat. Res. Thoughts* **2021**, *9*, 2. Available online: https://ijcrt.org/papers/IJCRT2102359.pdf.
26. Kołodziejczyk, J.; Grzegorczyk-Dłuciak, N.; Kuliga, E. Rule-based expert system supporting Individual Education-and-Therapeutic Program composition in SYSABA. *Procedia Comput. Sci.* **2022**, *207*, 4535–4544. https://doi.org/10.1016/j.procs.2022.09.517.
27. Shabani, N.; Beheshti, A.; Farhood, H.; Bower, M.; Garrett, M.; Alinejad-Rokny, H. A Rule-Based Approach for Mining Creative Thinking Patterns from Big Educational Data. *AppliedMath* **2023**, *3*, 243-267. https://doi.org/10.3390/appliedmath3010014
28. Hooshyar, D.; Druzdzel, M.J. Memory-Based Dynamic Bayesian Networks for Learner Modeling: Towards Early Prediction of Learners' Performance in Computational Thinking. *Educ. Sci.* **2024**, *14*, 917. https://doi.org/10.3390/educsci14080917
29. Xu, J.; Dadey, N. Using Bayesian Networks to Characterize Student Performance across Multiple Assessments of Individual Standards. *Appl. Meas. Educ.* **2022**, *35*, 179–196. https://doi.org/10.1080/08957347.2022.2103134.
30. Chanthiran, M.; Ibrahim, A.B.; Abdul Rahman, M.H.; Mariappan, P. Bayesian Network Approach in Education: A Bibliometric Review Using R-Tool and Future Research Directions. *The Eurasia Proceedings of Educational and Social Sciences* **2022**, *25*, 17–25. https://doi.org/10.55549/epess.1191900.
31. Hegazi, M.O.; Almaslukh, B.; Siddig, K. A Fuzzy Model for Reasoning and Predicting Student's Academic Performance. *Appl. Sci.* **2023**, *13*, 5140. https://doi.org/10.3390/app13085140
32. Vora, D.; Tulshyan, V. Evaluation of Student's Performance using Fuzzy Logic. *Int. J. Novel Res. Dev.* **2024**, *9*, 800–810. Available online: https://www.ijnrd.org/papers/IJNRD2404596.pdf.
33. Amelia, N.; Abdullah, A.; Mulyadi, Y. Meta-analysis of Student Performance Assessment Using Fuzzy Logic. *Indonesian J. Sci. Technol.* **2019**, *4*, 74–88. https://doi.org/10.17509/ijost.v4i1.15804.
34. Tariq, M.B.; Habib, H.A. A Reinforcement Learning Based Recommendation System to Improve Performance of Students in Outcome Based Education Model. *IEEE Access* **2024**, *12*, 36586–36605. https://doi.org/10.1109/ACCESS.2024.3370852.

35. Xie, J. The Role of Reinforcement Learning in Enhancing Education: Applications in Psychological Education and Intelligent Tutoring Systems. *Highlights in Sci. Eng. Technol.* **2025**, *124*, 123–131. https://doi.org/10.54097/rkxbvk42.
36. Osakwe, I.; Chen, G.; Fan, Y.; Rakovic, M.; Li, X.; Singh, S.; Molenaar, I.; Bannert, M.; Gašević, D. Reinforcement learning for automatic detection of effective strategies for self-regulated learning. *Comput. Educ. Artif. Intell.* **2023**, *5*, 100181. https://doi.org/10.1016/j.caeai.2023.100181.
37. Li, H. Evaluation system of vocational education construction in China based on linked TOPSIS analysis. *Heliyon* **2024**, *10*, e39369. https://doi.org/10.1016/j.heliyon.2024.e39369.
38. Wang, N.; Ren, Z.; Zhang, Z.; Fu, J. Evaluation and Prediction of Higher Education System Based on AHP-TOPSIS and LSTM Neural Network. *Appl. Sci.* **2022**, *12*, 4987. https://doi.org/10.3390/app12104987
39. Wang, T.-C.; Nguyen, T.T.; Phan, B.N. Analyzing higher education performance by entropy - TOPSIS method: A case study in Viet Nam private universities. *Meas. Control* **2022**, *55*, 385–410. https://doi.org/10.1177/00202940221089504.
40. Yu, G.; Fu, Z.; Huang, Y.; Lee, C.-T. Data analysis of Higher Education System Based on TOPSIS and Time Series Model. In *Proceedings of the 2021 5th International Seminar on Education, Management and Social Sciences (ISEMSS 2021)*; Atlantis Press: 2021; pp. 1014–1024. https://doi.org/10.2991/assehr.k.210806.194.
41. Jin, H.; Ma, Y.; Ma, W.; Zhu, H.; Fang, H. Dynamic evaluation of higher education system based on TOPSIS model. In *Proceedings of the 2021 2nd International Conference on Information Science and Education (ICISE-IE)*, Chongqing, China, 2021; pp. 916–920. https://doi.org/10.1109/ICISE-IE53922.2021.00210.
42. Koltharkar, P.; K.K., E.; Sridharan, R. Application of fuzzy TOPSIS for the prioritization of students' requirements in higher education institutions: A case study: A multi-criteria decision making approach. In *Proceedings of the 2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, Pondicherry, India, 2020; pp. 1–7. https://doi.org/10.1109/ICSCAN49426.2020.9262329.
43. Abood, N.; Faris, M.; Alashari, O.; Mohammed Naser, A. Evaluating educational services using TOPSIS method with eigenvalue methodology based on GIS. *Int. J. Nonlinear Anal. Appl.* **2023**, *14*, 329–336. https://doi.org/10.22075/ijnaa.2022.28699.4062.
44. Huang, T.-Y.; Chen, W.-K.; Nalluri, V.; Huynh-Cam, T.-T. Evaluating E-Teaching Adoption Criteria for Indian Educational Organizations Using Fuzzy Delphi-TOPSIS Approach. *Mathematics* **2022**, *10*, 2175. https://doi.org/10.3390/math10132175
45. Yuhefizar, Y.; Sutrisno, S.; Rinaldi, U.; Aini, Q.; Indrasari, M.; Rahim, R. Analyzing the Benefits of MCDA-TOPSIS Decision Support System Method for Management Education. *J. Educ. Teach. Train.* **2023**, *14*, 111–116. https://doi.org/10.47750/jett.2023.14.02.010.
46. Kang, J. TOPSIS-Based Multi-Criteria Decision-Making Using 2-Tuple Linguistic Neutrosophic Numbers: A Comprehensive Evaluation of Industry-Education Integration in Applied Undergraduate Universities. *Neutrosophic Sets Syst.* **2025**, *77*, 1. Available online: https://digitalrepository.unm.edu/nss_journal/vol77/iss1/30.
47. Çelikbilek, Y.; Tüysüz, F. An in-depth review of theory of the TOPSIS method: An experimental analysis. *J. Manag. Anal.* **2020**, *7*, 281–300. https://doi.org/10.1080/23270012.2020.1748528.
48. Mbuli, N. A Survey of Applications of Analytic Hierarchy Process (AHP) in Wind Generation Research. In *Proceedings of the 2024 4th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Male, Maldives, 2024; pp. 1–5. https://doi.org/10.1109/ICECCME62383.2024.10796673.
49. Hidayat, T.; Kurniawan, H.; Astuti, I.A.; Pravitasari, R.; Kristyawan, Y.; Syahadiyanti, L. The Effect and Impact of the Electre Method for Sensitivity Testing Based on the Case Study Selection of Outstanding Students. In *Proceedings of the 2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*, Jakarta, Indonesia, 2022; pp. 118–122. https://doi.org/10.1109/IC2IE56416.2022.9970134.
50. Uzhga-Rebrov, O.; Grabusts, P. Application of the PROMETHEE Method with Missing Criteria Values. In *Proceedings of the 2023 IEEE 64th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS)*, Riga, Latvia, 2023; pp. 1–7. https://doi.org/10.1109/ITMS59786.2023.10317689.
51. Huan, J.; et al. Structural Vulnerability Study of AC-DC Hybrid Grids Considering the VIKOR Method. In *Proceedings of the 2024 International Conference on HVDC (HVDC)*, Urumqi, China, 2024; pp. 854–858. https://doi.org/10.1109/HVDC62448.2024.10723052.
52. Huang, J.-J.; Chen, C.-Y. Using Markov Random Field and Analytic Hierarchy Process to Account for Interdependent Criteria. *Algorithms* **2024**, *17*, 1. https://doi.org/10.3390/a17010001
53. Balali, V.; Zahraie, B.; Roozbahani, A. Integration of ELECTRE III and PROMETHEE II Decision-Making Methods with an Interval Approach: Application in Selection of Appropriate Structural Systems. *J. Comput. Civil Eng.* **2014**, *28*, 297–314. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000254.

54. Taherdoost, H.; Madanchian, M. VIKOR Method—An Effective Compromising Ranking Technique for Decision Making. *Macro Manag. Public Policies* **2023**, *5*, 27–33. https://doi.org/10.30564/mmpp.v5i2.5578.
55. Sheng, X.; Lan, K.; Jiang, X.; Yang, J. Adaptive Curriculum Sequencing and Education Management System via Group-Theoretic Particle Swarm Optimization. *Systems* **2023**, *11*, 34. https://doi.org/10.3390/systems11010034
56. Xu, S.; Sun, M.; Fang, W.; Chen, K.; Luo, H.; Zou, P.X.W. A Bayesian-based knowledge tracing model for improving safety training outcomes in construction: An adaptive learning framework. *Develop. Built Environ.* **2023**, *13*, 100111. https://doi.org/10.1016/j.dibe.2022.100111.
57. Williams, A. Delivering effective student feedback in higher education: An evaluation of the challenges and best practice. *Int. J. Res. Educ. Sci. (IJRES)* **2024**, *10*, 473–501. https://doi.org/10.46328/ijres.3404.