DOI: https://doi.org/10.15276/hait.08.2025.24
UDC 37.091.3:004.9:378.018.43

Analysis of methods for supporting personalization in IT education

Andrii V. Khyzhniak[1)]
ORCID: https://orcid.org/0009-0008-8655-3736; alf.andrey@stu.cn.ua
Volodymyr V. Kazymyr[1)]
ORCID: https://orcid.org/0000-0001-8163-1119; vvkazymyr@stu.cn.ua. Scopus Author ID: 56644727300
[1)] Chernihiv Polytechnic National University, 95, Shevchenko Str, Chernihiv, 14030, Ukraine

ABSTRACT

The increasing complexity of digital learning environments and the growing demand for individualized educational experiences underscore the importance of developing effective methods for personalizing practical tasks. The purpose of this study is to analyze existing approaches to the personalization of practical tasks in IT education and to identify their advantages, limitations, and potential for integration into digital learning ecosystems. The main objectives include the identification of technological solutions that enable automated generation and assessment of personalized practical tasks, and their effectiveness for supporting adaptive and scalable learning processes. The research methodology is based on a comparative analysis of scholarly sources, existing educational platforms, and practical implementations of personalized tasks. Special attention is given to the development of conceptual frameworks that enable personalization not only at the level of task complexity but also in terms of content variability, learning pathways, and assessment mechanisms. The review identifies the most effective strategies for achieving scalable personalization in digital education. The analysis also reveals the limitations of current solutions. The critical limitations are: insufficient support for comprehensive practical task parametrization, high technical barriers for instructors, incomplete learning management system integration, security vulnerabilities in local deployment scenarios, and limited automation in assessment processes. These constraints highlight the need for more sophisticated, user-friendly solutions. The findings provide a foundation for developing integrated information technologies that combine personalization methods with adaptive learning and artificial intelligence tools, supporting the evolution of digital learning ecosystems toward more effective, scalable, and integrity-focused educational approaches. The scientific novelty of the research lies in the systematic comparison of personalization methods and the identification of criteria for their effectiveness in IT education. The practical significance is determined by the possibility of integrating the proposed approaches into digital platforms, thereby improving the quality of training, optimizing the learning process, and ensuring the flexibility required by modern educational ecosystems.

Keywords: methods; analysis; personalized assignments; automated generation; automated assessment; academic integrity; parameterized task

For citation: **Khyzhniak A. V., Kazymyr V. V. "Analysis of methods for supporting personalization in IT education".** ***Herald of Advanced Information Technology*****. 2025; Vol.8 No.3: 366–381. DOI: https://doi.org/10.15276/hait.08.2025.24**

INTRODUCTION

In digital learning ecosystems (DLE) [1], practical tasks play a critical role in the education of IT specialists. The acquisition of durable, labor-market-relevant skills – such as programming, network administration, and data analysis – and the development of expertise in these fields largely depend on hands-on assignments. The objectives of higher education, as defined in Ukrainian legislation, align with this principle. Higher education institutions are expected to equip students with the competencies necessary for effective professional practice in a given specialty or field of knowledge. However, data from the Ministry of Digital Transformation of Ukraine indicate that nearly 50 % of IT graduates end up working outside their field due to a mismatch between curricula and labor market needs [2]. This underscores the importance of embedding practical cases into programs, as they contribute to students' skill development and employability (as highlighted in studies on integrating professional standards and real-world tasks into IT education programs [3]).

Holdgraf, C., et al. [4] examined the use of containerized and cloud-based environments for Data Science education, demonstrated that hands-on computational instruction significantly enhances students' mastery of data analysis tools by providing practical experience. Therefore, in disciplines such as computer and engineering sciences, laboratory-based work should constitute a core component of the educational process. Laboratory-oriented courses in computer science effectively link disciplinary learning outcomes with skills that are directly relevant to the labor market.

Issues such as plagiarism and poor-quality task completion have long been a concern. Traditionally, person instructors supervised students during in-assessments, and most practical work was completed under observation in classrooms. Even when tasks were performed independently at home, students were usually required to present and defend their work in person. The shift to remote learning has introduced new challenges. In particular, asynchronous online education exacerbates concerns about academic integrity, as it limits instructors' ability to verify that the work was completed personally by each student. Ensuring that students complete practical tasks personally – and thus acquire the intended skills - has become increasingly challenging. Academic literature notes that detecting misconduct in online learning environments is more complex [5].

Common forms of online academic dishonesty include soliciting assistance from third parties during exams, sharing login credentials, and using AI to generate responses [6], [7]. Empirical evidence from the authors corroborates these trends. In the "Unix Operating Systems" course (2023/2024), 6.7% of students used AI to falsify answers, while 16.7% employed graphic editors. These reflect only detected cases; the actual prevalence is likely higher. An anonymous survey [8] revealed that 60 % of students admitted to regularly cheating during online exams.

Various strategies have been implemented to counter academic dishonesty, including specialized proctoring software that monitors students' behavior, such as eye movements, posture, keystrokes, and the task execution environment. However, these tools still require instructor supervision. The authors argue that implementing automated personalized learning formats offers a promising avenue for mitigating academic misconduct.

Furthermore, the potential of adaptive learning should be leveraged. This study focuses on enhancing the training of engineering specialists through improved methods for the automated generation and verification of personalized practical tasks. When combined with adaptive learning, these methods not only enhance student engagement and performance but also foster academic integrity and improve the overall quality of higher education. This analysis serves as a foundation for future research in this area.

THE PURPOSE AND OBJECTIVES OF THE RESEARCH

The objective of the study is to investigate and analyze existing methods in the field of personalization of practical tasks, with a particular focus on approaches, techniques, and the direct implementation of processes for the automated generation and verification of such tasks. The study aims to identify and outline the main limitations of current methods and systems for automated generation and verification of practical tasks (if such limitations are found), substantiate the need for improving these methods and systems, and determine the primary directions for potential future research. The research aims to adopt best practices and apply them to address further scientific and applied research challenges. Based on the outlined problem statement, this study addresses the following research questions:

RQ1: What methods are currently applied to support personalization in IT education in general?

RQ2: Which methods appear to be the most promising for supporting personalized practical tasks while ensuring the prevention of academic dishonesty?

RQ3: What specific methods exist for the automated generation and assessment of practical tasks?

RQ4: What are the limitations of existing automated generation and assessment methods?

LITERATURE ANALYSIS

The definition of personalized learning has only recently appeared in Ukrainian educational policy documents. According to the recommendations of the MESU [9], personalization is defined as "the ability to create a learning environment that enables students to pursue their own goals, pace, and/or mode of learning."

Personalized learning is an approach that aims to develop an effective trajectory for knowledge acquisition, aligning with a learner's strengths while addressing their weaknesses to achieve desired outcomes. This approach involves adapting the content, complexity, format, and pace of tasks to the individual characteristics of each student, such as prior knowledge, learning style, interests, and academic goals. Tasks are not merely pre-adjusted for each learner but are dynamically shaped during the learning process based on the student's progress [10], [11]. A key feature of personalization in practical tasks is dynamic adaptation: learning materials and assignments are not static but continuously evolve in accordance with the student's growing knowledge, experience, and performance [10], [12]. Another essential principle is the student-centered perspective: unlike traditional systems, personalized learning prioritizes the learner's outcomes [11].

Advances in modern technology have made personalized learning increasingly adaptable, and adaptive learning, in turn, has become increasingly personalized. In line with this trend, a revised approach has emerged - personalized adaptive learning (PAL) [13]. PAL is a pedagogical model that leverages technology to dynamically adjust instructional strategies through real-time monitoring [Peng]. Its defining features include recognition of each learner's unique traits, monitoring and adapting instruction based on prior knowledge and real-time progress, aligning learning with students' personal long-term goals, and dynamically modifying strategies and content according to evolving needs. In the era of artificial intelligence (AI), machine learning (ML), and big data analytics, technology plays a decisive role in strengthening personalized education. Within digital learning environments [1], such systems can reorganize the sequence and pace of assignments based on earlier performance, adjust task complexity through ML and AI tools, and provide personalized hints, feedback, or alternative tasks.

The integration of personalization methods into digital learning environments brings several advantages. It ensures a learning experience adapted to individual needs, preferences, and progress. Personalization also enhances learning outcomes by tracking performance, identifying knowledge gaps, and recommending content or exercises for recursive reinforcement [14]. In addition, such systems provide real-time, constructive feedback that helps learners recognize and correct mistakes promptly, thereby improving the efficiency of the learning process. The application of AI/ML can further increase student engagement and motivation through features such as gamification and dynamically generated, recommendation-based content that analyzes large repositories of educational resources and suggests materials suited to individual learner profiles [14]; [15].

Within the field of personalized learning, a wide variety of methods and techniques are employed. However, it is crucial to distinguish between pedagogical methods – didactic approaches that define how the learning process should be organized to be learner-oriented and responsive to individual needs [16] – and technological or analytical methods, which are used to operationalize personalization in specific environments. The latter focus on data processing, analysis, or generation, serve as tools within personalized strategies, and require careful integration into the educational context.

RESEARCH METHODOLOGY

The methodological framework of this study is based on a systematic analysis of existing approaches to personalization in digital learning environments and a comparative analysis of methods. A comprehensive literature review was conducted to identify existing methods and models of learning personalization.

Identified approaches were systematically compared according to several criteria: the degree of personalization achievable through parametrization, adaptability to learners' individual profiles and performance data, integration with automated assessment mechanisms, scalability, and suitability for practical tasks in IT training contexts. This comparative framework enabled the identification of the strengths and limitations of different personalization techniques.

In addition, the study incorporates insights from the work of A.H. Nabizadeh [17], which systematizes methods of learning path personalization, identifies personalization parameters, elaborates recommendation strategies, and describes approaches for monitoring learning effectiveness.

ANALYSIS OF METHODS OF PERSONALIZATION

The following section provides a closer examination of methods applied for learning personalization purposes and answering on RQ1. Combined with personalization theories (Mastery Learning [18], [19], Self-Determination Theory [20], Metacognitive Theory [21], Model-Centered Instruction [22], Activity Theory / Cultural-Historical Activity Theory [23], [24], Challenge-Point Framework [25]), these methods enable the design of effective personalized environments for the completion and assessment of practical tasks.

Models and methods for supporting learning personalization

1. Rule-Based Adaptation [26], [27], [28]. This method is applied for simple and interpretable adaptation: if a student fails to complete a task, the system selects an easier variant according to a set of condition–action rules (IF-THEN) based on student behavior (performance, time spent, use of hints). Interpretable rules are employed to determine the next exercises based on topic and difficulty, and the rule-based mechanism provides transparent and flexible recommendation logic.

Advantages: transparency, interpretability, low entry barrier, manual configuration possibility, and no need for large datasets.

Limitations: limited scalability and flexibility, requires maintenance when content changes, and is difficult to improve without human intervention.

 ISSN 2663-0176 (Print) ISSN 2663-7731 (Online)

2. Stereotype → Overlay Method [29], [30], [31], [32]. This method utilizes the concept of stereotypes, which are basic profiles assigned to a student at the beginning of the learning process. For example, a student may be assigned the stereotype "beginner in Unix", which defines initial knowledge values and accelerates the start of individual modeling.

More commonly, a hybrid approach is used following the "stereotype → overlay" principle, where as more data are collected (through learning progress and increased information about the student), the initial stereotypes are gradually replaced with more flexible overlay models that more accurately reflect the student's knowledge.

3. Probabilistic Methods – Bayesian Networks, Naïve Bayes Classifier, and Fuzzy Logic Approach. Probabilistic methods are applied to model a student's knowledge, skills, behavior, or cognitive states in terms of probabilities. Unlike rigid binary decisions ("knows" / "does not know"), these methods enable the modeling of uncertainty in student competence, allow the assessment of the degree of knowledge acquisition, support the prediction of actions, and facilitate decision-making within personalized educational systems.

3.1. Bayesian Knowledge Tracing (BKT). This model estimates the probability that a student has mastered a given concept after each attempt to solve a task. It also takes into account the probability of guessing the correct answer and the probability of making an error despite having knowledge of the material [33].

3.2. Bayesian Networks. This approach models relationships between concepts and allows the estimation of the probability of mastering a complex concept, given knowledge of whether the student has or has not mastered related concepts [34].

3.3. Naive Bayes Classifier. This model evaluates the probability that a student belongs to a certain class based on input features (such as the number of mistakes, task completion time, or initial test results).

3.4. Hidden Markov Models. This method determines the probabilities of transitions between "hidden states" of student knowledge, taking into account the sequence of task completion.

3.5. Item Response Theory (IRT). This approach predicts student performance, where the probability of a correct answer depends on both task difficulty and student ability. Task difficulty is automatically estimated from statistical data on student responses using calibration methods. Student ability is automatically assessed based on responses to a series of tasks with known difficulty parameters [35].

3.6. Fuzzy Logic Method makes it possible to smoothly evaluate the degree of knowledge acquisition — not just "knows" / "does not know," but also "partially knows" or "almost mastered" (as a value between 0 and 1) [36].

4. Latent Probabilistic Methods [37]. Matrix Factorization (MF) and Tensor Factorization (TF) are not classical probabilistic knowledge models, such as Bayesian Knowledge Tracing (BKT) or Bayesian Networks, but they are employed to predict the probability of student success based on latent factors.

In Matrix Factorization (MF), student performance on a task is estimated as the predicted probability of a correct response, calculated as the scalar product of latent factor vectors (each student and each task are represented as vectors of hidden factors). Tensor Factorization (TF) extends MF by introducing a temporal dimension: it incorporates timestamps to model the dynamics of knowledge change. The student–task–time representation forms three-dimensional structures, enabling the prediction of the probability of a correct response at a specific moment in time.

5. Collaborative Filtering (CF) [38], [39], [40], [41] is a recommendation method that, unlike content-based approaches, does not rely on the intrinsic characteristics of tasks but rather on the similarity between students or between tasks, based on prior actions or ratings. CF personalizes task recommendations by leveraging the collective experience of students who share similar performance patterns. Experimental results on educational datasets demonstrate the high relevance and accuracy of such recommendations.

6. Multi-Criteria Decision Making (MCDM) is a family of approaches for solving multi-criteria decision problems that can be combined with machine learning to support personalized recommendations or adaptive learning pathways. MCDM has been widely adopted in educational technologies, particularly for personalized resource selection, evaluation of learning platforms, and the construction of individualized learning trajectories [42], [43].

Advantages: the ability to simultaneously account for multiple aspects (both qualitative and quantitative), transparency and traceability of decisions, and flexibility in adjusting weights or criteria.

Limitations: requires careful selection of criteria and accurate weight definition, introduces subjectivity (many methods rely on expert judgments), and may not scale efficiently to very large sets of alternatives.

7. Difficulty Ranking Model (DRM) is a statistical approach for personalization in online learning, which generates a ranked list of tasks according to their difficulty for each student. The main goal of DRM is to determine the individualized difficulty of a task, taking into account not only general statistics but also the specific interactions each student has with the content. Unlike a general difficulty ranking, DRM provides a personalized ranking for each user. For example, the EduRank algorithm [41] personalizes the sequence of tasks based on the student's individual level. Tasks are ranked from easier to harder, improving learning efficiency.

Unlike Item Response Theory (IRT), which estimates the probability of a correct response based on task difficulty and student ability, DRM does not require a priori task calibration: it learns from actual responses, making it suitable for scaling in online environments with a large number of dynamic tasks. However, DRM requires a large number of task-pair comparisons (i, j) for each student; if a student has completed only a few tasks, the ranking may be unreliable or impossible. Additionally, DRM does not consider learning context, learning style, or time spent on tasks, so personalization is based solely on historical successes or failures, ignoring other important adaptive factors.

8. Dynamic Programming Method (DP) [44], [45], [46] is a method for solving complex optimization problems by decomposing them into simpler subproblems and sequentially combining optimal solutions of these subproblems. In the context of personalized learning, DP helps to identify optimal learning trajectories or task sequences that best suit a particular student, considering their current knowledge level, goals, and constraints (time, difficulty).

Advantages: ability to find an optimal or near-optimal solution for problems with a well-defined structure, efficiency through memorization of intermediate results, and the capacity to incorporate multiple criteria and constraints into the model. Limitations: applicable only to problems that can be decomposed into overlapping subproblems with an optimal substructure property, and computational complexity can increase sharply as the number of parameters grows.

DP is a powerful tool for modeling and optimizing adaptive learning trajectories, enabling the formation of personalized sequences of learning tasks that account for individual student characteristics and resource constraints. In the study by [46], DP is used to construct optimal learning pathways by decomposing the problem into subproblems and applying Bellman updates. In combination with reinforcement learning (RL) and cognitive graphs, DP supports dynamic personalization of the learning path, where cognitive graphs represent dependencies between concepts and resources, allowing the model to understand the context of learning interactions, while RL algorithms are used to automatically update the policy for selecting the next content based on student interactions.

In [47] DP (specifically DQN – Deep Q-Network) is employed to develop a personalized strategy for recommending learning content in an adaptive educational environment. DQN is an approximate variant of DP in which a neural network estimates the action-value function (Q-function), and the Bellman equation is applied during training. This approach exemplifies a modern hybrid of DP and deep learning in the context of adaptive educational solutions [48].

9. Automated Task Generation Methods involve the creation of learning tasks or test questions based on algorithmically defined rules or by leveraging AI. They represent a key mechanism of personalization in DLEs, enabling adaptation of content, complexity, and task types to the individual needs of students. Task generation can be implemented through templates, language models, domain-specific languages, or integrated into broader adaptive systems (intelligent tutoring systems, RL frameworks).

Approaches to automated task generation include:

1) Template- and rule-based methods: static templates with parametrization;

2) Natural Language Processing (NLP): generating questions from text using transformer or seq2seq models;

3) Domain-Specific Languages (DSL): using formalized task descriptions that can be automatically compiled into individual variants;

4) Dynamic task generation: adjusting tasks in real time based on student performance (e.g., success rate, common errors).

Automated task generation supports the creation of dynamically adapted assignments without manual intervention from instructors, while accounting for difficulty level, learning pace, and individual knowledge gaps. It also enhances task diversity and uniqueness, which is especially important in large-scale online learning and assessment scenarios.

10. Automated Assessment Methods [49], [50], [51], [52], [53] are a set of approaches designed to evaluate learning tasks without direct instructor involvement, in order to provide fast and objective feedback, update student knowledge models, and support adaptive learning environments. Automated assessment is an essential prerequisite for scalable personalization, enabling the dynamic adaptation of content, complexity, and task types to each student's individual profile.

Such methods include [54]: comparison with a predefined solution (gold standard), unit testing, code-based evaluation, NLP-based grading for open-text answers, and ML-based evaluation techniques. Hybrid approaches are also emerging. For example, [53] presents a hybrid method that combines NLP capabilities with Random Forest Regression to continuously assess the quality of textual responses.

Automated task generation and assessment methods are most effective when applied in combination, forming a key area of our research focus, particularly for practice-oriented tasks in IT disciplines. Among the existing solutions with capabilities for both automated generation and assessment of tasks, the following stand out and will be examined in more detail in the next section:

1) SERA framework – a flexible platform for conducting lab exercises in security-related courses, with automated evaluation [49];

2) APG (Automatic Problem Generation) – a system that automatically generates personalized practice tasks for students and provides tools for automated configuration of lab environments with unique setups for each student [50];

3) PAGE system – a tool for simplifying the creation and administration of personalized digital assessments in the Canvas LMS [51];

4) MCTest – an assessment tool that uses parameterized tasks and automated solution checking in introductory programming courses [52].

11. AI/ML Methods [55], [56] are widely applied in IT education. In the context of personalizing practical tasks, AI/ML technologies enable the analysis of large datasets on students (including their knowledge, behavior, and task performance) to automatically generate individualized tasks and recommendations, adapt task difficulty levels, and select suitable materials and task formats. Based on behavioral and cognitive characteristics, AL/ML methods help construct the most effective learning trajectories and tasks. This allows practical assignments to be dynamically personalized to individual needs, thereby improving motivation and learning outcomes.

Among the ML approaches, the following should be highlighted:

11.1. Reinforcement Learning (RL) [57], [58] is a class of ML methods that models the learning process through interaction between an agent and an environment. The agent performs actions, receives feedback (in the form of rewards), and attempts to maximize cumulative rewards by following a particular policy.

In an educational setting, an RL agent can function as an adaptive tutor, analyzing student responses, selecting subsequent tasks or explanations, and updating its policy according to observed learning outcomes.

Advantages of RL in personalization include:

- real-time adaptation: the model updates its policy continuously as new data is collected;
- long-term planning: RL optimizes entire sequences of actions rather than just the next task;
- flexibility and scalability: RL can adapt to different types of students or courses.

However, RL also presents certain limitations:

- high sample complexity: it often requires a large number of learning episodes to converge, which may not be feasible in real educational settings;
- reward formalization challenges: defining an appropriate reward function (e.g., the "usefulness" of a task) is difficult;
- knowledge state modeling: accurately representing a student's knowledge state remains a significant challenge.

11.2. Deep Reinforcement Learning (DRL) is being increasingly explored for complex personalization scenarios. For example, Deep Q-Networks (DQN) have been applied to multimodal educational data. In the study by Govea et al. [59], DQN was employed to process images and audio, enabling the recognition of students' emotions in hybrid learning settings and facilitating the adaptation of educational interventions based on emotional responses.

11.3. Clustering is an emerging and important technique in Educational Data Science [60] and educational data analytics. The primary operation of clustering involves grouping students based on shared characteristics (similarity measures) to identify latent patterns in educational data, thereby enabling personalized group recommendations. Clustering is applied to a wide range of educational tasks [61], including the analysis of student behavior, interactions, engagement, motivation, and emotions, as well as performance analysis, success and dropout prediction, learning support, feedback and recommendation provision, collaborative task support, and assessment of students' physical and mental health.

Clustering groups students with common features (e.g., behavioral types, learning style, pace) to create personalized groups. Individual learning trajectories - flexible learning paths – are assigned to different groups or classes based on cluster analysis. Support modes, such as pace, content, and assessment format, are adapted according to cluster characteristics.

Clustering combined with individualized learning trajectories represents a novel approach that enables the creation of adaptive pathways at the level of interactive systems, supporting logical structures and analytics. Common clustering methods [61] include k-means, hierarchical clustering, fuzzy c-means, EM clustering, self-organizing maps (SOM), spectral clustering, DBSCAN, and others.

11.4. Graph Neural Networks (GNNs) are a type of neural network specifically designed to operate on graph-structured data, where entities (nodes) are connected via edges (relationships). GNNs facilitate information exchange between nodes, considering both local context and the overall graph structure [62]. In personalized learning, GNNs can be utilized for analyzing student knowledge graphs (concepts ↔ mastery), tracing learning progress [63], identifying relationships among learning components, and predicting performance based on the graph structure of student actions [58]. The relationship between complexity and personalization level is shown on Fig 1.

The analysis reveals a strong positive correlation between implementation complexity and achievable personalization levels across all examined methodological categories. Advanced AI/ML approaches, including tensor factorization, reinforcement learning, deep Q-networks, and graph neural networks, demonstrate the highest personalization capabilities but require substantial implementation complexity, positioning them in the high-complexity, high-personalization quadrant. Intermediate approaches such as Bayesian networks and matrix factorization occupy a balanced position with moderate-to-high complexity and corresponding personalization levels, representing viable solutions for production environments with established technical infrastructure. Entry-level methodologies, including rule-based adaptation and collaborative filtering, exhibit lower complexity requirements with variable personalization

Fig. 1. **Relationship between complexity and personalization level**
Source: **compiled by the authors**

outcomes. Notably, the analysis identifies an absence of methods in the low-complexity, high-personalization quadrant, indicating that achieving superior personalization inherently requires accepting increased implementation complexity – a fundamental trade-off that constrains methodological selection in resource-limited educational technology deployments.

A wide range of methods is applied for the purposes of learning personalization, spanning from simple variable exercises to deep cognitive adaptation using student models and AI. However, the issue of academic dishonesty remains, which is why the authors of this study focus on addressing a complex challenge: enhancing the effectiveness of engineering education with an emphasis on practical tasks while acknowledging the existence of academic misconduct. The use of methods such as task parametrization, automated generation, and automated assessment of personalized practical assignments constitutes an effective mechanism to counteract cheating.

To directly address RQ2, the most promising approach is strategic integration of advanced AI methods with existing automated generation and assessment methods, which help to create next-generation personalized learning environments.

ANALYSIS OF METHODS OF AUTOMATED GENERATION AND ASSESSMENT

We focus on methods for the automated generation and assessment in more detail for answering RQ3. In these studies, the authors aim to automate the creation of assignments with unique datasets for each student as a means to reduce cheating and enhance engagement. It is important to clarify that the authors consider parameterized tasks (including practical ones) as a type of personalized assignment that maintains a consistent didactic objective while being unique for each learner. Uniqueness is achieved either through different sets of input parameters or through varying initial conditions for each student. Such an approach provides a unique parameterized task for each student while preserving the overall methodological structure and difficulty level. This approach is commonly referred to as a “parameterized task” or “variant-based personalization”.

1. SERA. SERA framework [49] provides a flexible platform for defining laboratory exercises in security courses and automating their assessment. The system consists of several components (SERA Engine = Generator + Validator + Monitor), modules, and utilities for creating and evaluating personalized tasks. The authors highlighted limitations of traditional approaches, where all students receive the same assignment, enabling answer copying. They emphasize the need for task customization and automation to save instructors’ time and provide students with immediate feedback.

The proposed solution, the SERA Framework, integrates with existing LMS platforms via the LTI standard to obtain course student information. The framework enables the creation of exercises using activity files – task descriptions (activity templates) from which individualized tasks (activity instances) are generated for each student. Expected results are defined according to these templates (activity submissions). The system uses JSON-based templates to automate task generation and file validation. Key quality indicators include tracking the average completion time, the number of attempts, the average time per attempt, and the longest attempt to monitor potential issues with task design.

Features of the method:

- definition of individualized laboratory exercises;
- ease of use, allowing instructors to easily improve or modify exercises with minimal programming skills;
- integration with Moodle, providing access to all exercises directly from the LMS environment without additional registration;
- automatic assessment with precise progress tracking and immediate feedback on errors.

Despite its advantages (task parameterization, automated generation, activity description language), SERA has limitations: it has a weak LMS connection (unidirectional for student lists), stores activities within the framework, rather than in the LMS, and provides no feedback integration with the LMS. It only supports exercises that produce file artifacts, whose structure and parameters are verified for correctness rather than representing a fully configurable learning environment.

2. APG. This study presents a toolset for automated laboratory environment configuration with a unique setup for each student. The system automatically generates personalized practical assignments, reducing academic dishonesty and unauthorized collaboration while increasing student engagement.

APG method (Automatic Problem Generation) [50] enables the creation of individualized student tasks and a set of software tools for deploying the lab environment. Python scripts parameterize virtual machine configuration files, allowing each student to run a unique virtual machine for their lab work on their local OS. A stand-alone server waits for individual student submissions, which are only available after completing tasks in their local VM. Thus, each submission is unique due to the environment parameterization.

Metrics for monitoring potential cheating include task dependencies – dividing tasks into stages and tracking the sequence of student submissions. Additionally, authors suggest monitoring downloads of correct answers from other students, as well as submission time/location (IP addresses) to detect potential collaboration. They note, however, that this criterion may be unreliable (e.g., students living together may submit around the same time honestly). Other limitations include the local nature of the environment (allowing reverse engineering) and the semi-automated assessment process requiring manual input, as well as a lack of LMS integration.

Methods used in the study:

- Automatic Problem Generation (APG) for creating unique task variants, preventing cheating;
- Instrumental modeling and software development, including Environment Generator (via Vagrant, Ansible, VirtualBox) and Submission Server (CTFd) modules;
- Case study with 207 students in an introductory security course, detecting 7 suspicious submissions;
- Collection of participant feedback indicating that students experienced no technical difficulties and instructors found the approach suitable for large groups;
- Implementation as an open-source IT solution to ensure replicability and transportability.

3. PAGE. The study [51] presents PAGE, a tool designed to simplify the creation and management of personalized digital assessments in the Canvas LMS. The system generates unique assignments for each student, reducing cheating and increasing engagement.

The authors apply a research-design methodology [64], where the primary value lies in creating an innovative artifact (the tool) and evaluating it in a real context – in this case, Canvas LMS. PAGE generates personalized tests with unique questions and answers tailored to each student.

The process includes:

1) generating a dataset of facts for each test to ensure different answers for the same question;

2) selecting questions and answers from the dataset to create a unique test version;

3) deploying the test in the LMS with randomized questions and answers.

Empirical evaluation in a real course (~100+ students, 7 Canvas tasks) confirmed scalability. Advantages of PAGE include LMS integration, facilitating use in DLEs and MOOCs. Limitations include the need for instructors' programming skills (JavaScript for question/answer generation), data storage demands, and increased LMS processing time under maximum personalization. To mitigate this, the system allows lowering personalization levels (1 variant per 2–10 students), which reduces infrastructure load but may slightly increase opportunities for academic dishonesty. PAGE does not perform true parameterized generation; tasks are randomly selected from a large pre-created pool.

4. MCTest. The focus of MCTest [52] is on implementing the Automated Assessment Method. It supports task parameterization and automated solution evaluation in programming courses. While multi-choice questions are straightforward, evaluating open-ended tasks is more complex. The study examines various task types, including essays (manually graded), programming exercises (automatically graded code), and iterative computational scenarios (loops and table tests). MCTest utilizes templates with parameters (wildcards) to generate variants using LaTeX, Python, and C++, and integrates these tools with Moodle and VPL. This design-based approach aims to prevent academic dishonesty through technological innovation. A key achievement is the integration with Moodle VPL, enabling in-browser execution of multiple programming languages. Parameterized Programming Exercises are issued to students who write and submit code for automated evaluation. A challenge is indicating the task version after code submission.

Advantages include immediate student feedback, eliminating the need to wait for instructor grading. The use of parameterized Programming Exercises improved average course grades compared to a control group (n = 30 each), although prior preparation or prior achievement levels were not reported. The method is applied as a case study in real courses.

LIMITATIONS OF EXISTING SOLUTIONS

To substantiate the need for improving existing methods and answering RQ3, the main limitations have been identified and formulated, along with potential directions for further research:

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

1) limited functionality of automatic generation – in many cases, systems rely on random selection of tasks from a pre-existing set rather than true automated generation [51];

2) insufficient support for practical task parametrization – not all solutions are suitable for parameterizing practical assignments, which restricts their applicability in real-world training scenarios [49], [51];

3) high entry barrier for instructors – in certain systems, educators must possess not only subject-matter expertise but also technical skills such as working with template engines, task generation scripts, and scripting languages [49], [51], [52];

4) integration challenges – many systems are difficult to integrate with local LMS and lack tools that effectively incorporate personalized assessment and additional resources, such as large datasets, into existing educational platforms [49], [50];

5) security vulnerabilities in local deployment – when learning environments are deployed locally, students may exploit reverse engineering to obtain correct answers without completing the tasks themselves [50];

6) incomplete automation of assessment – in some cases, instructors and students must manually input correct answers into the system for validation, and teachers are required to participate directly in the evaluation process [50]; comparison of methods for the automatic generation and assessment is shown in Table 1.

CONCLUSIONS

In the article, the authors investigate approaches and methods in the field of personalized learning in IT education. The analysis of contemporary scientific solutions in this area demonstrates significant scholarly and practical interest in the problems of automated generation of personalized practical tasks and adaptive learning.

Particular attention is given to methods for automated generation, evaluation, and deployment of virtual learning environments, in which personalized tasks are executed. Researchers worldwide are actively working on the development and refinement of these methods, which also represent a key area of interest in our scientific field. At the same time, the authors emphasize the importance of leveraging the capabilities offered by adaptive learning and AI, motivating our work to integrate these approaches into a unified information technology [65] that combines personalization methods with adaptive learning and AI tools. This integrated approach positively impacts not only student performance and engagement but also academic integrity and the overall quality of higher education.

The main limitations of existing methods for automated task generation and assessment are identified, and directions for potential further research are outlined. Despite the successful implementation of such methods in DLEs, challenges remain related to technical complexity, limited automation of assessment, incomplete LMS integration, and the risk of task compromise during local execution. These issues underscore the need to refine existing methods and develop flexible, scalable, and user-friendly solutions to support personalized [66] learning in DLEs.

Technological constraints are general and include online system security concerns, insufficient feedback for incorrect responses, and practical difficulties with questions sensitive to spelling or case errors. There are also significant times and resource limitations, as programming and content preparation for these platforms require substantial effort. Moreover, PAL may increase the workload

Table 1. **Comparison of methods for the automatic generation and assessment**

Study/Capabilities	individual learning environment	automated deployment	specification language	integration with LMS	gamification	scalability	simple templates
SERA	-	-	+	+/-	-	+	-
APG	+/-	+/-	-	-	+	+/-	+
PAGE	-	-	-	+	-	-	-
MCTest	-	-	-	+	-	-	+/-
GRADE	+	+	+	+	+	+	+

Source: **compiled by the authors**

for instructors and administrators, potentially necessitating higher levels of self-regulation among students. However, this issue can be mitigated through the automation of most processes.

Additional challenges include the complexity and cost of integrating such platforms into existing educational infrastructures, as well as the need for advanced expertise in AI and ML. Serious privacy and data security concerns arise due to the extensive collection and analysis of sensitive student data. The scalability and applicability of these systems require further investigation, particularly regarding the "cold start" problem (absence of initial data for new students). Finally, there is a risk of over-reliance on technology, which may overshadow essential human elements of teaching and learning.

Prospects for further research will focus on creating an integrated information technology that combines an intelligent automated task generation method based on generative AI with sophisticated deployment, assessment and monitoring capabilities.

REFERENCES

1. Zabasta, A., Kazymyr, V., Kunicina, N. & Velihorskyi, O. "Digital learning ecosystem: online concept for engineering education". *IEEE 65th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*. 2024. p. 1–7. DOI: https://doi.org/10.1109/RTUCON62997.2024.10830866.

2. "Report on the results of an express analysis of the current state of IT education". *Ministry of Digital Transformation of Ukraine*. 2020. – Available from: https://thedigital.gov.ua/storage/uploads/files/page/community/docs/Звіт_щодо_результатів_експрес_аналізу_поточного_стану_IT_освіти.pdf. – [Accessed: Aug, 2024].

3. Nahorniuk, O. & Levytska, L. "Integrating professional and educational standards for developing IT curriculum". *Universum*, 2024; (4): 160–168. – Available from: https://archive.liga.science/index.php/universum/article/view/699. – [Accessed: Aug, 2024].

4. Holdgraf, C., Culich, A., Rokem, A., Deniz, F., Alegro, M. & Ushizima, D. "Portable learning environments for hands-on computational instruction: Using container- and cloud-based technology to teach data science". *ArXiv*. 2017. DOI: https://doi.org/10.48550/arXiv.1703.04900.

5. Stevens, R., Silver, L., Richards, R. & Campbell, K. "A comparison of faculty and student perspectives of academic integrity in an online environment: A pilot study". *Journal of Business Administration*. 2022; 16 (2): 14–27. – Available from: https://www.atu.edu/business/jbao/fall2022/202202%2002%20Stevens%20Silver%20Richards%20Campbell.pdf. – [Accessed: Aug, 2024].

6. Valizadeh, M. "Cheating in online learning programs: learners' perceptions and solutions. Turkish". *Online Journal of Distance Education*. 2022; 23 (1): 195–209. DOI: https://doi.org/10.17718/tojde.1050394.

7. Naidu, K. & Sevnarayan, K. "ChatGPT: an ever-increasing encroachment of artificial intelligence in online assessment in distance education". *Online Journal of Communication and Media Technologies*. 2023; 13 (1). DOI: https://doi.org/10.30935/ojcmt/13291.

8. Malik, A. A., Hassan, M., Rizwan, M., Mushtaque, I., Lak, T. A. & Hussain, M. "Impact of academic cheating and perceived online learning effectiveness on academic performance during the COVID-19 pandemic among Pakistani students". *Frontiers in Psychology*. 2023; 14. DOI: https://doi.org/10.3389/fpsyg.2023.1124095.

9. "Recommendations for the implementation of blended learning in vocational pre-higher and higher education institutions". *Ministry of Education and Science of Ukraine*. – Available from: https://mon.gov.ua/static-objects/mon/sites/1/vishcha-osvita/2020/zmyshene%20navchanny/zmishanenavchannia-bookletspreads-2.pdf. – [Accessed: Aug, 2024].

10. Shemshack, A. & Spector, J. M. "A systematic literature review of personalized learning terms". *Smart Learn. Environ.* 2020; 7 (33). DOI: https://doi.org/10.1186/s40561-020-00140-9.

11. Maghsudi, S., Lan, A., Xu, J. & Schaar, M. "Personalized education in the Artificial Intelligence era: what to expect next". *IEEE Signal Processing Magazine*. 2021. DOI: https://doi.org/10.1109/MSP.2021.3055032.

12. Clément, B., Sauzéon, H., Roy, D. & Oudeyer, P.-Y. “Improved performances and motivation in intelligent tutoring systems: combining machine learning and learner choice”. *E-print ArXiv.* 2025. DOI: https://doi.org/10.48550/arXiv.2402.01669.

13. Peng, H., Ma, S. & Spector, J. “Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment”. *Foundations and Trends in Smart Learning*. 2019. p. 171–176. DOI: https://doi.org/10.1186/s40561-019-0089-y.

14. du Plooy, E., Casteleijn, D., Franzsen, D. “Personalized adaptive learning in higher education: A scoping review of key characteristics and impact on academic performance and engagement”. *Heliyon*. 2024; 10. DOI: https://doi.org/10.1016/j.heliyon.2024.e39630.

15. Song, C., Shin, S.-Y. & Shin, K.-S. “Implementing the dynamic feedback-driven learning optimization framework: a machine learning approach to personalize educational pathways”. *Appl. Sci*. 2024; 14: 916. DOI: https://doi.org/10.3390/app14020916.

16. Walkington, C. & Bernacki, M. L. “Appraising research on personalized learning: Definitions, theoretical alignment, advancements, and future directions”. *Journal of Research on Technology in Education*. 2020; 52 (3): 235–252. DOI: https://doi.org/10.1080/15391523.2020.1747757.

17. Nabizadeh, A. H., Leal, J. P., Rafsanjani, H. N. & Shah, R. R. “Learning path personalization and recommendation methods: A survey of the state-of-the-art”. *Expert Systems with Applications*. 2020; 159: 113596. DOI: https://doi.org/10.1016/j.eswa. 2020.113596.

18. Bloom, B. S. “Learning for mastery instruction and curriculum”. *Regional Educational Laboratory for the Carolinas and Virginia, DN. Topical Papers and Reprints, Number 1.* 1968; 1 (2). – Available from: http://www.researchforteachers.org.uk/sites/default/files/Docs/Bloom%20(1968)%20Learning%20for%20Mastery_0.pdf. – [Accessed: Aug, 2024].

19. Vasilev, I. “Bloom's master learning model: a case study example of application in the TVET”. *Knowledge International Journal*. 2024; 66.2. – Available from: https://www.researchgate.net/profile/Iliyan-Vasilev/publication/384761613_Bloom's_master_learning_model_a_case_study_example_of_application_in_the_tvet/links/6706732298b3cf6ba1573173/blooms-master-learning-model-a-case-study-example-of-application-in-the-tvet.pdf. – [Accessed: Aug, 2024].

20. Deci, E. & Ryan, R. “Self-determination theory: a macrotheory of human motivation, development, and health”. *Canadian Psychology / Psychologie canadienne*. 2008; 49: 182–185. DOI: https://doi.org/10.1037/a0012801.

21. Agustianto, K., Permanasari, A. E., Kusumawardani, S. S. & Hidayah, I. “Design adaptive learning system using metacognitive strategy path for learning in classroom and intelligent tutoring systems”. *AIP Conf. Proc.* 2016; 1755 (1): 070012. DOI: https://doi.org/10.1063/1.4958507.

22. Gibbons, A. S. “Model-centered instruction, the design and the designer”. In *Understanding Models for Learning and Instruction*. 2008. p. 161–173. DOI: https://doi.org/10.1007/978-0-387-76898-4_8.

23. Sannino, A. & Engeström, Y. “Cultural-historical activity theory: founding insights and new challenges”. *Cultural-Historical Psychology*. 2018; 14: 43–56. – Available from: https://www.researchgate.net/publication/328398841_Cultural-historical_activity_theory_Founding_insights_and_new_challenges/fulltext/5bca8d27458515f7d9cb973e/Cultural-historical-activity-theory-Founding-insights-and-new-challenges.pdf. – [Accessed: Aug, 2024].

24. Chen, K.-Z. & Tseng, J.-Y. & Oakley, B. “Transforming online teacher training through expansive learning: a case study applying cultural-historical activity theory and the change laboratory method”. *The Asia-Pacific Education Researcher.* 2024; 34; 1413–1430. DOI: https://doi.org/10.1007/s40299-024-00954-y.

25. Guadagnoli, M. A. & Lee, T. D. “Challenge point: a framework for conceptualizing the effects of various practice conditions in motor learning”. *Journal of motor behavior.* 2004; 36 (2): 212–224. DOI: https://doi.org/10.3200/jmbr.36.2.212-224.

26. Sayed, W. S., Noeman, A. M., Abdellatif, A., et al. “AI-based adaptive personalized content presentation and exercises navigation for an effective and engaging E-learning platform”. *Multimedia Tools and Applications.* 2023; 82: 3303–3333. DOI: https://doi.org/10.1007/s11042-022-13076-8.

27. Kolekar, S. V., Pai, R. M. & Pai M. M. “Rule based adaptive user interface for adaptive E-learning system”. *Education and Information Technologies.* 2019; 24. DOI: https://doi.org/10.1007/s10639-018-9788-1.

28. Pelánek, R., Effenberger, T. & Jarušek, P. “Personalized recommendations for learning activities in online environments: a modular rule-based approach”. *User Modeling and User-Adapted Interaction*. 2024; 34: 1399–1430. DOI: https://doi.org/10.1007/s11257-024-09396-z.

29. Zakaria, M., Moore, A., Ashman, H., Stewart, C. & Brailsford, T. “The Hybrid Model for Adaptive Educational Hypermedia”. *Adaptive Hypermedia and Adaptive Web-Based Systems, Conference Paper*. 2002. p. 580-585. DOI: https://doi.org/10.1007/3-540-47952-X_89.

30. Grubišić, A. , Stankov, S. & Žitko, B. “Stereotype student model for an adaptive e-learning system”. *International Journal of Computer and Information Engineering*. 2013; 7 (4): 440–447. – Available from: https://zenodo.org/records/1081417/files/13596.pdf. – [Accessed: Aug, 2024].

31. Al-Hmouz, A. “An adaptive framework to provide personalisation for mobile learners”. *University of Wollongong. Thesis*. 2012. – Available from: https://hdl.handle.net/10779/uow.27662952.v1. – [Accessed: Aug, 2024].

32. Hohl, H., Böcker, H. D. & Gunzenhäuser, R. “Hypadapter: an adaptive hypertext system for exploratory learning and programming”. *User Modeling and User-Adapted Interaction*. 1996; 6 (2): 131–156. DOI: https://doi.org/10.1007/BF00143965.

33. Corbett, A. T., & Anderson, J. R. “Knowledge tracing: modeling the acquisition of procedural knowledge”. *User Modeling and User-Adapted Interaction*. 1994; 4 (4): 253–278. DOI: https://doi.org/10.1007/BF01099821.

34. Millán, E. & Pérez-de-la-Cruz, J. L. “A Bayesian diagnostic algorithm for student modeling and its evaluation”. *User Modeling and User-Adapted Interaction*. 2002; 12*:* 281–330. DOI: https://doi.org/10.1023/A:1015027822614.

35. Johns, J., Mahadevan, S. & Woolf, B. “Estimating student proficiency using an item response theory model”. *Intelligent Tutoring Systems, Lecture Notes in Computer Science*. 2006; 4053: 473–480. DOI: https://doi.org/10.1007/11774303_47.

36. Karac, A. “Intelligent tutoring system model based on fuzzy logic and constraint-based student model”. *Neural Computing and Applications*. 2019; 31. DOI: https://doi.org/10.1007/s00521-017-3311-2.

37. Thai-Nghe, N., Drumond, L. & Horvath, T., Nanopoulos, A. & Schmidt-Thieme, L. “Matrix and tensor factorization for predicting student performance”. *CSEDU 2011 - Proceedings of the 3rd International Conference on Computer Supported Education.,* 2011; 1. – Available from: https://www.researchgate.net/profile/Tomas-Horvath-3/publication/221130749_Matrix_and_Tensor_Factorization_for_Predicting_Student_Performance/links/00b49538cb93031204000000/Matrix-and-Tensor-Factorization-for-Predicting-Student-Performance.pdf. – [Accessed: Aug, 2024].

38. Manouselis, N., Drachsler, H. & Verbert, K. & Duval, E. “Recommender systems for learning”:. *Springer*, 2012. – Available from: https://www.researchgate.net/profile/Hendrik-Drachsler/publication/234057357_Recommender_Systems_for_Learning/links/02bfe50ead1856fc87000000/Recommender-Systems-for-Learning.pdf. – [Accessed: Aug, 2024].

39. Salau, L., Hamada, M., Prasad, R., Hassan, M., Mahendran, A. & Watanobe, Y. “State-of-the-art survey on deep learning-based recommender systems for e-learning”. *Applied Sciences,* 2022; 12 (23): 11996. DOI: https://doi.org/10.3390/app122311996.

40. Li, J. & Ye, Z. “Course recommendations in online education based on collaborative filtering recommendation algorithm”. *Complexity.* 2020. p. 1–10. DOI: https://doi.org/10.1155/2020/6619249.

41. Segal, A., &al, K., Shani, G. & Shapira, B. “A difficulty ranking approach to personalization in e-learning”. *International Journal of Human-Computer Studies*. 2019; 130. DOI: https://doi.org/10.1016/j.ijhcs.2019.07.002.

42. Srivastava, S., Tripathi, A. & Arora, N. “Multi-criteria decision making (MCDM) in diverse domains of education: a comprehensive bibliometric analysis for research directions”. *International Journal of System Assurance Engineering and Management.* 2024. DOI: https://doi.org/10.1007/s13198-024-02332-9.

43. Syed Hassan, S. A. H., Tan, S. C. & Yusof, K. M. “MCDM for Engineering Education: Literature Review and Research Issues”. In: *Engineering Education for a Smart Society*. *GEDC WEEF 2016 2016. Advances in Intelligent Systems and Computing, Springer, Cham*. 2018; 627. DOI: https://doi.org/10.1007/978-3-319-60937-9_16.

44. Bellman, R. “Dynamic Programming”. *Princeton University Press, Princeton*. 1957. – Available from: https://gwern.net/doc/statistics/decision/1957-bellman-dynamicprogramming.pdf. – [Accessed: Aug, 2025].

45. Onwuegbuchunam, D., Amuji, H., Okoroji, L., Nwachi, C. & Mbachu, J. “Development/application of dynamic programming model for students' academic planning and performance in the universities”. *Journal of Science Education and Technology*. 2023; 12: 20–25. DOI: https://doi.org/10.5281/zenodo.7765105.

46. Lou, H., Yue, P. & Chen, J. “Research on adaptive education path dynamic programming algorithm based on reinforcement learning and cognitive graphs”. *IEEE Access*. 2025; 13: 105446–105462. DOI: https://doi.org/10.1109/ACCESS.2025.3578389.

47. Tan, C., Han, R., Ye, R. & Chen, K. “Adaptive learning recommendation strategy based on deep q-learning”. *Appl Psychol Meas*. 2019; 44 (4): 251–266. DOI: https://doi.org/10.1177/0146621619858674.

48. Song, X., Chen, K., Bi, Z., et al. “Mastering reinforcement learning: foundations, algorithms, and real-world applications”. *Preprint*. – Available from: https://www.researchgate.net/profile/Junhao-Song-3/publication/391499126_Mastering_Reinforcement_Learning_Foundations_Algorithms_and_Real-World_Applications/links/6823a7dbdf0e3f544f54459f/Mastering-Reinforcement-Learning-Foundations-Algorithms-and-Real-World-Applications.pdf. – [Accessed: Aug, 2024].

49. Agudo, I., Rios, R., & Nieto, A. “Personalized computer security tasks with automatic evaluation and feedback”. *Proceedings of the 2019 AIS SIGED International Conference on Information Systems Education and Research*. 2019. – Available from: https://aisel.aisnet.org/siged2019/1. – [Accessed: Aug, 2024].

50. Vykopal, J., Švábenský, V., Seda, P. & Čeleda, P. “Preventing cheating in hands-on lab assignments”. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1 (SIGCSE 2022)*. 2022; 1: 78–84. DOI: https://doi.org/10.1145/3478431.3499420.

51. Chan, J. & Teng, S. “The Design and Implementation of PAGE: Personalised Assessment Generative Engine”. *Pacific Journal of Technology Enhanced Learning*. 2024; 6 (2): 33–46. DOI: https://doi.org/10.24135/pjtel.v6i2.203.

52. Zampirolli, F. de A., Pisani, P. H., Josko, J. M., et al. “Parameterized and automated assessment on an introductory programming course”. In: *Simposio Brasileiro de Informatica na Educacao*. 2020; 31: 1573–1582. DOI: https://doi.org/10.5753/cbie.sbie.2020.1573.

53. Azahar, M., & Ghauth, K. “A hybrid automated essay scoring using NLP and random forest regression”. *Proceedings of the International Conference on Computer, Information Technology and Intelligent Computing*. 2022. p. 448–457. DOI: https://doi.org/10.2991/978-94-6463-094-7_35.

54. Combéfis, S. “Automated code assessment for education: review, classification and perspectives on techniques and tools”. *Software*. 2022; 1 (1): 3–30. DOI: https://doi.org/10.3390/software1010002.

55. Ersozlu, Z., Taheri, S. & Koch, I. “A review of machine learning methods used for educational data”. *Educ Inf Technol*. 2024; 29: 22125–22145. DOI: https://doi.org/10.1007/s10639-024-12704-0.

56. Zhang, J. & Fan, Y. “Machine learning based big data analytics for education in curriculum reforms”. *Applied Mathematics and Nonlinear Sciences*. 2025; 10. DOI: https://doi.org/10.2478/amns-2025-0135.

57. Tang, X., Chen, Y., Li, X., Liu, J. & Ying, Z. “A reinforcement learning approach to personalized learning recommendation systems”. *British Journal of Mathematical and Statistical Psychology*, 2019; 72: 108–135. DOI: https://doi.org/10.1111/bmsp.12144.

58. Das Adhikary, P. & Metsämuuronen, J. “Knowledge tracing: big picture, timeline, and categories in digital learning”. *Preprint*. 2025. DOI: https://doi.org/10.13140/RG.2.2.17277.27364.

59. Govea, J., Navarro, A., Sánchez-Viteri, S. & Villegas, W. “Implementation of deep reinforcement learning models for emotion detection and personalization of learning in hybrid educational environments”. *Frontiers in artificial intelligence*. 2024; 7. DOI: https://doi.org/10.3389/frai.2024.1458230.

60. McFarland, D. A., Khanna, S., Domingue, B. W. & Pardos, Z. A. “Education data science: past, present, future”. *AERA Open*. 2021; 7. DOI: https://doi.org/10.1177/23328584211052055.

61. Le Quy, T., Friege, G. & Ntoutsi, E. "A review of clustering models in educational data science toward fairness-aware learning". In *Educational Data Science: Essentials, Approaches, and Tendencies. Big Data Management. Springer, Singapore*. 2023. DOI: https://doi.org/10.1007/978-981-99-0026-8_2.

62. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. & Yu, P. S. "A comprehensive survey on graph neural networks". In *IEEE Transactions on Neural Networks and Learning Systems*. 2021; 32 (1): 4–24. DOI: https://doi.org/10.1109/TNNLS.2020.2978386.

63. "Graph neural networks for knowledge tracing". *Medium Official Site*. 2023. – Available from: https://medium.com/stanford-cs224w/graph-neural-networks-for-knowledge-tracing-ef31fdaa5f00. – [Accessed: Aug, 2025].

64. Peffers, K., Tuunanen, T., Rothenberger, M. A. & Chatterjee, S. "A design science research methodology for information systems research". *Journal of Management Information Systems*. 2007; 24 (3): 45–77. DOI: https://doi.org/10.2753/MIS0742-1222240302.

65. Khyzhniak, A. V. & Prila, O. A. "Designing a system for automated generation and automated assessment of parameterized practical assignments". *Technical Sciences and Technologies*. 2025; 2 (40): 221–233. DOI: https://doi.org/10.25140/2411-5363-2025-2(40)-221-233.

66. Khyzhniak, A. V. & Kazymyr, V. V. "A generalized classification of personalization levels in practical assignments for IT-education". *Nauka i Tehnica Syogodni*. 2025; 7 (48): 1932–1949. DOI: https://doi.org/10.52058/2786-6025-2025-7(48)-1932-1949.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 14.07.2025
Received after revision 17.09.2025
Accepted 25.09.2025

DOI: https://doi.org/10.15276/hait.08.2025.24
УДК 37.091.3:004.9:378.018.43

Систематичний огляд методів підтримки персоналізації в ІТ-освіті

Хижняк Андрій Васильович[1)]
ORCID: https://orcid.org/0009-0008-8655-3736; alf.andrey@stu.cn.ua
Казимир Володимимр Вікторович[1)]
ORCID: https://orcid.org/0000-0001-8163-1119; vvkazymyr@stu.cn.ua. Scopus Author ID: 56644727300
[1)] Національний університет "Чернігівська Політехніка", вул. Шевченка, 95. Чернігів, 14030, Україна

АНОТАЦІЯ

Зростаюча складність цифрових навчальних середовищ та зростаючий попит на індивідуалізовані освітні програми підкреслюють важливість розробки ефективних методів персоналізації практичних завдань. Метою цього дослідження є аналіз існуючих підходів до персоналізації практичних завдань в ІТ-освіті та визначення їхніх переваг, обмежень і потенціалу для інтеграції в цифрові навчальні екосистеми. Основні завдання включають визначення технологічних рішень, що забезпечують автоматичне створення та оцінювання персоналізованих практичних завдань, а також їхню ефективність для підтримки адаптивних і масштабованих навчальних процесів. Методологія дослідження базується на порівняльному аналізі наукових джерел, існуючих освітніх платформ та практичних реалізацій персоналізованих завдань. Особлива увага приділяється розробці концептуальних рамок, що забезпечують персоналізацію не тільки на рівні складності завдань, але й з точки зору варіативності змісту, шляхів навчання та механізмів оцінювання. Огляд визначає найефективніші стратегії для досягнення масштабованої персоналізації в цифровій освіті. Аналіз також виявляє обмеження існуючих рішень. Критичними обмеженнями є: недостатня підтримка комплексної параметризації практичних завдань, високі технічні бар'єри для викладачів, неповна інтеграція LMS, вразливість безпеки в сценаріях локального розгортання та обмежена автоматизація процесів оцінювання. Ці обмеження підкреслюють необхідність більш досконалих, зручних для користувача рішень. Результати дослідження становлять основу для розробки інтегрованих інформаційних технологій, що поєднують методи персоналізації з адаптивним навчанням та інструментами ШІ, підтримуючи еволюцію цифрових навчальних екосистем у напрямку більш ефективних, масштабованих та орієнтованих на цілісність підходів до освіти. Наукова новизна дослідження полягає в систематичному порівнянні методів персоналізації та визначенні критеріїв їх ефективності в ІТ-освіті. Практичне значення визначається можливістю інтеграції запропонованих підходів у цифрові платформи, що дозволить поліпшити якість навчання, оптимізувати навчальний процес та забезпечити гнучкість, необхідну сучасним освітнім екосистемам.

Ключові слова: методи; аналіз; персоналізовані завдання; автоматичне генерування; автоматична оцінка; академічна доброчесність; параметризоване завдання

ABOUT THE AUTHORS

Andrii V. Khyzhniak - PhD student, Senior Lecturer, Department of Information and Computer Systems. Chernihiv Polytechnic National University, 95, Shevchenko Str, Chernihiv, 14030, Ukraine
ORCID: https://orcid.org/0009-0008-8655-3736; alf.andrey@stu.cn.ua
***Research field*:** Computer science; digital learning ecosystem developing; AI and ML methods; personalized adaptive learning

Хижняк Андрій Васильович - аспірант, старший викладач кафедри Інформаційних та комп’ютерних систем. Національний університет “Чернігівська Політехніка”, вул. Шевченка, 95, Чернігів, 14030, Україна

Volodymyr V. Kazymyr - Doctor of Engineering Sciences, professor, Department of Information and Computer Systems. Chernihiv Polytechnic National University, 95, Shevchenko Str. Chernihiv, 14030, Ukraine
ORCID: https://orcid.org/0000-0001-8163-1119; vvkazymyr@stu.cn.ua. Scopus Author ID: 56644727300
***Research field*:** Computer science; digital learning ecosystem developing; AI and ML methods

Казимир Володимир Вікторович - доктор технічних наук, професор кафедри Інформаційних та комп’ютерних систем. Національний університет “Чернігівська Політехніка”, вул. Шевченка, 95, Чернігів, 14030, Україна