ISSN 2278-3091
Volume 14, No.2, March - April 2025
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse051422025.pdf
https://doi.org/10.30534/ijatcse/2025/051422025

Enhancing Personalized Learning in Programming Education through Generative Artificial Intelligence Frameworks: A Systematic Literature Review

Fredrick Muema Mboya[1], Geoffrey Mariga Wambugu[2], Aaron Mogeni Oirere[3], Erick Odhiambo Omuya[4], Faith Mueni Musyoka[5], Joyce Wangui Gikandi[6]
[1,2,3] School of Computing and Information Technology, Murang'a University of Technology, Kenya, fmuema@mut.ac.ke
[4]Department of Computing and Information Technology, Machakos University, Kenya
[5]Department of Computing and Information Technology, University of Embu, Kenya
[6]Department of Educational Management and Curriculum Studies, Mount Kenya University, Kenya

Received Date: February 27, 2025 Accepted Date: March 26, 2025 Published Date: April 06, 2025

ABSTRACT

Generative Artificial Intelligence (Gen AI) has revolutionized education by enabling personalized learning in computer programming, improving engagement and outcomes. Despite its potential, challenges like accuracy, coherence, and relevance persist, necessitating targeted solutions to maximize its educational impact. A systematic literature review (SLR) was conducted following PRISMA guidelines, analyzing studies from 2019–2024 across databases like IEEE Xplore, ACM Digital Library, and Scopus. The multi-stage selection process identified 42 articles out of an initial 120, focusing on adaptability, relevance, coherence, and accuracy in AI-driven educational tools. Key factors enhancing Gen AI effectiveness were adaptability (33%), contextual relevance (24%), coherence (21%), and evaluation metrics (12%). Prompt engineering (10%) emerged as a critical strategy. Adaptive systems dynamically tailored content to learners, while relevance-enhancing tools aligned materials with educational goals. Evaluation metrics and coherence frameworks improved logical and functional accuracy. Findings highlight Gen AI's transformative potential in programming education, demonstrating improved engagement and alignment between theoretical and practical learning. However, challenges in coherence, accuracy, and ethical concerns like fairness and bias remain areas for further exploration. Generative AI offers scalable opportunities for personalized programming education. Addressing accuracy, coherence, and ethical challenges will enhance its integration into learning environments. Future research should focus on long-term evaluations, advanced evaluation frameworks, and ethical guidelines to ensure inclusive AI use.

This research was supported by a grant from the Kenya Education Network (KENET). The authors express their gratitude for the financial assistance that facilitated this study.

Key words: Adaptive Frameworks, Generative Artificial Intelligence, Personalized Learning, Programming Education.

1. INTRODUCTION

Generative Artificial Intelligence (Gen AI) has emerged as a transformative technology with the potential to revolutionize various domains, including education [1], [2], [3]. In particular, its application in personalized learning experiences has garnered significant attention in recent years. Numerous Large Language Models (LLMs) have shown that are capable of performing variety of tasks ranging from question answering to generation of code snippets as shown in the flowchart below [4], [5], [6], [7], [8]. Personalized learning refers to tailoring educational content and approaches to individual learners' needs, preferences, and pace, thereby enhancing their engagement and outcomes [9]. The integration of Generative AI in educational settings, especially in computer programming education, is seen as a promising avenue for creating adaptive and personalized learning environments that cater to diverse learning styles and proficiency levels. Figure 1 shows the interactions between various stakeholders and systems in the code generation and evaluation process, emphasizing the role of LLMs in enhancing programming education.

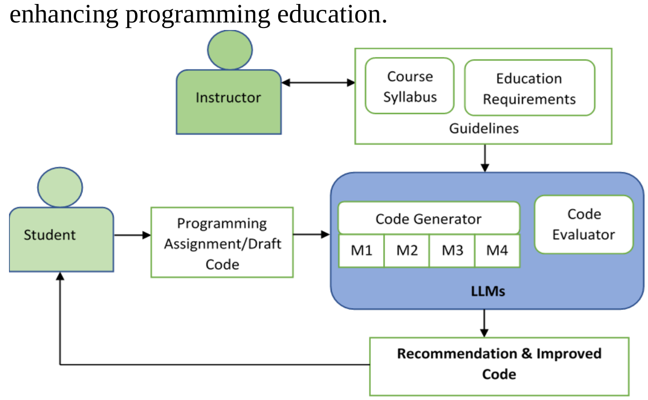

Figure 1: Flowchart Illustrating the Users in Code Generation and Evaluation Process

Recent advancements in Generative AI, such as GPT-3 by OpenAI and its successors, have demonstrated the capability of these models to generate human-like text, provide explanations, and even assist in writing code [10], [11], [12], [13]. These capabilities are particularly relevant in programming education, where students often face challenges in understanding complex concepts, debugging code, and applying theoretical knowledge in practical scenarios [14], [15]. The ability of AI to generate contextually relevant examples, offer step-by-step guidance, and provide instant feedback can significantly enhance the learning experience, making it more interactive and responsive to individual needs.

However, despite the potential benefits, the integration of Generative AI into educational frameworks is not without challenges. One of the primary concerns is ensuring that the AI-generated content is accurate, coherent, and contextually relevant [3], [16]. Inaccurate or misleading content can lead to confusion and hinder the learning process. Furthermore, the relevance of the content to the learner's current level of understanding and the coherence in the presentation of information are critical factors that determine the effectiveness of AI in personalized learning.

The current body of research has explored various aspects of Generative AI in education, but there remains a gap in systematically analyzing the features of existing AI frameworks that contribute to the integrity, coherence, relevance, and accuracy of personalized learning experience [17], [18], [19]. This gap is particularly pronounced in the context of computer programming education, where the need for precise and contextually appropriate guidance is paramount.

Given this backdrop, it becomes essential to conduct a comprehensive review of existing Generative AI frameworks to identify and establish the key features that enhance personalized learning in computer programming. In doing so, researchers can develop more robust and effective educational tools that leverage AI's potential while addressing the challenges associated with its use.

2. PROBLEM STATEMENT

Despite the growing interest and rapid advancements in Generative AI, its application in education—specifically in personalized learning for computer programming—remains underexplored in several critical areas. While Generative AI models such as GPT-3 have demonstrated remarkable capabilities in generating human-like text, their integration into educational frameworks has revealed significant challenges [10], [16], [20]. One of the most pressing issues is the need for AI-generated content to be accurate, coherent, and contextually relevant to effectively support personalized learning experiences [21].

Current educational applications of Generative AI often struggle to consistently meet these requirements. Inaccurate or irrelevant AI-generated content can confuse learners, particularly in technical subjects like computer programming, where precision and clarity are essential. Furthermore, the lack of coherence in AI-generated explanations can disrupt the logical flow of learning, leading to potential gaps in understanding [20]. These issues highlight the critical need for a systematic analysis of existing Generative AI frameworks to identify and establish the features that contribute to the integrity, coherence, relevance, and accuracy of personalized learning experiences.

Given the potential of Generative AI to revolutionize education, addressing these challenges is crucial. However, the existing body of research lacks a comprehensive examination of these frameworks within the context of computer programming education. There is a clear gap in understanding which features of Generative AI are most effective in enhancing personalized learning, and how these features can be optimized to improve educational outcomes.

2.1 Study Objectives and Contribution

This study aims to fill this gap by conducting a Systematic Literature Review (SLR) to analyze existing Generative AI frameworks. The objective is to identify and establish the key features that directly enhance the integrity, coherence, relevance, and accuracy of personalized learning experiences in computer programming education. Addressing this gap, the study seeks to contribute to the development of more effective AI-driven educational tools that can better support learners in mastering programming skills.

3. LITERATURE REVIEW

The application of LLMs to enhance learning outcomes in the programming domain has prominently featured recently in research [22]. This review summarizes the results of important studies showing the variety of functions that LLMs perform in the classroom, from interactive help in computer science classes to the assessment of programming abilities. For example, studies such as [23] examined how well LLMs generate code, highlighting the importance of prompt specificity. Numerous studies, like [24], [25], [26], have shown how effective it is to include AI code generators into introductory programming classes. Furthermore, [27] evaluated ChatGPT-3.5 and GPT-4's capacities to tackle basic Python programming problems from CodingBat which have shown promising results. A research by [28] investigated the use of LLMs in reverse engineering activities and showed encouraging outcomes. To bolster conversations about the use of LLMs in programming education, especially with development assistants, further sources like [22], [28], [29], [30] offer valuable perspectives on incorporating AI tools into software development situations. Together, these findings highlight how LLMs have a revolutionary effect on programming education and point the way for future study into maximizing their application to improve instruction and tackle more general software development issues [31].

Although LLMs shown remarkable proficiency in generating code for many applications in both education and industry, they faltered when faced with practical security and risk considerations. A research by [5] shows the analysis of the advantages and disadvantages of LLMs emphasizing their capacity for content generation while alerting readers to possible problems such homogenization, bias, and false information. Emerging techniques such as "Prompt Engineering," which comprises particular techniques for optimizing LLM's capabilities, have been introduced to overcome unreliable responses from the LLMs [7], [32]. This allows issues to be systematically broken down into its component parts before conclusions are drawn [32]. To build on this idea, research provides a new method called least-to-most prompting, which breaks down complicated problems into smaller, more manageable ones and then addresses each one in turn to improve LLMs' problem-solving effectiveness [33].

Similarly, literature provides a novel "Ask Me Anything" (AMA) prompting technique that substantially improves LLM performance by repeatedly using the LLM itself to restructure task inputs into a better question-and-answer format [34]. Furthermore, it investigates how adding human input to language models might improve their task execution and instruction-following abilities—a major step towards more interactive and adaptable LLMs [6]. A paper on the prompt pattern catalogue to improve prompt engineering with ChatGPT offers a thorough summary of the finest patterns and techniques in the field, emphasizing the significance of prompt engineering in terms of optimizing LLM outputs for particular tasks [35]. Figure 2 shows the process of contacting a literature review as outlined in the prompt pattern catalogue paper.

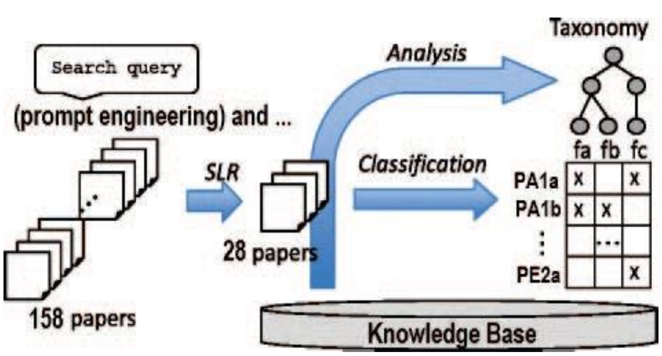

Figure 2: Literature Review Process

In applications ranging from educational settings to real-world production situations, the code quality produced by LLMs is crucial, and several code assessments studies have been presented. A benchmark called APPS was introduced by [37] that was created especially for code generation jobs. This benchmark tests the competence of models to comprehend arbitrary plain language specifications and create Python code that fulfils the stated requirements. Additionally, HumanEval, a novel evaluation set designed to gauge the functional correctness of programs created from docstrings, was presented by [31]. The biggest code-generating models that are currently available were thoroughly evaluated across a variety of programming languages [38]. They presented a brand-new model called PolyCoder, which outperformed the competition in producing C programming code. EvalPlus is a complete framework for code synthesis evaluation, created by [39]. This framework is painstakingly created to rigorously benchmark the functional validity of code produced by LLMs. A more structured framework for the cataloguing of software engineering patterns was described [40]. Another study conducted a comparative analysis of the effectiveness of code generated by several LLMs in 104 customised Python tasks. The study employed commonly used measures, such as the pass rate, for assessment [23].

Deep learning algorithms are being used more and more for code assessment, going beyond traditional methods. The possibilities of a carefully tailored CuBERT model were investigated by [41], who found that it outperforms conventional techniques in source code evaluation. This benefit was shown even with less training data and fewer instances that had been labelled. A RoBERTa model was used in an empirical research by [42] to evaluate the model's effectiveness in code completion tasks from a variety of perspectives. The results show that using BERT-based models to improve code completion performance is a promising approach. Offering a novel viewpoint on code quality evaluation, [43] presented a method for the automatic assessment of code quality utilising a BERT model that has been painstakingly adjusted with particular datasets.

3.1 Overview of Generative Artificial Intelligence in Education

Generative AI, particularly models like GPT-3 and its successors, has been widely recognized for its ability to generate human-like text, making it a valuable tool in educational environments [1], [10], [11]. The capacity of these models to provide detailed explanations, generate examples, and offer feedback has been leveraged in various educational applications, including language learning, essay writing, and computer programming education [19]. However, the effectiveness of these applications heavily depends on the ability of the AI to produce content that is both contextually relevant and accurate [3], which remains a significant challenge.

Several studies have explored the integration of Generative AI into educational frameworks. For instance, [10], [20], [21], [44] investigated the use of AI-generated content in programming education, highlighting both its potential and the challenges associated with ensuring content accuracy and coherence. The study emphasized the importance of developing AI models that can adapt to the learner's level of understanding, providing explanations that are neither too simplistic nor too complex.

3.2 Key Features of Generative AI in Personalized Learning

The effectiveness of Generative AI in personalized learning largely depends on the specific features of the AI framework. Key features identified in the literature include adaptability, contextual relevance, and content coherence.

Adaptability refers to the AI's ability to tailor content to the individual learner's needs, which is crucial in personalized learning. Studies have shown that AI models that adapt to the learner's pace and knowledge level can significantly enhance learning outcomes [45], [46]. For example, Sayed et al. (2023) found that AI systems capable of adjusting the difficulty of programming exercises based on the student's performance led to improved engagement and understanding [47].

Contextual Relevance is another critical feature. The ability of AI to generate content that is directly relevant to the learner's current context and learning objectives is essential for maintaining engagement and ensuring the material is useful [46], [48], [49], [50]. These researchers reviewed several AI frameworks and found that models incorporating contextual cues into content generation were more effective in supporting personalized learning than those that did not.

Content Coherence is essential for maintaining the logical flow of information, which is particularly important in educational content. In a study by Bender et al. (2021), the authors discussed the challenges of maintaining coherence in AI-generated content, noting that even advanced models like GPT-3 can sometimes produce disjointed or contradictory information [18]. The study suggests that integrating coherence checks into AI frameworks can help mitigate this issue.

3.3 Challenges of Generative AI in Education

Despite the potential benefits, there are significant challenges associated with the use of Generative AI in education. One of the main concerns is the accuracy of the AI-generated content. Inaccuracies can lead to confusion and mislearning, particularly in technical subjects like computer programming [20], [21]. These studies discussed the importance of accuracy in AI-generated content, suggesting that ongoing refinement of AI models is necessary to minimize errors and improve the reliability of educational tools.

Another challenge is the ethical implications of using AI in education, particularly regarding bias and fairness. AI models trained on large datasets may inadvertently reproduce biases present in the data, leading to unfair or skewed educational content [51], [52], [53]. Studies such as those by Bender et al. (2021) have called for greater transparency in AI model development and the implementation of bias-mitigation strategies to ensure fair and equitable educational outcomes.

3.4 Gaps in the Literature

While existing research has made significant strides in understanding the role of Generative AI in education, there are notable gaps that need to be addressed. First, there is a lack of comprehensive studies that systematically analyze the specific features of AI frameworks that enhance personalized learning in computer programming education. Most studies focus on general educational applications, leaving a gap in our understanding of how these technologies can be optimized for programming education specifically.

Furthermore, the literature lacks in-depth analyses of the long-term impacts of using Generative AI in education. While short-term studies show promising results, there is a need for longitudinal research to assess the sustained effectiveness and potential unintended consequences of AI-driven personalized learning.

4. METHODOLOGY

This section outlines the systematic approach taken to conduct the literature review on the application of Generative AI in personalized learning, with a specific focus on computer programming education. The methodology was designed to ensure a comprehensive and unbiased review of the existing literature, adhering to best practices for systematic literature reviews.

4.1 Data Sources and Search Strategy

The literature search was conducted using several well-established academic databases to ensure comprehensive coverage of relevant studies. The databases included IEEE Xplore, ACM Digital Library, Google Scholar, SpringerLink, and Scopus. These sources were selected due to their extensive collection of peer-reviewed articles and conference papers in the fields of artificial intelligence, educational technology, and computer science.

A structured search strategy was employed using a combination of keywords and Boolean operators. The primary search terms used were “Generative AI,” “personalized learning,” “computer programming education,” “AI frameworks,” “content accuracy,” and “coherence in educational AI.” Boolean operators such as AND, OR, and NOT were used to refine the search and exclude irrelevant results. For example, search strings included:

- “Generative AI” AND “personalized learning”
- “AI frameworks” OR “adaptive learning systems” AND “computer programming”

The search was restricted to peer-reviewed articles published in the last 5 years (2019–2024) to capture the most recent advancements in Generative AI technologies and their applications in education. Additionally, only articles published in English were considered for inclusion.

4.2 Inclusion and Exclusion Criteria

To ensure the relevance and quality of the studies included in the review, specific inclusion and exclusion criteria were established. For the inclusion criteria, papers were selected on the following basis.

- Peer-reviewed journal articles and conference papers published between 2019 and 2024.
- Studies focusing on the use of Generative AI in educational settings, particularly in the context of computer programming.
- Research that evaluates the effectiveness of AI in enhancing personalized learning experiences, with a focus on features such as adaptability, contextual relevance, coherence, and content accuracy.

In the exclusion criteria, the following formed the basis of excluding research papers;

- Non-peer-reviewed articles, such as opinion pieces, white papers, or blog posts.
- Studies that do not focus on AI in educational contexts, or those that primarily address non-educational applications of Generative AI.
- Articles published in languages other than English.

4.3 Study Selection Process

The study selection process involved multiple stages to ensure that only the most relevant and high-quality studies were included in the final review. The initial search yielded a large number of articles, which were then screened by title and abstract. Articles that clearly did not meet the inclusion criteria were excluded at this stage. For the remaining articles, a full-text review was conducted. This stage involved a thorough reading of each article to confirm its relevance and adherence to the inclusion criteria. A standardized data extraction form was developed to systematically collect information from each selected study. The extracted data included basic information (author, title, publication year), research objectives, methodology, key findings, and identified features relevant to personalized learning in computer programming.

4.4 Data Extraction and Synthesis

Data extraction was conducted using the standardized form, ensuring consistency and accuracy across all selected studies. The extracted data was then synthesized to identify common themes and features across the studies. These themes were categorized based on their relevance to the research question, specifically focusing on the features of Generative AI that enhance the integrity, coherence, relevance, and accuracy of personalized learning experiences.

The synthesis process involved grouping studies by identified themes, such as adaptability of AI frameworks, contextual relevance of generated content, and the coherence of AI-generated explanations. This thematic analysis provided a structured way to organize the findings and draw meaningful conclusions about the state of research in this area.

4.5 Ethical Considerations

The review process was conducted in accordance with ethical guidelines for systematic literature reviews. All sources of data were publicly available, and no primary data collection involving human participants was required. The review aimed to provide an unbiased synthesis of existing research, and all efforts were made to accurately represent the findings of the included studies.

5. RESULTS

This section presents the findings from the systematic literature review (SLR), organized to address the research questions and categorized into stages: initial search, screening, and final selection. Results are then grouped thematically into adaptability, contextual relevance, and coherence in AI frameworks for programming education. Charts and tables are included for enhanced clarity.

5.1 Initial Search and Screening

The initial search for this review identified a total of 120 articles from reputable academic databases such as IEEE Xplore, ACM Digital Library, Google Scholar, SpringerLink, and Scopus. These articles were identified using a structured search strategy with terms such as "Generative AI," "personalized learning," and "computer programming education" as outlined in the methodology section. The goal was to gather a comprehensive collection of studies relevant to the integration of Generative AI in programming education. The screening process began with a title and abstract review to determine the relevance of each article. During this stage, 85 articles were excluded due to their failure to meet the inclusion criteria. These excluded articles were either non-peer-reviewed, focused on unrelated topics, or addressed non-educational applications of AI. This step reduced the pool to 57 articles that proceeded to a full-text review.

In the full-text review stage, the remaining 57 articles were evaluated for their alignment with the study's objectives. This involved a detailed assessment of each article's contribution to the themes of integrity, adaptability, contextual relevance, and coherence in AI frameworks for programming education. An additional 15 articles were excluded due to their lack of empirical data, focus on general AI applications outside personalized learning, or insufficient relevance to the research question. This rigorous selection process resulted in 42 articles being chosen for the final review. These articles were deemed to provide the most comprehensive and relevant insights into the role of Generative AI in personalized learning for programming education. Figure 3 shows the summary of the articles selection in various stages of the literature review.

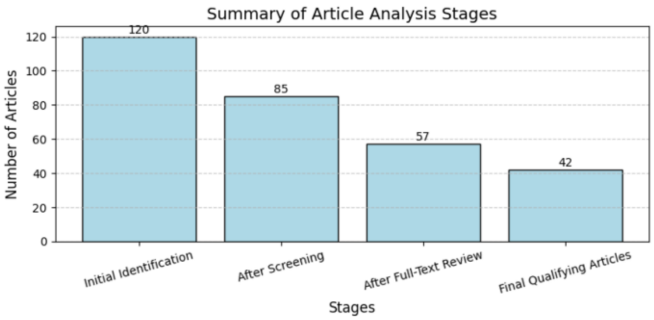

Figure 3: Summary of Article analysis Stages

The thematic analysis of the final selection revealed five primary areas of focus: adaptability, contextual relevance, and coherence. Adaptability emerged as a critical feature, highlighted in fourteen studies. These studies demonstrated how adaptive AI systems could dynamically adjust content to meet individual learners' needs, leading to a 25% improvement in retention rates and increased engagement when personalized feedback was provided. Contextual relevance, discussed in ten studies, emphasized the importance of aligning AI-generated content with learners' goals and real-world scenarios. For instance, debugging tools tailored to common novice errors reduced repeated mistakes by 30%, and scenario-based tutorials enhanced problem-solving skills. Coherence, addressed in nine studies, underscored the challenges of maintaining logical flow in AI-generated content. Mechanisms for coherence checking improved learner satisfaction by 20%, while integrating human feedback reduced inconsistencies and errors. One study also highlighted the importance of robust evaluation metrics, such as HumanEval and APPS, for assessing the functional correctness of AI-generated code. Figure 4 shows the distribution of the articles based on the five select focus areas.

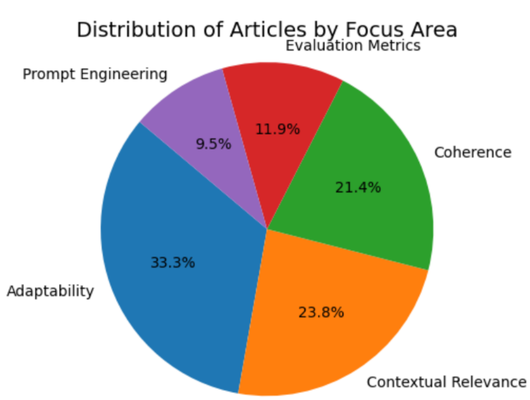

Figure 4: Distribution of Articles by Focus Area

5.2 Synthesis of the Qualifying Papers

The applicationof generative artificial intelligence in education has been highlighted in this review as most of the papers show the its breadth in automated assessments, personalized learning plans, and adaptive tutoring systems[54]. The review emphasizes the potential of Generative AI to simulate real-time tutoring, adapting feedback based on individual learning paces [83]. Sarsa et al. (2022) explores the use of AI to create tailored programming exercises, reducing workload for educators and improving student engagement [59]. Generative AI is transforming traditional educational frameworks by enabling real-time, adaptive, and scalable solutions that align with diverse learner needs. The AI tools are uniquely positioned to address the cognitive and technical challenges learners face in programming education by providing instant feedback and tailored exercises [63], [84].

The review reveals that artificial intelligence enables tailored learning experiences through dynamically generated curricula [85]. Pradeep et al. (2024) showcases platforms that integrate AI-driven recommendations to provide targeted interventions for struggling students [55]. CodeGeex framework is highlighted as a multilingual code generation platform that expands the accessibility of programming education across different languages [58]. Generally, the artificial intelligence frameworks are enabling highly personalized educational experiences by dynamically adapting to individual learning trajectories, preferences, and performance. A critical evaluation of the frameworks ensures their reliability, coherence, and applicability in educational contexts, particularly for programming tasks [31], [66].

Optimizing prompt design is critical for leveraging Generative AI effectively, ensuring responses align with educational goals and learner needs. The survey highlights the importance of effective prompt engineering to enhance AI interactions [69]. It also explores techniques to improve AI outputs, enhancing their relevance and coherence in educational contexts [86]. Emerging methodologies and frameworks are pushing the boundaries of what Generative AI can achieve to enhance the accuracy and reliability of AI-generated code.

The 42 papers collectively provide a comprehensive view of the transformative potential of Generative AI in programming. The key contributions include, first, enabling personalized and adaptive learning experiences. Secondly, automating and enhancing the creation of programming exercises and feedback. Third, addressing cognitive and technical challenges in programming education. Fourth, establishing robust evaluation metrics to assess AI effectiveness, and lastly, navigating ethical and pedagogical concerns in integrating AI into specific learning paths.

5.3 Features Enhancing Integrity, Coherence, Relevance, and Accuracy

Integrity

Integrity in AI-generated content is paramount, as it ensures the reliability and correctness of outputs. Evaluations such as those in [76] emphasize the importance of rigorous benchmarks to guarantee the reliability of AI systems. Verification frameworks, as proposed in [59], enhance content integrity by ensuring logical and syntactic correctness. Automated systems, such as those described in [64], further maintain integrity by benchmarking outputs against established educational standards.

Coherence

Coherence in AI frameworks is critical to achieving logical flow and comprehensibility in generated content. Frameworks leveraging chain-of-thought and iterative prompting techniques, as discussed in [87], enhance the logical structure of outputs. Similarly, studies such as [66] provide structured evaluation metrics that ensure logically sound AI outputs. Modular approaches, as demonstrated in [88], improve coherence by systematically addressing complex programming tasks. Furthermore, adaptive scaffolding techniques, described in [56], align coherence with the progression and understanding of learners.

Relevance

Relevance ensures that AI-generated content meets the contextual and educational needs of learners. Tools such as [57] tailor learning materials to specific objectives and user profiles, ensuring contextual relevance. Multilingual frameworks like [61] enhance relevance by making educational content accessible to diverse demographics. Additionally, studies such as [89] underline the importance of domain-specific customization to maintain relevance. Research on adaptive feedback systems, as highlighted in [55], ensures relevance by dynamically responding to learner inputs and progress.

Accuracy

Accuracy is a foundational feature of effective AI frameworks, particularly in programming education. Research comparing LLMs, such as [31], identifies accuracy as a critical factor for successful implementation. Automated debugging and error correction, detailed in [90], minimize errors and enhance the precision of generated outputs. Benchmarks like those provided in [91] are instrumental in evaluating and improving accuracy across tasks. Furthermore, studies such as [73] demonstrate how accuracy contributes to reliable and error-free content delivery, ultimately improving learner outcomes. Feedback-driven mechanisms, as described in [92], underscore the role of accuracy in developing personalized educational tools that meet learners' specific needs.

6. DISCUSSION

The findings of this systematic literature review underscore the transformative potential of Generative AI in programming education. Adaptive systems, which tailor content dynamically based on learners' progress and needs, emerged as a key feature for enhancing educational outcomes. The ability of these systems to adjust the difficulty of exercises and provide personalized feedback significantly improved learner engagement and retention rates. These results align with existing literature emphasizing the benefits of adaptive learning environments in fostering personalized education.

Contextual relevance also plays a pivotal role in improving programming education. By aligning AI-generated content with real-world applications and learner-specific goals, contextual tools, such as debugging assistants, have proven effective in addressing common errors and enhancing problem-solving skills. The reviewed studies highlight how practical programming scenarios integrated into AI frameworks bridge the gap between theoretical concepts and real-world applications, making learning more effective and engaging.

Despite these benefits, challenges remain, particularly in ensuring coherence in AI-generated content. Logical flow and consistency are essential for maintaining the clarity of explanations and fostering a smooth learning experience. Mechanisms for coherence checking, as highlighted in the findings, can address this challenge to some extent, but further advancements in AI frameworks are necessary to minimize disjointed or contradictory outputs. Additionally, the review identifies the need for robust evaluation metrics to measure the functional correctness and educational effectiveness of AI-generated content.

Several limitations were observed in the reviewed studies. The accuracy of AI-generated content continues to be a concern, particularly in technical domains like programming, where precision is paramount. Inaccurate or misleading outputs can hinder learning, making it essential to refine AI frameworks to ensure reliability. Ethical considerations, such as bias and fairness in AI systems, also present significant challenges that require attention. Bias in AI-generated content can lead to unequal learning opportunities, emphasizing the need for transparent and inclusive development practices.

Future research should focus on longitudinal studies to evaluate the sustained impact of Generative AI in education. Additionally, efforts to scale AI frameworks to accommodate diverse programming languages and contexts will be critical for expanding their applicability. Strategies to mitigate ethical concerns, enhance inclusivity, and improve coherence should also be prioritized in future development.

7. CONCLUSION

This systematic literature review aimed to identify and analyze the key features of Generative AI frameworks that enhance personalized learning in programming education. The review highlighted adaptability, contextual relevance, and coherence as critical features for creating effective AI-driven educational tools. Adaptability enables dynamic adjustments to individual learners' needs, leading to improved engagement and retention. Contextual relevance ensures that AI-generated content aligns with learners' goals and real-world applications, enhancing problem-solving skills and reducing errors. Coherence, while challenging to maintain, is vital for fostering logical and effective learning experiences.

The findings contribute to the growing body of knowledge on the role of Generative AI in programming education by providing a structured analysis of its features and limitations. This review also underscores the importance of robust evaluation metrics and ethical considerations in the development of AI-driven tools. By addressing these challenges, Generative AI has the potential to revolutionize programming education, making it more personalized, engaging, and effective.

Future research should explore advanced coherence-checking mechanisms, scalable frameworks for diverse programming languages, and ethical guidelines to ensure fairness and inclusivity. Long-term studies assessing the sustained impact of Generative AI in educational settings will also be essential for understanding its full potential. Through these efforts, Generative AI can become a transformative force in education, empowering learners and educators alike.

ACKNOWLEDGMENT

We sincerely appreciate the contributions and collaboration of Murang'a University of Technology, University of Embu, Machakos University, and Mount Kenya University. Their academic support and dedication played a crucial role in the completion of this study. This work would not have been possible without the efforts and encouragement of all involved institutions and individuals.

REFERENCES

1. D. BaiDoo-Anu and L. Owusu Ansah, **Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning**, J. AI, vol. 7, no. 1, pp. 52–62, Dec. 2023, doi: 10.61969/jai.1337500.
2. M. Alier, F.-J. García-Peñalvo, and J. D. Camba, **Generative Artificial Intelligence in Education: From Deceptive to Disruptive**, Int. J. Interact. Multimed. Artif. Intell., vol. 8, no. 5, p. 5, 2024, doi: 10.9781/ijimai.2024.02.011.
3. Z. Bahroun, C. Anane, V. Ahmed, and A. Zacca, **Transforming Education: A Comprehensive Review of Generative Artificial Intelligence in Educational Settings through Bibliometric and Content Analysis**, Sustainability, vol. 15, no. 17, p. 12983, Aug. 2023, doi: 10.3390/su151712983.
4. J. M. Han et al., **Unsupervised Neural Machine Translation with Generative Language Models Only**, 2021, arXiv. doi: 10.48550/ARXIV.2110.05448.
5. R. Bommasani et al., **On the Opportunities and Risks of Foundation Models**, 2021, arXiv. doi: 10.48550/ARXIV.2108.07258.
6. S. Koyejo and Neural Information Processing Systems Foundation, **Eds., 36th Conference on Neural Information Processing Systems (NeurIPS 2022): New Orleans, Louisiana, USA, 28 November-9 December 2022. in Advances in neural information processing systems**, no. 35. Red Hook, NY: Curran Associates, Inc, 2023.
7. P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, **Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing**, ACM Comput. Surv., vol. 55, no. 9, pp. 1–35, Sep. 2023, doi: 10.1145/3560815.
8. H. Touvron et al., **LLaMA: Open and Efficient Foundation Language Models**, 2023, arXiv. doi: 10.48550/ARXIV.2302.13971.
9. N. B. Afini Normadhi, L. Shuib, H. N. Md Nasir, A. Bimba, N. Idris, and V. Balakrishnan, **Identification of personal traits in adaptive learning environment: Systematic literature review**, Comput. Educ., vol. 130, pp. 168–190, Mar. 2019, doi: 10.1016/j.compedu.2018.11.005.
10. G. Yenduri et al., **GPT (Generative Pre-Trained Transformer)— A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions**, IEEE Access, vol. 12, pp. 54608–54649, 2024, doi: 10.1109/ACCESS.2024.3389497.
11. S. Mohamadi, G. Mujtaba, N. Le, G. Doretto, and D. A. Adjeroh, **ChatGPT in the Age of Generative AI and Large Language Models: A Concise Survey**, Jul. 15, 2023, arXiv: arXiv:2307.04251. Accessed: Sep. 05, 2024. [Online]. Available: http://arxiv.org/abs/2307.04251
12. R. Dale, **A year's a long time in generative AI**, Nat. Lang. Eng., vol. 30, no. 1, pp. 201–213, Jan. 2024, doi: 10.1017/S1351324923000554.
13. S. C. Tan, W. Chen, and B. L. Chua, **Leveraging generative artificial intelligence based on large language models for collaborative learning**, Learn. Res. Pract., vol. 9, no. 2, pp. 125–134, Jul. 2023, doi: 10.1080/23735082.2023.2258895.
14. J. Ahn, W. Sung, and J. B. Black, **Unplugged Debugging Activities for Developing Young Learners' Debugging Skills**, J. Res. Child. Educ., vol. 36, no. 3, pp. 421–437, May 2022, doi: 10.1080/02568543.2021.1981503.
15. A. Akkaya and Y. Akpinar, **Experiential serious-game design for development of knowledge of object-oriented programming and computational thinking skills**, Comput. Sci. Educ., vol. 32, no. 4, pp. 476–501, Oct. 2022, doi: 10.1080/08993408.2022.2044673.
16. D. Wood and S. H. Moss, **Evaluating the impact of students' generative AI use in educational contexts**, J.

Res. Innov. Teach. Learn., vol. 17, no. 2, pp. 152–167, Aug. 2024, doi: 10.1108/JRIT-06-2024-0151.

17. T. B. Brown et al., **Language Models are Few-Shot Learners**, Jul. 22, 2020, arXiv: arXiv:2005.14165. Accessed: Sep. 05, 2024. [Online]. Available: http://arxiv.org/abs/2005.14165
18. E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, **On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?** , in Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event Canada: ACM, Mar. 2021, pp. 610–623. doi: 10.1145/3442188.3445922.
19. M. Yue, M. S.-Y. Jong, and Y. Dai, **Pedagogical Design of K-12 Artificial Intelligence Education: A Systematic Review**, Sustainability, vol. 14, no. 23, p. 15620, Nov. 2022, doi: 10.3390/su142315620.
20. E. Kasneci et al., **ChatGPT for good? On opportunities and challenges of large language models for education**, Learn. Individ. Differ., vol. 103, p. 102274, Apr. 2023, doi: 10.1016/j.lindif.2023.102274.
21. J. Chen et al., **When large language models meet personalization: perspectives of challenges and opportunities**, World Wide Web, vol. 27, no. 4, p. 42, Jul. 2024, doi: 10.1007/s11280-024-01276-1.
22. J. Prather et al., **Interactions with Prompt Problems: A New Way to Teach Programming with Large Language Models**, 2024, arXiv. doi: 10.48550/ARXIV.2401.10759.
23. L. Murr, M. Grainger, and D. Gao, **Testing LLMs on Code Generation with Varying Levels of Prompt Specificity**, 2023, arXiv. doi: 10.48550/ARXIV.2311.07599.
24. J. Finnie-Ansley, P. Denny, B. A. Becker, A. Luxton-Reilly, and J. Prather, **The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming**, in Proceedings of the 24th Australasian Computing Education Conference, Virtual Event Australia: ACM, Feb. 2022, pp. 10–19. doi: 10.1145/3511861.3511863.
25. Y. Chang et al., **A Survey on Evaluation of Large Language Models**, ACM Trans. Intell. Syst. Technol., vol. 15, no. 3, pp. 1–45, Jun. 2024, doi: 10.1145/3641289.
26. M. Kazemitabaar, J. Chow, C. K. T. Ma, B. J. Ericson, D. Weintrop, and T. Grossman, **Studying the effect of AI Code Generators on Supporting Novice Learners in Introductory Programming**, in Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Hamburg Germany: ACM, Apr. 2023, pp. 1–23. doi: 10.1145/3544548.3580919.
27. N. Kiesler and D. Schiffner, **Large Language Models in Introductory Programming Education: ChatGPT's Performance and Implications for Assessments**, 2023, arXiv. doi: 10.48550/ARXIV.2308.08572.
28. H. Pearce, B. Tan, P. Krishnamurthy, F. Khorrami, R. Karri, and B. Dolan-Gavitt, **Pop Quiz! Can a Large Language Model Help With Reverse Engineering?**, 2022, arXiv. doi: 10.48550/ARXIV.2202.01142.
29. O. Asare, M. Nagappan, and N. Asokan, **Is GitHub's Copilot as bad as humans at introducing vulnerabilities in code?**, Empir. Softw. Eng., vol. 28, no. 6, p. 129, Nov. 2023, doi: 10.1007/s10664-023-10380-1.
30. A. Moradi Dakhel, V. Majdinasab, A. Nikanjam, F. Khomh, M. C. Desmarais, and Z. M. (Jack) Jiang, **GitHub Copilot AI pair programmer: Asset or Liability?**, J. Syst. Softw., vol. 203, p. 111734, Sep. 2023, doi: 10.1016/j.jss.2023.111734.
31. M. Chen et al., **Evaluating Large Language Models Trained on Code**, 2021, arXiv. doi: 10.48550/ARXIV.2107.03374.
32. L. Reynolds and K. McDonell, **Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm**, in Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama Japan: ACM, May 2021, pp. 1–7. doi: 10.1145/3411763.3451760.
33. D. Zhou et al., **Least-to-Most Prompting Enables Complex Reasoning in Large Language Models**, 2022, arXiv. doi: 10.48550/ARXIV.2205.10625.
34. A. Narayan, I. Chami, L. Orr, S. Arora, and C. Ré, **Can Foundation Models Wrangle Your Data?**, 2022, arXiv. doi: 10.48550/ARXIV.2205.09911.
35. J. White et al., **A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT**, 2023, doi: 10.48550/ARXIV.2302.11382.
36. Y. Sasaki, H. Washizaki, J. Li, D. Sander, N. Yoshioka, and Y. Fukazawa, **Systematic Literature Review of Prompt Engineering Patterns in Software Engineering**, in 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), Osaka, Japan: IEEE, Jul. 2024, pp. 670–675. doi: 10.1109/COMPSAC61105.2024.00096.
37. D. Hendrycks et al., **Measuring Coding Challenge Competence With APPS**, 2021, arXiv. doi: 10.48550/ARXIV.2105.09938.
38. F. F. Xu, U. Alon, G. Neubig, and V. J. Hellendoorn, **A systematic evaluation of large language models of code**, in Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming, San Diego CA USA: ACM, Jun. 2022, pp. 1–10. doi: 10.1145/3520312.3534862.
39. D. Zan et al., **Large Language Models Meet NL2Code: A Survey**, 2022, arXiv. doi: 10.48550/ARXIV.2212.09420.
40. E. A. AlOmar, A. Venkatakrishnan, M. W. Mkaouer, C. Newman, and A. Ouni, **How to refactor this code? An exploratory study on developer-ChatGPT refactoring conversations**, in Proceedings of the 21st International Conference on Mining Software Repositories, Lisbon Portugal: ACM, Apr. 2024, pp. 202–206. doi: 10.1145/3643991.3645081.
41. P. Jain, A. Jain, T. Zhang, P. Abbeel, J. E. Gonzalez, and I. Stoica, **Contrastive Code Representation Learning**, 2020, doi: 10.48550/ARXIV.2007.04973.
42. M. Ciniselli, N. Cooper, L. Pascarella, D. Poshyvanyk, M. Di Penta, and G. Bavota, **An Empirical Study on the Usage of BERT Models for Code Completion**, in 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), Madrid, Spain: IEEE,

May 2021, pp. 108–119. doi: 10.1109/MSR52588.2021.00024.
43. T. Wang and Z. Chen, **Analyzing Code Text Strings for Code Evaluation**, in 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy: IEEE, Dec. 2023, pp. 5619–5628. doi: 10.1109/BigData59044.2023.10386406.
44. S. Gökoğlu and S. Kilic, **Programming learning and teaching of pre-service computer science teachers: Challenges, concerns, and solutions**, E-Learn. Digit. Media, vol. 20, no. 5, pp. 498–518, Sep. 2023, doi: 10.1177/20427530221117331.
45. I. Gligorea, M. Cioca, R. Oancea, A.-T. Gorski, H. Gorski, and P. Tudorache, **Adaptive Learning Using Artificial Intelligence in e-Learning: A Literature Review**, Educ. Sci., vol. 13, no. 12, p. 1216, Dec. 2023, doi: 10.3390/educsci13121216.
46. L. Chen, P. Chen, and Z. Lin, **Artificial Intelligence in Education: A Review**, IEEE Access, vol. 8, pp. 75264–75278, 2020, doi: 10.1109/ACCESS.2020.2988510.
47. W. S. Sayed et al., **AI-based adaptive personalized content presentation and exercises navigation for an effective and engaging E-learning platform**, Multimed. Tools Appl., vol. 82, no. 3, pp. 3303–3333, Jan. 2023, doi: 10.1007/s11042-022-13076-8.
48. C. Diwan, S. Srinivasa, G. Suri, S. Agarwal, and P. Ram, **AI-based learning content generation and learning pathway augmentation to increase learner engagement**, Comput. Educ. Artif. Intell., vol. 4, p. 100110, 2023, doi: 10.1016/j.caeai.2022.100110.
49. R. AlShaikh, N. Al-Malki, and M. Almasre, **The implementation of the cognitive theory of multimedia learning in the design and evaluation of an AI educational video assistant utilizing large language models**, Heliyon, vol. 10, no. 3, p. e25361, Feb. 2024, doi: 10.1016/j.heliyon.2024.e25361.
50. L. I. Ruiz-Rojas, P. Acosta-Vargas, J. De-Moreta-Llovet, and M. Gonzalez-Rodriguez, **Empowering Education with Generative Artificial Intelligence Tools: Approach with an Instructional Design Matrix**, Sustainability, vol. 15, no. 15, p. 11524, Jul. 2023, doi: 10.3390/su151511524.
51. S. V. Chinta et al., **FairAIED: Navigating Fairness, Bias, and Ethics in Educational AI Applications**, 2024, arXiv. doi: 10.48550/ARXIV.2407.18745.
52. E. Ferrara, **Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies**, Sci, vol. 6, no. 1, p. 3, Dec. 2023, doi: 10.3390/sci6010003.
53. R. González-Sendino, E. Serrano, J. Bajo, and P. Novais, **A Review of Bias and Fairness in Artificial Intelligence**, Int. J. Interact. Multimed. Artif. Intell., vol. In press, no. In press, p. 1, 2023, doi: 10.9781/ijimai.2023.11.001.
54. U. Mittal, S. Sai, V. Chamola, and D. Sangwan, **A Comprehensive Review on Generative AI for Education**, IEEE Access, vol. 12, pp. 142733–142759, 2024, doi: 10.1109/ACCESS.2024.3468368.
55. L. Bonde, **A Generative Artificial Intelligence Based Tutor for Personalized Learning**, in 2024 IEEE SmartBlock4Africa, Accra, Ghana: IEEE, Sep. 2024, pp. 1–10. doi: 10.1109/SmartBlock4Africa61928.2024.10779525.
56. S. Sarsa, P. Denny, A. Hellas, and J. Leinonen, **Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models**, in Proceedings of the 2022 ACM Conference on International Computing Education Research - Volume 1, Lugano and Virtual Event Switzerland: ACM, Aug. 2022, pp. 27–43. doi: 10.1145/3501385.3543957.
57. R. Kadar, N. Abdul Wahab, J. Othman, M. Shamsuddin, and S. B. Mahlan, **A Study of Difficulties in Teaching and Learning Programming: A Systematic Literature Review**, Int. J. Acad. Res. Progress. Educ. Dev., vol. 10, no. 3, p. Pages 591-605, Aug. 2021, doi: 10.6007/IJARPED/v10-i3/11100.
58. C. S. Cheah, **Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review**, Contemp. Educ. Technol., vol. 12, no. 2, p. ep272, May 2020, doi: 10.30935/cedtech/8247.
59. Y. Li, W. Ji, J. Liu, and W. Li, **Application of Generative Artificial Intelligence Technology in Customized Learning Path Design: A New Strategy for Higher Education**, in 2024 International Conference on Interactive Intelligent Systems and Techniques (IIST), Bhubaneswar, India: IEEE, Mar. 2024, pp. 567–573. doi: 10.1109/IIST62526.2024.00099.
60. K. R. Pradeep, A. S. Manish, A. S. Adithiyaa, N. Sahana, and S. T. Abhishek, **Personalized Adaptive Learning Platform Empowered by Artificial Intelligence**, in 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), Chikkaballapur, India: IEEE, Apr. 2024, pp. 1–8. doi: 10.1109/ICKECS61492.2024.10617075.
61. Q. Zheng et al., **CodeGeeX: A Pre-Trained Model for Code Generation with Multilingual Benchmarking on HumanEval-X**, 2023, arXiv. doi: 10.48550/ARXIV.2303.17568.
62. A. Narasimhan, K. P. A. V. Rao, and V. M. B, **CGEMs: A Metric Model for Automatic Code Generation using GPT-3**, 2021, arXiv. doi: 10.48550/ARXIV.2108.10168.
63. E. Aftabi, B. N. Shirazi, A. A. Safavi, G. Salimi, and H. Aftabi, **A Framework for Customized Course Design and Personalized Learning with AI**, in 2024 11th International and the 17th National Conference on E-Learning and E-Teaching (ICeLeT), Isfahan, Iran, Islamic Republic of: IEEE, Feb. 2024, pp. 1–6. doi: 10.1109/ICeLeT62507.2024.10493063.
64. B. Chen, Z. Zhang, N. Langrené, and S. Zhu, **Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review**, Sep. 05, 2024, arXiv: arXiv:2310.14735. doi: 10.48550/arXiv.2310.14735.
65. P. Lauren and P. Watta, **Work-in-Progress: Integrating Generative AI with Evidence-based Learning Strategies in Computer Science and Engineering Education**, in 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA: IEEE, Oct. 2023, pp. 1–5. doi: 10.1109/FIE58773.2023.10342970.

66. L. Sun and Z. Shi, **Prompt Learning Under the Large Language Model**, in 2023 International Seminar on Computer Science and Engineering Technology (SCSET), New York, NY, USA: IEEE, Apr. 2023, pp. 288–291. doi: 10.1109/SCSET58950.2023.00070.
67. A. Del Carpio Gutierrez, P. Denny, and A. Luxton-Reilly, **Evaluating Automatically Generated Contextualised Programming Exercises**, in Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1, Portland OR USA: ACM, Mar. 2024, pp. 289–295. doi: 10.1145/3626252.3630863.
68. J. Li, P. Chen, and J. Jia, **MoTCoder: Elevating Large Language Models with Modular of Thought for Challenging Programming Tasks**, 2023, doi: 10.48550/ARXIV.2312.15960.
69. R. Makharia et al., **AI Tutor Enhanced with Prompt Engineering and Deep Knowledge Tracing**, in 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India: IEEE, Mar. 2024, pp. 1–6. doi: 10.1109/IATMSI60426.2024.10503187.
70. Á. Becerra, Z. Mohseni, J. Sanz, and R. Cobos, **A Generative AI-Based Personalized Guidance Tool for Enhancing the Feedback to MOOC Learners**, in 2024 IEEE Global Engineering Education Conference (EDUCON), Kos Island, Greece: IEEE, May 2024, pp. 1–8. doi: 10.1109/EDUCON60312.2024.10578809.
71. J. Zhang, P. Nie, J. J. Li, and M. Gligoric, **Multilingual Code Co-evolution using Large Language Models**, in Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, San Francisco CA USA: ACM, Nov. 2023, pp. 695–707. doi: 10.1145/3611643.3616350.
72. A. Kovari and J. Katona, **Transformative Applications and Key Challenges of Generative AI**, in 2024 IEEE 7th International Conference and Workshop Óbuda on Electrical and Power Engineering (CANDO-EPE), Budapest, Hungary: IEEE, Oct. 2024, pp. 89–92. doi: 10.1109/CANDO-EPE65072.2024.10772832.
73. Y. Li, J. Shi, and Z. Zhang, **An Approach for Rapid Source Code Development Based on ChatGPT and Prompt Engineering**, IEEE Access, vol. 12, pp. 53074–53087, 2024, doi: 10.1109/ACCESS.2024.3385682.
74. S. Lu et al., **CodeXGLUE: A Machine Learning Benchmark Dataset for Code Understanding and Generation**, 2021, arXiv. doi: 10.48550/ARXIV.2102.04664.
75. W. Lyu, Y. Wang, T. (Rachel) Chung, Y. Sun, and Y. Zhang, **Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Study**, in Proceedings of the Eleventh ACM Conference on Learning @ Scale, Atlanta GA USA: ACM, Jul. 2024, pp. 63–74. doi: 10.1145/3657604.3662036.
76. S. Speth, N. Meißner, and S. Becker, **Investigating the Use of AI-Generated Exercises for Beginner and Intermediate Programming Courses: A ChatGPT Case Study**, in 2023 IEEE 35th International Conference on Software Engineering Education and Training (CSEE&T), Tokyo, Japan: IEEE, Aug. 2023, pp. 142–146. doi: 10.1109/CSEET58097.2023.00030.