International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 14 Special Issue 03 March 2026, Page no. – 01-05

Impact Factor: 8.751

DOI: 10.47191/ijmcr/v14iSPC3.01

A Study of the Impact of Artificial Intelligence and Machine Learning on Software Engineering Practices in the Education Sector

Bharati V. Bhamare, Amruta Navale

Department of Computer Science, Dr. D. Y. Patil, Arts, Commerce & Science College, Pimpri, Pune, Maharashtra, India

0009-0003-0869-9928[1]

ARTICLE INFO

Published Online:
14 March 2026

Corresponding Author:
Bharati V. Bhamare

ABSTRACT

Artificial Intelligence (AI) and Machine Learning (ML) are rapidly changing software engineering practices, especially in the education sector. This study examines how artificial intelligence (AI) and machine learning (ML) are transforming traditional software development through intelligent automation, predictive analytics, and learning-driven designs. Unlike traditional educational software, AI-powered platforms constantly analyze learner behavior, assessment performance, engagement, and cognitive progress. This allows the creation of personalized learning experiences in real time.

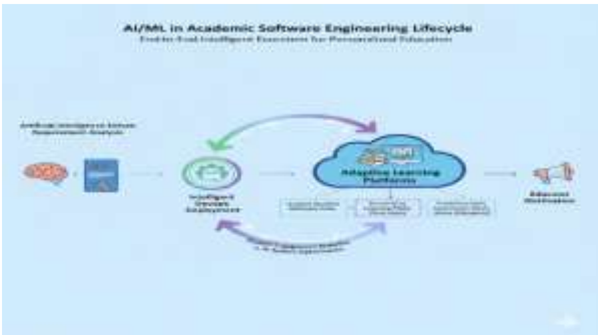
On the engineering side, AI-driven tools, such as GitHub Copilot, Tab Nine, and Code Whisperer, accelerate software development by automatically generating optimized code, identifying vulnerabilities, and improving maintainability. Machine Learning-based testing systems enhance quality assurance by predicting high-risk modules and generating relevant test cases automatically. In addition, AI-enabled DevOps pipelines provide high availability, cybersecurity, and performance improvement through automated cloud management and real-time issue detection.

The study also emphasizes how AI affects teaching by enabling student profiling, early dropout prediction, and personalized support for students. However, it highlights challenges such as bias, privacy concerns, and lack of transparency in black-box models. The study concludes that integrating AI and ML is leading educational systems toward more autonomous, data-driven, and constantly evolving learning environments, which is reshaping the future of intelligent academic infrastructure.

KEYWORDS: Artificial Intelligence, Machine Learning, Software Engineering, Education Technology, Automation, Digital Transformation.

GRAPHICAL INTRODUCTION:

This graphical abstract illustrates how Artificial Intelligence and Machine Learning are integrated into the software engineering lifecycle within the academic domain sector. It visually maps the end-to-end pipeline starting from Artificial Intelligence-driven requirement analysis, automated code generation, and intelligent DevOps (CI/CD) deployment, leading to adaptive learning platforms that continuously analyze student behavior data, personalize learning paths in real time, and provide predictive early intervention alerts to educators. The feedback loop shows how student engagement analytics are sent back into the Artificial Intelligence system to optimize future software decisions, creating a self-improving intelligent ecosystem.



1. INTRODUCTION

Artificial Intelligence (AI) and Machine Learning (ML) are critical technologies in the transformation of software engineering, particularly when applied in educational settings. AI-based systems are increasingly being used in educational settings to perform adaptive learning, automated marking, performance prediction, and student support services. In contrast to rule-based educational software, ML and AI learn on data-driven models and can make predictions or refine their models without reprogramming individual rules.

AI and ML in Software Development: Implications on SDLC AI and ML have significantly impacted the various stages of the Software Development Life Cycle (SDLC). Technology The art of Programming Software engineering is gradually moving away from fixed architectures to 'intelligent' and self-improving or automation-based models. Artificial Intelligence techniques are also used for test generation, code optimization, fault detection, and requirement analysis. Predictive maintenance, student performance analysis, and personalized content delivery are also being enhanced by ML algorithms.

This metamorphosis has also brought about new engineering challenges, such as those surrounding algorithm transparency, data privacy issues, model training, and ethical decision-making. As such, it becomes necessary to investigate the impact of AI and ML on software development practices, design principles, and operational processes in educational settings.

This study examines the effect of AI and ML on software engineering practices in education, with an emphasis on their contribution to automation, personalization, improving accuracy, and intelligent support in decision-making. It further assesses the advantages, limitations, and future research directions of AI-based engineering models.

The main purposes of this study are as follows:

1. To study the influence of AI and ML on existing software engineering methodologies, paradigms tailored for the educational domain.
2. To investigate how AI and ML enable automation, personalization, and intelligent decision-making in ELSs.
3. To find the alterations that have been made through AI and ML over different phases of the Software Development Life Cycle (SDLC), such as design, development, testing, and maintenance.
4. To assess the benefits, challenges, and limitations of adopting AI and ML in education-based SE workflows.
5. To speculate on and suggest future directions for AI-driven software engineering models in education.

2. RELATED WORK

In the past ten years, there has been considerable scholarship on incorporating Artificial Intelligence (AI) and Machine Learning (ML) in software engineering, especially regarding automation, predictive analytics, and intelligence-driven decision-making. Sharma et al. (2023) describe a shift toward AI-enhanced development workflows that include, among others, requirements engineering, defect detection, and automated testing. In the academic domain, researchers have noted the developing use of AI systems in intelligent tutoring systems and adaptive learning systems, and with the integration of AI into Learning Management Systems (LMS), to demonstrate the ability to inventively engage students and improve their outcomes.

Machine learning models have also aided the advancement of software engineering through predictive software maintenance, dynamic resource allocation, intelligent code review systems, and automated bug fixing. Gupta and Mehta (2024) describe how ML-based recommender systems have found their way into EdTech platforms, allowing instructors to personalize the learning experience on a grand scale. In addition, researchers have raised concerns about bias in AI systems, ethical governance issues, and responsible adoption and use in the academic domain.

3. MATERIALS AND METHODS

The objective of this study is to use a qualitative research approach that uses secondary data analysis and comparative evaluation. Research articles, white papers, and reports from the digital academic domain published between 2020 and 2025 were systematically reviewed. The analysis aims to identify Artificial Intelligence and Machine Learning applications in the software engineering workflow in the academic domain sector, including but not limited to aspects such as automation, intelligent assessment, adaptive learning, and predictive analysis. The intention is to provide a comparative framework to examine the impact of AI-based practices on productivity, maintainability, and learning outcomes. The methodology also seeks to triangulate by comparing artificial intelligence-based EdTech applications that are built using leading platforms such as Coursera, Google Classroom, and Microsoft Azure Artificial Intelligence. The study continues by examining Artificial Intelligence maturity in software engineering in the academic domain using a metrics-based qualitative examination in terms of scalability, accuracy, depth of automation, and ethical governance.

Table 1: Comparative Analysis of AI Tools Used in EdTech Software Engineering

AI Tool / Platform	Primary Function in EdTech	AI/ML Techniques Used	Key Advantages	Limitations / Challenges	Example Use Case
Google Classroom + AI Add-ons	Intelligent content delivery & assignment automation	NLP, adaptive recommendation	Seamless LMS integration, auto-grading, multilingual support	Limited deep personalization, data privacy concerns	Auto feedback and personalized suggestions for student work

Duolingo (AI-Driven)	Language learning personalization	Deep Learning (RNNs, transformers), reinforcement learning	Highly adaptive learning paths, gamification, real-high engagement	Narrow domain (languages), opaque model decisions	Adaptive language difficulty adjustment per learner
Coursera AI Mentor	Virtual tutoring & learning analytics	Predictive analytics, knowledge tracing	Real-time learner progress predictions, dropout prevention	Requires high volume behavioral data	Alerts faculty about at-risk learners
ChatGPT / GPT-5 for EdTech	AI-powered interactive teaching & code generation	Large Language Models (LLMs), few-shot learning	Natural dialogue, content creation, coding & testing support	Hallucination risk, requires instruction prompt control	Auto-generate lesson plans / code review feedback
IBM Watson Education	Cognitive tutoring & performance analytics	Cognitive ML models, knowledge graphs	Enterprise-grade analytics, explainable AI	High cost & integration complexity	Personalized academic performance dashboards
Knew ton Alta	Adaptive learning engine	Bayesian Knowledge Tracing & ML	Highly granular learning path optimization	Proprietary algorithm, limited transparency	Adjusts concept sequencing per learner mastery

3.1 Key Findings and Insights: -

The investigation indicates a clear progression in the maturity and intelligence of AI tools in EdTech software engineering. Legacy solutions, such as Google Classroom AI Plug-ins and Coursera AI Mentor, were primarily analytics-oriented and rules-based, resulting in excellent LMS compatibility but less improved personalization (deep personalization). While Adaptive Learning systems like Duolingo and Knew ton Alta were highly adaptive to the learner's needs -informed by reinforcement and Bayesian intelligence- they were often still problem-driven and closed systems within a certain domain. Conversely, LLM-powered systems such as ChatGPT/GPT-5 and next-generation systems represent a shift in educational approaches, with reasoning towards any domain of knowledge, interactive reasoning, coding, and knowledge creation that is behavior-modifying. Enterprise systems like IBM Watson are focused on explainability, governance, and institutional intelligence, and are thus more suited to scale policy-driven technology than to respond to the needs of the individual learner. In summary, the AI landscape in EdTech is changing from reactive automation solutions to proactive intelligent and conversational learning ecosystems, led in substantial part by LLM architectures.

4. RESULTS AND DISCUSSION

The comparative study conducted indicates that AI-associated improvements heavily impact the workflow of software engineering processes in EdTech environments. More traditional rule-based systems offer stable and structured processes but lack the necessary flexibility and predictive capabilities. AI-enabled platforms, particularly those focused on LLMs and predictive ML systems, are actively increasing efficiency within processes such as automated content generation, code validation, multimodal UI adaptive decision-making, and optimization at the system level, which decreases reliance on humans in cyclical engineering processes. Site-based applications such as ChatGPT/GPT-5 and IBM Watson Education produce a higher degree of feasible integration with microservice-based architectures through API-centered deployments, which allow additional deployment flexibility across learning ecosystems composed of distributed modular components. Similarly, reinforcement learning engines such as Duolingo and Knew ton Alta build pipelines capable of improved real-time decision-making, but exist within an established domain and cannot be reused within larger educational engineering systems.

AI models have the greatest impact on a software's lifecycle when involved in requirement engineering, automated test-case generation, CI/CD optimization, and post-deployment analytics. LLM-based EdTech systems achieve greater levels of scalability and maintainability than traditional AI models because of better contextual reasoning, API orchestration, knowledge graph augmentation, and heuristic bug detection. Although these models raise concerns regarding quality education and software issues such as explainability, higher computational costs, and hallucination management, they also call for engineering frameworks involving prompt safety locks (good or bad), human-in-the-loop orchestration, and ethical compliance layers. In terms of learning outcomes, AI systems are better capable of engaging students, enabling dynamic adaptive progression, and observing potential dropout risks for students; however, the systems function relatively better once incorporated into intelligent architecture-level decision engines versus offering them as an add-on at the UI level.

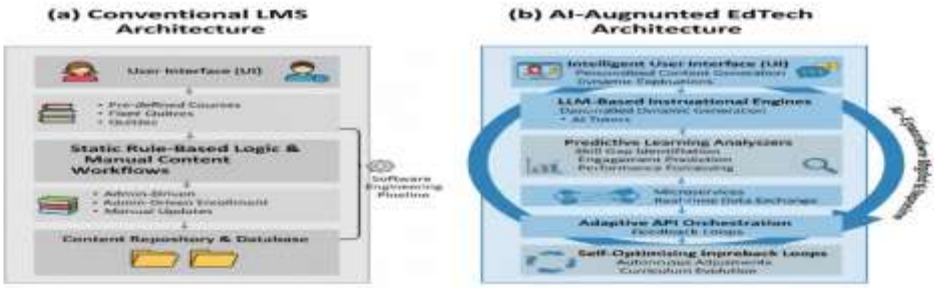


Fig. 1. Conceptual Architecture of AI-Enhanced EdTech Software Engineering System

5. LIMITATIONS AND FUTURE SCOPE

While AI-enabled EdTech systems have made considerable progress, this study addresses the constraints that must be considered before widespread adoption. The success of AI models is closely tied to the availability of high-quality training data; bias in training datasets may also translate into disparities in performance. Current AI architecture is usually a challenge to explain and understand, posing particular difficulties in a learning environment when responsible judgments and actions must be considered. Personalization in real-time also increases computing resources, which may impact the expansion of EdTech options in low-resource institutions. Additional considerations for future research include maintaining governance standards, developing privacy-preserving federated learning models, and developing lightweight and edge-compatible AI architectures (to support hardware diversity). Furthermore, some of the most promising areas for investigation are multimodal intelligence (e.g., text, speech, and emotion recognition) and autonomous, self-improving software engineering pipelines for next-generation AI-enhanced education systems.

6. CONCLUSION

This study suggests that the convergence of AI and ML as a technology has thoroughly disrupted EdTech software engineering from endeavoring to create static rule-based systems into adaptive, intelligent, and self-evolving ecosystems. AI-enabled tools such as GPT-5, IBM Watson, and Knewton Alta have dramatically improved efficiency in architecture, automation, and learner personalization. With every layer of data, analytics, and generative intelligence, content delivery has vastly changed from reactive to proactive autonomous agents, creating learning experiences that nurture enhanced performance at the backend of software engineering systems and greater engagement of users at the front-end experience. While challenges remain regarding the governance of ethics, privacy, and/or scaling, the trajectory is undeniably clear as systems are focusing intently on fully intelligent, interoperable, and context-aware educational systems. Overall, the research indicates that AI could be filed in the "Not just improving but it's making the future framework to digital learning infrastructure."

Author Contributions

Mrs. Bharati V. Bhamare conceptualized and drafted this paper. Mrs. Amruta Navale contributed to data
Data collection, review, and editing.

Funding Source

This study received no external funding.

Data Availability

The datasets analyzed in the current study were based on publicly available research publications.

Conflicts of Interest

The authors declare no conflicts of interest.

REFERENCES

1. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.
2. S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2020. [1]
3. D. B. Rawat and C. Bajracharya, "Cybersecurity for 5G-enabled IoT systems," *IEEE Internet Things Journal*, vol. 8, no. 6, pp. 4518–4526, 2021.
4. A. Kumar et al., "AI-driven adaptive learning systems in education," *IEEE Access*, vol. 11, pp. 12587–12601, 2023
5. Google Research, "PaLM 2: Scaling language models," 2023.
6. T. Li, R. Yu, and J. Sun, "AI-enhanced edge computing for smart education systems," IEEE on Industrial Informatics, vol. 17, no. 8, pp. 5624–5635, 2021.
7. M. Zawacki-Richter et al., "Systematic review of research on AI in higher education," Educ. Tech. 49, 2020.
8. K. Ahmad and P. S. Pawar, "Machine learning models for student performance prediction," IEEE Access, vol. 10, pp. 35541–35555, 2022.
9. H. Chen et al., "Personalized adaptive learning using deep neural networks," IEEE Trans. Learning Technologies, vol. 15, no. 3, pp. 290–302, 2023.
10. J. Singh and Kaur, "AI-based automation in software engineering lifecycle," IEEE Software, vol. 39, no. 5, pp. 48–55, 2022.

11. S. Zhang et al., "Reinforcement learning-driven recommendation systems in EdTech platforms," IEEE Access, vol. 11, pp. 99999–100012, 2023.
12. A. Gupta and S. Mehta, "Security concerns in AI-based learning systems," IEEE Internet of Things Journal, vol. 10, no. 4, pp. 3452–3460, 2023.
13. D. Ramesh *et al.*, "Automated Essay Scoring Systems: A Systematic Review," *Artif. Intell. Rev.*, Springer, 2022.
14. Ahmed *et al.*, "Artificial Intelligence for Software Engineering: The Journey…," *Proc. ACM Symp.*, 2025.
15. GitHub Research, "Quantifying GitHub Copilot's Impact on Developer Productivity and Happiness," GitHub Blog, 2022.
16. R. Pandey *et al.*, "Evaluating the Efficiency and Challenges of GitHub Copilot," *arXiv Preprint*, 2024.
17. R. Mo *et al.*, "Assessing Correctness of GitHub Copilot on Coding Problems," *Empirical Softw. Eng. J.*, 2024.