Understanding Machine Learning's Impact on Software Development through AI-Driven Code Generation

Jatin Verma[1] Kashish Kaushal[2] Devanshu[3] Isha Malhotra[4] Joginder Singh[5]
[1,2,3,4]Department of Computer Science and Engineering [5]Department of Applied Science
[1,2,3,4,5]CGC- College of Engineering, Landran Mohali Punjab, India

Abstract — The most notable transformation in software development practice over the years stems from the advancements in artificial intelligence. Modern methods utilize machine learning algorithms in ways that reduce workforce input, enhance software performance, and automate code writing. This paper focuses on two crucial aspects related to the development of AI in code generation. These aspects include methods and purposes of AI implementation along with its challenges and possibilities. In this study, we detail a number of AI tools such as OpenAI Codex, DeepCode, and PolyCoder, as well as their possible impacts on software engineering. We also examine the norms of applying AI to the software development processes and questions of morality.

Keywords: AI-Powered Code Generation, Machine Learning, Software Development, Neural Code Synthesis, Large Language Models, Automated Programming, Ethical Considerations, And Future Prospects

I. Introduction

The field of software development is experiencing a very rapid transformation as a result of AI, which provides innovative services and products that increase productivity and ease of coding tasks [1]. AI is changing and improving the way people develop, test, and enhance software applications with multi-level machine learning models through AI-powered code authoring. To reduce the burden of coding for humans, AI can automatically generate extensively structured, relevant, and useful code snippets from already available code bases. Along with improving productivity, this development has made it possible for people without any programming skills to participate in software development, which vastly expands the scope of programming.

The essence of AI-powered code generation is the large-scale trained Machine Learning models of different programming languages and frameworks. Codex Openai and DeepCode are examples of these models. This model uses deep learning to create a code that understands and interprets the guidelines of natural languages [2]. By integrating these AI systems into the development environment, software engineers can accelerate the project offering time, allocate important time for important construction problems, and focus on forging code lines. The introduction of these models is a serious change in the software development environment, which increases the accessibility of new developers and increases age productivity [3]. The Impact on software development through AI driven code generation is shown in Figure 1.

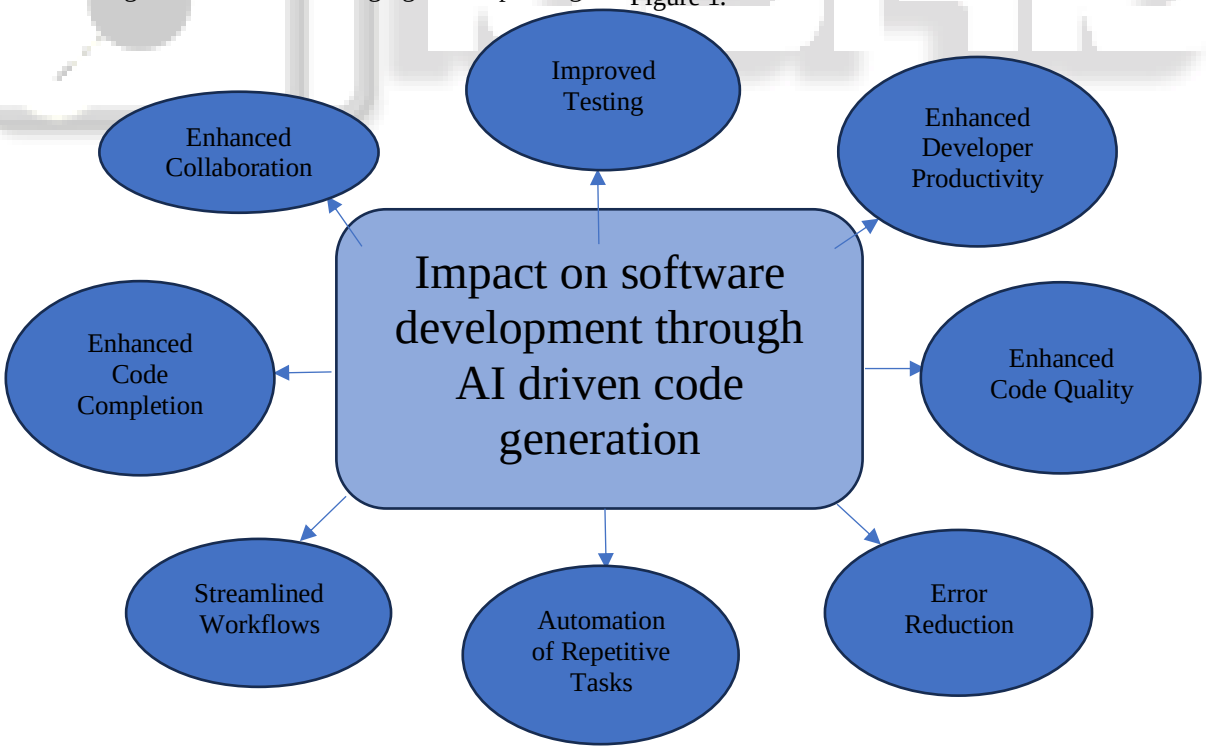

Fig. 1: Impact on software development through AI driven code generation

In addition to creating code based on AI, applications range from debugging and refactoring to code optimization. AI optimizes software, captures errors, provides improvements, and automates the entire fixed process [4]. The effects of AI in software development are very important for sensitive sectors such as finance, medical and cyber security and are especially useful. The implicit understanding of the long -term improvement of the AI model

and the complex code models of the meaning of the meaning of the meaning of the meaning is to create good recommendations and automate automated software development [5].

Along with the clear advantages, AI AutoCoding also indicates a problem. The ethical concerns regarding code quality, intellectual propertyrights, and AI adoption need to be addressed. There is no denying that AI-assisted code does come at a security risk if it's not fully tested, and the answers concerning AI-generated content ownership are still out there. Plus, heavy reliance on AI tools might be fatal. It can lead to poor human coding skills, which can constrain the knowledge spectrum of many upcoming developers and result in the loss of technical prowess [6]. With the continued growth of AI tools, developers have shifted from traditional types of coding to monitor and review diagrams generated by AI [7].Human programmers are not outdated, but are permitted to create innovative things, solve complex problems, and solve architectural systems. The partnership between developers and AI solutions must be perfectly balanced. This allows AI to provide written support, but there are people there who are intelligently in the process when achieving complex tasks [8]. Code un with AI Tools is one of the key technologies that change programming by automating activities and increasing developer productivity [9]. Well-trained machine learning models for large coded data banks can create labor fragments, debug applications, and increase software protection [10]. In this article, we will show you how the inclusion of AI tools can improve the effectiveness of writing software by switching from secular tasks to critical, complex problem solving.

II. AI-Powered Code Generation Models

AI-powered code era models rent deep, getting to know methodologies, especially neural networks and transformer-primarily based architectures, to generate, optimize, and refine software program code[11]. These fashions are skilled on widespread datasets such as open-source and proprietary code, allowing them to understand styles and bring syntactically accurate and purposeful code snippets. By leveraging series-to-collection studying techniques, these AI fashions can translate natural language activates into executable code, assisting developers in a wide range of programming tasks, from code of entirety to complete-fledged software program development [12].

One of the most commonly known models for AI-driven code generation is the OpenAI Codex, which serves as the basis for GitHub Copilot. Codex is trained on a large body of published code and uses deep learning to create context-related code suggestions. Similarly, Code Gen in Code5 and Salesforce use transformer-based architectures to improve programming efficiency. Not only can these models help you write code, but they can also help you recognize errors, redesign existing code, and generate documentation. This makes the development of software faster and more efficiently.

Another category that contributes to AI is focused on applications related to the domain. For example, DeepCode specializes in the analysis and optimization of the code. This shows AI's shortcomings and improvements of disability. Polycode, on the other hand, emphasizes the adaptability of AI in many programming languages, ensuring the universality and accessibility of developers working in various technical fields. This model increases the diversity and technology of the solution to create AI control code. After producing code with AI, the researchers and developers increase and self -considering the accuracy, adaptability and ethical considerations of these models. The integration of reinforcement learning and transmission learning techniques improves the ability of AI to adapt to new coding paradigms, making AI-generated code more reliable and context-related [13]. Future progress in the production of AI control code is expected to further improve programming efficiency, improve software performance, and promote seamless human AI cooperation in software engineering [14].

III. Applications of AI in Software Development

The advent of AI-powered code generation is redefining the software engineering landscape. Automated code completion, where AI models generate code snippets based on developer input, appears to be a particularly impactful application, as it streamlines productivity by minimizing time spent on repetitive code structures and ensures syntactic accuracy. Considering these factors, AI-driven tools such as GitHub Copilot and IntelliCode have empowered developers to prioritize logic and problem-solving aspects, allowing AI to manage syntax and boilerplate code. This enables developers to concentrate on higher-value tasks, fostering innovation and efficiency.

Another transformative area of influence is bug detection and fixes, where AI-powered tools employ machine learning techniques to scrutinize code for vulnerabilities and provide targeted solutions to augment security. Notably, these tools can detect prevalent security flaws such as SQL injection, buffer overflows, and authentication issues, thereby mitigating the risk of cyberattacks. By expediting the identification and rectification of errors, automated debugging solutions enhance the reliability and robustness of software applications, effectively strengthening stakeholder alignment and operational scalability.

AI also plays a key role in code refactoring, optimizing existing code to improve efficiency and maintainability. AI-controlled refactoring tools analyze code structures, recognize layoffs, and propose optimizations that improve performance [15]. By improving code readability and modularity, AI developers support a high-quality codebase, reduce technical debt, and promote smoother collaboration between development teams.

Additionally, AI drives platform development using low-code and no-code platforms, allowing non-developers to create applications through an intuitive interface. These platforms use AI to generate function code functions based on user input, enabling companies and people with limited programming skills to develop software solutions. This democratization of software development is particularly advantageous for startups and small businesses that require customer-specific applications without the need for comprehensive coding expertise.

There are various applications of Artificial Intelligence (AI), Machine Learning (ML) and Deep

Learning (DL). Some of the performance data of these methods in advanced robotics are discussed below [16]:

1) Object Lensing: Object Lensing is an important task in robotics, which is important for autonomous navigation and manipulation. Deep learning methods such as harmful neural networks (CNNs) achieve impressive results in object recognition.
2) Movement planning: Movement planning is a key task of robotics, which includes the search for a path without hindrance for the robot in order to go from one point to another. Renegortion Learning (RL) is a powerful automated learning technique used to achieve impressive results in exercise planning. For example, we used a deep deterministic policy gradient (DDPG) algorithm to generate smooth and effective paths for the robotic manipulator.
3) Control: Control is another important task in robotics, and involves regulating robotics. Deep training (DRL) was used to achieve impressive results in control tasks. For example, the algorithm for proximal optimization of politics (PPO) was used to teach a robotic arm to understand and move objects.
4) Location: Location is the process of determining the location of the robot within the environment. Automatic learning methods, such as auxiliary vector machines (SVM) and random forests, have been used to obtain impressive results in location tasks. For example, a random method based on the forest reached 98.8% precision in the location of the robot.
5) Object Discovery: Object Discovery is the process of discovering and localizing objects in an image. The faster R-CNN and Yolo learning techniques have obtained impressive results in object detection tasks.

IV. Requirements Gathering and Analysis

Thanks to the transformation of an unstructured language into structured formats, the treatment of a natural language (NLP) transforms the extraction of the requirements [17]. The named recognition of the essence (NER) finds important components, and the preparation of the text is used to attract content. Studies of syntax and meaning improve understanding by learning language and meaning. Relation extraction helps document traceability. Automating this workflow decreases manual effort, enhances accuracy, and accelerates the process. While NLP minimizes ambiguity, challenges related to language diversity and specialized terminology remain. It simplifies the development process by ensuring a consistent and clear comprehension of requirements [18]. Sentiment analysis can be valuable as it reflects the tone of the requirements. In summary, natural language processing (NLP) is an essential resource for effective and precise requirement management [19].

Reviewing requirements based on AI will automate and improve processes and ensure system accuracy. AI evaluates the requirements of sequence, integrity, and ambiguity, thereby reducing human error [20]. The verification of traceability and automated test of test cases guarantee appropriate implementation and tests. The simulations and the verification of models facilitate early identification of defects, which reduces the need for touch - ups. Risk assessments allow you to prioritize requirements, leading to the best distribution of resources. Thanks to understanding natural language, AI includes the intent of requirements to increase verification accuracy. The advantages include greater efficiency, improved monitoring and excellent quality of the system. The problems include maintaining data quality, the guarantee of the interpretation of models, the requirement of experience in the field of the field and the management of the complexity of natural language[21]. Ultimately, AI simplifies auditing and leads to more reliable software.

V. Challenges and Ethical Considerations

One of the main challenges when it comes to AI-powered code generation is the importance of maintaining code quality and security. Unoptimized or vulnerable code could potentially introduce security threats, underscoring the necessity for developers to meticulously assess AI-generated code. By combining AI-driven validation techniques with human expertise, it is possible to reduce these risks and enhance the dependability of AI-generated code.

Intellectual property and copyright issues pose another significant concern. AI models are trained on vast datasets that include publicly available code, raising questions about the ownership of AI-generated content [22]. Developers must navigate legal and ethical considerations when using AI-generated code, ensuring compliance with licensing agreements and avoiding potential copyright infringement. Establishing clearer legal frameworks for AI-generated software will be crucial in addressing these concerns.

Ensuring bias and fairness in code generated by artificial intelligence is a major ethical issue. AI models trained on biased data sets have the potential to produce discriminatory or inaccurate code, leading to ethical and legal repercussions [23]. It is essential to guarantee that training data sets are varied and inclusive to reduce bias. Researchers and developers are obliged to integrate fairness-conscious algorithms and consistently evaluate the results of AI systems to prevent unintentional biases in software applications [24].

Finally, developer dependence and deterioration in skills are growing issues. As AI tools become more demanding, there is a risk that developers can be over generated with A-generated code generated by AI, leading to a decline in traditional programming knowledge. The promotion of a balanced approach in which AI functions as an auxiliary device as an alternative to human knowledge is of great importance to maintain a strong foundation for developers' coding skills[25].

VI. Role of Human Developers in AI-Assisted Coding

Despite AI's advancements, human developers remain essential in software development:

- Quality Assembly and Verification: Developers must strictly modify the AI code created to ensure the accuracy, safety, and compliance of the industrial standard. Human directors are important for determining errors or vulnerabilities that AI may be invisible.
- Creative solutions to problems and innovation: AI cannot develop creative and innovative solutions. Developers

need to solve problems for unique use options, algorithm design, and code optimization[26].

- AI's ethical and responsible use: human intervention is needed to comply with ethical standards, avoid prejudice, and support legal requirements. Developers must actively evaluate and relieve ethical issues related to the development of software controlled by artificial intelligence. The generated AI code contains hidden disadvantages or has no explanatory opinions. Developers play an important role in debugging, refactoring, and maintaining codebase AVOs and play an important role in improving readability and long-term stability.
- Settings and Situation Recognition: AI can create code that lacks understanding of a specific project. Developers provide knowledge about the situation needed to adjust the AI and business requirements, software architecture, and the best industries and code [27].
- Continuous education and development of technology: As AI tools are developed and developed tools are developed, developers should continue to study new technologies, adapt to changing technologies, and improve their experience in effective use of AI within the framework of the work process[28].

VII. Future prospects

The future of AI in software development holds promising advancements that will further transform the industry. Improved Context Awareness will allow AI models to understand project-specific requirements, resulting in more accurate and relevant code generation [29]. Integrating situation data, AI provides individual recommendations that correspond to developers' needs to increase the efficiency and accuracy of software development [30].

Improving cooperation with Person-Ai will stimulate the smooth partnership between AI and developers. Instead of replacing human programmers, AI acts as an intellectual assistant, helping engineers make the best coding decisions, optimize work processes and reduce errors [31]. Since the AI model continues to study the developer's interaction, it provides more intuitive sentences and creates software development with more interactive and effective processes.

Personalized assistants of artificial intelligence are designed for developers' individual preferences and provide specially developed coding tools. These AI systems are involved in the style and habits of coding developers and provide users with more accurate and specific recommendations [32]. The integration of reinforcement training enables AI coding to continue to develop and improve the ability to support developers effectively for AI coding.

Finally, the expansion of the AI -based development platform allows more professionals and companies to use AI in the work process. From intellectual IDE to automated software test tools, AI is an essential element of modern software development, increasing innovation and industrial performance [33].

VIII. Limitations

While AI-driven code generation is a promising technology, it has limitations:

- Lack of situation perception: AI often lacks a deep understanding of the requirements of nuances related to projects, business logic and domain [34]. As a result, you can create a phrase phrase, accurate but functionally inappropriate code.
- Non -government code sentence: AI production code does not always correspond to the best coding method or claimed design, which leads to the inefficient development process [35]. Changes in output quality may require additional debugging and modifications.
- Ethical problem: AI can produce unintended plagiarism or patent code without proper property. This raises legal and ethical issues, especially in software development for companies where intellectual property protection is important.
- False costs for performance: Integrating AI tools into the development process development can lead to delay and additional computing costs [36]. Continuous code generation code using A-assistance may require significant computing resources that affect the overall efficiency of the development conveyor[37].
- Difficulties in the generated code debugging: The generated AI code is often written without explicit documents or explanations, so debugging and maintaining these code can be complicated[38]. Developers can do their best to understand the justification of a particular sentence created by AI, which eliminates long problems.
- Security vulnerabilities: The code created by AI can intentionally introduce security defects such as inappropriate input or vulnerable authentication mechanisms [39]. Without the appropriate supervision, these vulnerabilities can be used by the actor.
- The adaptability to development technology development is limited. The AI model often needs to be re -educated frequently to follow new programming languages, frameworks and best practices [40]. Without continuous updates, AI -controlled code creation can eventually be old or inefficient.

In addition, the investigating how machine learning impacts the software development lifecycle through "AI-Driven Code Generation." By focusing on this specific application of AI, the paper seeks to understand its innovations, benefits, and challenges within the realm of software creation [41]. Further, the paper provide review of graphene, detailing its history, characteristics, applications, and future potential [42]. However, study on additive manufacturing, covering its methods, costs, challenges, and industrial uses, which, like AI-driven code generation, represents advancement in automated production [43]. Recently the use of computer-aided education to improve learning within the medical field, highlighting its innovative aspects, advantages, and the obstacles encountered. Similar to how computer-aided tools enhance education, AI-driven code generation aims to augment and transform the software development process [44]. Moreover AI driven code generation can be employed in electronic commerce,

cybercrime and cybersecurity, and data security, classification, and control measures within the realm of information technology [45-48].Some researchers only explored the impact of social media on student life and political campaigning, along with a study on IoT-based smart agriculture [49-51]. However, the AI-driven code generation could be used to develop tools for automated sentiment analysis of social media data related to political campaigns [52]. The underlying software and algorithms for instructional chatbots might be quickly prototyped and developed via AI-driven code creation [53, 54]. This can entail writing code for the chatbot's natural language processing, dialogue control, and personalized learning functionalities [53]. Additionally, Jindal et al. [55] provide an overview of AI's role, but AI-driven code generation is a specific application of AI that could be further explored within the context of software development as one of those "distinct sectors [56].

IX. Conclusion

AI-based code production exhausts software development, increases efficiency, and improves the quality of the code. This progress allows developers to automate repeated work, optimize code, improve safety, and ultimately optimize the software development process. Nevertheless, in order to ensure responsible AI relocation, we need to carefully control problems such as potential safety gaps, intellectual property, and ethical considerations.

After the AI model is developed, the ability to understand the complex logic of programming and to adapt to various coding requirements improves the effect. Integrating artificial intelligence management tools into the work process of software development stimulates the cooperation between human programmers and artificial intelligence systems to improve productivity and innovation. In addition, the current conditions of research and normative structures are inherent in order to determine the main principles of the ethical use of the code using the signs that have reduced risks and created AI.

The automation of AI and the balance of human directors can use artificial intelligence technology in future software development. This synergy effect is a more reliable, effective and inexpensive programming solution, which is the next intellectual development era.

References

[1] Alenezi M, Akour M. AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Applied Sciences.* 2025; 15(3):1344. https://doi.org/10.3390/app15031344
[2] Alenezi, Mamdouh, and Mohammed Akour. 2025. "AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions" *Applied Sciences* 15, no. 3: 1344. https://doi.org/10.3390/app15031344
[3] https://www.researchgate.net/publication/267711880_A_Comparative_Overview_of_the_Evolution_of_Software_Development_Models
[4] G. Fan, X. Xie, X. Zheng, Y. Liang, and P. Di, "Static code analysis in the AI era: An in-depth exploration of the concept, function, and potential of intelligent code analysis agents," arXiv preprint, arXiv:2310.08837, Oct. 13, 2023. [Online] Available: https://arxiv.org/abs/2310.08837.
[5] https://doi.org/10.1016/j.jiixd.2024.01.002. (https://www.sciencedirect.com/science/article/pii/S2949715924000027)
[6] Zviel-Girshin R. The Good and Bad of AI Tools in Novice Programming Education. *Education Sciences*. 2024; 14(10):1089.
[7] Alenezi, Mamdouh, and Mohammed Akour. 2025. "AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions" *Applied Sciences* 15, no. 3: 1344.
[8] https://doi.org/10.1016/j.bushor.2022.03.002. (https://www.sciencedirect.com/science/article/pii/S0007681322000222)
[9] conference: International Workshop on Software Engineering in 2030; November 2024; Puerto Galinàs (Brazil)
[10] Hassija, V., Chamola, V., Mahapatra, A. *et al.* Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. *Cogn Comput* 16, 45–74 (2024).
[11] Kotsiantis S, Verykios V, Tzagarakis M. AI-Assisted Programming Tasks Using Code Embeddings and Transformers. *Electronics.* 2024; 13(4):767. https://doi.org/10.3390/electronics13040767
[12] Zhang, J., Bu, H., Wen, H. *et al.* When LLMs meet cybersecurity: a systematic literature review. *Cybersecurity*,55(2025).
[13] Alenezi, M.; Akour, M. AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Appl. Sci.* 2025, *15*, 1344.
[14] Alenezi, Mamdouh, and Mohammed Akour. 2025. "AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions" *Applied Sciences* 15, no. 3: 1344.
[15] Baumgartner N, Iyenghar P, Schoemaker T, Pulvermüller E. AI-Driven Refactoring: A Pipeline for Identifying and Correcting Data Clumps in Git Repositories. *Electronics*. 2024; 13(9):1644.
[16] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN COMPUT. SCI.* 2, 160 (2021).
[17] Khurana, D., Koli, A., Khatter, K. *et al.* Natural language processing: state of the art, current trends and challenges. *Multimed Tools Appl* 82, 3713–3744 (2023).
[18] Krasadakis P, Sakkopoulos E, Verykios VS. A Survey on Challenges and Advances in Natural Language Processing with a Focus on Legal Informatics and Low-Resource Languages. *Electronics*. 2024; 13(3):648.
[19] Faccia, Alessio, Julie McDonald, and Babu George. 2024. "NLP Sentiment Analysis and Accounting Transparency: A New Era of Financial Record Keeping" *Computers* 13, no. 1: 5.
[20] Kalogiannidis S, Kalfas D, Papaevangelou O, Giannarakis G, Chatzitheodoridis F. The Role of Artificial Intelligence Technology in Predictive Risk Assessment for Business Continuity: A Case Study of Greece. *Risks*. 2024; 12(2):19.

[21] Aldoseri A, Al-Khalifa KN, Hamouda AM. Re-Thinking Data Strategy and Integration for Artificial Intelligence: Concepts, Opportunities, and Challenges. *Applied Sciences*. 2023; 13(12):7082.
[22] Al-kfairy M, Mustafa D, Kshetri N, Insiew M, Alfandi O. Ethical Challenges and Solutions of Generative AI: An Interdisciplinary Perspective. *Informatics*. 2024; 11(3):58.
[23] Ferrara E. Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies. *Sci*. 2024; 6(1):3.
[24] Ferrara, Emilio. 2024. "Fairness and Bias in Artificial Intelligence: A Brief Survey of Sources, Impacts, and Mitigation Strategies" *Sci* 6, no. 1: 3.
[25] Perifanis N-A, Kitsios F. Investigating the Influence of Artificial Intelligence on Business Value in the Digital Era of Strategy: A Literature Review. *Information*. 2023; 14(2):85.
[26] Alenezi, Mamdouh, and Mohammed Akour. 2025. "AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions" *Applied Sciences* 15, no. 3: 1344.
[27] Perifanis N-A, Kitsios F. Investigating the Influence of Artificial Intelligence on Business Value in the Digital Era of Strategy: A Literature Review. *Information*. 2023; 14(2):85.
[28] Kamalov, F., Santandreu Calonge, D., & Gurrib, I. (2023). New Era of Artificial Intelligence in Education: Towards a Sustainable Multifaceted Revolution. *Sustainability*, *15*(16), 12451.
[29] Sauvola, J., Tarkoma, S., Klemettinen, M. *et al*. Future of software development with generative AI. *Autom Softw Eng* 31, 26 (2024).
[30] Alenezi, M., & Akour, M. (2025). AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Applied Sciences*, *15*(3), 1344.
[31] Alenezi, M.; Akour, M. AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Appl. Sci.* 2025, *15*, 1344.
[32] Sajja R, Sermet Y, Cikmaz M, Cwiertny D, Demir I. Artificial Intelligence-Enabled Intelligent Assistant for Personalized and Adaptive Learning in Higher Education. *Information*. 2024; 15(10):596.
[33] Alenezi, Mamdouh, and Mohammed Akour. 2025. "AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions" *Applied Sciences* 15, no. 3: 1344.
[34] Tominc, P.; Oreški, D.; Čančer, V.; Rožman, M. Statistically Significant Differences in AI Support Levels for Project Management between SMEs and Large Enterprises. *AI* 2024, *5*, 136-157.
[35] Barenkamp, M., Rebstadt, J. & Thomas, O. Applications of AI in classical software engineering. *AI Perspect* 2, 1 (2020).
[36] Mohammad, Abdulghafour, and Brian Chirchir. 2024. "Challenges of Integrating Artificial Intelligence in Software Project Planning: A Systematic Literature Review" *Digital* 4, no. 3: 555-571.
[37] Elahi, M., Afolaranmi, S.O., Martinez Lastra, J.L. *et al*. A comprehensive literature review of the applications of AI techniques through the lifecycle of industrial equipment. *Discov Artif Intell* 3, 43 (2023).
[38] Bulla L, Midolo A, Mongiovì M, Tramontana E. EX-CODE: A Robust and Explainable Model to Detect AI-Generated Code. *Information*. 2024; 15(12):819.
[39] Humphreys, D., Koay, A., Desmond, D. *et al.* AI hype as a cyber security risk: the moral responsibility of implementing generative AI in business. *AI Ethics* 4, 791–804 (2024).
[40] Sinha, S., Lee, Y.M. Challenges with developing and deploying AI models and applications in industrial systems. *Discov Artif Intell* 4, 55 (2024).
[41] Kumar, S., Kumar, R. A Comprehensive Study on Additive Manufacturing Techniques, Machine Learning Integration, and Internet of Things-Driven Sustainability Opportunities. J. of Materi Eng and Perform (2025). https://doi.org/10.1007/s11665-025-10757-x
[42] Rai R. and Kumar S. (2024), "A Comprehensive Review on Graphene: Historical Development, Properties, Applications, Challenges and Future Scope", Journal of Management and Engineering Sciences, Vol. 1, Iss. 4 (2024) 150-166, DOI: 10.61552/JMES.2024.04.004
[43] Kumar, S., Kushawaha, M.K. and Kumar, R. (2022), "Comprehensive Study on Additive Manufacturing Process Methods, Cost Comparison, Challenges and Industrial Applications, *i-manager's Journal on Future Engineering & Technology*, Vol. 17 (4), pp. 31-44. https://doi.org/10.26634/jfet.17.4.18909
[44] Singh, C., Kumar, R. and Kumar, S. (2024), "Leveraging Computer-Aided Education for Enhanced Learning: Innovations, Benefits, and Challenges in the Medical Sector", Asian Journal of Engineering and Applied Technology, Vol.12 No.2, 2023, pp.40-49. DOI: https://doi.org/10.51983/ajeat-2023.12.2.4074
[45] Irshaad Jada, Thembekile O. Mayayise, The impact of artificial intelligence on organisational cyber security: An outcome of a systematic literature review, Data and Information Management, Volume 8, Issue 2, 2024, 100063, ISSN 2543-9251, https://doi.org/10.1016/j.dim.2023.100063.
[46] Tuli, B., Jyoti, Gautam, N. and Kumar, S. (2022), "Overview of Electronic Commerce (E-Commerce)", i-manager's Journal on Information Technology (JIT), Vol.11 (2), pp.1-14.
[47] Tuli, B., Kumar, S. and Gautam, N. (2022), "An Overview on Cyber Crime and Cyber Security", Asian Journal of Engineering and Applied Technology, Vol 11 (1), pp. 36-45. DOI: https://doi.org/10.51983/ajeat-2022.11.1.3309.
[48] Sharma, M., Jindal, H., Kumar, S. and Kumar, R. (2022), "Overview of data security, classification and control measure: A study", i-managers Journal on Information Technology, Vol. 11 (1), pp. 17-34.https://imanagerpublications.com/article/18557/13.
[49] Tuli, B., Jyoti, Gautam, N. and Kumar, S. (2022), "Impact of social media on student life", i-managers Journal on Information Technology, Vol. 11 (1), pp. 41-53.
[50] Jindal, H., Garg, Y., Kumar, S.,Gautam, N., and Kumar, R. (2021), "Social media in political campaigning: A Study", I-manager's Journal on Humanities & Social

Sciences, Vol. 16(1), 49-60. https://imanagerpublications.com/article/18266/.

[51] Jindal, H., Kaur, A., Arshita, Kumar, S*., Gautam, N. and Kumar, R. (2021), "IOT based smart agriculture: A study", I-manager's Journal on Information Technology, Vol. 9 (431-), pp. 31-38. https://doi.org/10.26634/jit.10.1.18345.

[52] Margarita Rodríguez-Ibánez, Antonio Casánez-Ventura, Félix Castejón-Mateos, Pedro-Manuel Cuenca-Jiménez, A review on sentiment analysis from social media platforms, Expert Systems with Applications, Volume 223, 2023, 119862, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2023.119862.

[53] H. Abdulla, A. M. Eltahir, S. Alwahaishi, K. Saghair, J. Platos and V. Snasel, "Chatbots Development Using Natural Language Processing: A Review," 2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC), Crete, Greece, 2022, pp. 122-128, doi: 10.1109/CSCC55931.2022.00030. keywords: {Computers;Productivity;Ecosystems;Oral communication;Linguistics;Chatbots;Planning;natural language processing;chatbot;artificial intelligence;machine linguistics;text analysis},

[54] Beenu, Jindal, H., Kumar, R., Kushawaha, M. K. and Kumar, S*. (2021), "Utilization of Chabot in an Educational System", Asian Journal of Electrical Sciences, Vol.10 (1), pp.5-13.

[55] Jindal, H., Kumar, D., Ishika, Kumar, S. and Kumar, R. (2021), "Role of Artificial Intelligence in Distinct Sector: A Study", Asian Journal of Computer Science and Technology, Vol. 10 (1), pp. 1-12. https://doi.org/10.51983/ajcst-2021.10.1.2696

[56] Alenezi, M.; Akour, M. AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Appl. Sci.* 2025, *15*, 1344. https://doi.org/10.3390/app15031344

