



Original Article

Artificial Intelligence in Software Testing: A Study on Efficiency

Rutuja Kadam

Department of Computer Science, Baburaoji Gholap College Sangvi, Pune

Manuscript ID:
IBMIIRJ -2026-030123

Submitted: 08 Dec. 2025

Revised: 12 Dec. 2025

Accepted: 07 Jan. 2026

Published: 31 Jan. 2026

ISSN: 3065-7857

Volume-3

Issue-1

Pp. 121-123

January 2026

Correspondence Address:
Rutuja Kadam
Department of Computer Science,
Baburaoji Gholap College Sangvi, Pune
Email: rutujakadam299@gmail.com



Quick Response Code:



Web: <https://ibrj.us>



DOI: 10.5281/zenodo.18950756

DOI Link:

<https://doi.org/10.5281/zenodo.18950756>



Creative Commons

Abstract

The growing complexity of modern software systems has reduced the effectiveness and scalability of traditional software testing methods. Artificial Intelligence (AI) has emerged as a promising approach for enhancing software testing by improving efficiency, accuracy, and test coverage. This paper investigates the role of AI in software testing and presents a comparative analysis between AI-based testing approaches and conventional testing techniques. AI-driven methods, including machine learning, deep learning, and predictive analytics, enable automated test case generation, intelligent defect detection, and adaptive test execution. These capabilities reduce manual effort and support continuous testing in Agile and DevOps environments. However, the adoption of AI in software testing introduces challenges related to data quality, algorithmic bias, ethical concerns, and limited transparency in AI decision-making. This study reviews existing research to analyse the benefits and limitations of AI-based testing, and identifies key research gaps, and highlights future research directions. These findings emphasise the importance of combining human expertise with AI-driven automation to achieve efficient and reliable software testing.

Keywords: Artificial Intelligence, Software Testing, AI-Based Testing, Machine Learning, Test Automation, Software Quality

Introduction

Software testing is a critical phase in the software development lifecycle because it ensures that applications function correctly, securely, and efficiently. Traditional software testing methods, including manual and rule-based automation testing, often require significant effort and time. With the rapid growth of software systems and frequent updates in modern development environments, these traditional approaches face limitations in terms of scalability and efficiency.[6] Artificial Intelligence (AI) has introduced new possibilities in software testing by enabling systems to learn from data, identify patterns, and make intelligent decisions. AI-driven testing tools can automatically generate test cases, detect defects, and prioritize high-risk areas of software. These capabilities are particularly useful in Agile and DevOps environments, where continuous integration and testing are essential.[1] This study explores how AI is transforming software testing, compares AI-driven approaches with traditional methods, and discusses the benefits and challenges associated with AI adoption. This study also emphasizes the importance of human-AI collaboration and identifies areas where further research is needed.[3]

Literature Review

Several studies have examined the application of AI in software testing and its benefits. Existing research highlights that AI-based testing improves test coverage, reduces testing time, and enhances defect detection accuracy. Machine learning techniques have been widely used for test case generation, defect prediction, and test prioritization.[2],[7] Deep learning models have been effective in handling complex software behaviors and identifying hidden patterns in large datasets.[7] Predictive analytics has been used to forecast defect-prone modules, allowing testing teams to focus on the critical components. Despite these advancements, most existing studies are review-based and lack experimental validation [8]. The literature also identifies challenges such as dependency on high-quality data, difficulty in explaining AI decisions, and ethical concerns related to biased algorithms. These limitations indicate the need for more practical, experimental, and transparent AI-driven software testing research.[5][9]

Creative Commons (CC BY-NC-SA 4.0)

This is an open access journal, and articles are distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International Public License](https://creativecommons.org/licenses/by-nc-sa/4.0/), which allows others to remix, tweak, and build upon the work noncommercially, as long as appropriate credit is given and the new creations are licensed under the identical terms.

How to cite this article:

Kadam, R. (2026). Artificial Intelligence in Software Testing: A Study on Efficiency. *InSight Bulletin: A Multidisciplinary Interlink International Research Journal*, 3(1), 121–123. <https://doi.org/10.5281/zenodo.18950756>

Methodology

1. **Machine Learning:** Machine learning uses algorithms that learn from previous test outcomes and the behaviour of the software. Based on this learning, the system can automatically create test cases, identify errors, and gradually make the testing process faster and more effective than conventional methods of testing.[2]
2. **Deep learning** involves the use of neural networks with multiple layers that can understand complex patterns by analysing large amounts of test data and the software behaviour.[8]
3. **Predictive Analytics:** Predictive analytics involves studying past test results and software data to forecast the locations where defects are most likely to occur. This approach helps testing teams focus on high-risk areas and plan testing activities. [1]
4. **Software Quality:** Software quality refers to how well a software system meets user requirements and performs its intended functions. High software quality ensures reliability, usability, efficiency, security, and maintainability. Software testing plays a key role in maintaining software quality throughout the software development life cycle. Traditional quality assurance methods rely on manual testing and predefined test cases, which are often time-consuming and less effective for complex systems [6].

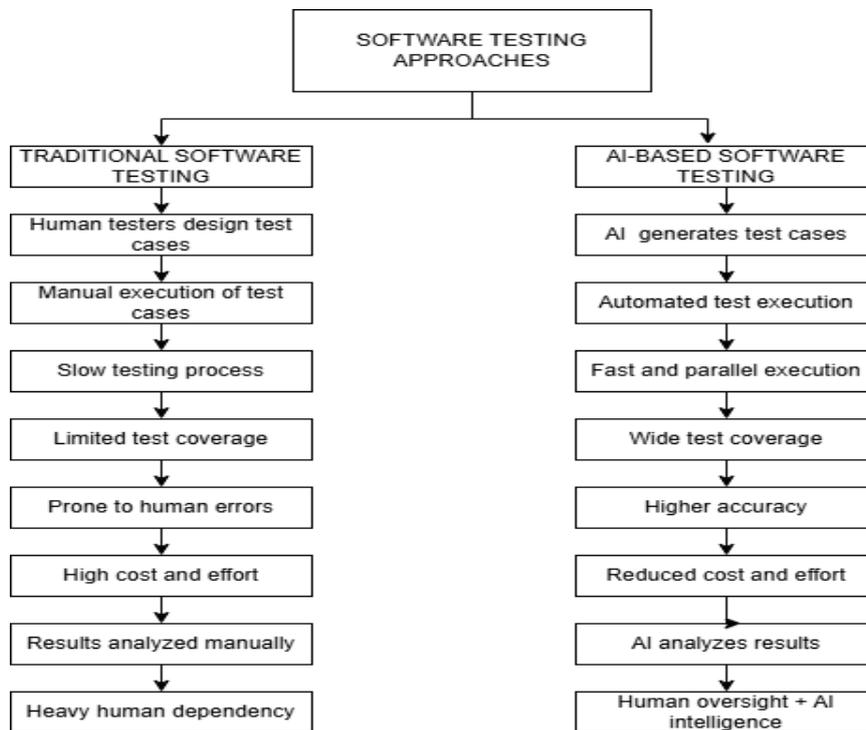
Traditional vs AI-Based Software

Traditional Software Testing

Traditional testing relies heavily on human testers to design and execute the test cases. Although it allows for creativity and contextual understanding, it is often slow, costly, and prone to human error. Manual testing also struggles to achieve comprehensive test coverage in large and complex systems.[6]

AI-Based Software Testing

AI-driven testing automates several testing activities using learning algorithms. It can automatically generate test cases, execute tests efficiently, and adapt to software changes. AI-based testing offers better scalability, faster execution, and improved accuracy than traditional methods. However, it still requires human oversight to ensure ethical and accurate decision-making.[3][4]



Research Gaps and Future Directions and Comparative Study of AI-Driven Testing Method

Existing System	AI-driven System
AI-driven test automation and defect prediction are the main focus	Efficiency, challenges, and prospects of AI in software testing
Mostly literature reviews and conceptual studies	Analytical comparison with explicit gap identification
Limited comparative discussion	Clear comparison between traditional and AI-based testing
Mainly Machine Learning and Deep Learning	ML, DL, Predictive Analytics, and NLP
Focus on Agile and DevOps-based continuous testing	Explicitly defined and categorized .

Challenges in Software Testing

- Data Quality Issues: AI models depend on large and accurate datasets.
- Algorithmic Bias: Biased data can lead to unfair or incorrect testing outcomes.
- Lack of Explainability: AI decisions are often difficult to interpret.[8]
- Ethical and Regulatory Concerns: Privacy, fairness, and accountability must be ensured.
- Integration Issues: Incorporating AI into legacy systems can be complex and challenging.[7]
Addressing these challenges is essential for the responsible adoption of AI in software testing.

Conclusion

Artificial Intelligence has significantly transformed software testing by improving its efficiency, accuracy, and test coverage. AI-driven techniques reduce manual effort and support continuous testing in modern development settings. However, challenges related to data quality, ethics, and transparency must be carefully addressed. This study highlights the importance of human-AI collaboration and identifies key research gaps that require further investigation in the future. With responsible implementation and continued research, AI has the potential to redefine software testing and quality assurance [1][8][10].

Acknowledgment

I would like to express my sincere gratitude to my faculty members of the Computer Science Department, Baburaoji Gholap College, Sangvi, Pune, for their continuous guidance, support, and encouragement throughout the completion of this study.

Financial support and sponsorship

Nil.

Conflicts of interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. Kulkarni, Y. (2024). Artificial Intelligence in Software Testing. *International Journal of Innovative Science and Research Technology*, 9(6). <https://doi.org/10.38124/ijisrt/IJISRT24JUN606> [1]
2. Panichella, A. et al. (2018). Automated Test Case Generation as a Search Problem. *IEEE Transactions on Software Engineering*, 44(10), 963–990. [2]
3. Testim.io (2023). AI-Based Testing: The Future of Test Automation. <https://www.testim.io> [3]
4. Parasoft (2022). A Practical Guide to AI in Software Testing. <https://www.parasoft.com> [4]
5. Melillo Consulting (2021). Benefits and Challenges of AI in Software Testing. <https://melilloconsulting.com> [5]
6. Garousi, V., Felderer, M., & Mäntylä, M. (2020). Guidelines for Industrial Software Testing. *IEEE Software*, 37(2), 77–87. [6]
7. Bavota, G., et al. (2019). Machine Learning for Software Engineering: A Systematic Review. *IEEE Transactions on Software Engineering*, 45(11), 1055–1076. [7]
8. ArXiv.org (2022). Artificial Intelligence in Software Testing: Opportunities and Challenges. <https://arxiv.org> [8]
9. Sculley, D., et al. (2015). Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems (NeurIPS)*. [9]
10. IEEE Computer Society (2021). Ethical Considerations in AI-Based Software Systems. IEEE Standards Association. [10]