iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.76900

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

A Survey on Modern Techniques in Software Testing

Sneha B. Sahane
Master of Computer Application, Dr.Moonje Institute of Management & Computer Studies, Nashik, India

\

Abstract: Software testing is a critical phase in the software development lifecycle that ensures the delivery of reliable, defect-free software. With the increasing complexity of software systems, traditional testing methods are no longer sufficient to meet quality requirements. This paper presents a comprehensive survey of modern techniques in software testing, including automated testing, model-based testing, mutation testing, and machine learning-aided testing. We compare these approaches in terms of effectiveness, efficiency, and applicability across different software domains. Finally, we discuss open challenges and future research directions.

Keywords: Software testing, automated testing, model-based testing, machine learning, mutation testing, test coverage.

I. INTRODUCTION

Software testing plays a vital role in verifying that software behaves according to specifications and meets user expectations. With ever-growing system complexity and reduced time-to-market pressures, more advanced testing methodologies have emerged to improve both the quality and speed of testing. Traditional manual testing is becoming expensive and ineffective in large systems; therefore, automated and intelligent testing solutions are increasingly adopted.

This paper surveys key methods in modern software testing and evaluates their pros and cons.

II. BACKGROUND AND MOTIVATION

Software testing can be broadly classified into static and dynamic testing. Static testing includes code review and static analysis, while dynamic testing involves executing code with test cases. With agile practices and CI/CD pipelines becoming mainstream, automated tools and intelligent techniques are essential to maintain quality at speed.

III. MAJOR TECHNIQUES IN SOFTWARE TESTING

A. Automated Testing

Automated testing uses scripts and tools to execute test cases without human intervention. Selenium, JUnit, and TestNG are widely used tools for automated functional and regression testing.

1) Advantages

Reduces manual effort

Enables frequent regression testing

2) Limitations

Initial setup cost

Requires skilled engineers

B. Model-Based Testing (MBT)

In MBT, tests are derived from models representing desired behavior. Models can be state machines, UML diagrams, or formal specifications.

1) Benefits

Promotes early detection of inconsistencies

Improves test coverage

2) Challenges

Model creation can be time-consuming

©IJRASET: All Rights are Reserved

C. Mutation Testing

Mutation testing introduces small changes (mutants) to the original program to evaluate the effectiveness of test cases. A strong test suite should be able to detect and eliminate mutants.

1) Insights

Provides precise evaluation of test quality
Helps identify weak test cases

D. Machine Learning in Testing

Machine learning (ML) assists in optimizing test case selection, predicting defects, and generating test data. Reinforcement learning and neural networks are increasingly used for test prioritization.

1) Examples

Test case prioritization using historical defect data
Automated generation of input data

IV. COMPARATIVE ANALYSIS

Technique	Coverage	Effort	Scalability	Cost
Automated Testing	High	Medium	High	Medium
Model-Based Testing	Very High	High	Medium	High
Mutation Testing	Very High	Very High	Low	Very High
ML-aided Testing	Medium to High	Medium	High	Medium

V. CHALLENGES AND FUTURE DIRECTIONS

Despite significant advances, several challenges persist:

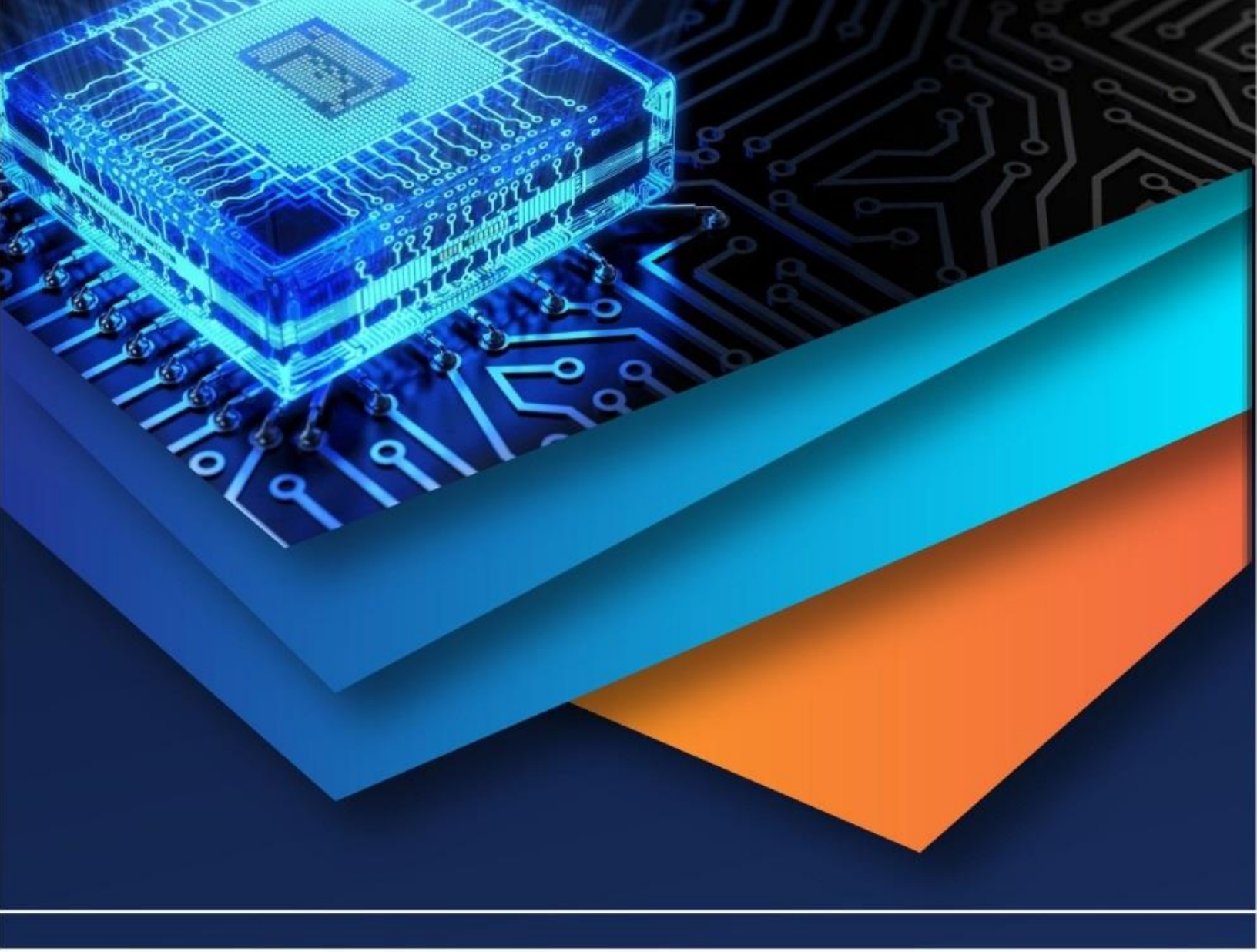
1) Test Oracle Problem: Determining expected outcomes for complex systems remains difficult.
2) High False Positives: Especially in ML based test tools.
3) Scalability: Techniques like mutation testing struggle with large codebases.

VI. CONCLUSION

Modern software testing techniques improve quality and efficiency compared to traditional methods. Automated and intelligent testing approaches are essential to cope with the evolving complexity of software. However, challenges such as test oracle issues and scalability remain open

REFERENCES

[1] G. J. Myers, C. Sandler, and T. Badgett, The Art of Software Testing, 3rd ed., Wiley, 2011.
[2] B. Beizer, Software Testing Techniques, 2nd ed., Van Nostrand Reinhold, 1990.
[3] P. Ammann and J. Offutt, Introduction to Software Testing, Cambridge University Press, 2016.
[4] A. Bertolino, “Software Testing Research: Achievements, Challenges, Dreams,” 2010 Future of Software Engineering (FOSE), 2010, pp. 85–103.
[5] L. C. Briand, Y. Labiche, and P. Omar, “A UML-based Approach to System Testing,” ACM Trans. Softw. Eng. Methodol., vol. 15, no. 1, pp. 1–38, 2006.
[6] M. Harman et al., “Search-Based Software Engineering: Trends, Techniques and Applications,” ACM Comput. Surv., vol. 45, no. 1, 2012.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)