IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, June 2025
Impact Factor: 7.67

A Review on Revolutionizing Test Case Generation: Integrating AI and ChatGPT for Enhanced Software Testing

Akash Wavhal, Aniket Gunjal, Akshay Hinge, Prof. Raut Sumedha.
Department of AI & DS Engineering,
Jaihind College of Engineering, Kuran
Akashwavhal18@gmail.com, gunjalaniket88@gmail.com,
akshayhinge45r@gmail.com, sumedharautjcoe@gmail.com

Abstract: *Using artificial intelligence (AI) and language models like ChatGPT is changing how we test software. Traditional methods of creating test cases often take a long time, can have mistakes, and might not cover everything. This paper looks at how AI and ChatGPT can solve these problems by automatically creating detailed test cases, quickly adapting to changes, and finding risky parts of the code. AI-driven testing makes regression testing more efficient, increases overall test coverage, and allows for continuous testing with self-healing features. By automating repetitive tasks and generating realistic test data, ChatGPT saves time and improves the accuracy and thoroughness of testing. The paper delves into how AI algorithms can analyze vast amounts of code and user interactions to identify potential test scenarios that might be overlooked by human testers. ChatGPT, with its advanced natural language processing capabilities, can assist in creating detailed and contextually relevant test cases based on user requirements and specifications. Furthermore, the review highlights the benefits of this integration, such as reduced testing time, improved coverage of test scenarios, and the ability to quickly adapt to changes in the software. It also addresses potential challenges, including the need for high-quality training data and the importance of maintaining the security and privacy of the software being tested*

Keywords: Artificial Intelligence (AI),ChatGPT Test Case Generation, Software Testing ,Test Automation ,AI- driven Testing

I. INTRODUCTION

In recent years, artificial intelligence technology represented by machine learning (ML) algorithms has been constantly developing and innovating, and has empowered various aspects of people's daily life. It has achieved re markable success in various tasks such as computer vision, natural language processing (NLP), speech recognition and intelligent medical care [1]. In today's fast-paced software development world, ensuring that applications are reliable and bug-free is more important than ever [2]. Traditional methods of creating test cases, which often involve a lot of manual work, can struggle to keep up with the complexity and scale of modern software [3]. This is where advanced technologies like Artificial Intelligence (AI) and Natural Language Processing (NLP) come into play, offering new ways to enhance software testing. One exciting development in this field is the use of AI models like ChatGPT for generating test cases. ChatGPT, developed by OpenAI, is a powerful language model that can understand and generate human like text[1][2]. By integrating ChatGPT into the test case generation process, we can create more intelligent, automated, and comprehensive test cases. This not only saves time and effort but also improves the accuracy and coverage of software testing.

DOI: 10.48175/IJARSCT-27348

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, June 2025
Impact Factor: 7.67

II. METHODOLOGY

The motivation behind this study is to explore the trans formative potential of AI in the realm of software testing. As software systems become increasingly complex, traditional testing methods often fall short in terms of efficiency and coverage. AI-based software testing offers a promising solution by automating and enhancing various aspects of the testing process. By understanding the current state and future possibilities of AI-based software testing, we aim to provide valuable insights that can help software development teams, small firms, and organizations make informed decisions about adopting these innovative techniques. [4] This study seeks to highlight the benefits, address the challenges, and showcase the practical applications of AI in software testing, ultimately contributing to the advancement of the field and the development of more reliable and robust software systems.

Approaches used for building System :

[1]. Define Objectives and Requirements Start by clearly defining the objectives of your system. Determine what you aim to achieve with AI-enhanced test case generation, such as improving test coverage, reducing manual effort, or increasing the accuracy of test cases.

[2]. Data Collection and Preparation Gather a diverse set of data that includes various software requirements, user stories, and existing test cases. This data will be used to train and fine tune the AI models. Ensure the data is clean, well-labelled, and representative of the scenarios you want to test

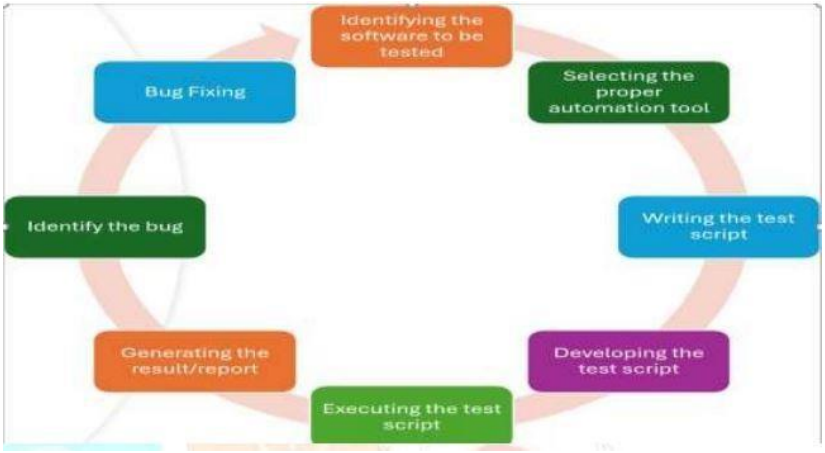
[3]. Model Selection and Training Choose appropriate AI models for different tasks. For natural language processing tasks, such as understanding requirements and generating test cases, models like GPT-4 can be highly effective. Train these models on your collected data to ensure they understand the context and can generate relevant test cases.

[4]. Integration of ChatGPT Integrate ChatGPT into your system to facilitate interactive test case generation. ChatGPT can be used to interpret user inputs, generate test cases based on natural language descriptions, and provide suggestions for improving test coverage. This integration can be achieved through APIs that allow seamless communication between ChatGPT and your testing framework.

[5]. Automated Test Case Generation Develop algorithms that leverage the trained AI models to automatically generate test cases. These algorithms should be capable of understand ing software requirements and translating them into executable test cases. Ensure the generated test cases cover a wide range of scenarios, including edge cases and potential failure points.

[6]. Validation and Debugging Implement mechanisms to validate the generated test cases. This can include running the test cases against the software to check for correctness and coverage. Use AI-based debugging tools to identify and fix issues in the test cases, ensuring they are accurate and reliable.

[7]. Continuous Learning and Improvement Set up a feed back loop where the system continuously learns from the results of executed test cases. Use this feedback to improve the AI models and the overall test case generation process.



IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, June 2025
Impact Factor: 7.67

Regularly update the models with new data and insights to keep them relevant and effective.

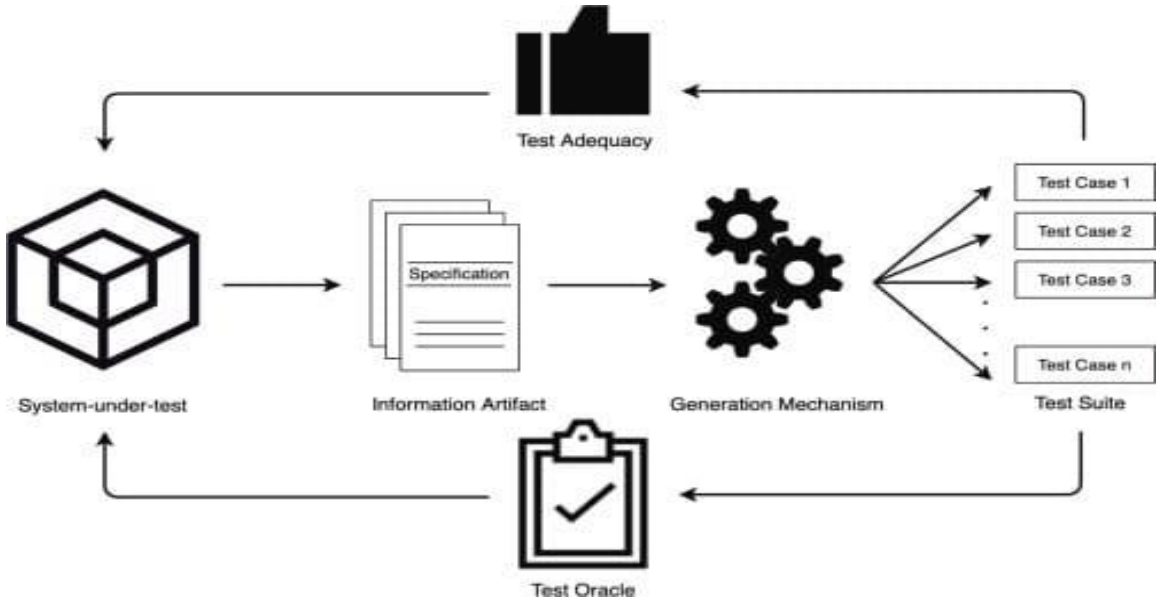
[8]. Documentation and Reporting Document the entire process, including the methodologies used, the data collected, and the results obtained. Provide detailed reports on the performance of the AI generated test cases, highlighting areas of improvement and success. Use visualizations to make the reports more accessible and understandable

III. SYSTEM ARCHITECTURE

All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right- justified.

Identifying the software to be tested: This is the initial step where the software product or feature to be tested is clearly defined. It involves understanding the requirements, scope, and objectives of the testing process

Selecting the proper automation tool: Segmentation focuses on identifying and isolating regions of interest, particularly the hands and eyes, from the processed input. In this project, the convex hull algorithm is utilized to detect and segment the hand by generating a contour that encloses the hand shape. This approach enables precise augmentation of the hand, which is essential for recognizing gestures like finger pointing or swiping. For eye tracking, segmentation isolates the eye region for further feature extraction.



Writing the test script: Test scripts are created to outline the specific actions and expected outcomes during testing. These scripts can be manual or automated, depending on the chosen approach.

Developing the test script: In the case of automated testing, the test scripts are developed using the selected automation tool. This involves coding the necessary instructions and logic to execute the test cases.

Executing the test script: Once the test scripts are ready, they are executed against the software under test. This involves running the scripts and verifying if the actual results match the expected outcomes.

Generating the result/report: The results of the test execution are captured and analyze. A comprehensive report is generated summarizing the test outcomes, including pass/fail rates, defects found, and overall test coverage.

Identifying the bug: If defects are discovered during testing, they are identified, documented, and reported to the development team for resolution

IV. CONCLUSION

AI-based software testing is revolutionizing the field by enhancing the efficiency, accuracy, and coverage of test cases. By integrating AI and machine learning, we can automate the generation, validation, and debugging of test cases,

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 1, June 2025
Impact Factor: 7.67

making the process faster and more reliable. This approach is particularly beneficial in complex scenarios like automated vehicle testing, where systematic assignment of test cases ensures comprehensive validation.

The future of software testing lies in leveraging AI to create intelligent test systems that can adapt and learn from previous tests, continuously improving their effectiveness. Tools like ChatGPT can further enhance this process by providing natural language interfaces for test case generation and analysis, making it more accessible and user-friendly.

In summary, AI-powered software testing promises a future where testing is not just a phase in development but an ongoing, intelligent process that ensures higher quality and reliability in software products

V. ACKNOWLEDGMENT

We sincerely thank the editorial board of IJARSCT for the opportunity to publish my paper, “A Review on Revolutionizing Test Case Generation. Integrating Ai and Chat gpt For Enhanced Software Testing ” I am grateful to my guide, faculty, and all researchers whose work supported this study. Special thanks to my peers and family for their constant encouragement.

REFERENCES

[1]. GAO1 , Chao SHEN1*, Weipeng JIANG1 , Chenhao LIN1 , Qian LI1 , Qian WANG2 , Qi LI3 Xiaohong GUAN1 ”Fairness in machine learning: definition, testing, debugging, and application” September 2024, Vol. 67, Iss. 9, 191201:1–191201:21 https://doi.org/10.1007/s11432- 023-4060-x

[2]. MARKUS STEIMLE 1 , NICO WEBER 2 , AND MARKUS MAURER 1.”Toward Generating Sufficiently Valid Test Case Results: A Method for Systematically Assigning Test Cases to Test Bench Configurations in a Scenario-Based Test Approach for Automated Vehicles” Digital Object Identifier 10.1109/ACCESS.2022.3141198

[3]. Mohammad Baqar (baqar22@gmail.com) Rajat Khanda (rajat.mnnit@gmail.com)”The Future of Software Testing: AI-Powered Test Case Generation and Validation”

[4]. Saquib Ali Khan1 , Nabilah Tabassum Oshin2 , Md Masum Musfique3 , Mahmuda Nizam4 , Ishtiaque Ahmed5 , Dr. Mahady Hasan6,”AI based Software Testing.”

[5]. Kishore Sugali, Chris Sprunger and Venkata N Inukollu,”SOFTWARE TESTING: ISSUES AND CHALLENGES OF ARTIFICIAL INTELLI GENCE MACHINE LEARNING.”

[6]. Mahesha Pandit 1 , Deepali Gupta 1 , Divya Anand 2 , Nitin Goyal 1,*, Hani Moaiteq Aljahdali 3 , Arturo Ortega Mansilla 4,5 , Seifedine Kadry 6 and Arun Kumar 7,”Towards Design and Feasibility Analysis of DePaaS: AI Based Global Unified Software Defect Prediction Framework .”

[7]. Muhammad Khatibsyarbini1 , Mohd Adham Isa2 , Dayang N. A. Jawawi3 , Haza Nuzly Abdull Hamed4 , Muhammad Dhiauddin Mohamed Suf f ian5,”Test Case Prioritization Using Firefly Algorithm for Software Testing.”