

SURVEY ON ARTIFICIAL INTELLIGENCE IN SOFTWARE TESTING

Prashant Hule¹

Pursuing master's degree program in
IT from Pillai HOC College of Arts,
Science, & Commerce, Rasayani.
E-mail: prashant.hule6999@gmail.com
PH - 7666888533.

Darshana Wajekar²

Asst. Professor, Department Of IT/ CS
Pillai HOC College of Arts, Science
& Commerce, Rasayani.
E-mail: darsh.wajekar90@gmail.com
PH - 7208404353.

Abstract - Artificial intelligence has significantly aided the process of the automation of different software process. Artificial Intelligence (AI) plays an important role in our life and touch base most of our surrounding applications and systems. Software testing is an important process that guarantees customer satisfaction within an application and helps in safeguarding against potential failures that may prove to be detrimental down the line. It is a planned process where the application is assessed and analyzed under certain conditions to understand the overall threshold and risks involved in its implementation. There is no other choice than to test smarter and not harder in this day and age. The usage of AI techniques not only reduces the cost but it also guarantees better quality as well as thorough testing. Testing helps to safeguard an application against potential application fail-overs which may turn out being harmful to the application and the organization later on. Software debugging completed, once an application thoroughly tested. Testing, however, is slowly transitioning to more automation to ensure maximum accuracy in the journey towards Digital Transformation. A more complex software applications have been built, time is becoming a critical factor to release applications that must be fully tested and comply with Business Requirements.

Index Terms - Artificial Intelligence, Software Testing.

1.1 INTRODUCTION

AI plays a key role in Software Testing and can get more accurate results and saves time. AI Software Testing will reduce time to market and will increase the efficiency of the organization to produce more sophisticated software and will create smarter automated testing. The application of AI in software testing tools is focused on making the software development lifecycle easier. Through the application of reasoning, problem solving, and, in some cases, machine learning, AI can be used to help automate and reduce the amount of mundane and tedious tasks in development and testing. AI software testing refers to diverse quality testing activities for AI-based software systems using well-defined quality validation models, methods, and tools. AI major objective is to validate system functions and features developed based machine learning models, techniques and technologies. AI software testing includes the goals such as Establish AI function quality testing requirements and assessment criteria, Detect AI function issues, limitations, and quantitative and quality problems, Gain the quality confidence of AI functional features developed based on

AI techniques and machine learning models, Evaluate AI system quality against well-established quality requirements and standards.

1.2 USE OF AI IN SOFTWARE TESTING

- **Unit tests** - Unit testing is very important to make sure that the build is stable and testable. With AI- powered unit test tools like RPA, a developer can get reduce the flaky test cases and maintenance of unit tests.
- **API testing**- API testing saves time and effort by getting into the root cause of the issue. The problem with UI tests is that they are not reliable anymore as UI keeps changing in agile, while API tests give a deeper insight into the application and directly hit the root cause of an issue eventually making the application more robust.

There are many tools which are using artificial intelligence to help take the complexity out of API testing by converting manual UI tests into auto- mated API tests, lowering the technical skills re- quired to adopt API testing and helping organiza- tions build a comprehensive API testing strategy that scales.

- **UI testing** - The first step in automation is to convert manual UI tests into automated tests. There are tools which leverage AI to run the test cases on multiple platforms and browsers and also learn from the functional flow, reducing the maintenance effort and making testing more reliable.

1.3 TOOLS

AI Powered testing tools - There are various testing tools which are using AI, though not harnessing the best of AI, they are still able to help testers a lot-

- **Applitools** - It is an AI-powered visual testing and monitoring tool that can run tests on different browsers and platforms. It uses AI to identify the meaningful changes in UI and also identify them as bugs/ desired changes. It also leverages ML/AI-based for automated maintenance (being able to group together similar groups of changes from different pages/browsers/devices)
- **Testim** - It leverages machine learning into the most critical part of automation which is execution and maintenance of tests.
- **Sealights** - Sealights uses AI and machine learning to analyze the code and run tests which cover the impacted area. It can be any kind of test- unit, functional, performance, manual, etc.
It provides a useful insight ‘Quality Risks’ which focuses user efforts on the things that matter by letting him or she know exactly which files/methods/lines have changed in the last build that wasn’t tested by a specific test type (or any test type).
- **Test.AI** - Test.AI is building as a tool that will add an AI brain to Selenium and Appium. It was created by Jason Arbon, co-author of How Google Tests Software and the founder of Appdiff. Tests are defined in a simple format similar to the BDD syntax of Cucumber, so it requires no code and no need to mess with element identifiers.
AI just like a real person, IDENTIFIES the screens and elements in your app. AI EXECUTES user scenarios — test on-demand whenever you’re ready. AI RECOGNIZES elements so that even if things change, your test doesn’t break.
- **MABL** - Like the other AI-based test automation tools, MABL can automatically detect whether elements of your application have changed, and dynamically updates the tests to compensate for those

changes. You just need to show the workflow that has to be tested and MABL does the rest.

- **Retest**- Retest propagates an innovative testing approach, which is a combination of “intelligent” monkey testing and “difference testing” and works actually more like a GUI version management than

conventional testing.

This tool does Monkey testing whereby the monkey (is called Surili) is artificially intelligent and can be trained by users by capturing user actions.

- **ReportPortal** - ReportPortal, as the name suggests, is an AI-powered automation tool which focuses more on report analysis and management. As per its website it-

Manage all your automation results and reports in one place & Make automation results analysis actionable & collaborative, Establish fast traceability with defect management, Accelerate routine results analysis, Visualize metrics and analytics Make smarter decisions together

- **Functionize** - Functionize provides an overall solution for seamless automation with less/no efforts in maintenance all with the help of AI. Its AEA tool finds and fixes the broken test scripts thus eliminating the manual maintenance.

Functionize uses machine learning for functional testing and is very similar to other tools in the market regarding its capabilities such as being able to create tests quickly (without scripts), execute multiple tests in minutes, and carry out in-depth analyses. It also gives scalability to test suites by maintaining them in the functionize test cloud.

presents storage and analytics challenges in addition to creating noisy datasets

- AI systems rely on data gathered during unanticipated events which is extremely difficult to collate thus posing training challenges

1.4 CHALLENGES OF AI IN TESTING

- The number of possible solutions involving AI technologies. Testing these systems requires a tailored approach, adjusted for each specific bespoke scenario and client need.
- The amount of data required to test the system. Limited data items will not provide statistical assurance of the system.
- The question of what kind of skills should a tester have, and how should they interact with these systems of that complexity level.
- AI model test scenarios should be equipped to identify and remove human bias which often becomes part of training and testing datasets.
- Massive volumes of collected sensor data

2.1 THE ROLE OF AI IN SOFTWARE TESTING

The rise of test automation has coincided with the adoption of agile methodologies in software development. This enables teams to deliver robust and bug-free software in small batches. Manual testing is limited to business acceptance testing only. Test Automation along with DevOps helps agile teams to ship a fail-safe product for SaaS/ cloud deployment via a CI/ CD pipeline.

In software testing, AI is a combination of cognitive automation, reasoning, machine learning, natural language processing and analytics. Cognitive automation leverages various technological approaches like text analytics, semantic technology, data mining, natural language processing and machine learning. For example, RPA (Robotic Process Automation) is one such connecting link between Cognitive Computing and AI.

With the advent of AI, the drive for its use in software development, specifically in software testing, is only natural. Although the role of AI in testing is still in its infancy, thought leaders in this community are talking about self-generating, self-executing, and self-adapting testing frameworks. So, let's see how AI has changed the traditional way of testing software. Automating Visual Validation. A method of testing that is getting more and more popular every day is image-based testing using automated visual validation tools. There are many ML-based visual validation tools that can detect minor UI anomalies that human eyes are likely to miss.

The main purpose of UI testing is to make sure that each UI element looks good with the right shape, color, size and position and is not physically overlapping with other UI elements. All of these visual bugs can be checked by even a simple ML test without a tester having to intervene. Automatically Writing Test Cases

The biggest application of ML/AI in test automation has been in automatically writing test cases for software. In earlier days, we have heard about web crawlers and "spidering" (browsing a software/ web in an automated and methodical manner using an automated script or program) which helped us find 404 dead pages. Now, AI/ML tools have gone far ahead to learn the business usage scenarios of the application under test. They just need to be pointed to the software. While learning the application, they automatically crawl and collect useful data like screenshots, HTML pages and page loading time. Over time they

collect enough data from the application so that they can train the ML model for expected patterns of the app. When they are run/ executed, the current state of the app is compared with the known or saved patterns. If there is any error, visual difference, slow run time or similar issue, then the system automatically marks it as a potential issue. However, in some cases the differences might be valid. In that

case, the tester needs to validate the bug or issue.

Improving Reliability:

Are you one of those whose UFT or Selenium tests fail because of small changes to the application (like re-naming or resizing a field) made by the developers? If yes, then don't worry, this is a problem most testers face. Now AI can correct the code and make it more reliable and maintainable so that you don't have to change the test each time developers make a small change.

AI/ ML tools can read the changes made to the application and understand the relationship between them.

Such self-healing scripts observe changes in the application and start learning the pattern of changes and then can identify a change at runtime without you having to do anything. As the app evolves the ML scripts adjust automatically, reducing flakiness and fragility of test automation.

Reduced UI-based Testing:

Another change brought by AI/ML to automation testing is automation without user interface. Non-functional tests like Unit Integration, performance, security and vulnerability are also no exception. AI/ML based techniques can be applied for generating tests in these layers. In addition, AI/ML applied on various application logs like source code and production monitoring system logs, helps in developing bug prediction, early notification, self-healing and auto scaling capabilities in the overall software eco-system.

AI-based testing reduces overall testing cost, error, time and scripting. Isn't it exactly what we wish for? There is no doubt that AI and ML are game changers in the software industry and therefore it will become a trend in the market soon. Its high time software teams move towards an AI-based approach for Software Development, Testing and Management.

Application of artificial intelligence techniques in Engineering and testing of the software is a progressive area of research that leads to the cross-fertilization of ideas in the middle of the two fields. Applications of artificial intelligence methods in software testing have

been reported in several finished works.

3.1 MAIN ADVANTAGES OF AUTOMATED SOFTWARE TESTING

Following are the major advantages of automated software testing:

- Reduce software development and testing cycles, allowing the product to market faster.
- High test efficiency, make full use of hardware

IJSER

resources.

- Save manpower resources, reduce cost of test.
- Enhance the stability and reliability of the test.
- Improve the accuracy and precision of software testing, software increased confidence.
- Software testing tools to make the test work is relatively easy, but it can produce higher quality results.
- Cannot be done by hand, automated testing can do, such as load and performance testing. As mentioned above, there are many advantages of software testing automation that can bring very significant benefits, but under the current circumstances, automated testing software cannot solve all problems, but also has the following limitations or problems.
- Cannot replace manual testing.
- Manual testing found defects more than automated testing.
- Greatly dependent on the quality test.
- Test automation may restrict software development.
- The tool itself has no imagination, and cannot take the initiative to find defects. In addition, manual testing is superior to the other aspects of the test tool is that it can deal with unexpected events. Although the tool is also able to handle part of the unusual event, but there are real emergencies that cannot be solved by automated software testing. Typically, when the software development process exhibits the following conditions, it is necessary consider introducing of automated test:
 - Test cases generation: including test inputs, test outputs, test operation instructions, etc.
 - Execution and control tests.

advantages being offered by artificial intelligence domain when applied in software testing.

4.1 CONCLUSION

In this research paper various factors responsible for the classification of techniques have been reviewed and discussed. This paper has also evaluated IoT implementation techniques along with Artificial Intelligence technique. The test cases are generated for various applications to check their functionalities. Genetic algorithm is one of the most efficient algorithms that can be used for test case generation. This position paper presents the case of using artificial intelligence techniques for automation of the process of testing. Paper gives the brief description of tools, ad-

5.1 REFERENCES

1. International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7 Issue-4S2, December 2018 - Software Testing Techniques with Artificial Intelligence in Iot Applications.
2. The 9th International Conference on Computer Science & Education (ICCSE 2014) August 22-24, 2014. Vancouver, Canada.
3. 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE) - What is AI Software Testing? and Why

IJSER