Artificial Intelligence in Software Test Automation Tools and Technologies (Survey)

Arifa Shamim, Mehak Abbas, Saifullah Adnan
Department of CS & SE, Jinnah University for Women, Karachi, Pakistan
*E-mail: arifashamim254@gmail.com

ABSTRACT

Smart devices and applications are being widely used in this modern era of technology. Human beings are crowded and engaged with various smart applications. Pace of the development of smart applications and devices is very fast, technologies are rapidly evolving day by day. Along with the advancement of technologies their proper testing must be done. Quality of smart devices and applications must be ensured. Initially, it was done manually with a high rate of human intervention. But, nowadays instead of manual testing, automation testing is a main focus. For this purpose, Artificial Intelligence is used which can automate the testing process, make it fast with a lesser rate of human involvement. The purpose of this survey is to study that how artificial intelligence can be used for software test automation and which tools and technologies are more suitable for this purpose.

Keywords: Test Automation; Artificial Intelligence; Machine Learning; self-adaptive systems; end-to-end; web testing.

INTRODUCTION

Development of innovative and smart software applications is being highly trending since many years. But only the development of such software(s) is not enough. It is highly essential to ensure the proper quality of software products. For this purpose, software testing has to be done. Since past several years, it has been experienced that software testing is done manually with an involvement of a complete human intervention. But in this modern era of technology, test automation is highly recommended. Software test automation can accelerate process and reduce the cost and minimize human intervention which ultimately increased the productivity (Serna M. et al., 2019).

Artificial intelligence (AI) and Machine Learning (ML) are self-adaptive systems. In this study, we will discuss that how Artificial Intelligence is used in software test automation. Automated software testing has significant issues that will be thoroughly covered in this study. These issues include premature human involvement and lack of intelligence and it won't be possible to spot test errors, code problems, or other significant challenges in the testing environment. When it comes to testing, Artificial Intelligence enables users to overcome their present issues and increase their efficiency. Instead of prioritizing the feature's testing, usually the teams just assign that task to someone who is available at the moment to test effectively (Sugali et al., 2021).

LITERATURE REVIEW

An Overview of Artificial Intelligence for Software Testing

Software testing is used to ensure the customer satisfaction and automated test is carried out in some specified situation and enabling testers to evaluate the threshold and potential risks connected with the deployment of the software. In software testing, Artificial Intelligence (AI) helps to prevent applications from failure. This can be more expensive for both the programmer and the company but helps to improve the quality of AI

products. Moreover, the testing by using Artificial intelligence (AI) is more crucial in this era because AI is incorporated into various aspects of our daily life (Serna M. et al., 2019).

How is Artificial Intelligence Shaping the Dynamics of Software Testing?

Artificial intelligence is commonly used these days, to secure applications and we might be handling over a lot of testing for it. Hence, we're moving away from human-driven testing and stepping forward towards a scenario where robots run test scripts instead of humans.

Automated testing by using machine learning techniques require limited amount of human provided inputs. Hence, the creation of a group that focused on the Grand Dream of Testing has become critical, where technology offer superior testing to current application testing teams. Everything is automated with no human involvement by introducing smart software testing tools as shown in Figure 1. Consider going one step further and visualizing an environment where software can test, diagnose and cure itself on its own (Trudova et al., 2020).

Artificial Intelligence Enhanced Software Testing Tools

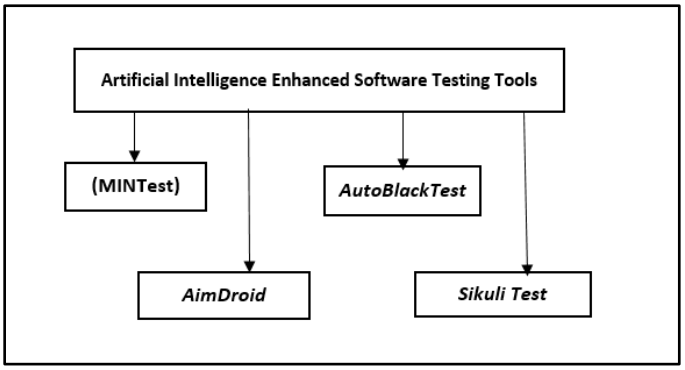

Figure 1. (Software Testing Tools).

Model-Inference Driven Testing

(MINTest) is a tool for automating software testing and it may be used to create test cases by using the C4.5 technique and implemented for the Linux operating system (OS). On its website, it promotes itself as a framework for unit and integration testing (Trudova et al., 2020).

AutoBlackTest

(Automated Black-Box Testing) uses supervised learning approach known as QLearning. The main objective of the tool is to automatically generate GUI test cases for interactive applications. According to GitHub, it is only compatible with Windows versions [7, 8] of IBM Rational Functional Tester.

AimDroid

Google developed AimDroid, a GUI testing environment for Android applications. It implements reinforcement learning guided random approach. The tool allows users to run tests and receives reported results. By eliminating activity transitions, each episode of the tests that this tool creates generates a finite amount of events and concentrates on a single activity. The Major drawback with AimDroid is that the smartphone must be rooted. To fix this GUI breakage by Vista, it can recover test scripts that malfunction on a later version of an application by comparing their present state to the previously documented state (Yasin et al., 2021).

Sikuli Test:

Another Automated tool that can be used for GUI tests is silkuli. It is based on an effective algorithm that enables the usage of the visual notation. It can be compatible with any operating system. It can be used to test mobile (Android) apps, desktop computers, and web services.

Morphy:

Morphy is another testing tool used to test all types of software's including AI applications and demonstrates in the form of case studies and provides advanced test automation facilities such as testing strategies and test scripting. Its architecture enables several testing tools and techniques to be unified by wrapping existing testing tools as method in test specification class that invoke the tools (Zhu et al., 2019).

Automated End to End Web Testing

It has been observed that the use of Web applications is very common since several years and their proper end-to-end testing is to be done which is considered as a black box testing due to the involvement of several steps including the insertion of username and password, clicking the login button, etc. (Leotta et al., 2016) Suggested approaches and tools for end-to-end web testing. Test automation is time saving and is very crucial to deliver quality product with less number of defects as per the expectations of users. Test automation results in the fast execution of set of tests. Various approaches have been suggested for end-to-end web testing and their selection is dependent upon some factors for instance: the tools and the cost required undergoing test automation.

Web browsers used by the users are crowded with various web applications on a daily basis. The frequent usage of web applications emphasizes upon the quality of these applications. Hence, end-to-end quality testing of web applications must be ensured. It has been observed that proper end-to-end quality testing requires more time and cost, due to which it is quite challenging for developing companies to completely perform end-to-end web testing. Manual testing of web applications was in practice since many years but it was more time consuming and failed to remove cent percent errors. Therefore, the developing teams decided to automate this testing process by using test automation tools and approaches (Leotta et al., 2016) as shown in Figure 2.

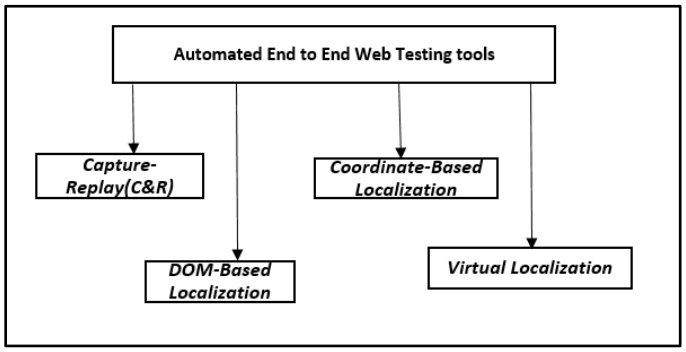

Figure 2. (End-to-End Web Testing Tools).

Capture-Replay (C&R)

It involves the recording of all actions performed by the tester on the GUI of web application which ultimately generates a test script that can repeat all those actions without human involvement.

Programmable

Web testing combines both traditional and web testing. Developers write automated test scripts themselves by using particular testing frameworks. These frameworks enable developers to create an interaction with web

pages and its elements. Test scripts created by developers can automatically fill and submit forms or click on links, hyperlinks, buttons, etc.

Coordinate-Based Localization

Screen coordinates of web page elements are to be recorded by the tools which device this approach and later on can be used to trace the elements during test case replay. But this approach is now considered as an obsoleted one as the test scripts created by this approach are exceptionally insubstantial.

DOM-Based Localization

Tools including Selenium IDE and Selenium Web Driver device this approach. These tools locate and suggest ways to locate the elements of a web page on the basis of information present in Document Object Model (DOM).

Visual Localization

Recently developed tools including Sikuli IDE3 and Sikuli API4 are implementing this approach by using image recognition techniques for the identification and controlling of GUI components.

Artificial Intelligence help to Reduce Issues of Test Automation

In the past few years, many different software testing tools have been created and made available in the market. Additionally, there has been a considerable advancement in the practice of software test automation (TA), from record-and-replay methods to support automated testing of graphical user interfaces to unit test framework (Ricca et al., 2021). Test Automation is used to enable a number of testing activities across numerous software products, such as online or mobile apps, including automated code analysis, unit testing, acceptability testing, and performance testing. However, the limitation of TA tool such as selenium was discovered by developers who wanted to use them to create sophisticated test suites. First, these tools still demand programming competence and non-trivial testing knowledge as they provide limited support in the generation of high-quality test code. For instance, testers still spend the majority of their time manually creating robust locators or deterministic test scripts (Li et al., 2020). Second, in case of constantly evolving requirements and software evolution, the technological constraints affecting such automated tests. Test maintenance is still a laborious and time-consuming task because the level of automation of existing tools in this respect is either very restricted or completely absent (Leotta et al., 2016).

Artificial Intelligence (AI) lead the promise of changing the way TA is performed. It influences the whole testing phase by facilitating automated testing activities such as test planning, authoring, development, and maintenance by producing quality use of statistical methods to enhance testers' tasks. The arrival of AI has opened up a wide range of possibilities for solving problems, from the automatic creation of test cases to their automatic repair. AI based automated testing considerably faster than traditional automated testing as testers save time to generating code and allows companies to enhance their ability to run test and deploy quickly (Ricca et al.).

OBJECTIVES

The main objective behind this literature survey is to identify the role of AI in software test automation along with the study of automated software testing issues. This study will throw an ample light on the role of AI to minimize test automation issues.

METHODOLOGY

For this literature survey, methodology which authors have followed was based on searching, exploring, reading and gathering points or evidences. Authors searched and thoroughly read multiple articles and research

papers related to the use of Artificial Intelligence and Machine Learning for test automation process and how the use of AI can reduces the issues of test automation.

CONCLUSION

Software test automation is in great demand since recent years. Artificial intelligence based tools and technologies are serving their level best for this purpose. From this literature survey we have studied and analyzed the performance of AI based software test automation tools and technologies. MINtest can perform best as a framework for unit and integration testing. Automated Black-Box Testing has some compatibility issues as it is only compatible with Windows 7 and 8. AimDroid initially showed rooting issues i.e. smart phones are needed to be rooted but later on this issue was resolved. Coordinate-Based Localization which was previously used for web testing has been obsoleted because it creates fragile test scripts. Among all software and web testing tools, the one which was found to be more effective is sikuli and it has no compatibility issues. It can be used for the testing of android applications, web services and desktop computers.

REFERENCES

1. Leotta, M., Clerissi, D., Ricca, F., & Tonella, P. (2016). Approaches and Tools for Automated End-to-End Web Testing. In Advances in Computers (1st ed., Vol. 101). Elsevier Inc. https://doi.org/10.1016/bs.adcom.2015.11.007
2. Li, J. J., Ulrich, A., Bai, X., & Bertolino, A. (2020). Advances in test automation for software with special focus on artificial intelligence and machine learning. Software Quality Journal, 28(1), 245–248. https://doi.org/10.1007/s11219-019-09472-3
3. Ricca, F., Marchetto, A., & Stocco, A. (2021). AI-based test automation: A grey literature analysis. Proceedings - 2021 IEEE 14th International Conference on Software Testing, Verification and Validation Workshops, ICSTW 2021, 263–270. https://doi.org/10.1109/ICSTW52544.2021.00051
4. Ricca, F., & Stocco, A. (2021). Web Test Automation: Insights from the Grey Literature. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12607 LNCS, 472–485. https://doi.org/10.1007/978-3-030-67731-2_35
5. Serna M., E., Acevedo M., E., & Serna A., A. (2019). Integration of properties of virtual reality, artificial neural networks, and artificial intelligence in the automation of software tests: A review. Journal of Software: Evolution and Process, 31(7), 1–12. https://doi.org/10.1002/smr.2159
6. Sugali, K., Sprunger, C., & N Inukollu, V. (2021). Software Testing: Issues and Challenges of Artificial Intelligence & Machine Learning. International Journal of Artificial Intelligence & Applications, 12(1), 101–112. https://doi.org/10.5121/ijaia.2021.12107
7. Trudova, A., Dolezel, M., & Buchalcevova, A. (2020). Artificial intelligence in software test automation: A systematic literature review. ENASE 2020 - Proceedings of the 15th International Conference on Evaluation of Novel Approaches to Software Engineering, 6(12), 181–192. https://doi.org/10.5220/0009417801810192
8. Yasin, H. N., Hamid, S. H. A., & Yusof, R. J. R. (2021). Droidbotx: Test case generation tool for android applications using q-learning. Symmetry, 13(2), 1–30. https://doi.org/10.3390/sym13020310
9. Zhu, H., Bayley, I., Liu, D., & Zheng, X. (2019). Morphy: A Datamorphic Software Test Automation Tool.