Global Scientific JOURNALS

GSJ: Volume 9, Issue 9, September 2021, Online: ISSN 2320-9186

www.globalscientificjournal.com

TESTING SOFTWARE SYSTEMS USING ARTIFICIAL INTELLIGENCE COMPONENTS: A COMPARATIVE STUDY

Sam Gildas HUMURIZA[1], Dr. Papias NIYIGENA[1, 2]

[1]University of Lay Adventists of Kigali

[2]Rwanda Coding Academy

ABSTRACT

With the growing complexity of today's software applications injunction combined with the increasing competitive pressure has pushed the quality assurance of developed software towards new heights. After the development of the code, it is mandatory to test the software to identify all the errors, and they must be debugged before the release of the software; the achievements of other researchers were reviewed and assessed to see if any gap found, the new researchers we have a must of contributing to that gap and provide the right and constructive feedback and solutions to them. This research is about to determine which is the best technique to be used when testing the software as defined in research objectives.

The main objective of this study is to compare Manual software testing techniques with AI-based (automated) techniques, by examining their performance on software testing just to help any person interested in software testing to get an appropriate technique to use. For understanding the current situation on software testing, data extracted from www.kaggle.com was used to form a dataset with 350 instances; Matplotlib library for Python is used for data plotting to facilitate its visualization. Findings indicate that Manual testing techniques are preferred highly than Automated testing techniques by 87% of software testers and in different testing cases such as User Interface and APIs testing cases with 49% while 20% only prefer to use Automated techniques for Security and Compatibility testing cases.

Keywords: *Software testing, AI techniques, data analysis, Python*

INTRODUCTION

Do you think that the manufacturer of Mercedes Benz has produced it in one as a whole car? A car manufacturer does not produce the car as a whole car; each component of the car is manufactured separately, like seats, steering, mirror, brake, cable, engine, car frame, wheels, etc. After manufacturing each item, it is tested independently whether it is working the way it is supposed to work and that is called Unit testing (STH, 2021).

Now, for that car, when each part is assembled with another part, that assembled combination is checked if assembling has not produced any side effect to the functionality of each component and whether both components are working together as expected and that is called integration testing.

According to the ANSI/IEEE 1059 standard, the right definition of testing is that it is the process of analyzing a software item to detect the differences between existing and required conditions, defects/errors/bugs, and to evaluate the features of the software item. The purpose of testing is verification, validation, and error detection to find problems and the purpose of finding those problems is to get them fixed (Komal, 2018).

The growing complexity of today's software applications injunction combined with the increasing competitive pressure has pushed the quality assurance of developed software towards new heights (Arun, 2020). After the development of the code, it is mandatory to test the software to identify all the errors, and they must be debugged before the release of the software (Nahid, A. & Susmita, K., 2019).

Testing is one of the most frequently used techniques in practice to assure the quality and reliability of software systems. It is used not only during the development of such systems but also during their operation (Husnu, Y. et al, 2019).

Today, many machine learning models and artificial technologies have been developed to build smart application systems based on multimedia inputs to achieve intelligent functional features, such as recommendation, object detection, classification, and prediction, natural language processing, and translation, and so on. This brings strong demand in quality validation and assurance for AI software systems (Jerry, G. et al, 2019).

Simply saying, software testing is an inevitable part of the Software Development Lifecycle, and keeping in line with its criticality in the pre and post-development process makes it something that should be catered with enhanced and efficient methodologies and techniques. This current study will focus on comparing the use of Artificial Intelligence components for testing software

quality by assessing the goodness of UI testing, Testing APIs, Performance Testing, Security Testing, and Compatibility Testing.

This study main duty is associated with the fulfillment of the following objectives;

1) Review the current software testing techniques preferences by testers.
2) Study both the manual and automated software testing via testing cases

All these objectives will be assessed and achieved via the analysis to be carried using selected methodologies and tools as described below.

RELATED WORKS

The following are the reviewed researches;

Software Testing Techniques with Artificial Intelligence in IoT Applications (Sathyavathy, V. and Shanmuga, P. D. , 2018)*;* In their research paper, authors argue that the main goals of Internet of things are control, management and co-ordination of various fields in a comfortable, effective and secure way. Software development companies in India use AI testing, Testing is carried out in all the aspects of software development lifecycle where various Categories of Embedded system using IoT application are required before testing, the black box testing is carried out on the application that checks the input and output of the system in order to compare the safety critical character of the embedded system. The test cases are generated for various applications to check their functionalities; also, Genetic algorithm is one of the most efficient algorithms that can be used for test case generation.

A comparative study of Artificial Neural Networks and Info-Fuzzy Networks as automated oracles in software testing (Deepam, A. et al, 2012); from their observations they concluded that software quality is one of the main concerns of software users. Hence, software testing is an utterly important phase in the software development life cycle. Nevertheless, manual evaluation of program compliance with its specification may be prohibitively time consuming. As a remedy, several software testing systems are using an automatic oracle to confirm that the developed software complies with its specification and determine whether a given test case exposes faults. The use of Artificial Neural Networks and InfoFuzzy Networks as automated oracles has been explored.

The use of pattern recognition techniques for automated black-box testing is an established and ongoing area of research. This paper has concentrated on comparing two of these methods; Artificial Neural Networks and InfoFuzzy Networks; ROC analysis, training time, and dispersion

analysis are used to compare the two approaches. However, the IFN clearly outperforms the ANN with respect to training time. Nevertheless, more experiments can be done to assess the utility of automated oracles based on alternative supervised and unsupervised clustering methods.

Artificial Intelligence Applied to Software Testing: A Literature Review (Rui, L. et al, 2020); In the last few years Artificial Intelligence (AI) algorithms and Machine Learning (ML) approaches have been successfully applied in real-world scenarios like commerce, industry and digital services, but they are not a widespread reality in Software Testing. Due to the complexity of software testing, most of the work of AI/ML applied to it is still academic.

The analysis procedure of the AI and ML methods used for this purpose during the last three years is based on the Scopus Elsevier, web of Science and Google Scholar databases. Algorithms used in software testing have been grouped by test types. After all they were brought to a conclusion stating that the black-box testing is, by far, the preferred method of software testing, when AI is applied, and all three methods of ML (supervised, unsupervised and reinforcement) are commonly used in black-box testing being the "clustering" technique, Artificial Neural Networks and Genetic Algorithms applied to "fuzzing" and regression testing.

In general, you see that various researches were conducted with aim of improving the way software testing is carried out and even comparing different techniques used currently in software testing with newly developed software testing techniques and methods; in my research I will focus on comparing those currently used methods and Artificial Intelligence components for testing software quality by assessing UI testing, Testing APIs, Performance Testing, Security Testing, and Compatibility Testing.

METHODOLOGY

Research approaches

This study is analytical and empirical research at once, where results are expected to be drawn from information to be generated from data analysis and literature, which means the research approach to be followed is a quantitative one.

Data collection

Data in use are secondary data from existing textbooks, research papers, datasets, and thesis, supported by data collected from the www.kaggle.com database to form a dataset.

Data analysis

This study is scientific and analytical research that means it is a necessity to present and explain each step to be carried out when working on the research objectives which will help to respond to its questions as well; main steps such as assess and understanding the research problem, selecting tools, data collection, data, data visualization, and interpretation and results in discussion.

Step 1 – Assess and understanding the research problem; The problem to be solved by this research has to be assessed and understood particularly so that other following procedures would be reasonable and well-conducted to help in research objectives achievement.

Step 2 – Selecting tools; after understanding the problem to be solved by the research, it is necessary to select tools that will help in data collection and analysis even those to be used in experiments. Matplotlib, a python library is used in results presentation under Anaconda software.

Step 3 – Data collection; For understanding the current situation on software testing, data is collected to be analyzed and extract insights data so that we become to conclude with appropriate factors; Data in use in this study are collected from the www.kaggle.com database to form a dataset.

Step 4 – Data preparation; When data are collected in any way, they have been harmonized regarding the time and objectives of the research.

Step 5 – Data visualization; Data scientists when analyzing their data just to see how are insights from data significant to the problem studied, use different ways, and in this research Matplotlib library for Python environment is used for dataset examination and visualization in such way data will be plotted for comparison between the software testing techniques, manual usually used and AI-based ones.

Step 6 – Interpretation and results from the discussion; From previous phases, after visualizing data and compare both techniques, then make decisions on which techniques among different studied ones are best than others even can be recommended to the users for testing their software.

Through the above steps, the following results were obtained and the analysis is carried out to achieve the set objectives for this study.

RESULTS AND DISCUSSIONS

As introduced above, software testing is important for each developed software before it is deployed to the users or generally on the market for being used; to carry out this activity there are various techniques used including manual and automated ones, this research is assessing currently used manual techniques to carry out their software testing comparatively to AI-based techniques which mainly rely on automation of the processes; the collected data is about the survey conducted on 350 testers responding on the Experience of testers, their usually used methods and techniques of testing, and identified test-cases per each used technique.

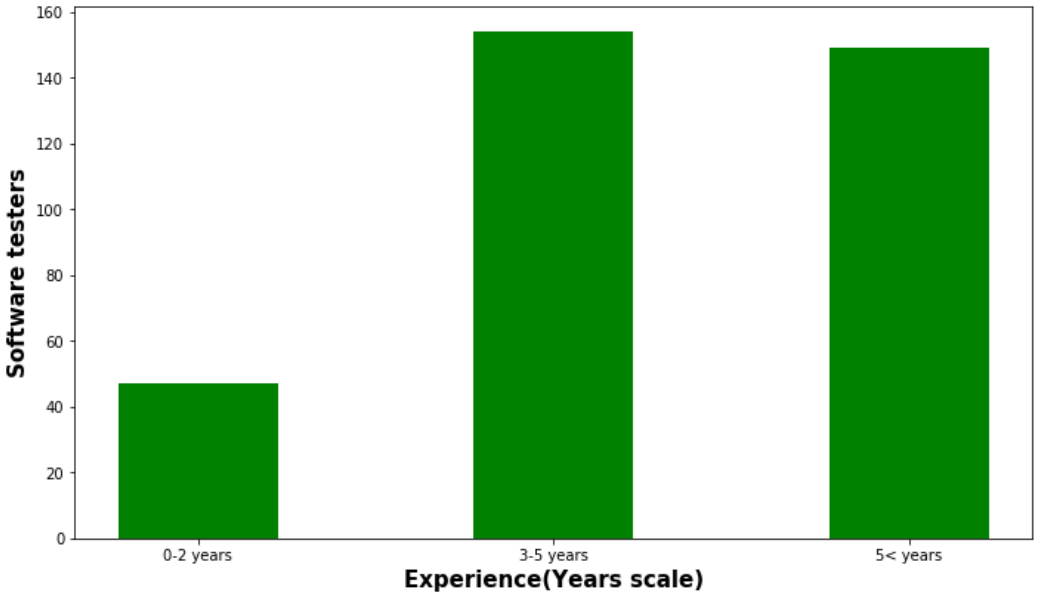

Figure 1 - Testers experience (Years scale)

As presented in Fig. 1, you see that testers participated in the survey when classified in 3 classes depends on the scale of years, those of 0-2, 3-5, and then above 5 years; findings indicated that testers are experienced on the 13.43%, 44.00%, and 42.57% respectively for the cited classes. As the top class is of 3 – 5 years with 44.00% and 42.57% for testers above 5 years of experience which makes 86.57% of all respondents, it means that the results from the survey are trustful as it involves experienced people.

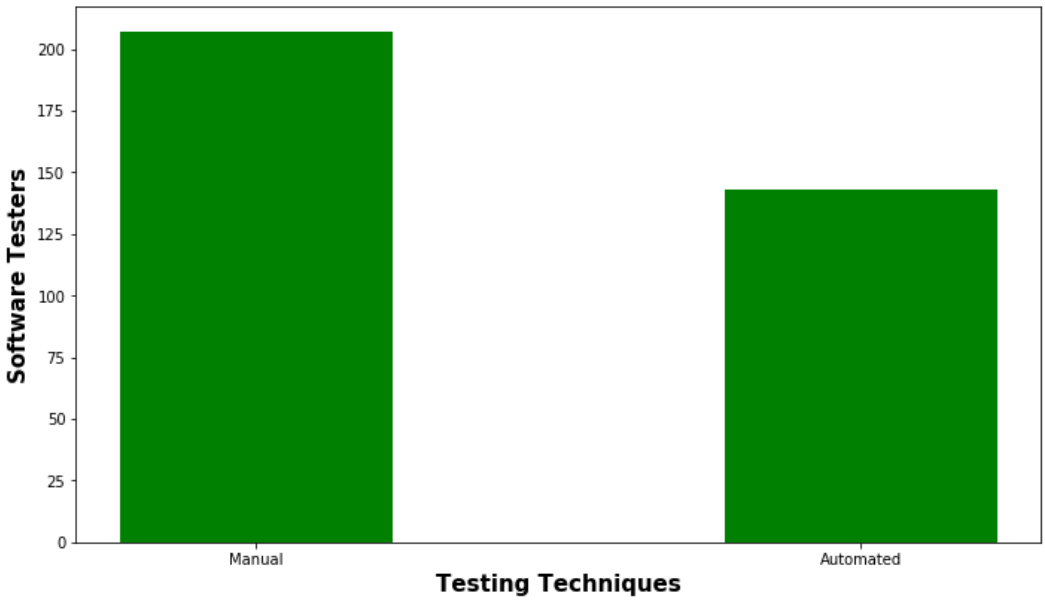

Figure 2 - Testers per Testing techniques

Fig. 2 presents the image of software testers by their preferences on techniques of testing software in their daily activities such that it can help readers to understand the popularity and development of techniques to be used when testing your software; findings tell us that 59.14% of testers prefer of using manual techniques in their activities over 40.86% who use automated testing techniques; Aligned to the first objective of this research, we can conclude partially that manual software testing techniques are more preferable than automated ones.

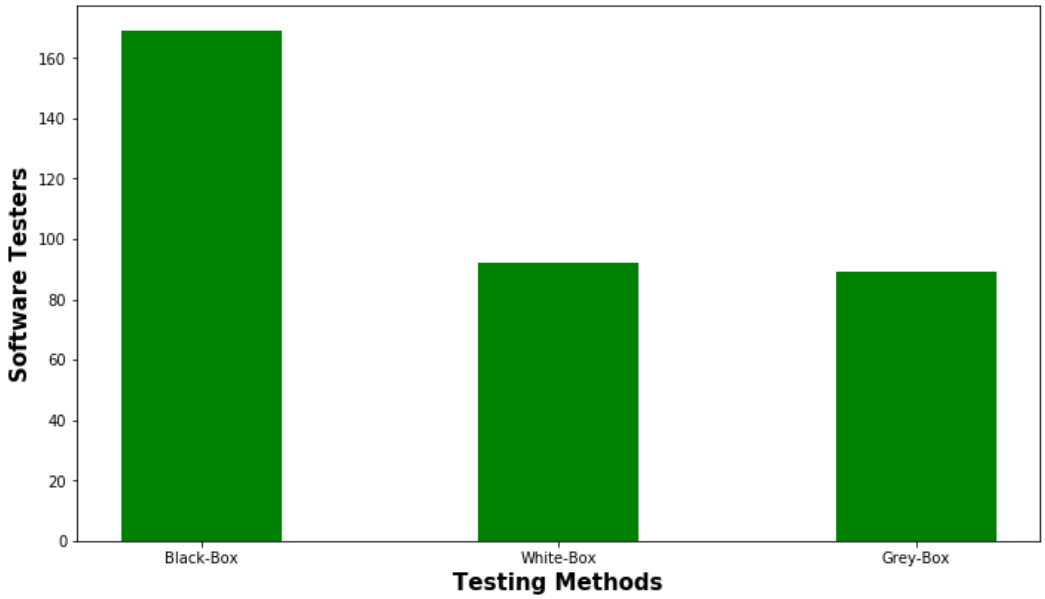

Figure 3 - Testers per Testing methods

To understand the results from the preferences of testers and application of both automated and manual software testing techniques, we saw that software testers are preferably choosing to use manual testing techniques instead of automated techniques; you will see that in regards to the methods of testing as started here and reviewed by previous researchers such as (Tsuyoshi, Y. et al, 2013), (Mika, 2012), (Deepak, D. and Heena, 2014), (Martín, 2017) and many others reflected on three methods of software testing as well as in this study, Black-Box, White-Box, and Grey-Box were reviewed.

Findings from the dataset studies indicate that 48.29%, 26.29%, 25.43% of testers are respectively using Black-Box, White-Box, and Grey-Box; this leads us to conclude that most of the software testers who participated in the survey prefer to use Black-box over two other testing methods studied together with this one; additionally, the black-box is preferred by many testers because it doesn't require code access, rather a tester interacts only with the user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. This advantage lets many testers prefer to use this method when working on their projects.

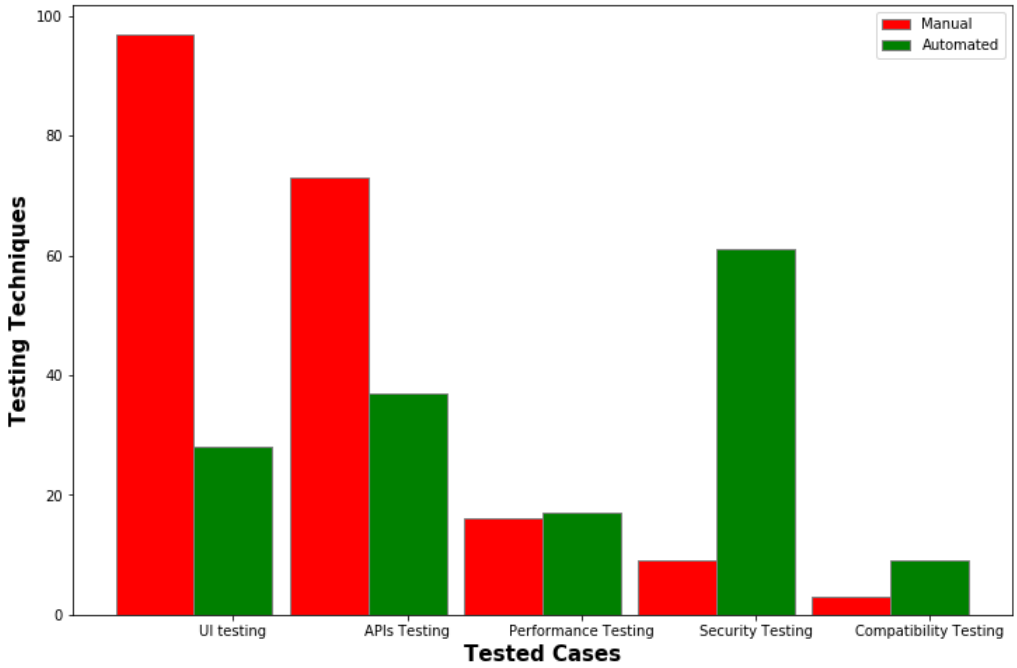

Figure 4 - Tested cases per Testing techniques

By fulfilling the second objective of this study, the tested cases per testing techniques were drawn to visualize the insights inside data through study of five tested cases to cite: UI, APIs, Performance, Security and Compatibility tests, and testers were questioned about their experience towards those tests with the use of Manual and Automated techniques; Fig. 4 demonstrates that when using Manual testing techniques the results are 27.71%, 20.86%, 4.57%,

2.57%, 0.86% respectively for UI, APIs, Performance, Security, and Compatibility tests against 8.00%, 10.57%, 4.86%, 17.43%, 2.57% for testers who use Automated testing techniques.

Through visualization of results, you see that User Interface (UI) tested-cases are the top with 28% of respondents prefer to use Manual testing over 8.00% that prefer, for APIs testing, the Manual testing is preferred at 20.86% against 10.57% of respondents who prefer Automated techniques, etc. globally, the findings indicate that Manual testing is preferred to UI, and APIs testing cases and Automated testing is preferred for Security and Compatibility testing cases, while for Performance testing cases both techniques are preferred or used on the same scale with 4.86% of testers participated at this survey.

CONCLUSION

To achieve this study's main goal as provided in the research proposal, which is to compare Manual software testing techniques with AI (Automated) techniques, by examining their performance on software testing just to help any person interested in software testing to get an appropriate technique to use.

A dataset of 350 instances was used in this study, and through the analysis of collected data, it was found that regards the experience many participants in the survey they are experienced at least two years of software testing, findings tell us that 59.14% of testers prefer of using manual techniques in their activities, indicate that 48.29% of software testers participated in survey prefer to use Black-box over other testing methods, and finally the findings indicate that Manual testing is preferred to UI, and APIs testing cases and Automated testing is preferred for Security and Compatibility testing cases, while for Performance testing cases both techniques are preferred or used on the same scale with 4.86% of testers participated at this survey.

By concluding we take a common understanding that Manual testing techniques are preferred highly by software testers than Automated testing techniques in different testing cases such as User Interface testing while few testers, 17.43% and 2.57% what makes 20.00% only prefer to use Automated techniques for Security and Compatibility testing cases.

REFERENCES

Arun, K. A. (2020). Software Testing Techniques & New Trends. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, 54-67.

Deepak, D. and Heena. (2014). A STUDY OF WHITE BOX AND BLACK BOX SOFTWARE TESTING. *Journal of Emerging Technologies and Innovative Research*, 679-683.

Deepam, A. et al. (2012). A comparative study of Artificial Neural Networks and Info-Fuzzy Networks as automated oracles in software testing. *IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans*, 1-17.

Farah, N. R. (2009). Artificial Intelligence Techniques in Software Engineering (AITSE). *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I*, 1-3.

Husnu, Y. et al. (2019). Guest Editorial: Special issue on Testing Software and Systems. *SpringerLink*, 9-17.

Hussam, H. (2019). The Impact of Artificial Intelligence on Software Testing. *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)* (pp. 565-570). Amann, Jordan: IEEE.

Jerry, G. et al. (2019). Invited Paper: What is AI Software Testing? and Why. *2019 IEEE International Workshop on Service-Oriented System Engineering, SOSE* (pp. 147-160). IEEE.

Komal. (2018). REVIEW PAPER ON SOFTWARE TESTING. *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)*, 1 - 7.

Martín, L. (2017). Black-Box Testing Technique for Information Visualization. Sequencing Constraints with Low-Level Interactions. *Journal of Computer Science and Technology* , 1-11.

Mika, K. (2012). *Model-Based GUI Testing*. Retrieved from Science Direct: https://www.sciencedirect.com/topics/computer-science/black-box-testing

Nahid, A. & Susmita, K. (2019). Review Paper on Various Software Testing Techniques & strategies. *Global Journal of Computer Science and Technology: Computer Software & Data Engineering*, 1-7.

Rui, L. et al. (2020). Artificial Intelligence Applied to Software Testing: A Literature Review. *Conference Paper · June 2020* (pp. 1-6). Castelo, Portugal: Instituto Politécnico de Viana do Castelo.

Sathyavathy, V. and Shanmuga, P. D. . (2018). Software Testing Techniques with Artificial Intelligence in IoT Applications. *International Journal of Recent Technology and Engineering (IJRTE)*, 291-293.

STH. (2021, May 30). *What Is System Testing – A Ultimate Beginner's Guide*. Retrieved from Software testing help: https://www.softwaretestinghelp.com/system-testing/

TP. (2021). *Software Testing Tutorial*. Retrieved from Tutorial point: https://www.tutorialspoint.com/software_testing/index.htm

Tsuyoshi, Y. et al. (2013). A Test Analysis Method for Black Box Testing Using AUT and Fault Knowledge. *Procedia Computer Science*, 551-560.