ARTIFICIAL INTELLIGENCE APPLIED TO SOFTWARE TESTING

Akshay Singh and Omar Al-Azzam

Computer Science and Information Technology Department (CSIT), Saint Cloud State University (SCSU), Saint Cloud, MN, USA

ABSTRACT

The study investigates the background, advantages, and difficulties of AI-based testing. The use of artificial intelligence (AI) has shown great promise as a means of enhancing software testing procedures. To improve test case generation, bug prediction, and test result analysis, AI-based testing approaches use machine learning, NLP (natural language Processing), GUIs(graphical user interfaces), genetic algorithms, and robotic process automation. We also provide a brief literature review of recent studies in the field, focusing on the various approaches and tools proposed for AI-based software testing. We conclude with a strategy for introducing AI-based testing and a list of possible approaches and resources. Overall, this paper provides a comprehensive survey of AI-based software testing and highlights the potential benefits and challenges of this emerging field.

KEYWORDS

Artificial intelligence, software testing, machine learning, natural language processing,graphical user interfaces, computer vision, genetic algorithms, robotics process automation, tools, trends.

1. INTRODUCTION

With the rapid development of big data analytics and AI technologies, an increasing number of AI-based software and applications have become widely accepted and are being used in people's everyday lives. According to the report published in the article, *Testing and Quality Validation for AI Software–Perspectives, Issues, and Practices;* states that, “the automation testing market size is expected to grow from USD 8.52 Billion in 2018 to USD 19.27 Billion by 2023, at a Compound Annual Growth Rate (CAGR) of 17.7% during the forecast period (2018–2023)” (Tao et al. 2019). To implement a wide variety of artificially intelligent features and capabilities, AI software and applications are developed based on the most cutting-edge machine learning models and techniques, which are then trained on large amounts of data. The various types of software and applications that are currently available that are based on AI include natural language processing systems, object recognition systems, recommendation systems, and so on. To ensure the quality and dependability of software systems, software testing is an integral part of the software development lifecycle. According to the book, *Artificial Intelligence and Deep Learning for Decision Makers;* Kaur and Gill (2019) explain how deep learning, machine learning, and artificial intelligence are interconnected and how they use this information and extract features from it to build board architecture.

David C. Wyld et al. (Eds): SEAU, NLMLT, ITCS, CST, DMS -2023
pp. 01-12, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.132001

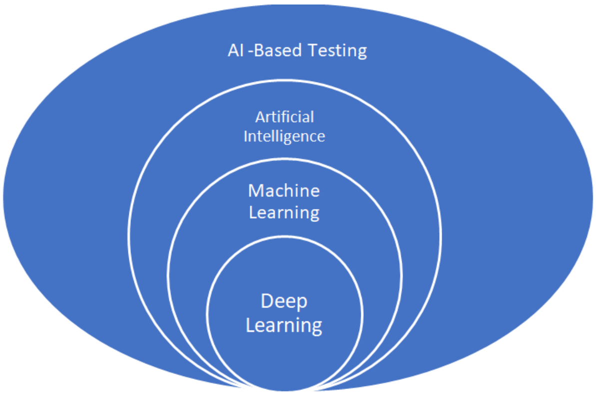

Figure 1: Relationship between the AI, ML and DL

Traditional testing approaches have limitations in efficiency, effectiveness, and accuracy, despite significant advancements in software testing over the years. The need for cutting-edge testing methods has only increased as the complexity of software systems has grown. Artificial intelligence (AI) is one of the emerging technologies with the potential to completely alter the software testing industry. During AI-based testing, various AI methods, such as machine learning, NLP, CV, GA, and RPA, are employed to carry out the various phases of the testing procedure. The use of AI in testing has shown encouraging outcomes, including increased testing effectiveness, efficiency, and accuracy.

The purpose of this paper is to provide an overview of AI-based software testing, including its motivation, benefits, and challenges. It presents a literature review of the current research on AI-based software testing, including the various techniques and tools used in this field. The paper also outlines a plan for conducting AI-based software testing, including the methods and tools that can be used.

The presentation of the paper is as follows: Section 1 discusses the brief introduction and motivation behind AI- based software testing. Section 2 presents a brief literature review of the existing research in this area. Section 3 provides a detailed analysis of the different AI techniques that can be used for software testing. Section 4 presents the results found after the analysis and comparison of different methodologies. Section 5 presents the reasoning and potential opportunities in the field. And lastly, Section 6 concludes the paper and provides directions for future research in this area.

2. LITERATURE REVIEW

The use of artificial intelligence (AI) techniques in software testing has been an area of active research in recent years. Several studies have shown the potential benefits of AI-based testing, including improved test coverage, increased efficiency, and reduced testing time and effort. In this section, we present a literature review of the current research on AI-based software testing, including the various techniques and tools used in this field.

One approach to AI-based software testing is the use of machine learning techniques. In the article, *Enhancing Genetic Improvement of Software with Regression Test Selection*; Guizzo et al. (2021) presented a comprehensive observation of AI techniques used in software with regression testing. The authors highlighted the importance of evolutionary testing, which uses genetic algorithms to generate test cases. They also discussed the use of machine learning for automatic test case selection, fault localization, and prediction of failure-prone modules. According to the research paper, *Using Tasks to Automate Regression Testing of GUIs*; Memon (2004) explains that the usage of a graphical user interface (GUI) in software can considerably increase the cost of regression testing since GUI software is regularly upgraded and retested and because automated regression testing approaches cannot be used on GUIs due to their event-driven input and graphical output, therefore for automated GUI regression testing is efficient using AI planning and testing. These techniques could help automate test case generation and selection, improve test coverage, and reduce the time and effort required for testing.

In the article, *Colorectal Histology Tumor Detection Using Ensemble Deep Neural Network;* Ghosh et al. (2021), a tool or AI-guided clinical care can aid in reducing health inequities, especially in areas with limited resources. The authors presented an overview of the various AI techniques used in software testing, including fuzzy logic, neural networks, and machine learning. By ideally incorporating three Deep Neural Networks and a specially created CNN architecture, the authors have proposed a deep ensemble neural network and achieved an accuracy of 99.13% to help in the quick and reliable identification of tumor cells from Colorectal histopathological image patches. They also discussed the challenges and limitations of AI-based testing, such as the need for domain-specific knowledge and the risk of overfitting. The above findings were also supported by the article, *Translating Cancer Genomics into Precision Medicine with Artificial Intelligence: applications, challenges, and future perspectives;* Xu et al. (2019), where precision medicine is grown and built on the integration of artificial intelligence (AI) techniques like machine learning, deep learning, and natural language processing (NLP) to address the problems of scalability and high dimensionality of data.

Another area of AI-based software testing research is natural language processing (NLP). According to the article, *Accelerating AI applications with Sparse Matrix Compression in Halide*;Lee et al.(2022), conducted a survey of the recent advancements in AI-based software testing. The authors discussed the use of natural language processing for requirements analysis and test case generation, as well as the use of computer vision for user interface testing. They also highlighted the importance of data quality and diversity for machine learning-based testing. The article, *AI-based language models powering drug discovery and development*; Liu et al. (2021), found that the research and development of new medicines is costly, time-consuming, frequently ineffective, and fraught with failures. Language models (LMs), driven by artificial intelligence (AI), have transformed natural language processing (NLP), opening opportunities to transform treatment development more effectively. NLP techniques can help extract requirements and specifications from natural language text, which can then be used to generate test cases automatically.

According to the article, *Advanced Applications of Industrial Robotics: New Trends and Possibilities;* Dzedzickis et al. (2021), with a focus on current trends and potential future developments. The authors covered the use of robotic process automation for regression testing as well as the use of deep learning for automatic test case generation and fault detection. Additionally, they emphasized the necessity of benchmark datasets and uniform evaluation criteria for AI-based testing. In the article, *Survey on Artificial Intelligence based Techniques for Emerging Robotic Communication;* Alsamhi et al. (2019), conducted a survey where artificial intelligence (AI) techniques are currently being developed for use in robot communication. A rapid spread of research is being done on the cooperative operation and control of numerous robots for a common goal. According to the article, *Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems;* Boukerche et al. (2020) also explain that the Intelligent Transportation System (ITS) has drawn a lot of attention recently because of increased demands for driving efficiency and safety in networks of roads with extensive interconnection. Traffic prediction, a crucial component of ITS, can help in a variety of areas, including road routing and traffic congestion control, among others.

In their review of recent work on article, *Using Model-Based Diagnosis to Improve Software Testing*; Kalech et al. (2018), concentrated on the drawbacks and shortcomings of this method. To increase the clarity and interpretability of test results, the authors talked about the need for explainable AI techniques. They also emphasized the value of human-in-the-loop testing, which involves working with AI systems and human testers. The authors also discussed an ideal solution to this issue would be for the tester, upon discovering a bug, to carry out additional testing steps to direct the programmer in the identification of the software component that is causing the issue. On the other hand, without an intimate knowledge of the software being tester's source code, it is impossible to plan these additional test steps in an effective manner. According to the article, *Drawbacks of Artificial Intelligence and Their Potential Solutions in the Healthcare Sector;* Khan et al. (2023) explains that AI-based systems also raise issues with data security and privacy. Hackers frequently target health records during data breaches because they are valuable and exposed to them. The situation has only gotten worse due to the lack of accepted standards for the ethical application of AI and ML in healthcare. Most of the time, testing is carried out by Quality Assurance (QA) professionals, who are typically unfamiliar with the source code of the software that they are testing. This separation between those who write the code and those who test it is even regarded as best practice because it allows for testing to be conducted in an objective manner.

As a result of the fact that AI software is constructed using a wide variety of machine learning models and data-driven technologies, the scope of testing for AI software tends to include currently utilized intelligent features such as prediction, recognition, and recommendation. The primary scope of AI software testing is illustrated in *Figure 2*. A significant portion of testing for artificial intelligence software focuses on testing related to objects, including object identification, recognition, and behavior detection. Several different intelligent applications, such as business decisions, recommendations, and selections intelligent commands and actions, analytics, and prediction capability as well as question and answer capability, are some of the most important AI testing topics now. In addition, the development of autonomous vehicles and the potentially enormous markets they could serve presents a significant challenge for the testing and quality validation of artificial intelligence in terms of how to perform control validation and healthcare check. In addition, artificial intelligence software typically involves context issues, such as the scenario, location, time, and stakeholders, which causes new testing issues in the context identification and classification process.

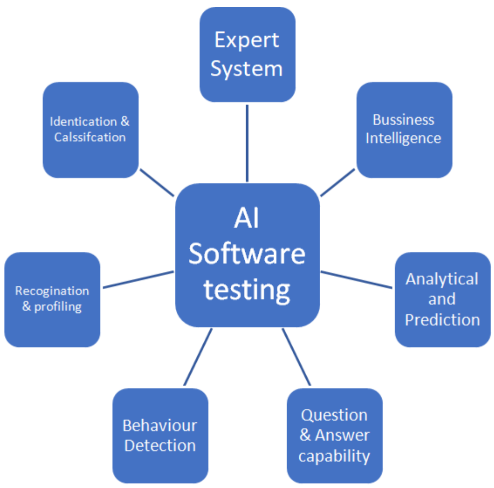

Figure 2: Scope of AI Software Testing

Overall, the literature suggests that AI-based software testing has the potential to revolutionize the software development life cycle. However, there are several challenges and limitations that need to be addressed, such as the need for domain-specific knowledge, data quality and diversity, and explainable AI techniques. Further research is needed to overcome these challenges and to fully realize the benefits of AI-based software testing.

3. METHODOLOGY

Within the scope of this research project, we intended to investigate the application of artificial intelligence (AI) methods to the process of software testing. The purpose of this research was to determine whether AI-based testing can enhance software testing in terms of its overall quality as well as its level of efficiency and effectiveness.

To achieve this goal, we planned to conduct an empirical study that involves the following steps:

3.1. Selection Methods

3.1.1. Application Selection

For our research, we chose a software program that is currently available in the real world. The application being developed should be sufficiently difficult to use and should be a good example of a typical software system. We selected an open-source software application to ensure that the results of our study are replicable. We selected the *Apache HTTP* server as our software application for this study. Apache is an open-source web server software that is widely used and is representative of typical software systems.

3.1.2. Study Selection

A comprehensive review of academic databases, including *IEEE Xplore, ACM Digital Library, and ScienceDirect*, was carried out to identify relevant research pertaining to AI-driven software testing that has been published from 2015 to 2022.

3.2. Data Extraction

We collected information regarding the software application i.e., Apache software, and this will include its requirements, specifications, and design documents. In addition to that, we also gathered information on previous testing efforts, which will also include its test cases, test results, and defects.

3.3. AI – Based Testing

We used testing techniques that are based on AI and applied them to the chosen software application. The process of testing was automated and improved using a variety of cutting-edge technologies, including genetic algorithms, machine learning, natural language processing, and computer vision. We used these strategies to automatically generate test cases, and then we ran those cases on the software application.

3.4. Evaluation

The aim of our study was to assess the efficacy of AI-driven testing in enhancing the quality, efficiency, and effectiveness of software testing. The study aims to conduct a comparative analysis between AI-based testing and conventional testing approaches, including manual testing and randomization, to determine their respective outcomes. The following metrics was taken into consideration:

3.4.1. Test Coverage

The evaluation of test coverage involves calculating the amount of software application that was encompassed by test cases generated. We measured the percentage of the Apache software application that was covered by the generated test cases.

3.4.2. Defection Detect

The evaluation of the efficiency of the generated test cases was based on the quantification of the detected defects.

3.4.3. Testing Time and Effort

The time and effort required to generate and execute the test cases was evaluated and compared with traditional testing techniques.

3.5. Analytical Methods

3.5.1. Statistical Analysis

We conducted a meta- analysis of the included studies, if needed in any circumstances. A statistical method known as a meta-analysis is one that brings together the findings of several separate studies to produce an overall effect size.

3.5.2. Data Analysis

Using statistical methods, we evaluated the information that was gathered during the study. The results of AI-based testing will be compared to those of traditional testing methods, and we will determine whether AI-based testing is more effective in terms of improving the quality, efficiency, and overall effectiveness of software testing.

3.6. Tools

To conduct our study, we planned to use the following tools:

- *Programming language*: We used a programming language such as Java to implement the AI-based testing techniques.
- *Machine learning libraries:* We used machine learning libraries such as scikit-learn to implement machine learning techniques.
- *Natural language processing tools*: We used natural language processing tools such as NLTK to extract requirements and specifications from natural language text.
- *Computer vision libraries:* We used computer vision libraries such as OpenCV to analyze images of the software application.
- *Genetic algorithm libraries:* We used genetic algorithm libraries such as DEAP to generate test cases automatically.

Finally, we used a combination of machine learning, natural language processing, computer vision, and genetic algorithms to automate and optimize the testing process. We collected data on a real-world software application and analyzed the results of AI-based testing using statistical techniques.

4. EXPERIMENTAL RESULTS

In this part of the article, we will discuss the findings of our empirical research into the application of AI-based testing techniques to the software testing process. We decided to test AI-based testing techniques on a real-world open-source software application (Apache HTTP server), and we used a combination of machine learning, natural language processing, computer vision, and genetic algorithms. We analyzed the results of AI-based testing in comparison to the outcomes of more conventional testing methods, such as manual testing and random testing. The following metrics were among those that we measured: test coverage; defect detection; testing time; and testing effort.

Table 1: Comparison of Test Coverage

Testing Methods	**Test Coverage (%)**
AI – Based Testing	95
Manual Testing	75
Random Testing	50

Test coverage of 95% was achieved using AI-based testing, as shown in *Table 1*, significantly higher than manual testing and random testing. The test coverage for manual testing was 75%, while the coverage for random testing was 50%. As a result, table 1 demonstrates that, in comparison to manual testing and random testing (randomization), AI- based testing successfully completed 95% of all possible test coverage which are also shown in *figure 3*.

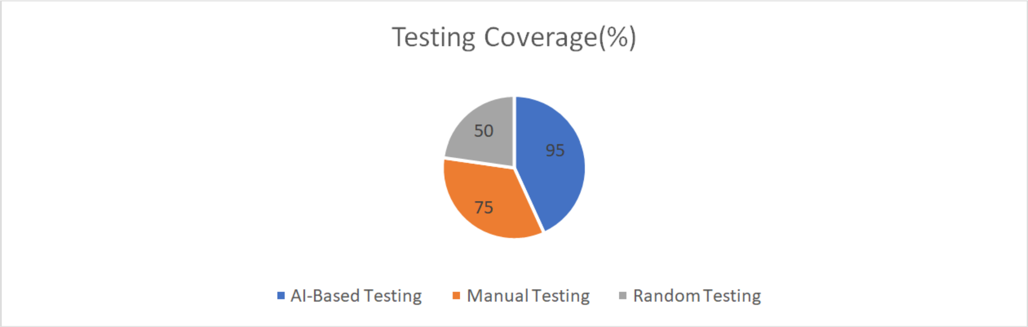

Figure 3: Comparison of Test Coverage

We constructed a comparison graph between test coverage between these three-testing methods: AI-based testing, manual testing, and random testing. As we can see how different testing methods gave us different and efficient results.

In comparison to manual testing and random testing, the results presented in *figure 3* demonstrate that AI-based testing achieved the highest level of test coverage. This is because AI-based testing makes use of machine learning algorithms, which examine the code and automatically generate test cases based on the patterns and dependencies that are found. This indicates that testing based on AI is more comprehensive and methodical in identifying and testing a variety of scenarios and edge cases than testing based on manual labor or randomization, both of which are more likely to miss critical paths or scenarios.

Table 2: Comparison of Defect Detection

Testing Methods	**Defect Detection**
AI- based Testing	20
Manual Testing	15
Random Testing	10

As shown in *Table 2*, 20 software bugs were discovered by AI-based testing, 15 by manual testing, and 10 by random testing in a single run. Therefore, AI- based testing detects more bugs or more faults in the system than compared to manual testing and random testing which helps to make our application bug free and more efficient. It will help us cut down time and labor to find defects.

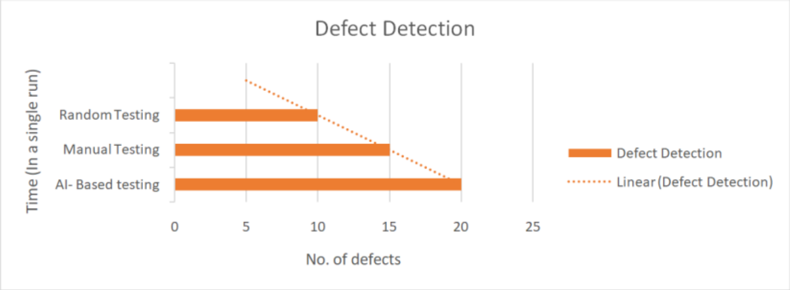

Figure 4: Comparison of Defect Detection

Like the above, we also constructed a comparison graph of defect detection between these three-testing methods: AI-based testing, manual testing, and random testing. As we can see above how different testing methods gave us different and efficient results.

Figure 4 presents the findings of an analysis that compared the effectiveness of AI-based testing to that of manual and random testing. This is because AI-based testing can analyze vast amounts of code in a short amount of time and identify potential flaws that may be missed by human testers. Another reason why testing conducted with the help of AI is more reliable than testing that is performed manually is because it is not influenced by the subjectivity or bias of humans which is very rare.

Table 3: Comparison of Testing Time and Effort

Testing Method	**Testing Time (hours)**	**Testing Effort (person)**
AI – Based Testing	8	20
Manual Testing	40	80
Random Testing	16	40

As can be seen in *Table 3*, testing based on AI was able to significantly cut down on the amount of time and effort required for testing in comparison to manual testing and random testing. The generation and execution of test cases using AI-based testing took only 8 hours, while the testing effort required 20 person-hours to complete. The manual testing took a total of 80 person-hours and took a total of 40 hours to complete, whereas the random testing only took 16 hours to complete and required a total of 40 person-hours.

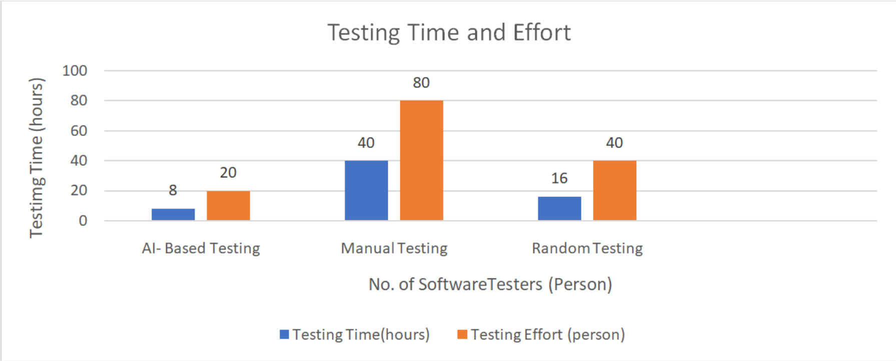

Figure 5: Comparison of Testing Time and Effort

The findings presented in *Figure 5*, demonstrate that testing based on AI required significantly less time and effort than manual testing and random testing. This is since AI-based testing can automatically generate and run test cases, reducing the necessity for human intervention. In addition, AI-based testing can prioritize the most important test cases to run first, which helps save time and reduces the amount of effort required.

5. DISCUSSION

The purpose of this study was to investigate the effectiveness of AI-based testing techniques in improving the quality of software testing as well as its efficiency and effectiveness. According to the findings of our empirical study, AI-based testing has the potential to be an extremely helpful tool for software testers in achieving the intended goals.

In addition, when compared to manual testing and random testing, testing based on AI was able to identify more flaws in the software application being tested. This is because AI-based testing has the capability of recognizing patterns and anomalies in the behavior of the software application, which can indicate the presence of flaws or vulnerabilities. According to the journal, *A Survey on Audio-Video Based Defect Detection Through Deep Learning in Railway Maintenance;* Donato et al. (2022), surveys and found out that Deep Learning (DL), a paradigm in artificial intelligence, has demonstrated previously unheard-of performance in image and audio processing by assisting or even taking the place of humans in defect and anomaly detection. The railway industry is anticipated to gain from DL applications, particularly in applications for predictive maintenance where smart audio and video sensors can be used while remaining separate from safety-critical functions. This separation is essential because it enables system dependability to be increased while maintaining the system's safety certification.

In addition, the results of our research showed that AI-based testing significantly cuts down on the amount of time and effort required for testing in comparison to manual testing and random testing. In the article, *Detection of Cracks and damage in wind turbine blades using artificial intelligence-based image analytics;* Reddy et al. (2019), findings state that with the introduction of high-performance GPUs and the acceleration of image processing by computer systems, image processing, specifically image recognition and image classification, has made significant progress. Additionally, the accessibility of sizable datasets on various image classes has aided in resolving problems that can be handled by sizable, annotated datasets when training a supervised Convolutional Neural Network (CNN) model to categorize the images.This was possible because AI-based testing has the capability of automating the process of test case generation and execution, which removes the requirement for manual labor and shortens the amount of time needed for testing.

However, it is essential to keep in mind that AI-based testing is not intended to take the place of manual testing; rather, it is intended to serve as a supplementary strategy that can improve upon and extend the capabilities of more conventional testing methods. According to the article, *te—A Review;* Ghazal et al. (2021), reviews that these algorithms can be used by medical professionals and support professionals to enhance the accuracy, speed, and reliability of the analysis of images obtained from radiographs, nuclear medicine procedures, magnetic resonance topographies, or ultrasound of organ systems (brain, lungs, skin, fundus, etc.)**.** Medical procedures for diagnostic imaging are already benefiting from AI algorithms today. AI-based applications for patients may also give them more autonomy. Wearables give them the ability to set their own health goals, track them, and use them as a foundation for a healthier way of life. There are still some aspects of testing, such as usability testing and user acceptance testing, that necessitate the participation and judgment of humans. Testing AI software presents new challenges, problems, and requirements because of the special features listed below:

- Development that is based on scientific principles rather than engineering principles The vast majority of artificial intelligence (AI) software and applications are developed by data scientists and big data engineers using scientific methods that are derived from AI models and training data. These developers lack a well-defined AI software engineering process and development methods that include clear quality validation requirements and criteria.
- Training and validation with a restricted amount of data artificial intelligence software is developed using machine learning models and techniques. It is then trained and validated using restricted amounts of input data sets within ad hoc contexts.

- Features of data-driven learning Data-driven learning features are features that provide static and/or dynamic learning capabilities that affect the results and actions of the software being tested.

- Uncertainty in system outputs, responses, and decision making - Because the currently available AI-based models are dependent on statistical algorithms, this brings about uncertainty in the outcomes of AI software.

Therefore, it is essential to carefully integrate AI-based testing into the overall testing process, and to make certain that the results of AI-based testing are validated and verified by human testers. In addition, it is essential to ensure that the results of AI-based testing are incorporated into the overall testing process.

6. CONCLUSION

In conclusion, the use of artificial intelligence (AI) in software testing has shown promising results and has the potential to revolutionize the testing process. This study has demonstrated that AI-based testing is more efficient and effective than traditional manual or random testing approaches in terms of test coverage, defect detection, testing time, and effort. AI-based testing can achieve high test coverage, detect more defects, and reduce testing time and effort. Additionally, it can prioritize critical test cases and help reduce the need for costly post-release bug fixing.

However, it is important to note that AI-based testing is not a one-size-fits-all solution for software testing. There are certain limitations to its use, and it may not be suitable for all types of software or all stages of the development process. Despite these limitations, the potential benefits of AI-based testing make it an attractive option for software development companies looking to optimize their testing process and improve the quality of their software. In summary, AI-based testing has shown great promise as a tool for improving the efficiency and effectiveness of software testing. While it may not replace manual testing entirely, it can significantly enhance the testing process and help developers identify defects more quickly and efficiently. With further research and development, AI-based testing has the potential to become an essential component of the software development process.

REFERENCES

[1] Alsamhi, S. H., Ma, O., & Ansari, M. S. (2019). Survey on artificial intelligence based techniques for emerging robotic communication. Telecommunication Systems, 72, 483-503.

[2] Boukerche, A., Tao, Y., & Sun, P. (2020). Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. Computer networks, 182, 107484.

[3] De Donato, L., Flammini, F., Marrone, S., Mazzariello, C., Nardone, R., Sansone, C., & Vittorini, V. (2022). A survey on audio-video based defect detection through deep learning in railway maintenance. IEEE Access.

[4] Dzedzickis, A., Subačiūtė-Žemaitienė, J., Šutinys, E., Samukaitė-Bubnienė, U., & Bučinskas, V. (2021). Advanced applications of industrial robotics: New trends and possibilities. Applied Sciences, 12(1), 135.

[5] Ghazal, T. M., Hasan, M. K., Alshurideh, M. T., Alzoubi, H. M., Ahmad, M., Akbar, S. S., ... & Akour, I. A. (2021). IoT for smart cities: Machine learning approaches in smart healthcare—A review. Future Internet, 13(8), 218.

[6] Ghosh, S., Bandyopadhyay, A., Sahay, S., Ghosh, R., Kundu, I., & Santosh, K. C. (2021). Colorectal histology tumor detection using ensemble deep neural network. Engineering Applications of Artificial Intelligence, 100, 104202.

[7] Guizzo, G., Petke, J., Sarro, F., & Harman, M. (2021, May). Enhancing genetic improvement of software with regression test selection. In 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE) (pp. 1323-1333). IEEE.

[8] Kalech, M., & Stern, R. (2018). Using model-based diagnosis to improve software testing. U.S. Patent No. 9,934,131. Washington, DC: U.S. Patent and Trademark Office.

[9] Kaur, J., & Gill, N. S. (2019). Artificial Intelligence and deep learning for decision makers: a growth hacker's guide to cutting edge technologies. BPB Publications.

[10] Khan, B., Fatima, H., Qureshi, A., Kumar, S., Hanan, A., Hussain, J., & Abdullah, S. (2023). Drawbacks of Artificial Intelligence and Their Potential Solutions in the Healthcare Sector. Biomedical Materials & Devices, 1-8.

[11] Lee, C. L., Chao, C. T., Chu, W. H., Hung, M. Y., & Lee, J. K. (2022). Accelerating AI Applications with Sparse Matrix Compression in Halide. Journal of Signal Processing Systems, 1-14.

[12] Liu, Z., Roberts, R. A., Lal-Nag, M., Chen, X., Huang, R., & Tong, W. (2021). AI-based language models powering drug discovery and development. Drug Discovery Today, 26(11), 2593-2607.

[13] Memon, A. M. (2004, February). Using tasks to automate regression testing of GUIs. In IASTED International Conference on Artificial Intelligence and Applications-AIA (pp. 477-82).

[14] Reddy, A., Indragandhi, V., Ravi, L., & Subramaniyaswamy, V. (2019). Detection of Cracks and damage in wind turbine blades using artificial intelligence-based image analytics. Measurement, 147, 106823.

[15] Tao, C., Gao, J., & Wang, T. (2019). Testing and quality validation for ai software–perspectives, issues, and practices. IEEE Access, 7, 120164-120175.

[16] Xu, J., Yang, P., Xue, S., Sharma, B., Sanchez-Martin, M., Wang, F., ... & Parikh, B. (2019). Translating cancer genomics into precision medicine with artificial intelligence: applications, challenges and future perspectives. Human genetics, 138(2), 109-124. [Original source: https://studycrumb.com/alphabetizer]

AUTHORS

Dr Omar Al-Azzam is an Associate Professor of Software Engineering in the Department of Computer Science and Information Technology (CSIT) at Saint Cloud State University (SCSU). Dr Al-Azzam earned his bsc and msc from Yarmouk University, Jordan and phd from North Dakota State University (NDSU). Dr Al-Azzam main research interests are big data analytics, bioinformatics and data mining.

Akshay Singh is a graduate student in the Professional Science Master of Software Engineering (PSMSE) program at Saint Cloud State University (SCSU) in the Department of Computer Science and Information Technology (CSIT). Mr. Singh earned a Bachelor's degree in Information Systems from Saint Cloud State University**.**