American Academic Scientific Research Journal for Engineering, Technology, and Sciences

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Using AI and Machine Learning in QA Testing

Nikita Klimov [a*]

[a] *Senior QA / QC Engineer in ADP inc.Miami, United States*

Abstract

The article will consider the possibilities of using artificial intelligence (AI) and machine learning (ML) technologies in the field of software quality control due to the fact that they are able to change the usual approaches to testing due to their abilities. Methods of using AI and ML to increase the effectiveness of quality assurance (QA) will be considered: automation of tests, detection of defects, prediction of anomalies. The methodology is based on the analysis of scientific papers, which will describe achievements in the application of these technologies during QA, including adaptive algorithms that automatically generate tests, clustering methods that systematize errors, and big data analysis that allows predicting defects. As part of the work, examples of organizations that demonstrate comparing user interface testing using a manual method and automated regression tests will also be considered. The data obtained show a decrease in the time spent on testing, a decrease in the probability of missing errors, and an improvement in the quality of processes. The information in the work will be useful to quality specialists, developers, and AI researchers working on optimizing testing. In conclusion, the article notes the success of applying such technological solutions in achieving QA goals.

Keywords: Artificial Intelligence, Machine Learning, QA Testing, Automation, Defect Prediction, Anomaly Analysis.

1. Introduction

In the context of evolving realities, the process of software development is being reshaped by the capabilities provided by AI and ML. The task is not merely to accelerate or automate testing but to provide adaptive mechanisms for comprehensive analysis and defect prediction. Traditional testing approaches currently reveal several vulnerabilities, especially as software architectures scale and become more complex. Firstly, they exhibit limited flexibility in adapting to dynamically changing parameters and interactions within complex systems. Secondly, the human factor introduces a risk of oversights and leads to excessive costs associated with routine processes.

Received: 4/18/2023
Published: 4/18/2023

----

* Corresponding author.

A significant portion of repetitive tasks, such as regression testing, consumes the time of specialists, making them susceptible to errors. Thus, the application of AI and ML in QA enhances the accuracy of testing. These algorithms are capable of analyzing large datasets, recognizing patterns, and predicting defects, thereby minimizing the likelihood of oversights. Such methods accelerate the testing process and make it more cost-effective.

Conversely, approaches based on rigidly fixed test scenarios do not account for stochastic changes, such as variability or dependencies between systems, thereby limiting the ability to timely detect hidden defects and deviations [2].

The relevance of this topic has grown alongside the increasing number of products incorporating AI and ML components, which require particularly thorough testing to ensure reliability, safety, and compliance with stated requirements. Their integration into QA optimizes testing processes and expands functional capabilities, including prediction and diagnostics, ultimately improving the overall quality of software.

The aim of this study is to examine the current AI and ML methods and algorithms used in QA and to analyze their effectiveness in enhancing software quality.

2. Materials and Methods

Several scientific methods were employed in this study. The analytical method enabled a comprehensive review of current approaches to applying AI and ML in QA. The comparative analysis method was used to evaluate the effectiveness of various algorithms and tools utilized in automated testing, including classification methods, anomaly detection, and test data generation.

The study by Santhanam P. [1] focuses on methods for enhancing the quality of AI systems, emphasizing fairness and robustness. The work examines approaches to reducing data bias and mitigating error probability, which are critical for developing socially safe AI-based technologies.

The work by Fujii G. et al. [2] analyzes the application of AI in industrial environments, where the deployment of ML models presents certain challenges. The recommendations proposed in the paper emphasize maintaining the quality of AI solutions in industrial settings, where technology integration requires reliable optimization approaches to ensure product stability.

Narita K. et al. [3] describes the Qunomon platform, designed to assess the quality of AI models. This tool integrates various testing methods, forming a comprehensive approach to evaluating model reliability and accessibility, which is in demand for universal assessment solutions.

In the article “Adaptive methods for machine learning-based testing of integrated circuits and boards,” Liu M., Chakrabarty K. [4] highlight adaptive ML methods in component testing. The authors demonstrate how algorithms adapt to object changes, facilitating defect detection and enhancing testing reliability.

Mulla N., Jayakumar N. [5] explores the use of AI for automating the software testing process, emphasizing how algorithms optimize tasks and reduce the workload on specialists.

The study by Bajaj Y. and Samal M. K. [6] examines the potential of these algorithms in generating test data and detecting defects, highlighting that such models enable the creation of diverse test scenarios.

As Overclockers [7] reported, Google plans to develop an AI-based system to assist QA specialists in software testing processes. This technology aims to automate the detection of errors and defects, potentially accelerating the development process and reducing the workload on QA teams.

Another critical aspect of testing is regression health analysis, which focuses on ensuring that changes made to the software do not disrupt the functionality of other components. IBM's experience is an example of this, as described in their official source [8]. The source discusses the use of AI for automating regression testing, enhancing analysis accuracy, speeding up execution, minimizing regression risks, and reducing routine tasks for engineers.

The reviewed studies present new approaches for quality assessment, centered on applying machine learning techniques.

3. Results and Discussion

The application of artificial intelligence and machine learning methods in QA involves not only classification and regression algorithms but also advanced data analysis techniques, such as anomaly detection, defect prediction, automatic test data generation, and adaptive testing methods. Below are the main areas where AI and ML are actively integrated into software quality assurance processes.

Anomaly detection algorithms help identify deviations from normal system behavior, allowing potential defects to be predicted before they manifest. Deep learning-based algorithms, such as autoencoders, classify behavioral patterns and detect data anomalies in a timely manner. The main challenge lies in model configuration, as errors can arise from both overfitting and insufficient sampling coverage. In such tasks, selecting the appropriate algorithm and effectively utilizing accuracy metrics are crucial for balancing sensitivity and specificity.

Synthetic test data generation using Generative Adversarial Networks (GANs) and other modeling techniques not only enhances the test suite but also considers rare edge cases. This approach produces a synthesis of data that closely resembles real-world scenarios, which is particularly valuable when access to actual data is limited or confidential. However, these methods necessitate rigorous quality control of the generated data, as modeling issues can lead to unrealistic scenarios that may fail to identify actual defects [1].

AI and ML also enable the automation of error classification, streamlining defect management. Using Natural Language Processing (NLP) algorithms for error log analysis and classification allows errors to be categorized and linked to specific root causes, accelerating the analysis process. This approach is particularly beneficial in programming contexts where the number of errors can accumulate to thousands in a short period. Additionally, clustering algorithms, such as K-means, along with graph theory techniques, assist in grouping similar issues, thereby reducing the workload for analysts.

Model-based testing (MBT), when combined with ML, facilitates not only test execution automation but also test generation. The system is represented as a model, which serves as the basis for creating scenarios. This approach is particularly effective when dealing with a large number of states or complex dependencies between components, as such algorithms aid in automating the model update processes.

Despite their clear advantages, the integration of AI and ML into QA presents several limitations that require a precise and tailored approach. One of the primary challenges is the sensitivity of ML models to data quality; low-quality, incomplete, or imbalanced data can lead to inaccurate outcomes and create a misleading impression of system stability. Furthermore, AI and ML models often exhibit a "black box" effect, rendering the decision-making processes opaque. This opacity increases the workload for analysts, as explaining specific system conclusions requires a thorough analysis of the model's logic, which is not always a straightforward task [3].

Another issue is the complexity of integration. Testing environments are often characterized by highly isolated processes and legacy tools, complicating the adoption of advanced ML models. Compatibility, architectural adaptation, and potential conflicts between existing tools and new algorithms must be considered.

Figure 1 below illustrates the main directions of AI and ML applications in QA.

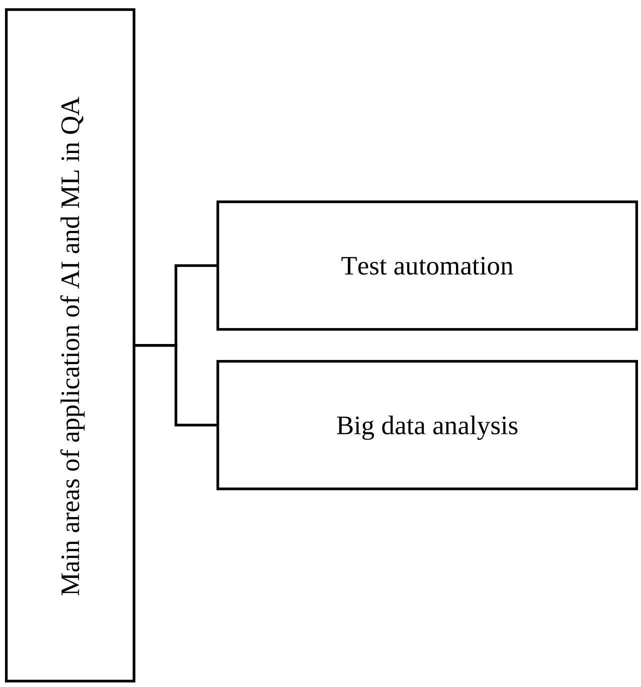

Figure 1: The main directions of application of AI and ML in QA [5]

Automated testing, in turn, enables the automation of tasks previously performed manually, such as test case generation, result analysis, and defect detection. This reduces testing time and lowers the risk of human error.

In the field of big data analysis, machine learning processes large data volumes, identifying hidden patterns and potential problem areas in software. This allows for the prediction of possible defects, thereby preventing them at early stages of development [6]. Table 1 below outlines the advantages of integrating AI into QA.

Table 1: Advantages of integrating AI into QA [5-6].

Advantage	Description
Increased efficiency	Automating routine tasks allows testers to focus on complex aspects of testing, improving overall product quality.
Reduced testing time	AI can execute tests faster and with fewer resource expenditures, accelerating software release.
Improved accuracy	Machine learning reduces the likelihood of missed defects, ensuring reliable testing.

Despite the clear advantages listed in Table 1, the implementation of AI in QA faces certain challenges, such as the need to train personnel in new skills and adapt existing processes to new technologies. However, given the rapid development of the field, it is expected that the role of AI in software testing will continue to grow, unlocking new opportunities to enhance the quality and efficiency of development.

Below is an example demonstrating the use of these algorithms with the scikit-learn machine learning library.

```
import numpy as np
from sklearn.ensemble import IsolationForest
from sklearn.metrics import classification_report
import pandas as pd

# Example data for log analysis and anomaly detection
# Assume we have metrics: Execution Time, Memory Usage, Error
Count-=
data = pd.DataFrame({
    'ExecutionTime': np.random.normal(loc=50, scale=5, size=100),
# Execution Time
    'MemoryUsage': np.random.normal(loc=200, scale=10, size=100),
# Memory Usage
    'ErrorCount': np.concatenate([np.random.poisson(1, 95),
np.random.poisson(5, 5)])  # Anomalous Error Counts
})

# Initialize the Isolation Forest model for anomaly detection
model = IsolationForest(contamination=0.05, random_state=42)  #
contamination=0.05 assumes 5% of data as anomalies
model.fit(data)

# Anomaly prediction
```

```
data['Anomaly'] = model.predict(data)

# Display results: 1 indicates normal data, -1 indicates
anomalies
print(data[['ExecutionTime', 'MemoryUsage', 'ErrorCount',
'Anomaly']])

# Generate a classification report
print(classification_report(data['Anomaly'],
model.predict(data)))

# Handling detected anomalies
anomalies = data[data['Anomaly'] == -1]
print("Detected anomalies:")
print(anomalies)
```

In this example, random data is generated to represent system performance metrics. The model is trained using the Isolation Forest algorithm, which is commonly used for anomaly detection. After training, the model predicts which records are anomalous. The code identifies logs that deviate from the norm, allowing QA engineers to focus on testing and resolving potential issues. This, in turn, reduces the time needed for product releases [4].

Examples of companies utilizing such technologies are provided below.

In 2023, Google filed a patent application for a technology aimed at conducting software testing using AI. It is expected that this technology could entirely replace QA specialists and function independently. Along with the patent, an accompanying report indicated that QA specialists do not always fulfill their duties completely, leading to project delays or releases containing code errors. The report also mentioned that as new projects become larger in scale, the complexity of testing increases.

The proposed solution is a new AI system from Google. However, even if this technology could operate without QA specialists, its capabilities would likely be limited to detecting only factual errors, not logical ones. For instance, the animation code for a character might be correctly written and functional, but incorrect parameter values could result in "awkward movements" of the character. Therefore, the future of this solution remains uncertain [7].

IBM leverages AI to automate the regression testing of software products. By using ML algorithms, the company analyzes code changes and automatically determines which tests need to be executed, significantly reducing testing time [8]. Thus, the application of AIe and ML in software QA enables a new level of automation, accuracy, and flexibility in testing, which is crucial given modern requirements for security and stability. Their integration into QA processes is becoming the standard for leading companies, helping to maintain product quality and meet increasing user expectations.

4. Conclusion

The use of artificial intelligence (AI) combined with machine learning (ML) methods in software quality assurance (QA) presents extensive opportunities for automating testing processes and enhancing their accuracy. The application of advanced ML algorithms, such as classification methods, anomaly detection, and test data generation, demonstrates the capability to identify defects at early stages of development.

The analysis results confirm that integrating AI and ML reduces time costs while simultaneously minimizing the impact of the human factor—an essential requirement for dynamic systems needing constant monitoring. It is necessary to consider the specific features of the tested products and conduct a detailed data analysis, which helps prevent false positives and improves testing accuracy.

References

[1] Santhanam P. Quality management of machine learning systems //Engineering Dependable and Secure Machine Learning Systems: Third International Workshop, EDSMLS 2020, New York City, NY, USA, February 7, 2020, Revised Selected Papers 3. – Springer International Publishing, 2020. – pp. 1-13.

[2] Fujii G. et al. Guidelines for quality assurance of machine learning-based artificial intelligence //International Journal of Software Engineering and Knowledge Engineering. – 2020. – vol. 30. – No. 11n12. – pp. 1589-1606.

[3] Narita K. et al. Qunomon: A FAIR testbed of quality evaluation for machine learning models //2021 28th Asia-Pacific Software Engineering Conference Workshops (APSEC Workshops). – IEEE, 2021. – pp. 21-24.

[4] Liu M., Chakrabarty K. Adaptive methods for machine learning-based testing of integrated circuits and boards //2021 IEEE International Test Conference (ITC). – IEEE, 2021. – pp. 153-162.

[5] Mulla N., Jayakumar N. Role of Machine Learning & Artificial Intelligence Techniques in Software Testing //Turkish Journal of Computer and Mathematics Education (TURCOMAT). – 2021. – Vol. 12. – No. 6. – pp. 2913-2921.

[6] Bajaj Y., Samal M. K. Accelerating Software Quality: Unleashing the Power of Generative AI for Automated Test-Case Generation and Bug Identification //International Journal for Research in Applied Science and Engineering Technology. – 2023. – vol. 11. – No. 7.

[7] Artificial intelligence-based analysis of eye movement for detecting abnormal human behavior: international patent application WO 2023/043493. Applicant: One Trust AI Inc. Published: 30 March 2023. Available at: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2023043493. (accessed 07.11.2024).

[8] Regression testing phase [Electronic resource]. Access mode: https://www.ibm.com/docs/en/devops-test-workbench/11.0.0?topic=playback-regression-testing-phase (accessed 07.11.2024).