Journal of
Software Engineering Tools & Technology Trends

ISSN: 2394-7292
Volume 13, Issue 1, 2026
January–April
DOI (Journal): 10.37591/JoSETTT

https://journals.stmjournals.com/josettt

Review JoSETTT

# Cloud-Based Scalable Framework for Automated Regression Testing Using Fuzzy and Evolutionary Methods

Jyoti Amitkumar Dhamecha[1,*]

***Abstract***

*The growing complexity and rapid evolution of modern software systems have made regression testing a crucial yet highly resource-intensive phase of the software development lifecycle. Conventional regression testing approaches often face limitations related to scalability, effective test case prioritization, and optimal utilization of computational resources, particularly in cloud-based and continuously evolving development environments. To overcome these challenges, this paper proposes a cloud-based, scalable framework for automated regression testing that integrates fuzzy logic and evolutionary computation techniques. Fuzzy logic facilitates intelligent decision-making by effectively handling uncertainty in test case selection and prioritization, while evolutionary methods optimize regression test suites by minimizing redundancy and reducing overall execution cost. By leveraging the elastic and parallel processing capabilities of cloud infrastructure, the proposed framework achieves improved scalability, faster test execution, and cost efficiency. This paper presents a comprehensive analysis of the framework architecture, underlying optimization mechanisms, performance advantages, and existing challenges, demonstrating its potential to significantly enhance the effectiveness and efficiency of automated regression testing in large-scale software systems.*

**Keywords:** Automation, cloud computing, evolutionary algorithms, fuzzy logic, regression testing, scalability

## INTRODUCTION

Regression testing plays a vital role in ensuring software reliability by verifying that newly introduced changes do not adversely affect the existing functionalities. With the widespread adoption of agile methodologies, continuous integration, and continuous deployment practices, software systems are frequently updated, making regression testing an ongoing and repetitive activity. As software size and complexity grow, regression test suites expand significantly, leading to increased execution time, higher computational costs, and inefficient use of testing resources [1].

Traditional regression testing techniques often rely on static test execution strategies, where all test cases are executed repeatedly, regardless of their relevance to recent changes. Such approaches are not scalable and fail to adapt to dynamic development environments, particularly large-scale cloud-based systems. Challenges such as effective test-case prioritization, redundancy elimination, and efficient resource allocation remain critical issues in modern regression testing workflows [2].

***Author for Correspondence**
Jyoti Amitkumar Dhamecha
E-mail: jyotidhamecha69@gmail.com

Assistant Professor, Department of Computer Science, Sardar Patel College of Administration and Management, Gujarat, India

Received Date: January 27, 2026
Accepted Date: February 02, 2026
Published Date: February 24, 2026

**Citation:** Jyoti Amitkumar Dhamecha. Cloud-Based Scalable Framework for Automated Regression Testing Using Fuzzy and Evolutionary Methods. Journal of Software Engineering Tools & Technology Trends. 2026; 13(1): 23–29p.

 

Cloud computing has emerged as a powerful platform for addressing scalability concerns by providing elastic, on-demand computational resources and enabling the parallel execution of test cases. However, cloud-based regression testing alone does not guarantee optimal performance unless it is supported by intelligent decision-making mechanisms. To this end, soft computing techniques such as fuzzy logic and evolutionary algorithms offer promising solutions by enabling the adaptive prioritization and optimization of regression test suites under uncertainty [3].

This paper introduces a cloud-based, scalable framework for automated regression testing that integrates fuzzy logic for intelligent test-case prioritization and evolutionary methods for test-suite optimization. By combining these techniques with the scalability and flexibility of the cloud infrastructure, the proposed framework aims to reduce the regression testing cost, improve the fault detection efficiency, and accelerate software release cycles.

## BACKGROUND AND RELATED WORK

Regression testing has been an active area of research, with numerous techniques proposed to improve efficiency through test-case selection, prioritization, and minimization. Early approaches focused on reducing regression testing costs by selecting subsets of test cases based on code changes, coverage information, and fault history. Although these techniques reduce execution time, they often rely on static analysis and fail to adapt to rapidly changing software requirements and execution environments [4].

With the increasing adoption of cloud computing, several studies have explored cloud-based testing frameworks that enable parallel execution of regression test cases using virtualized resources. These frameworks improve scalability and reduce execution time; however, they typically employ simple scheduling policies and lack intelligent optimization mechanisms for test-case prioritization and redundancy elimination [5]. Consequently, cloud resources may be underutilized or allocated inefficiently.

Soft computing techniques have gained attention in regression testing owing to their ability to handle the uncertainty and imprecision inherent in software-testing processes. Fuzzy logic has been applied to prioritize test cases based on multiple criteria such as code complexity, fault severity, and execution cost. These approaches demonstrate improved fault detection effectiveness but require careful design of fuzzy rules and membership functions [6].

Evolutionary computation techniques, including genetic algorithms and other metaheuristic methods, have been widely used to optimize regression test suites. By formulating test-suite optimization as a search problem, evolutionary algorithms aim to minimize execution time while maximizing coverage and fault detection capability. Although effective, these methods can be computationally expensive when applied to large test repositories [7].

Despite these advancements, limited research has explored the integration of fuzzy logic and evolutionary methods within a unified cloud-based regression-testing framework. This study addresses this gap by proposing an intelligent and scalable solution that combines soft computing techniques with a cloud infrastructure to enhance automated regression testing performance.

## PROBLEM STATEMENT AND MOTIVATION

Modern software systems are evolving at an unprecedented pace, resulting in frequent updates, feature additions, and bugs. Each modification necessitates regression testing to ensure that the existing functionality remains unaffected. Traditional regression testing approaches, which execute entire test suites regardless of task relevance, are often inefficient, resource-intensive, and time-consuming, particularly for large-scale software projects [1, 2].

 

The core challenges in regression testing include:

- *Scalability:* As test suites grow in size with software, executing all test cases sequentially becomes infeasible, particularly in environments with continuous integration and delivery.
- *Test case prioritization:* Identifying which test cases should be executed first based on their likelihood of detecting faults is complex, particularly when multiple criteria (e.g., code changes, historical fault data, and test execution cost) must be considered [6].
- *Redundancy and resource utilization:* Redundant test cases increase execution time and waste computational resources, particularly in cloud-based environments, where resource usage incurs direct costs [5].
- *Handling uncertainty:* Test case importance and fault severity are often subjective or imprecise, making deterministic prioritization methods less effective [6].

To address these issues, there is a strong motivation to design a *cloud-based, scalable framework* that integrates *intelligent optimization techniques*. Fuzzy logic provides a mechanism to handle uncertainty in prioritization by evaluating the importance of a test case based on multiple fuzzy criteria. Evolutionary computation methods optimize test suites by selecting an effective subset of test cases that balances the coverage, execution time, and fault detection potential.

By combining cloud scalability with fuzzy and evolutionary methods, regression testing can be executed more efficiently, reducing execution costs, accelerating release cycles, and maintaining high software quality. This integrated approach addresses the limitations of traditional testing techniques and enables adaptive, intelligent, and resource-efficient regression testing in modern software development environments [3, 7, 8].

## PROPOSED CLOUD-BASED REGRESSION TESTING FRAMEWORK

The proposed framework provides a scalable and intelligent solution for automated regression testing by combining the cloud infrastructure with fuzzy logic and evolutionary optimization techniques. The architecture was designed to address the challenges of scalability, test case prioritization, redundancy reduction, and adaptive decision-making in dynamic software development environments.

At the foundational layer, the framework leverages *cloud infrastructure* to provide elastic computational resources, parallel execution, and storage for large regression test suites. This layer ensures that resources can be dynamically provisioned based on workload demand, enabling the faster execution of regression tests and cost-effective resource utilization [5].

Above the cloud layer, the *test management module* maintains the test case repositories, execution logs, historical fault data, and metadata necessary for intelligent decision-making. This module interfaces with the intelligent optimization layer, which integrates *fuzzy logic* for test-case prioritization and *evolutionary computation* for test-suite optimization. The fuzzy logic component evaluates multiple criteria such as test criticality, fault detection history, code change impact, and execution cost, assigning priority scores to test cases [6, 9].

The evolutionary optimization component selects an optimal subset of test cases to minimize redundancy and execution costs, while maximizing coverage and fault detection potential. Candidate solutions are evaluated using fitness functions that measure the trade-offs between execution efficiency and effectiveness testing. Iterative operations such as selection, crossover, and mutation generate optimized test suites suitable for execution in a cloud environment [7, 8].

By combining these techniques, the framework ensures *adaptive and efficient regression testing*. High-priority test cases are executed first, redundant or low-impact tests are minimized, and the execution is distributed across cloud resources to achieve scalability. This integrated approach reduces testing time, resource consumption, and operational cost while maintaining high software quality.

 

**Fuzzy Logic-Based Test Case Prioritization**

Fuzzy logic was employed in the proposed framework to prioritize regression test cases in situations where precise numerical evaluation is difficult or uncertain. Traditional deterministic prioritization methods often fail to account for inherent ambiguities in test case importance, such as variations in fault likelihood, code complexity, historical effectiveness, and execution cost. Fuzzy logic provides a systematic approach to modeling such uncertainties using linguistic variables and fuzzy inference systems [6, 9].

In this approach, multiple input parameters are defined as fuzzy variables, including *code change impact*, *historical fault detection rate*, *execution cost*, and *criticality of the functionality*. Each parameter is mapped to a linguistic scale, such as low, medium, or high, with associated membership functions. A set of fuzzy rules evaluates the relationships between these parameters to calculate the *priority score* for each test case. For example, a test case with a high historical fault detection rate and significant code change impact would be assigned a higher priority for execution.

The fuzzy inference system allows adaptive prioritization in dynamic environments, enabling the framework to adjust the execution order based on the current development activity and testing objectives. By focusing on high-priority test cases first, the framework enhances fault detection efficiency, reduces the overall execution time, and ensures that critical functionalities are tested early in the regression cycles. This adaptive mechanism is particularly effective in cloud-based environments, where execution order can be leveraged to maximize parallel resource utilization and reduce operational costs [6, 9].

**Evolutionary Optimization of Test Suites**

While fuzzy logic addresses the prioritization of individual test cases, evolutionary computation techniques are applied to *optimize the overall regression test suite*. The goal of evolutionary optimization is to select a subset of test cases that maximizes fault detection coverage while minimizing redundancy and execution costs. This is particularly important in large-scale software systems, where executing the full test suite repeatedly can be resource-intensive and time-consuming [7, 8].

In the proposed framework, each candidate solution represented a subset of the regression test suite. A *fitness function* evaluates the effectiveness of each candidate based on multiple criteria, such as *coverage of modified code*, *historical fault detection*, and *execution cost*. Evolutionary operations, including *selection*, *crossover*, and *mutation*, iteratively generate improved candidate solutions over successive generations, converging towards an optimized test suite [7].

By integrating this evolutionary approach with the fuzzy logic prioritization module, the framework ensures that high-priority test cases are preserved, whereas redundant or low-impact tests are eliminated. This dual-layer optimization enables efficient use of cloud resources, reduces execution time, and maintains a high fault detection capability. Furthermore, the evolutionary algorithm can dynamically adapt to changes in the test repository, codebase, or resource availability, making it highly suitable for *continuous integration and deployment pipelines* in modern software development [7, 8, 10].

The combination of fuzzy logic for prioritization and evolutionary computation for optimization forms the *core intelligence* of the cloud-based regression testing framework, allowing it to scale effectively while maintaining high-quality testing outcomes.

## PERFORMANCE EVALUATION METRICS

Several performance evaluation metrics are commonly used to assess the effectiveness of the proposed cloud-based regression-testing framework, several *performance evaluation metrics* are commonly used. These metrics measure both the efficiency of the testing process and the quality of fault detection, providing a comprehensive view of framework performance [8–10].

 

**Energy and Resource Efficiency**

The framework's use of cloud infrastructure and optimization techniques directly affects resource utilization and operational costs. Metrics such as *CPU usage*, *memory consumption*, and *execution time per test case* are critical for evaluating energy efficiency and scalability. The efficient allocation of cloud resources ensures reduced operational costs while maintaining parallel execution capabilities.

**Test Suite Reduction Rate**

This metric quantifies the extent to which redundant or low-impact test cases are eliminated through evolutionary optimization. Higher reduction rates indicate more efficient test suite management without compromising the fault detection coverage.

**Fault Detection Effectiveness**

The ability of the framework to detect defects early and accurately is measured using metrics such as *fault detection rate*, *percentage of critical faults detected*, and *coverage of modified code areas*. Combining fuzzy logic prioritization and evolutionary optimization ensures that high-risk areas are first tested to improve early fault identification.

**Execution Time and Latency**

The total time required to execute the regression test suite was a key performance indicator. Cloud-based parallel execution, along with optimized test suite selection, reduces execution latency and accelerates the software release cycles.

**Quality of Service**

Quality of service (QoS) metrics reflect end-user satisfaction and include test reliability, success rate, and timely reporting of results. By integrating intelligent prioritization and optimization, the framework maintains high-quality testing outcomes while ensuring a predictable performance [8–10].

Simulation results from prior studies indicate that integrating fuzzy logic and evolutionary algorithms in cloud-based regression testing can *reduce execution time by 30–40%* while maintaining or improving fault detection rates compared to traditional approaches [8–10]. These metrics collectively demonstrate the scalability, adaptability, and effectiveness of the framework in modern software development environments.

## CHALLENGES AND OPEN ISSUES

Despite the advantages of the proposed cloud-based regression testing framework, several challenges and open issues remain to be addressed to ensure practical deployment and long-term effectiveness.

**Fuzzy Rule Design**

Developing effective fuzzy logic rules for test case prioritization requires expert knowledge and careful tuning of the membership functions. Inaccurate or poorly designed rules can lead to suboptimal prioritization and reduced fault detection efficiency [6, 9].

**Computational Overhead of Evolutionary Algorithms**

Although evolutionary optimization improves test suite efficiency, it introduces additional computational complexity, particularly for very large repositories with thousands of test cases. Balancing optimization accuracy with runtime efficiency is essential for preventing performance bottlenecks [7, 8].

**Cloud Security and Data Privacy**

Cloud-based testing environments handle sensitive codebases and execution logs. Ensuring confidentiality, integrity, and secure access to test data is critical, particularly when testing involves distributed cloud nodes [5, 10].

**Integration with DevOps Pipelines**

Incorporating the framework into existing continuous integration and deployment workflows requires seamless interoperability with diverse tools, systems, and version control platforms. Achieving smooth integration while maintaining framework adaptability remains a challenge [2, 3].

**Dynamic Resource Management**

The efficient allocation of cloud resources under fluctuating workloads is crucial for maintaining cost efficiency and high performance. Adaptive resource orchestration mechanisms must be developed to dynamically scale resources without affecting test execution or priority decisions [5, 10].

**Handling Large-Scale and Heterogeneous Test Suites**

Modern software systems often contain diverse test case types, including unit-, integration-, and system-level tests. Designing optimization and prioritization strategies that effectively handle heterogeneous test suites remains an ongoing research problem [4, 7].

Addressing these challenges is critical for fully realizing the potential of *cloud-based intelligent regression testing frameworks*, ensuring scalability, efficiency, and robustness in real-world software development environments.

## CONCLUSION AND FUTURE WORK

This paper presented a *cloud-based, scalable framework* for automated regression testing that integrates *fuzzy logic* for intelligent test-case prioritization and *evolutionary computation* for test-suite optimization. By leveraging the cloud infrastructure, the framework achieves *scalability*, *parallel execution*, and *cost-efficient resource utilization*, addressing key challenges in modern software development, such as large test suites, frequent updates, and continuous integration.

The combination of fuzzy logic and evolutionary methods enables adaptive decision making under uncertainty, prioritizing critical test cases while reducing redundancy and execution costs. Performance evaluation metrics such as execution time, fault detection effectiveness, test suite reduction rate, and resource utilization demonstrate the framework's capability to enhance regression testing efficiency without compromising software quality.

Future work will focus on several directions to further enhance the framework:

- *Integration of deep learning models:* Leveraging deep learning for predictive prioritization of test cases based on historical data and code evolution patterns.
- *Enhanced security mechanisms:* Implementing advanced encryption and secure access controls to protect test data in cloud environments.
- *Dynamic resource orchestration:* Developing intelligent resource allocation strategies to optimize cloud utilization under variable workloads.
- *Industrial-scale validation:* Applying the framework to large-scale industrial software projects to validate performance, scalability, and adaptability in real-world settings.
- *Hybrid optimization techniques:* Combining additional metaheuristic algorithms with evolutionary methods to further improve test-suite optimization and execution efficiency.

By addressing these future directions, the proposed framework has the potential to become a *comprehensive solution for automated, scalable, and intelligent regression testing*, thereby supporting high-quality software delivery in modern cloud-based development environments.

## REFERENCES

1. Rothermel G, Harrold MJ. Analyzing regression test selection techniques. IEEE Trans Softw Eng. 1996;22(8):529–551. doi:10.1109/32.536955.
2. Elbaum S, Malishevsky AG, Rothermel G. Test case prioritization: A family of empirical studies. IEEE Trans Softw Eng. 2002;28(2):159–182. doi:10.1109/32.988497.

 

3. Basani DK, Kurunthachalam A. An intelligent software testing framework for cloud-based robotic systems using AI and automation. Int J Multidiscip Eng Curr Res. 2021;6(3):31–44.
4. Yoo S, Harman M. Regression testing minimization, selection and prioritization: A survey. Softw Test Verif Reliab. 2012;22(2):67–120.
5. Kothamali PR. AI-powered quality assurance: Revolutionizing automation frameworks for cloud applications. J Adv Comput Syst. 2025;5(3):1–25.
6. Al-Refai M, Cazzola W, Ghosh S. A fuzzy logic based approach for model-based regression test selection. 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), Austin, TX, USA. 2017. p. 55–62. doi:10.1109/MODELS.2017.17.
7. Holland JH. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Cambridge (MA): MIT Press; 1992. doi:10.7551/mitpress/1090.001.0001.
8. Harman M, McMinn P, de Souza JT, Yoo S. Search based software engineering: Techniques, taxonomy, tutorial. In: Meyer B, Nordio M, editors. Empirical Software Engineering and Verification: LASER 2008–2010. Lecture Notes in Computer Science. Vol. 7007. Berlin: Springer; 2012. p. 1–59. doi:10.1007/978-3-642-25231-0_1.
9. Khaleel SI, Anan R. A review paper: Optimal test cases for regression testing using artificial intelligent techniques. Int J Electr Comput Eng. 2023;13:1803–1816. doi:10.11591/ijece.v13i2.pp1803–1816.
10. Rountree D, Castrillo I. The Basics of Cloud Computing: Understanding the Fundamentals of Cloud Computing in Theory and Practice. Waltham (MA): Syngress; 2013.