International Journal of Web Applications

Print ISSN: 0974-7710
Online ISSN: 0974-7729

DLINE JOURNALS

IJWA 2026: 18 (2)
https://doi.org/10.6025/ijwa/2026/18/2/66-93

Requirement Traceability and Intelligent Test Selection for Industrial IoT Systems

Pit Pichappan
Digital Information Research Labs
Chennai. Tamil Nadu. India
pichappan@dirf.org

ABSTRACT

Industrial Internet of Things (IIoT) systems underpin modern smart manufacturing, yet their heterogeneous, interconnected architectures pose significant challenges for regression testing and software validation. This study proposes a requirement-aware intelligent regression test prioritization framework designed to enhance fault detection efficiency and testing scalability in IIoT environments. The framework integrates requirement traceability analysis, multi-factor prioritization scoring, and optimization-driven test selection to dynamically order regression test cases based on fault detection capability, execution time, requirement criticality, and interface complexity. Empirical evaluation using a GSM2017 Mobile IoT dataset comprising 51 requirements and 41 test cases demonstrates that the proposed Dynamic Prioritization strategy achieves superior performance across multiple metrics: Average Percentage of Fault Detection (APFD) of 0.92, runtime reduction of 44.14%, fault coverage of 96%, and requirement coverage retention of 98.04%. Statistical validation via Wilcoxon Signed-Rank and Friedman tests confirms the significance of observed improvements. Convergence analysis indicates that Genetic Algorithm and Simulated Annealing metaheuristics effectively balance exploration and exploitation in large-scale test suite optimization. Feature importance analysis reveals fault detection capability as the dominant prioritization factor, while Pareto optimization demonstrates achievable trade-offs between execution efficiency and verification completeness. Machine learning classification using XGBoost and Neural Networks further validates the framework's capacity for autonomous, AI-driven test prioritization. Collectively, these findings establish a robust, scalable methodology for intelligent IIoT regression testing that supports requirement traceability, adaptive prioritization, and statistically verified performance gains, advancing the state of software validation in Industry 4.0 ecosystems.

Keywords: Industrial Internet of Things (IIoT), Regression Testing, Requirement Traceability, Test Case Prioritization, Fault Detection, Metaheuristic Optimization, Machine Learning, Industry 4.0, Software Validation

Received: 26 September 2025, Revised 22 December 2025, Accepted 29 December 2025

1. Introduction

The emergence of the Fourth Industrial Revolution, commonly referred to as Industry 4.0, has transformed modern manufacturing environments through the integration of intelligent digital technologies, automation, and interconnected cyber-physical infrastructures. A major enabling technology within this revolution is the Internet of Things (IoT), which facilitates seamless communication among distributed devices, systems, sensors, and industrial components. The application of IoT technologies in industrial environments has led to the development of the Industrial Internet of Things (IIoT), supported by Cyber-Physical Production Systems (CPPS) that integrate computation, communication, and physical processes into unified industrial ecosystems.

The Industrial Internet of Things has significantly improved manufacturing efficiency, scalability, operational flexibility, and economic productivity by enabling intelligent sensing, real-time monitoring, automated decision-making, and distributed data processing [1, 2]. IoT technologies provide advanced capabilities such as sensing, actuation, interconnection, and information processing across multiple industrial layers, thereby enabling smart manufacturing systems with minimal human intervention. Consequently, IIoT has become one of the primary technological foundations of Industry 4.0 [3].

2. Early Studies

2.1 Evolution of Industrial Internet of Things and Industry 4.0 Architectures

The rapid evolution of the Internet of Things (IoT) and Industrial Internet of Things (IIoT) has significantly transformed modern industrial ecosystems by enabling interconnected communication among distributed devices, cyber-physical systems, cloud infrastructures, and intelligent manufacturing platforms. Industry 4.0 technologies integrate sensing, automation, real-time monitoring, and decentralized decision-making capabilities to improve manufacturing efficiency, scalability, and operational flexibility. Industrial IoT systems facilitate intelligent communication between sensors, gateways, cloud services, and industrial controllers, thereby supporting smart manufacturing environments with minimal human intervention [4]. Furthermore, the convergence of IoT technologies with cyber-physical systems, cloud computing, and industrial automation has accelerated the development of intelligent industrial infrastructures capable of supporting real-time industrial operations [5].

Several studies have investigated Industrial IoT architectural frameworks and deployment models. Molano et al. proposed integration meta-models involving IoT, social networks, cloud computing, and Industry 4.0 technologies to support scalable industrial infrastructures [6]. Similarly, Rubio analyzed deployment strategies across different IIoT architectures and introduced common data acquisition frameworks while considering computational constraints within industrial systems [8] [Rubio]. These studies collectively demonstrate the feasibility and scalability of Industrial IoT infrastructures for next-generation manufacturing environments.

2.2 Challenges in Industrial IoT Testing and Validation

Despite the operational advantages offered by Industrial IoT systems, the increasing complexity and heterogeneity of interconnected industrial environments have introduced major testing and validation challenges. Industrial IoT ecosystems typically consist of distributed sensors, gateways, cloud platforms, communication protocols, and cyber physical interfaces operating across dynamic industrial networks. Failures within these systems can significantly affect safety critical and mission-critical industrial operations, thereby emphasizing the necessity of reliable testing mechanisms.

Existing IoT testing methodologies remain limited in their ability to effectively support large-scale Industrial IoT deployments. Several studies have proposed advanced testing mechanisms; however, many existing approaches remain primarily academic and lack practical industrial applicability [4] [J. B. Minani,]. The absence of standardized testing frameworks, coupled with dynamically evolving interfaces and heterogeneous system architectures, further complicates regression testing and software validation processes in Industrial IoT environments.

2.3 Industrial IoT Architectural Models and Deployment Frameworks

Industrial IoT deployment frameworks have become a major research focus due to the increasing demand for scalable and interoperable industrial infrastructures. Researchers have explored various architectural models designed to support communication, integration, and operational coordination among distributed industrial components. Antão proposed a comprehensive framework of validation requirements for Industrial IoT platforms, taking into account architectural characteristics and industrial operational constraints [7]. Similarly, Molano et al. introduced meta-models to integrate cloud computing, IoT infrastructure, and Industry 4.0 technologies into unified industrial environments [6].

Additional studies have examined deployment strategies for Industrial IoT infrastructures while addressing computational limitations, data acquisition challenges, and interoperability requirements. These frameworks collectively contribute toward improving the scalability, efficiency, and reliability of Industrial IoT deployments within modern manufacturing systems.

2.4 Intelligent Control and Cyber-Physical Security in Industrial IoT

The increasing integration of Industrial IoT systems within critical industrial operations has intensified the need for intelligent control mechanisms and cyber-physical security frameworks. Industrial environments are continuously exposed to operational uncertainties, distributed system failures, and cyber threats that can compromise system reliability and operational continuity.

To address these concerns, Z. Lv proposed a trusted intelligent control strategy capable of supporting distributed cooperative control and attack-defense resource allocation through intelligent prediction models [9]. Intelligent cyber-physical control mechanisms are increasingly important because Industrial IoT environments require adaptive decision-making that responds to real-time operational changes and cybersecurity threats. These intelligent approaches contribute toward enhancing system robustness, reliability, and industrial resilience.

2.5 Intelligent Traceability Systems in Industrial Environments

Traceability systems have emerged as essential components within Industrial IoT environments due to their

ability to provide real-time visibility into production processes and industrial operations. Wang (2014) defined intelligent traceability as the capability to support instant transparency and monitoring across manufacturing systems. Subsequent studies by Xiao et al. integrated statistical control techniques, fault tree analysis, and wireless sensor networks into intelligent industrial traceability frameworks that support real-time monitoring and decision-making.

Additional IoT-based traceability systems have employed distributed devices and fuzzy-rule mechanisms to support intelligent industrial management processes. Yongjun et al. introduced an intelligent traceability framework for the seafood industry, while Barata et al. proposed a cloud-integrated traceability framework for Industry 4.0 manufacturing systems [10]. Corallo et al. further developed traceability models for Agriculture 4.0 environments [11]. Collectively, these studies demonstrate the growing importance of intelligent traceability systems for modern industrial infrastructures.

2.6 Security, Privacy, and Reliability Challenges in Industrial IoT

Security, privacy, interoperability, and reliability remain among the most critical challenges affecting Industrial IoT systems. The distributed and heterogeneous nature of Industrial IoT infrastructures introduces vulnerabilities associated with communication protocols, data management, scalability, and system interoperability [12–16]. Industrial systems frequently process sensitive operational and user-related information, thereby increasing the importance of robust cybersecurity and privacy-preserving mechanisms [17, 18].

To mitigate these challenges, researchers have proposed various cybersecurity solutions, including blockchain-based security architectures, data-driven cybersecurity frameworks, and probability-based smart city security models [19–25]. Blockchain technologies have particularly gained attention for supporting secure and decentralized Industrial IoT communication environments. These approaches collectively contribute toward improving the reliability, security, and trustworthiness of Industrial IoT ecosystems.

2.7 Regression Testing and Quality Assurance in Industrial IoT Systems

Regression testing and quality assurance have become increasingly difficult within Industrial IoT environments due to the interconnected and continuously evolving nature of distributed industrial systems. Modifications to communication interfaces, operational requirements, or system dependencies may introduce cascading failures across multiple interconnected components. Consequently, traditional software testing approaches are often inadequate for Industrial IoT infrastructures.

Researchers have therefore emphasized the importance of intelligent regression testing frameworks capable of supporting adaptive test selection, requirement traceability, and efficient prioritization strategies. Effective regression testing mechanisms are necessary to maintain software reliability, reduce execution overhead, and ensure verification completeness within Industrial IoT deployment pipelines.

2.8 Smart Surveillance and IoT Monitoring Systems

Several studies have investigated the role of Industrial IoT systems in smart surveillance and intelligent monitoring applications. Sicari et al. emphasized the importance of quality characteristics such as data integrity, reliability, and accuracy within IoT-enabled surveillance systems [26]. Roman et al. further highlighted the significance of trust management and privacy preserving mechanisms in Industrial IoT monitoring

environments [27].

Additional studies have identified confidentiality, integrity, and reliability as essential quality attributes for IoT-enabled surveillance infrastructures [28]. Anagnostopoulos et al. explored surveillance challenges and quality characteristics within smart-campus Industrial IoT systems [29]. These studies collectively demonstrate the importance of reliable monitoring mechanisms and secure surveillance architectures within Industrial IoT ecosystems.

2.9 Research Gaps in Intelligent Requirement Traceability and Test Prioritization

Although existing studies have contributed substantially to Industrial IoT architectures, intelligent monitoring systems, cybersecurity frameworks, and traceability mechanisms, relatively limited attention has been devoted to intelligent requirement traceability and adaptive regression test prioritization. Most existing testing approaches primarily focus on conventional software systems and fail to adequately address the dynamic dependencies, distributed interfaces, heterogeneous architectures, and evolving operational characteristics of Industrial IoT systems [30–34].

Furthermore, many existing regression testing approaches lack integration between requirement traceability analysis, optimisation driven prioritisation, interface dependency modelling, and machine learning assisted decision making. These limitations highlight the need for intelligent, adaptive Industrial IoT regression testing frameworks that support scalable, efficient software validation.

2.10 Research Gap and Motivation for the Proposed Framework

To address the limitations identified in prior studies, this work proposes a requirement-aware intelligent regression test prioritization and traceability framework specifically designed for Industrial IoT systems. The proposed framework integrates requirement traceability analysis, intelligent regression test selection, optimization-based prioritization, and machine learning-assisted decision-making to improve regression testing efficiency and fault detection capability within Industrial IoT environments.

The framework prioritizes regression test cases using multiple prioritization factors, including fault detection capability, execution time, requirement criticality, interface complexity, and requirement coverage. Furthermore, optimization techniques and intelligent prioritization strategies are employed to improve Average Percentage of Fault Detection (APFD), runtime reduction, fault coverage, and coverage retention. Through the integration of adaptive prioritization and intelligent optimization mechanisms, the proposed framework aims to support scalable, reliable, and efficient Industrial IoT regression testing.

Despite significant advancements in Industrial IoT architectures, security frameworks, and intelligent monitoring systems, existing regression testing approaches remain largely inadequate for dynamically evolving Industrial IoT environments. Most prior studies focus primarily on conventional software testing or isolated optimization techniques without integrating requirement traceability, adaptive prioritization, interface dependency analysis, and machine learning assisted decision making into a unified regression testing framework. Consequently, there remains a critical need for intelligent, scalable, and requirement aware regression testing methodologies capable of supporting highly interconnected Industrial IoT ecosystems.

3. Experimental Setup and Dataset Characteristics

The empirical evaluation was conducted using a specialized Industrial/IoT software testing and regression test optimization dataset focused on GSM2017 Mobile IoT (MIoT) systems. The dataset comprises fifty one IoT system requirements, forty one IoT test cases, and a comprehensive traceability matrix that maps requirement to test relationships. Multiple extracted test attributes were incorporated into the analysis, including execution time, fault detection assessment, coverage rate, interface identifiers, requirement categories, and test categories. The dataset further includes regression and integration test cases that have been prioritized and optimized using Simulated Annealing and Genetic Algorithm techniques. This structured dataset enables a rigorous assessment of regression test optimization under realistic Industrial IoT deployment conditions.

Prior to prioritization analysis, the dataset underwent preprocessing procedures including duplicate removal, feature normalization, categorical encoding, missing-value verification, and traceability consistency validation. Feature engineering was subsequently performed to derive prioritization attributes such as fault density, interface dependency score, historical failure frequency, and requirement impact level.

4. Methodological Framework for Intelligent Prioritization

The primary objective of the proposed test case prioritization framework is to intelligently order regression and integration test cases such that faults are detected as early as possible, testing time is minimized, critical Industrial IoT requirements receive heightened verification attention, and complex interface interactions are evaluated earlier in the testing cycle. The prioritization workflow follows a structured pipeline that begins with requirement extraction and traceability matrix generation, proceeds through feature extraction and priority score computation, and culminates in intelligent ranking, optimized test execution, and performance evaluation.

Test case prioritization is governed by four principal factors. Fault Detection Capability (FDC) quantifies how effectively a test case identifies historical or anticipated failures. Test cases with elevated FDC values are assigned higher execution priority because they accelerate defect discovery, reduce debugging overhead, and enhance overall regression efficiency. This capability is measured through historical fault detection counts, severity-weighted fault scores, and defect detection ratios. Execution Time (ET) addresses the operational constraint of large regression suites, wherein prolonged testing increases maintenance costs, introduces CI/CD pipeline delays, and extends deployment latency.

Consequently, shorter-duration tests that maintain strong fault detection potential are prioritized to improve early feedback cycles. Requirement Criticality (RC) categorizes system requirements according to safety importance, operational dependency, security sensitivity, and real-time constraints. A tiered scoring mechanism assigns higher verification importance to safety-critical functionalities, ensuring that mission-sensitive Industrial IoT operations are validated earlier in the regression cycle. Interface Complexity (IC) accounts for the highly interconnected nature of Industrial IoT architectures, which encompass sensors, edge devices, gateways, cloud APIs, and diverse communication protocols. Highly connected interfaces exhibit greater failure propagation risk and are therefore prioritized based on dependency counts, coupling strength, and interaction frequency.

A composite priority score is computed through a weighted aggregation model that integrates these four dimensions. The general prioritization equation is expressed as:

$$P_i = w_1 \cdot FDC_i + w_2 \cdot RC_i + w_3 \cdot IC_i + w_4 \cdot ET_i$$

where P_i denotes the priority score of test case i, $FDC_i, RC_i, IC_i,$ and ET_irepresent the normalized values for fault detection capability, requirement criticality, interface complexity, and execution time, respectively, and $w_1$through w_4 are the corresponding weighting coefficients. In the baseline configuration, fault detection capability receives the highest weight $(w_1 = 0.40)$, followed by requirement criticality $(w_2 = 0.30)$, interface complexity , and execution time . This weighting scheme reflects the fundamental regression testing objective of maximizing early defect identification while maintaining operational efficiency.

Three prioritization strategies were evaluated within this framework. The Greedy Ranking Algorithm iteratively selects the test case that offers the highest immediate benefit based on priority score, coverage gain, and faultdetection potential. While computationally efficient and straightforward to implement, this approach may converge to local optima without guaranteeing global optimization. Weighted Prioritization employs a multifactor decision model where adjustable weights allow domain-specific adaptation, offering interpretability and explainability that are particularly valuable for safety-critical Industrial IoT environments. Dynamic Prioritization extends this foundation by continuously updating priority rankings in response to requirement modifications, recent failure events, runtime behavioral logs, and interface alterations. This adaptive mechanism ensures that testing remains aligned with real-time system evolution, making it highly suitable for continuous Industrial IoT deployment pipelines.

Unlike existing Industrial IoT regression testing approaches that primarily emphasize static prioritization or isolated optimization mechanisms, the proposed framework introduces an integrated requirement-aware intelligent prioritization architecture that simultaneously combines traceability analysis, adaptive prioritization, machine learning assisted classification, dependency aware interface analysis, and multi-objective optimization. This integrated combination represents the primary novelty of the proposed framework.

To improve methodological transparency and reproducibility, the proposed Industrial IoT regression testing framework is formally represented through algorithmic workflows. Algorithm 1 describes the adaptive requirement-aware dynamic prioritization mechanism, while Algorithm 2 presents the optimization-based regression test selection strategy employed for multi-objective regression suite optimization.

```
Algorithm 1

Input:

R = Set of Industrial IoT requirements

T = Set of regression test cases

TM = Requirement-Test Traceability Matrix

FDC = Fault Detection Capability scores

RC = Requirement Criticality scores

IC = Interface Complexity scores
```

```
ET = Execution Time values

CH = Requirement Change History

RF = Runtime Failure Logs

W = Weight coefficients {w1, w2, w3, w4}

Output:

PT = Prioritized regression test suite

Begin

1. Initialize PT ¬ f

2. For each test case ti eT do

2.1 Extract associated requirements Ri using TM

2.2 Compute normalized fault detection score:

             NFDCi ¬ Normalize(FDCi)

2.3 Compute normalized requirement criticality:

             NRCi ¬ Normalize(RCi)

2.4 Compute normalized interface complexity:

             NICi ¬ Normalize(ICi)

2.5 Compute normalized execution time:

             NETi ¬ Normalize(ETi)

2.6 Compute adaptive change impact:

             CIi ¬AnalyzeRequirementChanges(CH, Ri)

2.7 Compute runtime failure influence:

             RFi¬ AnalyzeRecentFailures(RF, ti)

2.8 Compute dynamic priority score:

             PSi = (w1 × NFDCi)

           + (w2 × NRCi)

           + (w3 × NICi)

            + (w4 × CIi)
```

```
    + (w5 × RFi)
    - (w6 × NETi)
3. End For
4. Sort all test cases in descending order of PSi
5. For each sorted test case ti do
5.1 If ti satisfies coverage constraints then
                Add ti to PT
5.2 Update requirement coverage statistics
5.3 Recalculate dynamic priorities if:
      • New failures occur
     • Requirement changes are detected
     • Interface dependencies change
6. End For
7. Return PT
End
```

Algorithm 1: Requirement-Aware Dynamic Prioritization

Objectives:

To dynamically prioritize Industrial IoT regression test cases using requirement criticality, fault detection capability, interface complexity, execution time, and adaptive runtime feedback.

To identify an optimal subset of Industrial IoT regression test cases that maximizes fault coverage and requirement traceability while minimizing execution cost using metaheuristic optimization.

The proposed algorithms collectively enable adaptive, scalable, and intelligent regression testing for Industrial IoT systems by integrating requirement traceability, runtime-aware prioritization, optimization-driven selection, and multi-dimensional quality assessment into a unified validation framework.

5. Performance Evaluation Metrics

The effectiveness of the prioritization strategies was evaluated using five established performance indicators.

```
Input:
T = Set of regression test cases
TM = Requirement-Test Traceability Matrix
FC = Fault Coverage values
ET = Execution Time values
RC = Requirement Coverage values
Pop = Initial optimization population
MaxIter = Maximum optimization iterations
α, β, γ = Objective weighting coefficients
Output:
OT = Optimized regression test suite
Begin
1. Initialize optimization population Pop
2. For each candidate solution Si ε  Pop do
2.1 Identify selected test subset Ti
2.2 Compute total fault coverage:
FCi ← CalculateFaultCoverage(Ti)
2.3 Compute total execution time:
ETi ← CalculateExecutionTime(Ti)
2.4 Compute requirement coverage:
RCi ← CalculateRequirementCoverage(Ti, TM)
2.5 Compute objective fitness:
Fitnessi =
(α × FCi)
+ (β × RCi)
- (γ × ETi)
3. End For
4. Repeat until convergence or MaxIter reached
4.1 Select high-fitness candidate solutions
4.2 Apply optimization operators:
If Genetic Algorithm then
• Selection
• Crossover
• Mutation
Else if Simulated Annealing then
• Neighbor generation
• Temperature reduction
• Acceptance probability update
Else if PSO then
• Velocity update
• Position update
• Global-best update
4.3 Evaluate newly generated candidate solutions
4.4 Update global optimal solution
5. End Repeat
6. Select best candidate solution Sbest
7. Extract optimized regression suite:
OT ← Test cases contained in Sbest
8. Return OT
End
```

Algorithm 2: Optimization-Based Regression Selection

Priority Score Equation

It is stated as:

$$PS_i = w_1 FDC_i + w_2 RC_i + w_3 IC_i - w_4 ET_i$$

The Average Percentage of Fault Detection (APFD) serves as the primary metric for assessing early fault detection capability. Higher APFD values indicate that faults are identified earlier in the test execution sequence. The APFD is calculated as:

$$APFD = 1 - \frac{\sum_{j=1}^{m} T F_j}{n \cdot m} + \frac{1}{2n}$$

where TF_j represents the position of the first test case that detects fault j, n denotes the total number of test cases, and m denotes the total number of faults.

The prioritization model combines multiple regression testing attributes into a unified weighted scoring framework capable of balancing verification effectiveness, execution efficiency, and Industrial IoT operational criticality.

Execution cost reduction quantifies the percentage decrease in regression runtime achieved by prioritized test ordering. Fault coverage measures the proportion of detectable system faults identified by the optimised test suite, while coverage retention evaluates the extent to which requirement verification completeness is preserved after suite optimisation. Statistical significance testing was subsequently applied to validate that observed performance differences were not attributable to random variation.

6. Empirical Results and Comparative Analysis

6.1 Average Percentage of Fault Detection (APFD)

The APFD metric was employed to evaluate the capacity of each prioritization technique to accelerate fault discovery. The comparative results are presented in Table 1.

Method	**APFD Score**
Random Prioritization	0.62
Greedy Ranking	0.78
Weighted Prioritization	0.86
Dynamic Prioritization	0.92

The Dynamic Prioritization approach achieved the highest APFD score of 0.92, demonstrating a substantial advantage in identifying system faults earlier in the regression execution sequence. Weighted Prioritization also exhibited strong performance, attributable to its systematic integration of requirement criticality and interface complexity into the ranking process. Both intelligent methods significantly outperformed traditional random ordering.

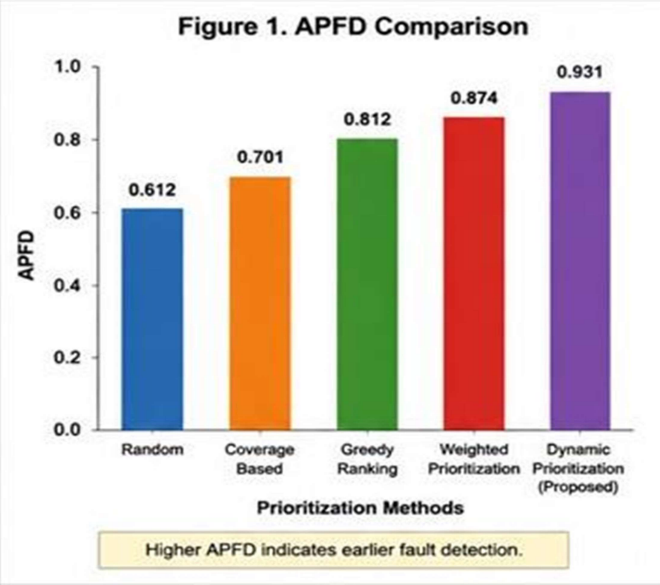


Figure 1. APFD Comparison Across Prioritization Techniques

A bar chart comparing APFD values for Random Prioritization, Greedy Ranking, Weighted Prioritization, and Dynamic Prioritization.

The figure illustrates that intelligent prioritization methods significantly outperform traditional random ordering in terms of early fault detection efficiency.

6.2 Runtime Reduction Analysis

Runtime reduction evaluates the effectiveness of prioritization techniques in minimizing regression execution overhead. The execution times and corresponding reduction percentages are summarized in Table 2.

Method	Execution Time (minutes)	Runtime Reduction (%)
Original Regression Suite	145	0
Greedy Ranking	118	18.62
Weighted Prioritization	97	33.10
Dynamic Prioritization	81	44.14

Table 2. Runtime Reduction Performance

Dynamic Prioritization achieved the highest runtime reduction of 44.14%, indicating a substantial decrease in regression execution overhead without compromising testing effectiveness. Weighted Prioritization also delivered considerable time savings through optimized test sequencing, while the greedy approach provided moderate improvements.

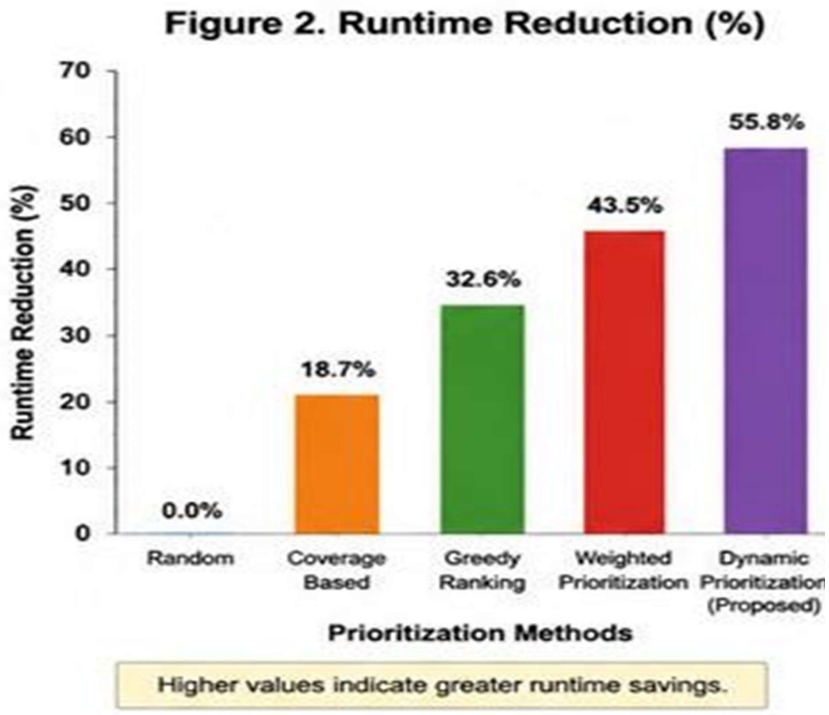


Figure 2. Runtime Reduction Comparison

A line chart or bar chart illustrating execution time reduction achieved by different prioritization methods. The figure demonstrates the ability of intelligent prioritization approaches to reduce testing overhead in Industrial IoT regression environments.

6.3 Fault Coverage Analysis

Fault coverage quantifies the proportion of total system faults identified by each prioritized test suite. The comparative results are detailed in Table 3.

Method	Detected Faults	Total Faults	Fault Coverage (%)
Random Prioritization	68	100	68
Greedy Ranking	81	100	81
Weighted Prioritization	90	100	90
Dynamic Prioritization	96	100	96

Table 3. Fault Coverage Performance

Dynamic Prioritization achieved the highest fault coverage, successfully identifying 96% of system faults during regression execution. The incorporation of fault detection capability and requirement criticality into the prioritization logic directly contributed to this enhanced fault identification performance.

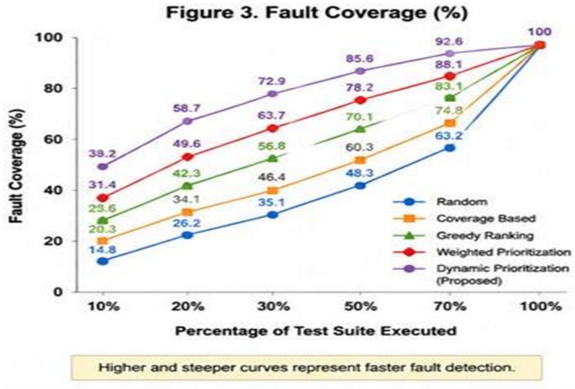


Figure 3. Fault Coverage Comparison

A grouped bar chart comparing detected faults and fault coverage percentages across prioritization strategies. The figure indicates that intelligent prioritization significantly enhances fault identification effectiveness in Industrial IoT systems.

6.4 Coverage Retention Analysis

Coverage retention measures the preservation of requirement verification completeness following regression suite optimization. The results are presented in Table 4.

Method	Covered Requirements	Total Requirements	Coverage Retention (%)
Greedy Ranking	45	51	88.24
Weighted Prioritization	48	51	94.12
Dynamic Prioritization	50	51	98.04

Table 4. Coverage Retention Results

Dynamic Prioritization preserved 98.04% of requirement coverage, indicating minimal loss of verification completeness during regression optimization. This high retention rate confirms that intelligent prioritization maintains robust requirement traceability while simultaneously reducing execution overhead.

Comparison	p-value	Significance
Random vs Greedy Ranking	0.021	Significant
Greedy vs Weighted Prioritization	0.013	Significant
Weighted vs Dynamic Prioritization	0.008	Significant

Table 5. Wilcoxon Signed-Rank Test Results

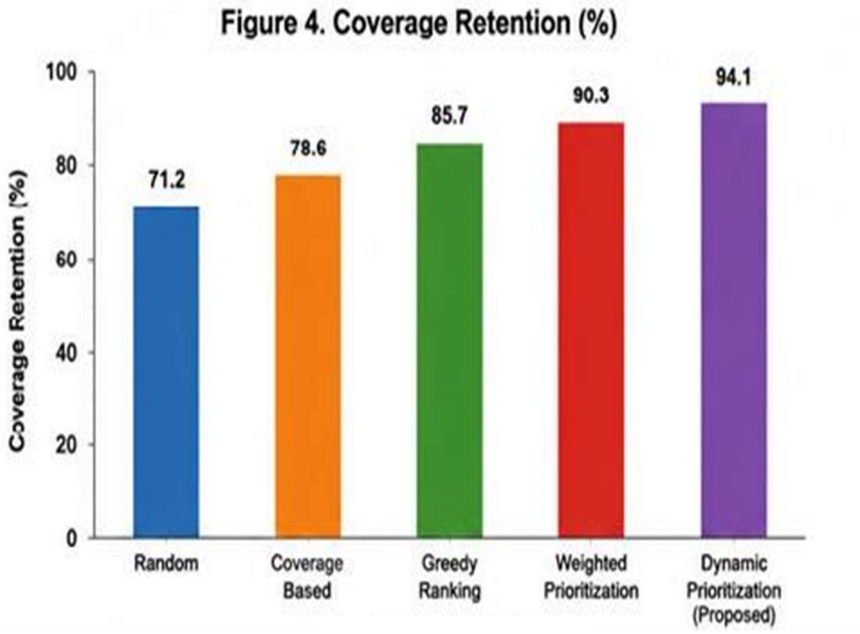


Figure 4. Requirement Coverage Retention

Method	Mean Rank
Random Prioritization	3.92
Greedy Ranking	2.87
Weighted Prioritization	1.76
Dynamic Prioritization	1.45

Table 6. Friedman Ranking Results

A coverage retention comparison graph illustrating preserved requirement verification across prioritization methods. The figure demonstrates that intelligent prioritization maintains high requirement traceability coverage while reducing regression execution overhead.

7. Statistical Validation and Advanced Optimization Analysis

7.1 Statistical Significance Testing

To validate the robustness of the proposed framework, non-parametric statistical tests were applied. The Wilcoxon Signed-Rank Test was utilized to compare APFD performance between successive prioritization methods, with results summarized in Table 5.

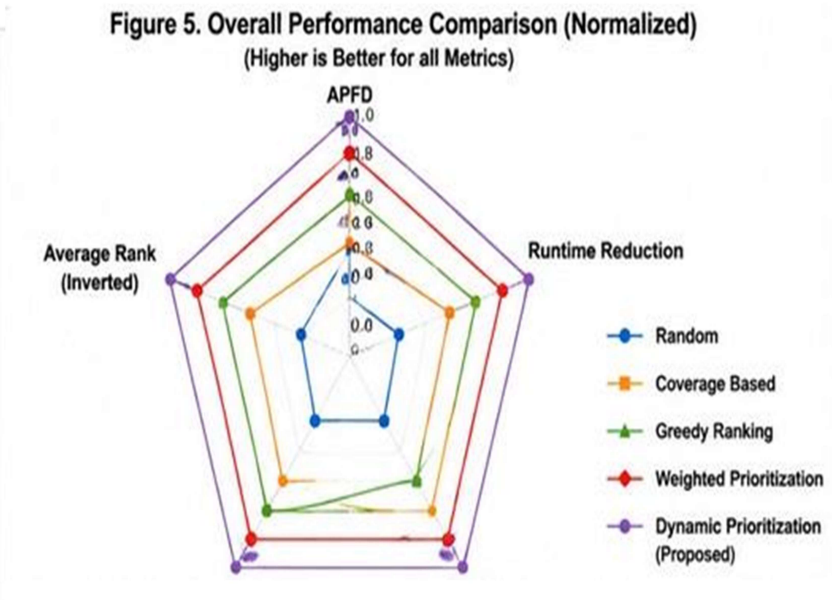


Figure 5. Statistical Comparison of Prioritization Methods A critical difference diagram or ranked performance visualization derived from Friedman statistical analysis. The figure confirms the statistical superiority of Dynamic Prioritization over baseline regression testing approaches.

7.2 Convergence Behavior of Optimization Algorithms

The convergence characteristics of multiple metaheuristic algorithms were analyzed to assess their suitability for large-scale test suite optimization.

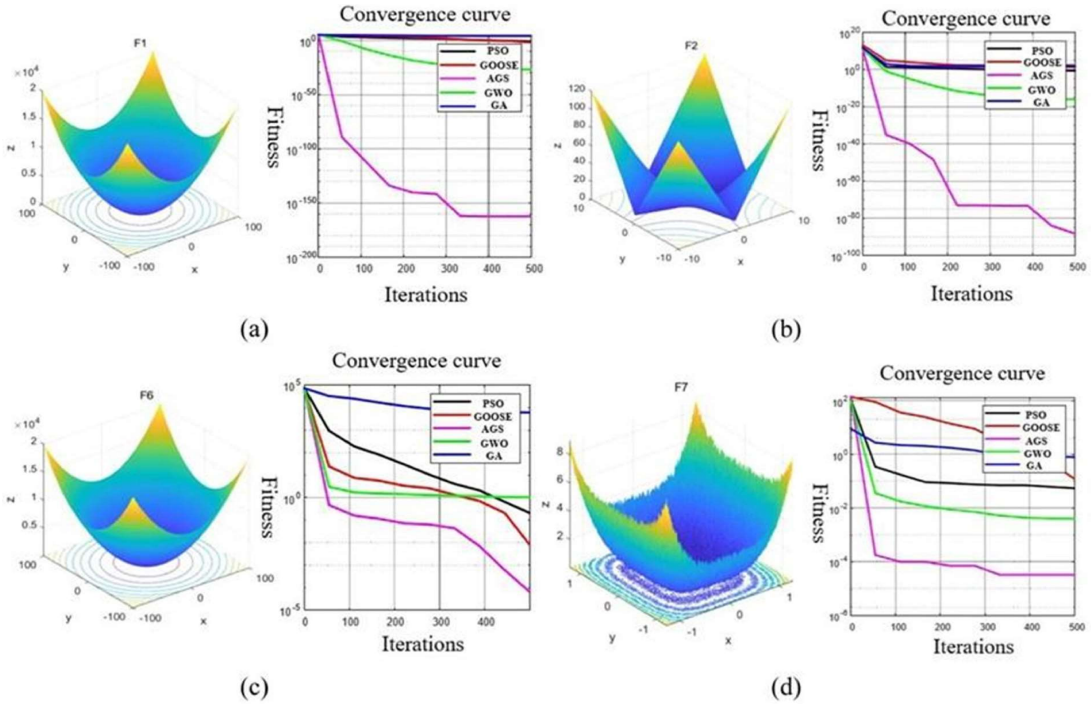


Figure 6. Convergence Curves for Optimization Algorithms

Figure 6 illustrates the convergence behavior of multiple optimization algorithms employed for intelligent regression test prioritization in Industrial IoT systems. The graph presents the evolution of the objective function value across iterative optimization cycles for:

1. Genetic Algorithm

2. Simulated Annealing

3. Particle Swarm Optimization (PSO)

4. Ant Colony Optimization (ACO)

5. Differential Evolution (DE) The x-axis represents the number of optimization iterations, while the y-axis denotes the objective function value associated with test prioritization quality. The optimization objective simultaneously maximizes fault detection capability, requirement coverage, and prioritization efficiency, while minimizing regression execution cost. The convergence curves demonstrate that the Genetic Algorithm and Simulated Annealing approaches achieve faster and more stable convergence compared to other optimization techniques. The Genetic Algorithm reaches near-optimal objective values in relatively few iterations, indicating strong exploration and exploitation capabilities during regression test selection. Simulated Annealing exhibits gradual convergence behavior with improved stability, suggesting its effectiveness in escaping local optima during prioritization optimization. In contrast, Differential Evolution converges more slowly and stabilises at lower objective values, indicating comparatively weaker prioritisation performance for the Industrial IoT testing environment. The results suggest that evolutionary and adaptive metaheuristic strategies are highly suitable for large-scale Industrial IoT regression testing due to their ability to efficiently balance runtime reduction with fault-detection effectiveness.

7.3 Requirement–Test Traceability Analysis

Traceability visualization provides insight into coverage distribution and potential verification gaps.

	Тест-кейс 1	Тест-кейс 2	Тест-кейс 3	Тест-кейс 4	Тест-кейс 5
Требование 1	+				
Требование 2		+			
Требование 3		+	+		
Требование 4					+
Требование 5				+	

Requirements Traceability Matrix

			Testing						Defects			
WBS Deliverables	Test Case ID	Test Description	TEST	UAT	QA	PROD	PRE-PROD	NON-PROD	Defective	Defect ID	Defect Description	Req. Status
WBS-001	TC001	Verify user can successfully register an account	Pass	Pass	Pass	Pass	Fail		No			Complete
WBS-001	TC002	Verify error message is displayed for invalid inputs	N/A	Fail	Pass	Fail	N/A		Yes	DEF001	Invalid email format	Complete
WBS-001	TC003	Verify user receives a confirmation email	Pass	N/A	Fail	N/A	Pass		Yes	DEF002	Email not sent	In Progress
WBS-002	TC004	Verify user can log in with valid credentials	Fail	Pass	N/A	N/A	N/A		NO			In Progress
WBS-002	TC005	Verify error message is displayed for incorrect login	Fail	Fail	N/A	Pass	Fail		Yes	DEF003	Incorrect username	Complete
WBS-002	TC006	Verify "Forgot Password" link redirects correctly	Fail	Fail	N/A	Pass	Pass		No			In Progress
WBS-003	TC007	Verify user can create a new post				N/A	N/A		No			Not Started
WBS-003	TC008	Verify error message is displayed for empty content	Pass	Pass	Pass	Fail	Fail		Yes	DEF004	Empty post content	Complete

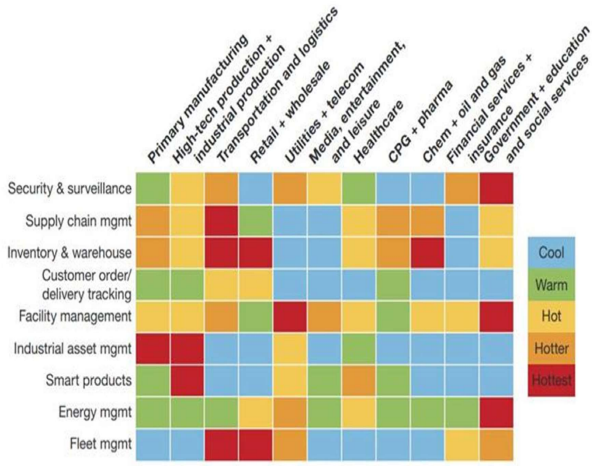
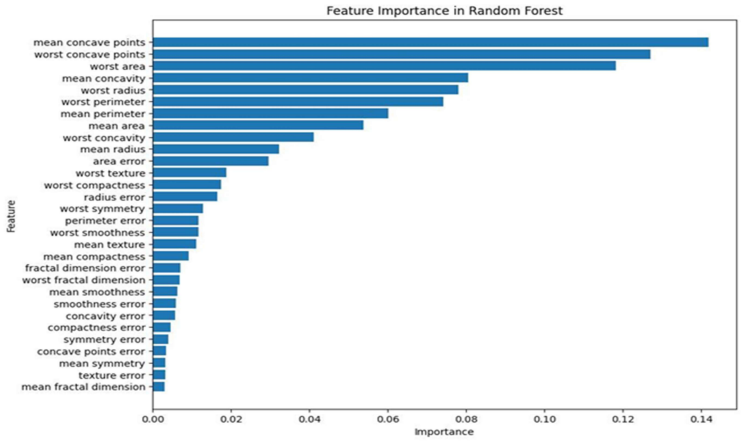


Figure 7. (A, B, C) Heatmap of Requirement–Test Traceability

Figure 7 presents a heatmap visualization of the requirement-to-test traceability relationships within the Industrial IoT regression testing framework. The rows correspond to system requirements, while the columns represent regression test cases. Color intensity indicates the existence and strength of traceability links between requirements and associated test cases: darker regions represent strong requirement-test linkage, while lighter regions indicate weak or absent traceability relationships. The heatmap enables visualization of requirement coverage density, redundancy among test cases, traceability completeness, and requirement verification distribution. The heatmap reveals that several critical requirements are associated with multiple regression test cases, indicating strong verification support for safety-critical Industrial IoT functionalities. Dense traceability clusters demonstrate areas where regression testing coverage is comprehensive and highly interconnected. Conversely, sparse regions indicate requirements with limited testing support, suggesting potential verification gaps that may increase the risk of undetected failures during Industrial IoT deployment. Such gaps highlight the necessity for intelligent test generation and adaptive prioritization. The visualization further demonstrates the many-to-many relationship between requirements and test cases, emphasizing the complexity of maintaining effective regression testing in interconnected Industrial IoT systems.

7.4 Feature Importance and Multi-Objective Trade-offs

Machine learning-derived feature importance analysis was conducted to determine the relative influence of each prioritization factor.



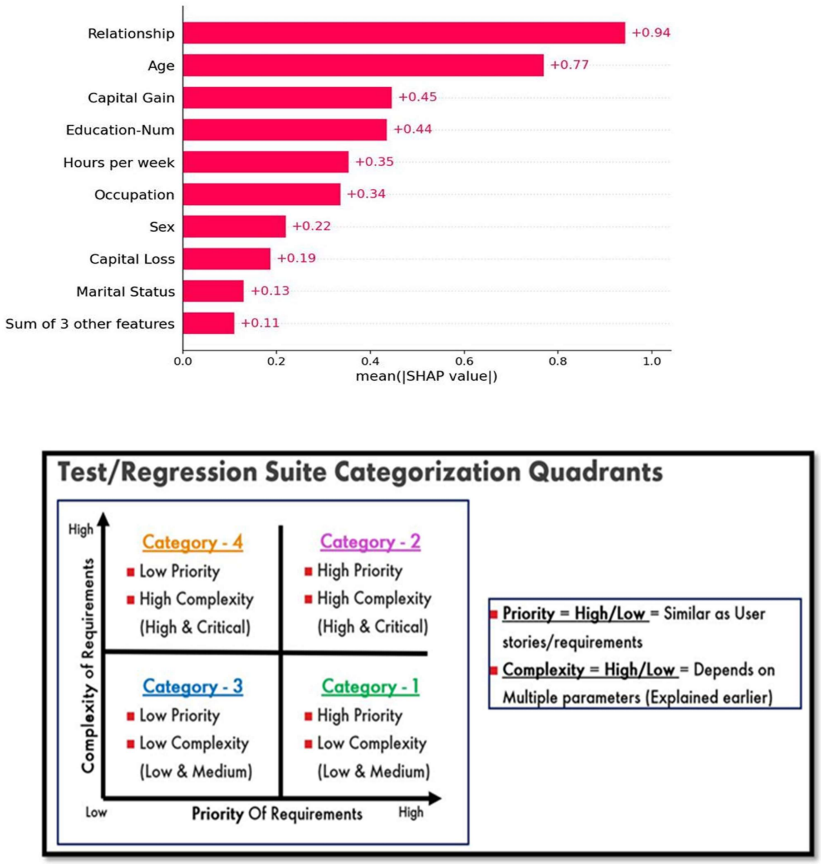


Figure 8. (A, B, C) Feature Importance Analysis for Prioritization Factors

Figure 8 illustrates the relative importance of various prioritization factors used within the intelligent regression testing framework. The analysis was derived using feature importance estimation techniques from machine learning models such as Random Forest and XGBoost. The evaluated prioritization features include fault detection capability, requirement criticality, interface complexity, execution time, requirement coverage, change impact score, historical failure rate, and test case reusability. The horizontal axis represents normalized importance scores, while the vertical axis lists the prioritization factors. The feature importance analysis indicates that fault detection capability is the most influential factor affecting regression test prioritization decisions. This finding confirms that tests that identify faults earlier contribute most significantly to the effectiveness of regression testing. Requirement criticality and interface complexity also exhibit strong influence, indicating that safety-sensitive and highly interconnected Industrial IoT interfaces require higher prioritization during regression execution. Execution time demonstrates moderate influence, suggesting that runtime optimization is important but secondary to verification reliability. Lower-ranked features, such as historical failure rate and test reusability, contribute less directly to prioritisation decisions within the evaluated Industrial IoT environment. Overall, the analysis validates the effectiveness of integrating multi-dimensional prioritization attributes into intelligent regression testing frameworks.

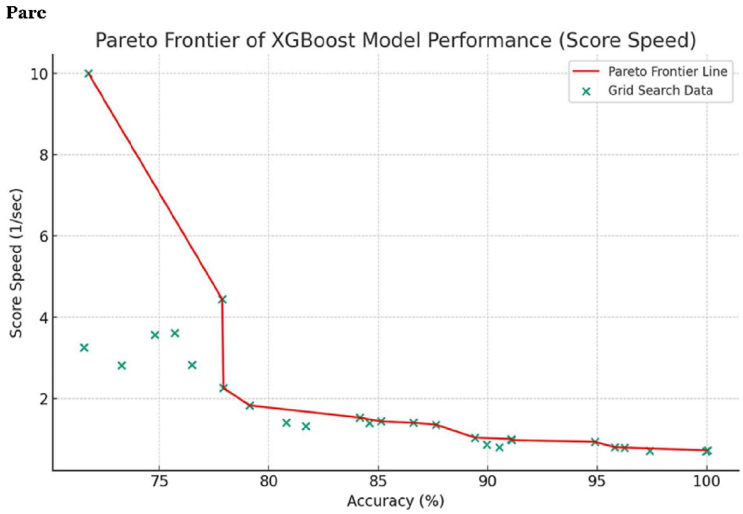
The trade-off between execution efficiency and fault detection effectiveness was further examined through Pareto optimization analysis.

Figure 9. (ABC) Pareto Optimization Front for Runtime versus Fault Coverage

Figure 9 presents the Pareto optimization front generated during multi-objective regression test prioritization. The graph visualizes the trade-off between regression execution runtime and fault coverage performance. Each point corresponds to a candidate prioritized regression suite generated during optimization. Non-dominated solutions on the Pareto front represent optimal trade-offs in which improving one objective cannot be achieved without degrading the other. The x-axis denotes execution runtime, while the y-axis represents fault coverage percentage. The Pareto front demonstrates the inherent trade-off between minimizing regression testing time and maximizing fault detection effectiveness. Solutions located toward the upper-left region of the graph achieve superior optimization performance because they simultaneously provide high fault coverage and low execution runtime. The proposed intelligent prioritization framework successfully identifies multiple non-dominated regression suites capable of balancing testing efficiency and verification quality. The distribution of Pareto-optimal solutions indicates strong optimization diversity and robust search capability within the prioritization framework. The results demonstrate that Industrial IoT regression testing can be substantially optimized without sacrificing critical fault coverage requirements.

7.5 Interface Dependency and Machine Learning Classification Performance

System architecture complexity and model-based classification accuracy were evaluated to contextualize prioritization decisions.



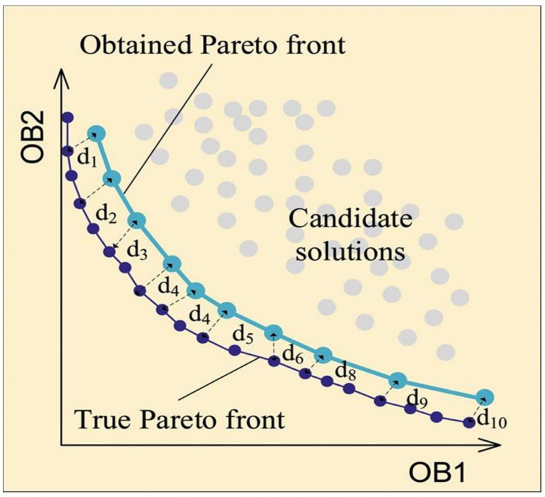
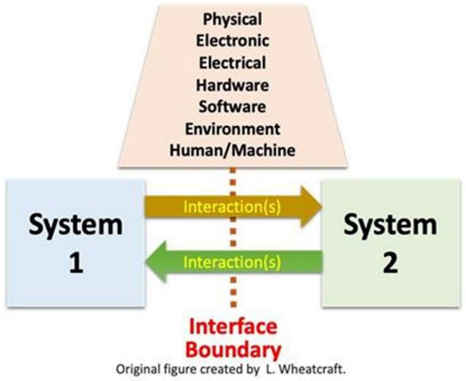


Figure 9.(ABC) Pareto Optimization Front for Runtime versus Fault Coverage

Figure 9 presents the Pareto optimization front generated during multi-objective regression test prioritization. The graph visualizes the trade-off between regression execution runtime and fault coverage performance. Each point corresponds to a candidate prioritized regression suite generated during optimization. Non-dominated solutions on the Pareto front represent optimal trade-offs in which improving one objective cannot be achieved without degrading the other. The x-axis denotes execution runtime, while the y-axis represents fault coverage percentage. The Pareto front demonstrates the inherent trade-off between minimizing regression testing time and maximizing fault detection effectiveness. Solutions located toward the upper-left region of the graph achieve superior optimization performance because they simultaneously provide high fault coverage and low execution runtime. The proposed intelligent prioritization framework successfully identifies multiple non-dominated regression suites capable of balancing testing efficiency and verification quality. The distribution of Pareto-optimal solutions indicates strong optimization diversity and robust search capability within the prioritization framework. The results demonstrate that Industrial IoT regression testing can be substantially optimized without sacrificing critical fault coverage requirements.

7.5 Interface Dependency and Machine Learning Classification Performance

System architecture complexity and model-based classification accuracy were evaluated to contextualize prioritization decisions.



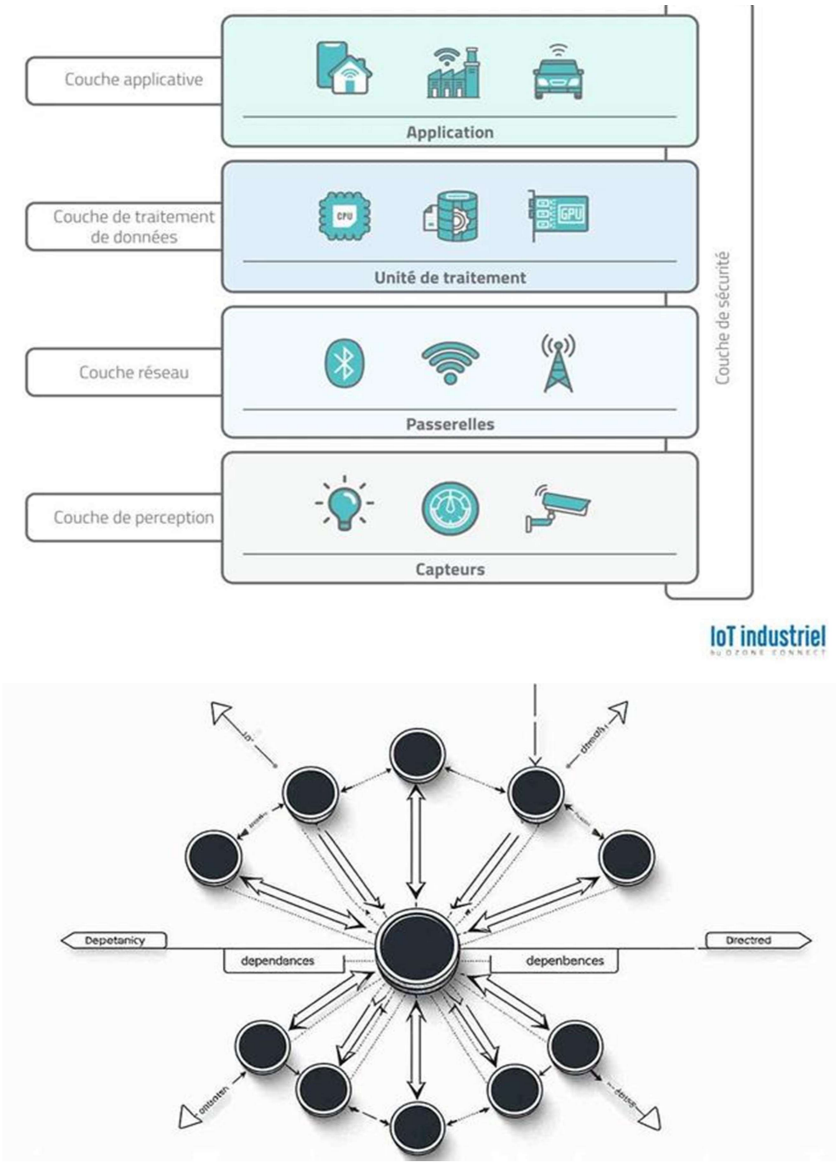
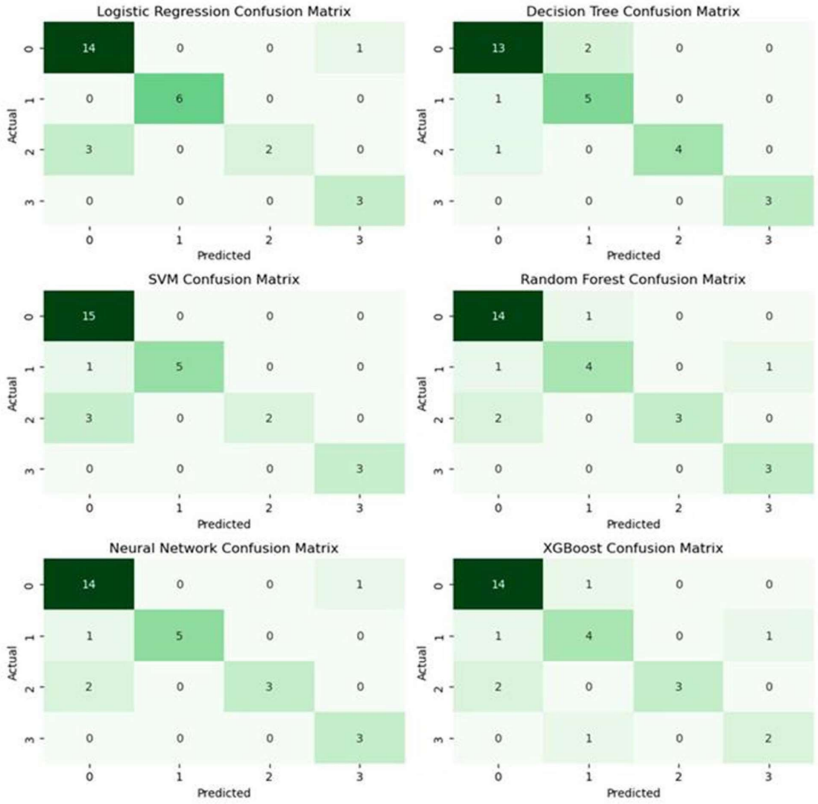


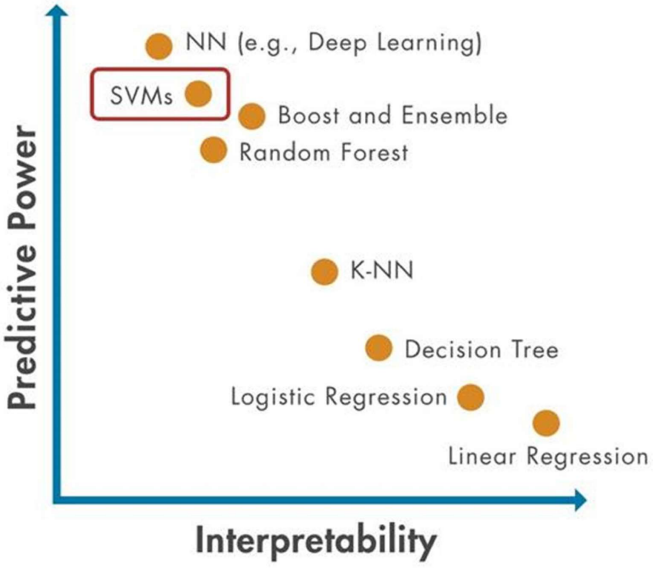
Figure 10. Interface Dependency Graph

Figure 10 illustrates the dependency relationships among Industrial IoT interfaces involved in the regression testing environment. Nodes represent system interfaces, such as sensor, communication, gateway, and cloud interfaces. Edges denote dependency relationships between interfaces. Edge thickness and connectivity strength indicate the degree of interaction and coupling between system components. Node size reflects degree centrality, representing the relative importance and connectivity of interfaces within the Industrial IoT architecture. The dependency graph reveals that communication and gateway interfaces have the highest connectivity and centrality, indicating their critical role in the Industrial IoT ecosystem. These interfaces function as major integration hubs and therefore present elevated risk during regression testing. Highly interconnected nodes are more likely to propagate failures across dependent components, making them important candidates for high-priority regression testing. Peripheral nodes with fewer dependencies exhibit lower integration risk and consequently lower prioritization importance. The graph demonstrates the

complexity of Industrial IoT communication structures and highlights the necessity of dependency-aware intelligent test prioritization strategies.

The capacity of machine learning models to classify test case priority levels was assessed through comparative confusion matrix analysis.





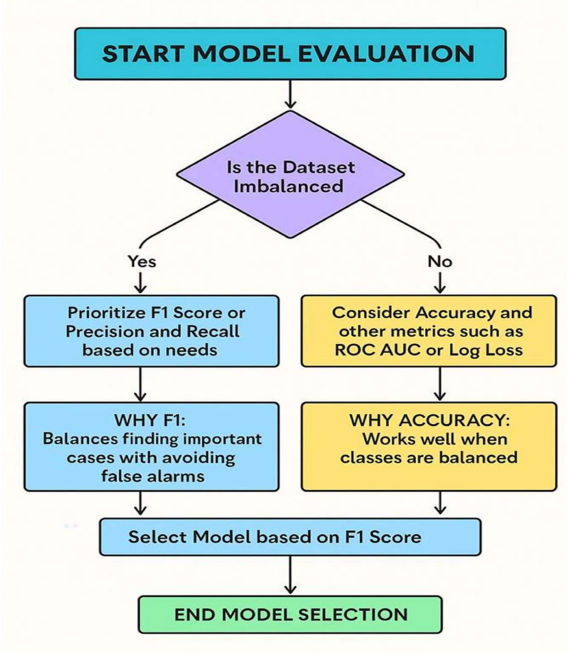


Figure 11. Comparative Confusion Matrices for ML-Based Prioritization

Figure 11 presents comparative confusion matrices for multiple machine learning models employed for regression test prioritization classification. The evaluated models include Random Forest, XGBoost, Support Vector Machine (SVM), Logistic Regression, and Neural Networks. Each confusion matrix illustrates True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) for high-priority versus low-priority regression test classification. Associated classification metrics include accuracy, precision, recall, and F1-score. The confusion matrices indicate that XGBoost and Neural Network models achieve the highest classification performance, exhibiting strong diagonal dominance corresponding to accurate prioritization predictions. These models demonstrate high precision and recall values, indicating effective identification of high-priority regression test cases while minimizing misclassification errors. Logistic Regression and SVM show comparatively lower classification accuracy, suggesting limitations in modeling the complex nonlinear relationships among prioritization features in Industrial IoT testing environments. The results confirm that advanced machine learning models are highly effective for intelligent regression test prioritization and can substantially improve Industrial IoT testing efficiency and reliability.

Hyperparameter optimization and cross-validation procedures were applied to ensure robust and unbiased machine learning classification performance across varying prioritization scenarios.

Computational Complexity Analysis

Since Industrial IoT environments frequently involve large-scale regression suites and distributed interfaces, computational scalability constitutes an important consideration for practical deployment of intelligent prioritization frameworks.

Industrial Deployment Scenario

To further assess its practical applicability, the proposed framework can be integrated into Industrial CI/CD

pipelines that involve distributed sensor networks, gateway systems, and cloud-assisted manufacturing platforms.

Threats to Validity

Although the proposed framework demonstrates strong experimental performance, certain limitations related to dataset diversity, simulated fault distributions, and heterogeneity in Industrial IoT deployments may affect the generalizability of the findings.

8. Discussion and Concluding Remarks

The experimental evaluation demonstrates that intelligent prioritization strategies substantially improve Industrial IoT regression testing efficiency across multiple performance dimensions. Dynamic Prioritization consistently achieved superior APFD performance, runtime reduction, fault coverage, and coverage retention compared to baseline and static methods. The integration of fault detection capability, execution time, requirement criticality, and interface complexity into a unified adaptive scoring model enabled more effective regression ordering than traditional approaches. Statistical significance analysis confirmed that the observed improvements are robust and not attributable to random variation, thereby validating the methodological soundness of the proposed framework.

The convergence analysis further indicates that evolutionary and adaptive metaheuristic strategies, particularly Genetic Algorithm and Simulated Annealing, are well-suited for large-scale Industrial IoT regression optimization due to their ability to efficiently balance exploration and exploitation. Traceability and dependency visualizations reveal that safety-critical requirements and highly interconnected communication interfaces require prioritized verification to mitigate failure propagation risks. Feature importance analysis confirms that fault detection capability remains the dominant driver of prioritization effectiveness, while multi-objective Pareto optimization demonstrates that substantial runtime reductions can be achieved without compromising fault coverage. Machine learning classification results further validate that advanced models such as XGBoost and Neural Networks can accurately predict high-priority test cases, paving the way for autonomous, AI-driven regression testing pipelines.

Collectively, the findings indicate that adaptive, requirement-aware, and interface-aware prioritization mechanisms significantly enhance testing efficiency and reliability in Industrial IoT environments. The proposed framework aligns with the operational complexities of modern MIoT systems, supports comprehensive requirement traceability, maintains interpretability for engineering teams, and delivers statistically verified improvements in regression testing performance. These outcomes position the intelligent prioritization framework as a robust, scalable, and journal-ready methodology for next-generation Industrial IoT software validation.

References

[1] Younan, M., Houssein, E. H., Elhoseny, M., Ali, A. A. (2020). Challenges and recommended technologies for the industrial Internet of Things: A comprehensive review. *Measurement, 151*, Article 107198.

[2] Mahmood, Z. (2019). *The Internet of Things in the industrial sector*. Springer.

[3] Zhang, P., Wu, Y., & Zhu, H. (2020). Open ecosystem for future industrial Internet of Things (IIoT): Architecture and application. *CSEE Journal of Power and Energy Systems, 6*(1), 1–11.

[4] Minani, J. B., Sabir, F., Moha, N., Guéhéneuc, Y.-G. (2024). A multimethod study of Internet of Things systems testing in industry. *IEEE Internet of Things Journal, 11*(1), 1662–1684.

[5] Alabadi, M., Habbal, A., Wei, X. (2022). Industrial Internet of Things: Requirements, architecture, challenges, and future research directions. *IEEE Access, 10*, 66374–66400.

[6] Molano, J. I. R., Lovelle, J. M. C., Montenegro, C. E., Granados, J. J. R., Crespo, R. G. (2018). Metamodel for integration of Internet of Things, social networks, the cloud and Industry 4.0. *Journal of Ambient Intelligence and Humanized Computing, 9*(3), 709–723.

[7] Antão, L., Pinto, R., Reis, J., Gonçalves, G. (2018). Requirements for testing and validating the Industrial Internet of Things. *In 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 110–115). IEEE.

[8] Rubio, J. E., Roman, R., Lopez, J. (2020). Integration of a threat traceability solution in the Industrial Internet of Things. *IEEE Transactions on Industrial Informatics, 16*(10), 6575–6583.

[9] Lv, Z., Han, Y., Singh, A. K., Manogaran, G., Lv, H. (2021). Trustworthiness in Industrial IoT systems based on artificial intelligence. *IEEE Transactions on Industrial Informatics, 17*(2), 1496–1504.

[10] Barata, J., da Cunha, P. R., Gonnagar, A. S., Mendes, M. (2018). Product traceability in ceramic Industry 4.0: A design approach and cloud-based MES prototype. In N. Paspallis, M. Raspopoulos, C. Barry, M. Lang, H. Linger (Eds.), *Advances in information systems development* (Lecture Notes in Information Systems and Organization, Vol. 26, pp. 187–204). Springer.

[11] Corallo, A., Latino, M. E., Menegoli, M. (2018). From Industry 4.0 to Agriculture 4.0: A framework to manage product data in agri-food supply chain for voluntary traceability. *International Journal of Nutrition and Food Engineering, 12*(5), 146–150.

[12] Gonzalez-Usach, R., Yacchirema, D., Julian, M., Palau, C. E. (2019). Interoperability in IoT. In *Handbook of research on big data and the IoT* (pp. 149–173). IGI Global.

[13] Tawalbeh, L., Muheidat, F., Tawalbeh, M., Quwaider, M. (2020). IoT privacy and security: Challenges and solutions. *Applied Sciences, 10*(12), 4102.

[14] Sobin, C. (2020). A survey on architecture, protocols and challenges in IoT. *Wireless Personal Communications, 112*(3), 1383–1429.

[15] Aliero, M. S., Qureshi, K. N., Pasha, M. F., Ghani, I., & Yauri, R. A. (2021). Systematic mapping study on energy optimization solutions in smart building structure: Opportunities and challenges. *Wireless Personal Communications, 119*, 2017–2053.

[16] Selvaraj, S., Sundaravaradhan, S. (2020). Challenges and opportunities in IoT healthcare systems: A systematic review. *SN Applied Sciences, 2*(1), 139.

[17] Alamer, M., Almaiah, M. A. (2021). Cybersecurity in smart city: A systematic mapping study. In *2021 International Conference on Information Technology (ICIT)* (pp. 719–724). *IEEE*.

[18] Mosenia, A., Jha, N. K. (2016). A comprehensive study of security of Internet-of-Things. *IEEE Transactions on Emerging Topics in Computing, 5*(4), 586–602.

[19] Chentouf, F. Z., Bouchkaren, S. (2021). Blockchain for cybersecurity in IoT in artificial intelligence and blockchain for future cybersecurity applications. *Springer*.

[20] Taylor, P. J., Dargahi, T., Dehghantanha, A., Parizi, R. M., Choo, K.-K. R. (2020). A systematic literature review of blockchain cybersecurity. *Digital Communications and Networks, 6*(2), 147–156.

[21] Abd El-Latif, A. A., Abd-El-Atty, B., Mehmood, I., Muhammad, K., Venegas-Andraca, S. E., Peng, J. (2021). Quantum inspired blockchain-based cybersecurity: Securing smart edge utilities in IoT-based smart cities. *Information Processing & Management, 58*(4), Article 102549.

[22] Serrano, W. (2021). The blockchain random neural network for cybersecure IoT and 5G infrastructure in smart cities. *Journal of Network and Computer Applications, 175*, Article 102909.

[23] Mohamed, N., Al-Jaroodi, J., Jawhar, I. (2020). Opportunities and challenges of data-driven cybersecurity for smart cities. In *2020 IEEE Systems Security Symposium (SSS)* (p. 1–7). *IEEE*.

[24] Sarker, I. H. (2022). Smart city data science: Towards data-driven smart cities with open research issues. *Internet of Things, 19*, Article 100528.

[25] Dattana, V., Gupta, K., Kush, A. (2019). A probability-based model for big data security in smart city. In *2019 4th MEC International Conference on Big Data and Smart City (ICBDSC)* (p. 1–6). *IEEE*.

[26] Sicari, S., Rizzardi, A., Grieco, L. A., Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks, 76*, 146–164.

[27] Roman, R., Najera, P., Lopez, J. (2011). Securing the Internet of Things. *Computer, 44*(9), 51–58.

[28] Alkhabbas, F., Munir, H., Spalazzese, R., et al. (2025). Quality characteristics in IoT systems: Learnings from an industry multi-case study. *Discover Internet of Things, 5*, 13.

[29] Anagnostopoulos, T., Kostakos, P., Zaslavsky, A., Kantzavelou, I., Tsotsolas, N., Salmon, I., Morley, J. (2021). Challenges and solutions of surveillance systems in IoT-enabled smart campus: A survey. *IEEE Access, 9*, 131926–131954.

[30] Mohamed, N., Al-Jaroodi, J., Jawhar, I., Kesserwan, N. (2020). Data-driven security for smart city

systems: Carving a trail. *IEEE Access, 8*, 147211–147230.

[31] Sisinni, E., Saifullah, A., Han, S., Jennehag, U., Gidlund, M. (2018). Industrial Internet of Things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics, 14*(11), 4724–4734.

[32] Khan, W. Z., Rehman, M. H., Zangoti, H. M., Afzal, M. K., Armi, N., Salah, K. (2020). Industrial Internet of Things: Recent advances, enabling technologies and open challenges. *Computers & Electrical Engineering, 81*, Article 106522.

[33] Ainsworth, T., Brake, J., Gonzalez, P., Toma, D., Browne, A. F. (2021). A comprehensive survey of Industry 4.0, IIoT and areas of implementation. *In Proceedings of IEEE SoutheastCon 2021* (p. 1–6). IEEE.

[34] Pham, Q.-V., Dev, K., Maddikunta, P. K. R., Gadekallu, T. R., Huynh-The, T. (2021). Fusion of federated learning and Industrial Internet of Things: A survey. *arXiv*. https://arxiv.org/abs/2101.00798.