SPECTRUM OF ENGINEERING SCIENCES
ISSN (E) 3007-3138 (P) 3007-312X VOLUME 4, ISSUE 8, 2026

# BEYOND CODE GENERATION: A REPOSITORY-AWARE LARGE LANGUAGE MODEL AGENT FOR AUTOMATED SOFTWARE DEFECT DETECTION, REPAIR, AND EXECUTION-BASED VALIDATION

**Mateen Ali**

*School of Computer Science and Informatics, De Mont Fort University Leicester UK*

mateenali6000@gmail.com

DOI: https://doi.org/10.5281/zenodo.24131466

**Keywords**
*AI coding agent; automated program repair; software debugging; repository awareness; large language models; defect detection; execution-based validation*

**Article History**
*Received: 01 February 2026*
*Accepted: 16 July 2026*
*Published: 27 August 2026*

**Copyright** @Author
**Corresponding Author:** *
**Mateen Ali**

**Abstract**
*The increasing complexity of software repositories, dependencies, and interactions among components makes debugging and maintenance increasingly demanding. Conventional debugging relies heavily on developer expertise and manual investigation, while many large language model (LLM)-based coding assistants operate with limited repository context and do not consistently validate proposed changes. This study presents the design, implementation, and controlled evaluation of a repository-aware AI coding agent for automated defect detection and repair. The proposed system integrates repository analysis, error understanding, LLM-based reasoning, repair suggestion, execution-based validation, and performance evaluation in a modular workflow. It was implemented using Python, Git-based repository management, and an LLM API, and evaluated using a CMake-based C/C++ repository containing 21 analysed files and two seeded defects. The full agent identified both seeded defects without false positives or false negatives, achieving 100% precision, 100% recall, and a 100% F1-score. The generated repair changed the repository from a failing automated-test state to a passing state. Repository analysis processed all 21 selected files, identified two code findings, and reported an overall repository score of 100/100 with a reliability score of 98%. In an LLM-only comparison conducted over three repetitions, recall remained 100%, but 13 false positives reduced precision to 31.58% and the F1-score to 48.00%. An efficiency comparison retained all findings and used seven model requests in both configurations; the efficient configuration reduced execution time by 15.86%, while token consumption increased from 10,305 to 10,640 (approximately 3.25%). The findings indicate that repository context and execution-based validation can improve the reliability of AI-assisted debugging, although the small, controlled evaluation limits claims about generalisation.*

## INTRODUCTION

Software debugging is essential to software reliability and correctness. As repositories expand in size and complexity, the number of dependencies, execution paths, and interactions among components increases the time and expertise required to identify and resolve defects (Sommerville, 2021; Pressman and Maxim, 2020). Conventional debugging methods include code inspection, fault localisation, failure reproduction, tracing, breakpoint analysis, patch implementation, and testing (Zeller, 2009; Myers, Sandler and

Badgett, 2011). These practices remain important, but they can become resource-intensive when a defect involves multiple files or dependencies.

LLM-based systems have advanced code generation, completion, explanation, and modification (Chen et al., 2021; Fan et al., 2023). However, code assistants often operate on limited fragments of a project and may not fully account for repository structure, dependencies, configuration, or expected behaviour. Consequently, a plausible or syntactically valid change may fail to resolve the underlying defect or may introduce compatibility and security problems (Pearce et al., 2025). Reliable debugging therefore requires more than code generation: it requires relevant project context and evidence that a proposed repair improves software behaviour.

Automated program repair (APR) research has explored search-based, template-based, and constraint-based approaches to generating patches (Monperrus, 2018; Gazzola, Micucci and Mariani, 2017). More recent repository-level agents extend code generation through planning, tool use, repository interaction, and execution feedback (Jimenez et al., 2024; Yang et al., 2024). Nevertheless, computational overhead, evaluation consistency, and dependable repair validation remain important challenges.

This research addresses these challenges through a modular AI coding agent that analyses repository structure, interprets diagnostic evidence, generates repair suggestions, and validates changes through testing and execution feedback. The study evaluates defect detection, repair behaviour, and computational efficiency, and compares the complete workflow with an LLM-only configuration.

### 1.1 Research aim and questions

The aim is to design, implement, and evaluate an efficient repository-aware AI coding agent capable of analysing software repositories, detecting programming defects, generating repair recommendations, and verifying proposed changes while monitoring computational requirements.

The study is guided by five questions: (1) How can existing AI-assisted debugging and APR approaches inform repository-level agent design? (2) How effectively can the agent analyse repositories and interpret source-code and error information? (3) To what extent does combining repository analysis, compiler/runtime feedback, documentation, reasoning, and validation improve debugging? (4) How effective is the agent in defect detection, repair, and computational efficiency? and (5) What limitations and future improvements affect practical application?

## 2. Related Work

Traditional debugging combines fault localisation, failure reproduction, program inspection, patching, and validation. Spectrum-based fault localisation ranks suspicious program elements by analysing their relationship with passing and failing tests (Abreu, Zoeteweij and Van Gemund, 2007; Wong et al., 2016). Although systematic, manual debugging depends on a developer's understanding of architecture and execution behaviour (Zeller, 2009).

APR automates some stages of defect correction. Search-based methods explore candidate changes and evaluate them against test suites; approaches such as GenProg use evolutionary search, while template-based approaches learn or apply recurring patch patterns (Le Goues et al., 2011; Kim et al., 2013; Martinez and Monperrus, 2019). Constraint-based methods formulate repair in terms of program specifications and execution constraints, but can incur substantial computational costs (Gazzola, Micucci and Mariani, 2017). Test-suite quality remains a central limitation because a patch can pass available tests without necessarily implementing the intended behaviour (Monperrus, 2018; Huang et al., 2024).

Machine-learning and deep-learning methods have supported defect prediction, code classification, code search, and code representation (Hall et al., 2011; Allamanis et al., 2018). Models such as code2vec, CodeBERT, and GraphCodeBERT learn representations that support code understanding (Alon et al., 2019; Feng et al., 2020; Guo et al., 2020). Code-generation models including Codex, CodeGen, StarCoder, and Code Llama have further expanded the range of programming tasks supported by LLMs (Chen et al., 2021; Nijkamp et al., 2022; Li et al., 2023; Roziere et al., 2023).

Interactive coding assistants can improve developer productivity, but their usefulness depends on the context supplied and on developers' ability to inspect their suggestions (Vaithalingam, Zhang and Glassman, 2022; Barke, James and Polikarpova, 2023). Repository-level agents aim to address context limitations by inspecting project files, identifying relevant code, planning changes, and interacting with execution tools. SWE-bench and SWE-agent illustrate the importance of evaluating agents on repository-level tasks and execution feedback (Jimenez et al., 2024; Yang et al., 2024). The present study builds on this direction by combining repository awareness with explicit repair validation and an efficiency comparison.

## 3. Materials and Methods

### 3.1 Research design

A Design and Development Research approach was adopted to design, implement, and refine the software artefact, with experimental assessment used to evaluate its performance (Richey and Klein, 2007; Wieringa, 2014). The evaluation followed a controlled workflow: repository selection, defect identification, agent execution, repair recommendation, repair validation, and recording of performance measures. Consistent configurations and controlled conditions were intended to support comparison and reproducibility (Wohlin et al., 2012; Keele, 2007).

### 3.2 Development environment

The system used Python for repository processing, automation, structured data handling, and integration; Git for repository management and controlled project access; and an LLM API for contextual reasoning and repair recommendations. Testing, build, and execution mechanisms were incorporated to assess proposed changes. The dissertation describes the implementation environment in terms of backend, repository-processing, LLM-interaction, and validation services; it does not provide a complete reproducible hardware specification or identify the exact LLM model/API version.

### 3.3 Agent architecture

The agent consists of six connected stages. First, the Repository Analysis Module scans source files, configuration, documentation, build files, and dependencies, then selects relevant files and prepares structured repository context. Second, the Error Understanding Module processes compiler errors, runtime exceptions, logs, and diagnostic information. Third, the AI Reasoning Module combines error evidence with repository context to interpret likely defects and formulate repair strategies. Fourth, the Repair Suggestion Module translates those strategies into code modifications. Fifth, the Validation Module evaluates changes through automated tests, build verification, and execution feedback. Finally, the Performance Evaluation Module records repair outcomes, detection metrics, token consumption, model requests, and execution time.

The architecture separates repair generation from repair acceptance. A generated patch is not treated as successful merely because the LLM proposes it; acceptance is based on observable validation evidence. The modular design also permits individual components to be compared or refined independently.

### 3.4 Experimental repository and procedure

The experimental evaluation reported in the results chapter used a CMake-based C/C++ repository. The system selected and analysed 21 relevant files, with no file-processing failures. Two seeded defects were used as known ground-truth issues. For each defect, the agent analysed the repository and diagnostic information, generated repair recommendations, applied a proposed change, and evaluated the modified repository using automated testing and execution feedback. The dissertation describes the wider selection rationale as small- and medium-sized repositories with source code, dependencies, build processes, testing infrastructure, and representative defects; however, the quantitative results presented for the main benchmark concern the single repository described above.

### 3.5 Evaluation measures

Defect detection was evaluated against seeded ground truth using precision, recall, and F1-score. Repair effectiveness was assessed by whether the target defect was addressed and whether the modified repository passed validation without

additional failures. Computational efficiency was assessed using token consumption, number of model requests, and execution time. The standard measures are defined as follows:

Precision = TP / (TP + FP); Recall = TP / (TP + FN); F1 = 2 × (Precision × Recall) / (Precision + Recall), where TP denotes true positives, FP false positives, and FN false negatives.

An ablation comparison evaluated the complete agent against an LLM-only configuration. The LLM-only ground-truth audit was repeated three times. A separate baseline-versus-efficient configuration comparison was performed under the same repository conditions.

## 4. Results

### 4.1 Repository analysis

The agent automatically identified the repository as a CMake-based C/C++ project, detected supported programming languages and available testing capabilities, and selected execution profiles using repository and toolchain evidence. All 21 selected files were analysed successfully, with zero failed files. One file contained potential issues, and two code-related findings were recorded. No security issues were identified by the repository analysis summary.

| Repository analysis measure | Observed result |
|---|---|
| Selected files | 21 |
| Analysed files | 21 |
| Failed files | 0 |
| Files with issues | 1 |
| Total findings | 2 |
| Security issues | 0 |
| Overall repository score | 100/100 |
| Code quality score | 100% |
| Security score | 100% |
| Performance score | 100% |
| Reliability score | 98% |
| Maintainability score | 100% |

The repository analysis dashboard reported an overall score of 100/100 and a reliability score of 98%, with full scores for code quality, security, performance, and maintainability. These are outputs of the implemented repository assessment interface and should be interpreted as system-generated assessment scores, rather than independent external audits of software quality or security.

### 4.2 Seeded defect detection

Against the two seeded ground-truth defects, the complete AI coding agent recorded two true positives, zero false positives, and zero false negatives. This corresponds to 100% precision, 100% recall, and a 100% F1-score. Thus, both seeded defects were identified and no unrelated finding was counted as a defect in this controlled benchmark.

| Configuration | TP | FP | FN | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|
| Full AI coding agent | 2 | 0 | 0 | 100% | 100% | 100% |
| LLM-only (three repetitions, aggregated audit) | Not separately stated | 13 | 0 | 31.58% | 100% | 48.00% |

The LLM-only configuration identified all seeded defects across three repeated runs (100% recall), but produced 13 false positives. Its precision was 31.58% and its F1-score was 48.00%. The contrast indicates that broad LLM reasoning could surface potential issues but, without the complete repository-aware workflow, was less effective at distinguishing the seeded defects from irrelevant findings.

### 4.3 Repair generation and validation

The repair interface identified two issues in the file mathlib.c and generated repair recommendations for the selected component. The source document reports that the changed file was written and subsequently validated. Before repair, the repository failed automated validation; after the generated modification was applied, the repository passed the validation phase with no failures identified. The repair was therefore considered successful according to the execution-based acceptance procedure.

The compiler/runtime evidence interface combined compiler output, test results, runtime data, and environment information. This evidence was associated with repository analysis results and used to support defect interpretation and the pre-/post-repair comparison. The code comparison described in the project indicated that modifications were focused on affected sections and did not restructure the overall program.

### 4.4 Computational efficiency

The baseline and efficient configurations each required seven model requests. The efficient configuration reduced execution time by 15.86% while retaining 100% of the findings. Token consumption increased from 10,305 to 10,640 tokens. This is an increase of approximately 3.25%, not a decrease; the original dissertation describes the change as a "-3.25% decrease," but the reported values show an increase. No absolute execution-time values are stated in the narrative results, so only the reported percentage reduction can be presented.

| Efficiency measure | Baseline | Efficient configuration | Reported outcome |
|---|---|---|---|
| Model requests | 7 | 7 | No change |
| Token consumption | 10,305 | 10,640 | Increase of approximately 3.25% |
| Execution time | Not stated | Not stated | 15.86% reduction |
| Finding retention | Reference configuration | 100% | No loss of findings |

## 5. Discussion

The evaluation supports the central proposition that repository awareness can improve the precision of AI-assisted defect analysis in a controlled setting. The complete agent combined file discovery, dependency and configuration context, structured error interpretation, LLM reasoning, and validation. It identified both seeded defects without false positives, whereas the LLM-only configuration retained full recall but generated a substantially larger number of false positives. This is consistent with concerns that LLM coding assistants may lack the project-specific context needed for dependable software analysis (Pearce et al., 2025; Fan et al., 2023).

The repair experiment also illustrates the importance of separating code generation from repair acceptance. The pre-repair repository failed automated tests, while the repaired repository passed. This provides behavioural evidence that the change improved the tested state of the software. It does not, however, establish that all possible defects were eliminated or that the patch is correct under untested execution paths. Test coverage and the quality of the test oracle remain important constraints in APR evaluation (Pacheco et al., 2007; Martinez and Monperrus, 2019).

The efficiency comparison suggests that workflow optimisation reduced execution time without reducing finding retention, although token consumption increased. Efficiency in LLM-based systems should therefore be interpreted as a trade-off among runtime, model interactions, token use, and output quality, rather than as a single resource measure (Bommasani et al., 2021). The unchanged number of requests and improved execution time

are promising, but the absence of absolute timing values and repeated timing statistics limits the strength of the efficiency claim.

Taken together, the findings support a system-level view of AI debugging: model capability alone is insufficient, and repository context, diagnostic evidence, and execution-based validation are important parts of a dependable workflow (Jimenez et al., 2024; Yang et al., 2024).

## 6. Limitations and Threats to Validity

Several limitations constrain interpretation. First, the principal quantitative benchmark used a small repository and two seeded defects. Such a controlled setup supports objective matching but may not represent the ambiguity, scale, and interacting failures found in naturally occurring software defects. Second, the reported results concern a CMake-based C/C++ environment, so generalisation to other languages, frameworks, and build systems has not been established. Third, the system depends on the underlying LLM and its availability and behaviour. Fourth, passing the available tests is evidence of improvement under the tested conditions, not proof of complete program correctness. Finally, the dissertation does not provide absolute execution times, full hardware details, model identity/version, or a broad multi-repository statistical analysis, which limits independent reproducibility and comparison.

## 7. Future Perspectives and Research Directions

The present study establishes a controlled proof of concept for repository-aware defect detection, repair generation, and execution-based validation. Its results should not, however, be interpreted as evidence of general performance across programming languages, repository scales, or industrial environments. The evaluation involved one CMake-based C/C++ repository, 21 analysed files, and two seeded defects. Accordingly, the most important future step is a broader and more rigorous validation programme using naturally occurring defects, multiple repositories, diverse build systems, and independently curated test suites. Benchmarking should include not only whether an agent can generate a patch, but also whether the patch preserves intended behaviour, avoids regressions, and remains maintainable under subsequent changes. This direction is consistent with the wider evolution of automated repair and empirical software engineering, where benchmark diversity, test-oracle quality, and transparent experimental protocols are central concerns (Huang et al., 2024; Zhang et al., 2024; Wohlin et al., 2012).

A second priority is to move from static repository snapshots towards continuous, longitudinal repository understanding. Future agents could maintain structured representations of dependencies, interfaces, build configurations, test histories, issue discussions, and version-control changes. Such representations may combine graph-based code relationships with semantic embeddings and execution traces, allowing the agent to reason about how a local change propagates across a system. Advances in code representation and machine learning for software repositories provide relevant foundations for this direction (Allamanis et al., 2018; Alon et al., 2019; Feng et al., 2020; Guo et al., 2020; Hall et al., 2011). In parallel, model diversity should be examined through controlled comparisons of general-purpose and code-specialized foundation models, with explicit reporting of model versions, prompts, context limits, and inference settings (Chen et al., 2021; Nijkamp et al., 2022; Li et al., 2023; Roziere et al., 2023; Bommasani et al., 2021).

A third direction concerns stronger validation and security assurance. Passing an available test suite is necessary evidence, but it cannot establish the absence of defects in untested paths. Future versions should therefore combine unit and integration tests with property-based testing, mutation testing, fuzzing, static analysis, dependency auditing, and security-specific checks. Where possible, the agent should generate additional tests from the intended behaviour and from the boundaries of the proposed patch, while keeping test generation distinct from the independent assessment of the patch. This is particularly important because AI-generated code can introduce security weaknesses even when it appears plausible or compiles successfully (Pearce et al., 2025; Ali and Butt, 2026). Fraud-detection research also illustrates the value of anomaly-

oriented pattern recognition in risk-sensitive digital systems, although its methods would need adaptation and software-specific validation before being applied to code security (Naeem, Butt and Javeid, 2026a).

Future work should also examine adaptive and human-supervised repair. Rather than treating the model output as an autonomous final decision, the agent could provide ranked hypotheses, evidence-linked explanations, patch previews, confidence estimates, and explicit approval checkpoints. Developer feedback could be retained as structured information about accepted, rejected, or modified recommendations, enabling evaluation of whether the system becomes more useful without silently reinforcing prior errors. Research on programmer interaction with code-generation tools, AI-assisted learning, adaptive learning platforms, and AI automation suggests that usability, human judgement, and organizational context are important determinants of successful adoption (Vaithilingam et al., 2022; Barke et al., 2023; Kamal et al., 2026; Faizi et al., 2026; Amanat and Butt, 2026; Naeem, Butt and Javeid, 2026b). Future studies should measure not only defect metrics but also review time, developer workload, comprehension, trust calibration, and the quality of human decisions.

Finally, reproducibility and resource efficiency should become first-class design objectives. The current efficient configuration reduced execution time by 15.86% but increased token consumption by approximately 3.25%; this illustrates why runtime, model calls, token use, energy demand, and repair quality should be reported together. Subsequent experiments should publish absolute timings, hardware specifications, model identifiers, prompts, repository commits, test configurations, and repeated-run variability. Ablation studies can isolate the contribution of repository indexing, diagnostic interpretation, reasoning, and execution feedback, while governance frameworks can define audit trails, access control, rollback procedures, and responsibility for accepted changes (Hossain, 2021; Wieringa, 2014; Bosch et al., 2021; Butt et al., 2026a). The long-term objective is not simply a faster code generator, but a dependable engineering assistant whose decisions are traceable, reproducible, secure, and proportionate to the risks of the software being modified.

**8. Multidisciplinary Integration and Cross-Domain Implications**

The architecture examined in this study is rooted in software engineering, but its underlying workflow—context acquisition, interpretation of heterogeneous evidence, hypothesis generation, intervention, and outcome validation—has conceptual parallels across scientific and professional domains. This does not imply that the present coding agent has been tested in those domains. Rather, it identifies a transferable computational pattern that could inform the design of domain-specific intelligent systems. Such transfer requires domain-appropriate data, validated models, specialist oversight, and evaluation criteria that reflect the consequences of error in each field.

In biomedical and clinical sciences, the closest parallel is the integration of multiple evidence streams before a decision is made. Clinical AI may combine laboratory biomarkers, medical images, molecular measurements, patient history, and diagnostic test outputs; a system that reasons over these sources must preserve provenance, resolve inconsistencies, and validate its conclusions against clinically meaningful reference standards. Work on multimodal cardiovascular diagnostics, CT-based lung-cancer classification, tuberculosis test interpretation, hepatitis C screening, antimicrobial resistance, and molecular or clinical determinants of disease illustrates the diversity of evidence that clinical decision-support systems must accommodate (Arif et al., 2026; Raoof et al., 2026; Sarwar et al., 2026; Nadeem et al., 2026; Naik et al., 2026; Farrukh et al., 2026; Riaz and Butt, 2026). The analogy to repository-aware debugging lies in contextual integration and verification, not in equating software tests with clinical diagnosis. In healthcare, validation must include clinical sensitivity, specificity, external validation, fairness, safety, and professional accountability. Research on low-carbohydrate recovery diets also illustrates the interaction of nutritional, hormonal, and musculoskeletal outcomes in human-performance contexts (Butt, Ishaq et al., 2026).

The same principle applies to nutrition, food science, and biotechnology, where product

performance and safety depend on interacting chemical, physical, microbial, nutritional, and process variables. Research on food composition and safety, probiotic fermentation, plant-based foods, hybrid proteins, meat analogues, and functional ingredients provides examples of complex systems in which isolated measurements may be insufficient to characterize overall quality (Khan et al., 2024; Butt et al., 2024; Ahmed et al., 2024; Butt et al., 2025a; Butt et al., 2025b; Riaz et al., 2026a; Riaz et al., 2026b; Butt et al., 2026b; Adnan et al., 2026; Butt et al., 2026c; Butt et al., 2026d; Zaidi and Randhawa, 2025). AI-guided fermentation and precision food processing further demonstrate the potential relevance of predictive modelling, process monitoring, and adaptive control in biotechnology (Khan et al., 2025; Butt et al., 2026e; Riaz et al., 2026c). A domain-specific agent could, in principle, integrate formulation records, process parameters, analytical results, microbial measurements, and acceptance criteria to identify deviations and recommend corrective actions. Such a system would require validated laboratory protocols and food-safety controls; a generated recommendation alone would not establish product safety. The sustainable hybrid-protein formulation work also highlights the relevance of process optimization and environmental considerations in food-system design (Butt, Shahzad et al., 2025). Ultrasound-assisted protein modification and ultrasonication-enhanced protein crosslinking provide further examples of process–structure–function relationships that could be monitored through data-integrated workflows (Butt, Arshad et al., 2026; Butt, Mustafa et al., 2026a; Butt, Mustafa et al., 2026b).

Molecular biology, genetics, and agricultural sciences offer another important connection. Genome and epigenome editing, crop disease resistance, drought response, salinity stress, and environmental molecular monitoring involve complex relationships between biological mechanisms, experimental conditions, and measured outcomes (Jabeen et al., 2025; Fatima et al., 2026; Shafeeq et al., 2026; Iqbal et al., 2024). Multi-omics investigations of aquatic organisms and studies of microplastic-associated toxicity further show how biological interpretation can require the integration of molecular, physiological, and histopathological evidence (Ullah et al., 2026a; Ullah et al., 2026b). A future scientific agent could assist with protocol-aware data organization, anomaly detection, reproducibility checks, and consistency assessment across experimental datasets. However, causal interpretation, biological significance, and experimental conclusions must remain grounded in domain expertise and independent validation.

In physics and engineering, simulation-based reasoning provides a particularly direct conceptual bridge to execution-based software validation. Numerical investigations of nonlinear differential equations and plasma instabilities depend on mathematical formulation, boundary conditions, numerical stability, and comparison of computed outputs with expected behaviour (Shah and Butt, 2026; Tahir et al., 2026). Photonic and metasurface device development, quantum information processing, and adaptive protection of power systems similarly involve coupled components whose performance must be evaluated under defined operating conditions (Butt and Ansar, 2026; Awais and Butt, 2026). These domains suggest opportunities for agents that inspect computational models, track parameter dependencies, identify inconsistent configurations, execute simulations, and compare outputs with physical constraints. Nevertheless, numerical convergence or successful execution is not by itself proof that a model accurately represents the physical system; validation against analytical solutions, experimental observations, and uncertainty estimates remains essential.

Cybersecurity and digital governance are cross-cutting requirements rather than optional extensions. Repository-aware agents may access source code, credentials, build pipelines, package registries, and deployment infrastructure, creating risks that extend beyond incorrect patches. Security-aware design should therefore include least-privilege permissions, sandboxed execution, secret detection, dependency provenance, signed changes, audit logs, human approval for high-impact operations, and reliable rollback. Research on security risks in AI-generated code and machine-learning-based fraud detection provides relevant perspectives on threat

identification and anomaly monitoring (Ali and Butt, 2026; Naeem, Butt and Javeid, 2026a). At the organizational level, risk governance and strategic controls can help define who authorizes automated actions, how incidents are escalated, and how accountability is maintained (Butt et al., 2026a). These safeguards become especially important when an agent is integrated into regulated or safety-critical environments.

The human and social sciences also shape whether intelligent agents can be adopted responsibly. Evidence concerning AI-assisted language learning, adaptive education, employee productivity, workforce sustainability, and healthcare human-resource practices indicates that automation interacts with learning, skills, work design, institutional capacity, and wellbeing (Kamal et al., 2026; Faizi et al., 2026; Amanat and Butt, 2026; Naeem, Butt and Javeid, 2026b; Rehman et al., 2026). Research addressing language, identity, and power further reminds developers that AI systems can reproduce biases embedded in data and institutional settings (Fatima et al., 2026b). In practical software teams, this means that agent performance should be assessed alongside accessibility, explainability, distribution of workload, professional autonomy, and the possibility of over-reliance. A technically accurate system may still be unsuitable if users cannot understand, challenge, or appropriately supervise its recommendations. Simulation-based AI venture decision systems and research on risk governance further suggest that intelligent recommendations must be evaluated within organizational strategy and accountability structures (Naeem, Butt and Javeid, 2026c; Butt, Yiwen and Javeid, 2026).

Finally, environmental, population-health, rehabilitation, and sustainability research broadens the set of outcomes that domain-specific intelligent systems may need to consider. Studies of environmental contaminants, biodiversity monitoring, nutritional status, migration-related health inequities, musculoskeletal rehabilitation, dental care, and sustainability performance illustrate that technical decisions often sit within larger ecological and social systems (Ullah et al., 2026a; Iqbal et al., 2024; Riaz et al., 2026d; Riaz et al., 2026e; Mahmood et al., 2026; Ehsan et al., 2026; Zaidi et al., 2026; Khan Swati et al., 2026a; Khan Swati et al., 2026b; Khurshid et al., 2026). The implication for AI-agent design is that system objectives should not be reduced to a single performance score. Depending on the domain, evaluation may need to balance accuracy, safety, equity, sustainability, interpretability, resource consumption, and long-term outcomes. Interdisciplinary collaboration is therefore necessary not merely to transfer algorithms, but to define the right problems, constraints, evidence standards, and measures of success. Comparative studies of plant-based patties and the valorization of food-processing by-products add further examples of integrating compositional, functional, sensory, and sustainability-related outcomes (Adnan et al., 2026; Butt, Mustafa et al., 2026c).

Population health, nutrition, and rehabilitation research further demonstrates why intelligent decision-support systems must account for variation between individuals and contexts. Studies addressing probiotic-yogurt consumption and metabolic health, dietary zinc and IGF-1, diet-related epigenetic markers, exercise recovery, student nutritional status, dietary quality, and nutritional biomarkers in rehabilitation provide examples of questions in which biological, behavioural, and clinical information may need to be interpreted together (Rashid et al., 2026; Butt, Ali et al., 2026a; Butt, Ali et al., 2026b; Butt, Ishaq et al., 2026; Riaz, Sohail et al., 2026; Anum et al., 2026; Zaidi et al., 2026). These studies do not validate the present software agent; rather, they illustrate the broader need for transparent data integration and carefully defined outcome measures in domain-specific analytical systems.

Other examples emphasize the importance of context-sensitive interpretation and the limits of single-indicator assessment. Research on gut-microbiome diversity and child malnutrition, migration-related health inequities, cancer-associated nutritional and molecular determinants, and musculoskeletal or dental interventions points to interacting biological, social, and treatment-related factors (Shazif et al., 2026; Riaz, Tariq et al., 2026; Riaz, Saim and Butt, 2026; Ehsan et al., 2026; Khan Swati et al., 2026a; Khan Swati et al., 2026b). In a comparable way, AI systems deployed

in B2B healthcare marketing or other institutional settings must be assessed in relation to their users, data governance, and service context rather than by predictive performance alone (Saboor et al., 2026).

Environmental and microbiological applications also offer useful examples of evidence fusion. Molecular monitoring of biodiversity, the detection of antimicrobial resistance, and studies of microbial dynamics during fermentation require careful interpretation of heterogeneous measurements and laboratory evidence (Iqbal et al., 2024; Naik et al., 2026; Riaz, Ye et al., 2026). In food formulation, mineral-fortified products and ultrasound- or enzyme-assisted modification demonstrate the importance of linking process conditions with compositional, functional, and stability outcomes (Zaidi and Randhawa, 2025; Butt, Mustafa et al., 2026a; Butt, Mustafa et al., 2026b; Riaz et al., 2026c). Likewise, crop salinity research and catalyst-assisted synthesis illustrate how computational or data-driven recommendations would need to be interpreted alongside experimental conditions and material-specific constraints (Shafeeq et al., 2026; Raza et al., 2026).

Overall, the multidisciplinary potential of repository-aware agents lies in the disciplined orchestration of context, reasoning, intervention, and verification. The present study demonstrates this pattern only for a narrowly defined software-debugging task. Its extension to medicine, biotechnology, agriculture, physics, engineering, education, or organizational decision-making remains a research proposition requiring domain-specific development and prospective evaluation. This distinction is important: the value of cross-disciplinary integration is not that one agent can solve every scientific problem, but that common principles of evidence integration, traceability, controlled action, and independent validation can guide the development of safer and more reliable intelligent systems.

## 9. Conclusion

This study presented a repository-aware AI coding agent that integrates repository analysis, error understanding, LLM-based reasoning, repair generation, execution-based validation, and performance evaluation. In the reported controlled C/C++ repository experiment, the agent processed 21 files, detected two seeded defects with 100% precision, recall, and F1-score, and changed the repository from a failing to a passing automated-validation state. The LLM-only comparison achieved 100% recall but produced 13 false positives, resulting in 31.58% precision and a 48.00% F1-score. The efficient configuration reduced execution time by 15.86% while retaining all findings, although token consumption rose by approximately 3.25%. These results indicate that repository context and behavioural validation can strengthen AI-assisted debugging, while the small benchmark and limited environment mean that broader testing is required before general claims about practical deployment can be made.

## REFERENCES

Abreu, R., Zoeteweij, P. and Van Gemund, A.J., 2007, September. On the accuracy of spectrum-based fault localization. In Testing: Academic and industrial conference practice and research techniques-MUTATION (TAICPART-MUTATION 2007) (pp. 89-98). IEEE.

Adnan, Muhammad, Kaifa Saddique, Misbah Khalid, Muhammad Abdullah Butt, Shazia Saeed, and Hafiza Ariba Javaid. "Comparative Evaluation of Oat Flour and Soy Protein Incorporation on the Nutritional, Textural, Antioxidant, and Sensory Properties of Plant-Based Patties." *Research Consortium Archive* 4, no. 2 (2026): 4147-4164. https://doi.org/10.5281/zenodo.22006945

Ahmed, Naveed, Muhammad Saeed, Aasma Asghar, Muhammad Abdullah Butt, Muhammad Afzaal, Farhan Saeed, Rizwan Wahab et al. "Utilization of Lactobacillus rhamnosus as probiotic adjunct culture for the development of tempeh." *International Journal of Food Properties* 27, no. 1 (2024): 1279-1289. https://doi.org/10.1080/10942912.2024.2398016

Ali, Mateen, and Muhammad Abdullah But. "Security Risks of AI-Generated Code in Modern Software Development." *Social Science Review Archives* 4, no. 1 (2026): 7334-7359. https://doi.org/10.70670/sra.v4i1.2598

Allamanis, M., Barr, E.T., Devanbu, P. and Sutton, C., 2018. A survey of machine learning for big code and naturalness. ACM Computing Surveys (CSUR), 51(4), pp.1-37.

Alon, U., Zilberstein, M., Levy, O. and Yahav, E., 2019. code2vec: Learning distributed representations of code. Proceedings of the ACM on Programming Languages, 3(POPL), pp.1-29.

Amanat, Asma, and Muhammad Abdullah Butt. "Evaluating the Impact of Adaptive Learning Platforms on Personalized Learning Outcomes and Cognitive Achievement in Higher Education." *Social Science Review Archives* 4, no. 2 (2026): 2991-3012. https://doi.org/10.70670/sra.v4i2.2481

Anum, Dn, Muhammad Gulzar, Areeba Fatima, Mahpara Ch, Dn Areesh, Muhammad Issa Khan, and Muhammad Abdullah Butt. "Association Between Diet Quality and Premature Gray Hair Among Young Adults: A Cross-Sectional Study in Faisalabad, Pakistan." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 5572-5588. https://doi.org/10.5281/zenodo.21871063

Arif, Neha, Hafsa Shafiq, Moiza Noor, Taha Shahbaz, Aswad Khan, Hafsa Noor, Khalil Ahmed, Eman Farrukh, and Muhammad Abdullah Butt. "Artificial Intelligence in Cardiovascular Diagnostics: Integration of Laboratory Biomarkers, Medical Imaging, and Molecular Data." *Pakistan Journal of Medical & Cardiological Review* 5, no. 2 (2026): 5665-5695. https://doi.org/10.66021/pakmcr1251

Awais, Muhammad, and Muhammad Abdullah Butt. "AI-DRIVEN ADAPTIVE PROTECTION SCHEMES FOR RESILIENT POWER SYSTEMS WITH HIGH PENETRATION OF DISTRIBUTED ENERGY RESOURCES." *Spectrum of Engineering Sciences* 4, no. 6 (2026): 1424-1434.

Barke, S., James, M.B. and Polikarpova, N., 2023. Grounded copilot: How programmers interact with code-generating models. Proceedings of the ACM on Programming Languages, 7(OOPSLA1), pp.85-111.

Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E. and Brynjolfsson, E., 2021. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258.

Bosch, J., Woods, E. and Mei, H., 2021. The future of software engineering. IEEE Software, 38, pp.14-21.

Butt, Muhammad Abdullah, Anam Ishaq, Rizwan Shukat, Qaisar Sohail, Muhammad Hamid Javed, Ambreen Saleem, Shazia Saeed, and Muhammad Mudassar Bashir. "Clinical Effects of Post-Exercise Low-Carbohydrate Recovery Diets on Bone Mineral Turnover, Hormonal Stability, and Lean Mass Preservation in Endurance-Trained Adults." *Research Consortium Archive* 4, no. 2 (2026): 1586-1599. https://doi.org/10.0000/

Butt, Muhammad Abdullah, Anas Mustafa, Ahmad Din, Rizwan Shukat, Talha Riaz, Ambreen Saleem, Shazia Saeed, Muhammad Mudassir, Qaisar Sohail, and Muhammad Hamid Javaid. "Effects of Ultrasound-Assisted Protein Modification on the Functional and Nutritional Properties of Soy Protein Isolate." *Pakistan Journal of Medical & Cardiological Review* 5, no. 2 (2026): 5678-5704. https://doi.org/10.5281/zenodo.21234624

Butt, Muhammad Abdullah, Anas Mustafa, Rizwan Shukat, Talha Riaz, Ambreen Saleeem, Shazia Saeed, Muhammad Mudassir, and Qaisar Sohail. "Valorization of Watermelon Peel Flavonoids: Kaempferol, Quercetin, and Rutin as Natural Antioxidants for Functional Food and Nutraceutical Applications: https://doi. Org/10.5281/zenodo. 21234696." *Journal for Current Sign* 4: 2761-2776.

Butt, Muhammad Abdullah, and Eliab Joshua Ansar. "Development Of Metasurface-Based Photonic Devices For High-Speed Optical Communication And Next-Generation Quantum Information Processing." *Spectrum of Engineering Sciences* 4, no. 6 (2026): 4091-4106.

Butt, Muhammad Abdullah, Feng Yiwen, and Umer Javeid. "Risk Governance Frameworks and Strategic Control Mechanisms for Managing Complexity in Global Business Operations." *Social Science Review Archives* 4, no. 2 (2026): 292-302. https://doi.org/10.70670/sra.v4i2.2056

Butt, Muhammad Abdullah, Muhammad Asif Ali, Anam Ishaq, Ambreen Saleem, Sawera Hayat, and Nida Khalil. "The Influence Of Dietary Zinc Supplementation On The Expression Of Insulin-Like Growth Factor 1 (Igf-1) In Adolescent Athletes: https://doi. org/10.5281/zenodo. 19438363." *Pakistan Journal of Medical & Cardiological Review* 5, no. 2 (2026): 12-19. https://pakjmcr.com/index.php/1/article/view/796

Butt, Muhammad Abdullah, Muhammad Asif Ali, Anam Ishaq, Ambreen Saleem, Shazia Saeed, and Mujahid Ul Islam. "Phytochemical-Rich Functional Diet Regulates Epigenetic Markers (DNA Methylation) Associated with Obesity and Insulin Resistance: https://doi. org/10.5281/zenodo. 19438403." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 2707-2715. https://doi.org/10.5281/zenodo.19438403

Butt, Muhammad Abdullah, Muhammad Hameez Shahzad, Samiyah Tasleem, Rabiya Riaz, Xianjiang Ye, Burhan Khalid, Muhammad Atiq Ashraf et al. "Design of a sustainable whey–corn hybrid protein powder for enhanced nutrition, functionality, and environmental stewardship." *Innovative Research in Applied, Biological and Chemical Sciences* 3, no. 2 (2025): 32-51. https://doi.org/10.62497/irabcs.171

Butt, Muhammad Abdullah, Muhammad Umair Arshad, Ali Imran, and Muhammad Afzaal. "Nutritional and biosafety assessment of a novel soy-whey hybrid protein crosslinked by microbial transglutaminase in Sprague Dawley rats." *TPM–Testing, Psychometrics, Methodology in Applied Psychology* 32, no. S7 (2025): Posted 10 October (2025): 597-608. https://doi.org/10.5281/zenodo.17430619

Butt, Muhammad Abdullah, Muhammad Umair Arshad, Ali Imran, and Muhammad Afzaal. "Ultrasonication-Enhanced Microbial Transglutaminase Crosslinking of Botanical and Foreign Protein for Sustainable High-Moisture Extruded Meat Analogue: A Comprehensive Multidimensional Characterization." *Genetics and Molecular Research* (2026). https://doi.org/10.4238/6rytc905

Butt, Muhammad Abdullah, Muhammad Umair Arshad, Samiyah Tasleem, Ali Imran, and Muhammad Afzaal. "Comparative analysis of chicken and meat analogue patties: evaluating physicochemical, cooking, textural, microbial, and sensory attributes." *TPM–Testing, Psychometrics, Methodology in Applied Psychology* 32, no. S6 (2025): Posted 15 September (2025): 1274-1285.

Butt, Muhammad Abdullah, Rizwan Shukat, Muhammad Afzaal, Farhan Saeed, Ali Imran, Aftab Ahmed, Fakhar Islam et al. "Comparative evaluation of the quality and safety attributes of local and branded beef seekh kabab." *Cogent Food & Agriculture* 10, no. 1 (2024): 2360769. https://doi.org/10.1080/23311932.2024.2360769

Butt, Muhammad Abdullah, Talha Riaz, Muhammad Abdullah Sadiq, Anam Ishaq, Ambreen Saleem, and Iza Almas. "AI-Assisted Nutrigenomic Modeling of Blueberry Anthocyanin–Gut Microbiome Interactions to Predict Butyrate Production and Insulin Sensitivity Improvement in Type II Diabetes." *Social Science Review Archives* 4, no. 2 (2026): 2159-2177. https://doi.org/10.70670/sra.v4i2.2309

Butt, Muhammad Abdullah, Talha Riaz, Muhammad Abdullah Sadiq, Anam Ishaq, Ambreen Saleem, and Iza Almas. "Development of Polyphenol-Rich Pomegranate Peel Nutraceuticals for Glycemic Regulation, Insulin Sensitivity Improvement, and Oxidative Stress Reduction in Type II Diabetes Mellitus." *Journal for Current Sign* 4: 2398-2413.

Butt, Muhammad Abdullah, Wajid Hussain, Muhammad Junaid Anwar, Anam Ishaq, Luqman Khan, Muhammad Hammad Anwar, Talha Riaz et al. "AI-Guided Precision Fermentation for the Development of Personalized Functional Foods: A Food Technology Framework for Nutritional Optimization." *Research Consortium Archive* 4, no. 2 (2026): 2778-2795. https://doi.org/10.5281/zenodo.20765535

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P.D.O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G. and Ray, A., 2021. Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374.

Ehsan, Amna, Muhammad Abdullah Butt, Shazia Saeed, Talah Riaz, Muhammad Hamid Javed, Ambreen Saleem, Mudassar Bashir, and Qaisar Sohail. "Comparative effectiveness of manual therapy and therapeutic exercise for improving shoulder function in patient with adhesive capsulitis." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 5879-5898. https://doi.org/10.5281/zenodo.22105778

Faizi, Waqar Un Nisa, Saira Akhtar, and Muhammad Abdullah Butt. "The Impact of Artificial Intelligence-Assisted Learning on Academic Performance and Critical Thinking Skills Among Undergraduate Students." *Social Science Review Archives* 4, no. 2 (2026): 2925-2948. https://doi.org/10.70670/sra.v4i2.2401

Fan, A., Gokkaya, B., Harman, M., Lyubarskiy, M., Sengupta, S., Yoo, S. and Zhang, J.M., 2023, May. Large language models for software engineering: Survey and open problems. In 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE) (pp. 31-53). IEEE.

Farrukh, Eman, Laiba Amjad Malik, Moiza Noor, Aneesa Ajmal, Hina Ali Ahmed, Afroz Rais, and Muhammad Abdullah Butt. "Prevalence, Antimicrobial Resistance Profiles, and Virulence Determinants of Staphylococcus aureus Isolated from Hospitalized Patients." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 5875-5903. https://doi.org/10.5281/zenodo.21903952

Fatima, Ambreen, Nadia Jabeen, Muhammad Abdullah Butt, Muhammad Noman, Talha Riaz, Shazia Saeed, Ambreen Saleem, and Qaisar Sohail. "CRISPR-Cas12a Mediated Epigenome Editing of DNA Methylation at the DREB1A Promoter Enhances Drought Survival Rate by≥ 35% in Zea mays Seedlings: https://doi. org/10.5281/zenodo. 20031798." *Research Consortium Archive* 4, no. 2 (2026): 1093-1102. https://doi.org/10.5281/zenodo.20031798

Fatima, Anees, Muhammad Awais, Shazeen Khalid Raja, and Muhammad Abdullah Butt. "Decolonizing Gendered Subjectivities in Mohammed Hanif's Rebel English Academy A Postcolonial Feminist Analysis of Language, Power, and Identity." *Social Science Review Archives* 4, no. 1 (2026): 6246-6269. https://doi.org/10.70670/sra.v4i1.2496

Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D. and Zhou, M., 2020, November. Codebert: A pre-trained model for programming and natural languages. In Findings of the association for computational linguistics: EMNLP 2020 (pp. 1536-1547).

Gazzola, L., Micucci, D. and Mariani, L., 2017. Automatic software repair: A survey. IEEE Transactions on Software Engineering, 45(1), pp.34-67.

Guo, D., Ren, S., Lu, S., Feng, Z., Tang, D., Liu, S., Zhou, L., Duan, N., Svyatkovskiy, A., Fu, S. and Tufano, M., 2020. Graphcodebert: Pre-training code representations with data flow. arXiv preprint arXiv:2009.08366.

Hall, T., Beecham, S., Bowes, D., Gray, D. and Counsell, S., 2011. A systematic literature review on fault prediction performance in software engineering. IEEE Transactions on Software Engineering, 38(6), pp.1276-1304.

Hamad Raza, Muhammad Abdullah Butt, Iza Almas, Hafiza Ariba Javaid, Talha Riaz, Ambreen Saleem, & Shazia Saeed. (2026). Catalyst-Assisted Synthesis of Functionalized Organic Molecules for Sustainable Pharmaceutical and Industrial Applications. *Social Science Review Archives*, *4*(1), 6589–6608. https://doi.org/10.70670/sra.v4i1.2523.

Hossain, M. (2021) 'A review of ablation studies in machine learning research', IEEE Access.

Huang, K., Xu, Z., Yang, S., Sun, H., Li, X., Yan, Z. and Zhang, Y., 2024. Evolving paradigms in automated program repair: Taxonomy, challenges, and opportunities. ACM Computing Surveys, 57(2), pp.1-43.

Iqbal, Saira, Muhammad Abdullah Butt, Muhammad Hamid Javed, and Talha Riaz. "Environmental DNA-Based Detection and Molecular Monitoring of Biodiversity in Freshwater and Terrestrial Ecosystems." *Pakistan Journal of Medical & Cardiological Review* 3, no. 3 (2024): 1-19. https://doi.org/10.66021/pakmcr1793

Jabeen, Nadia, Musaffa Shahzadi, Muhammad Taha, Nida Shahzadi, and Muhammad Abdullah Butt. "CRISPR-Cas mediated genome editing for disease resistance in crops: advances and challenges." *Pakistan Journal of Medical & Cardiological Review* 4, no. 3 (2025): 2677-2689. https://doi.org/10.5281/zenodo.20956124

Jimenez, C.E., Yang, J., Wettig, A., Yao, S., Pei, K., Press, O. and Narasimhan, K., 2024, May. Swe-bench: Can language models resolve real-world github issues?. In International Conference on Learning Representations (Vol. 2024, pp. 54107-54157).

Kamal, Numra, Muhammad Abdullah Butt, and Umer Javeid. "An empirical study on the effectiveness of artificial intelligence tools in English language acquisition and teaching strategies within an ESG framework." *Social Science Review Archives* 4, no. 1 (2026): 3562-3568. https://doi.org/10.70670/sra.v4i1.1922

Keele, S., 2007. Guidelines for performing systematic literature reviews in software engineering. https://www.researchgate.net/profile/Barbara-Kitchenham/publication/302924724_Guidelines_for_performing_Systematic_Literature_Reviews_in_Software_Engineering/links/61712932766c4a211c03a6f7/Guidelines-for-performing-Systematic-Literature-Reviews-in-Software-Engineering.pdf

khan Swati, Menahil, Musaffa Shahzadi, Muhammad Taha, Nida Shahzadi, and Muhammad Abdullah Butt. "Silver diamine fluoride for caries arrest in pediatric and special needs populations: a decade of clinical evidence." *Pakistan Journal of Medical & Cardiological Review* 5, no. 2 (2026): 693-704. https://doi.org/10.5281/zenodo.21506870

khan Swati, Menahil, Musaffa Shahzadi, Muhammad Taha, Nida Shahzadi, and Muhammad Abdullah Butt. "The impact of orthodontic treatment on periodontal health: gingival recession, bone loss and patient-specific risk factors." *Research Consortium Archive* 4, no. 2 (2026): 1198-1213. https://doi.org/10.5281/zenodo.21506552

Khan, Luqman, Muhammad Abdullah Butt, Ambreen Saleem, and Talha Riaz. "Artificial Intelligence-Guided Optimization of Food Fermentation: Fermentation Kinetics, Microbial Dynamics and Process Control." *Social Science Review Archives* 3, no. 4 (2025): 4766-4787. https://doi.org/10.70670/sra.v3i4.2648

Khan, Waqas Ahmad, Muhammad Inam-ur-Raheem, Hina Rasheed, Muhammad Abdullah Butt, Farhan Saeed, Muhammad Afzaal, Faiyaz Ahmed, Noor Akram, Aasma Asghar, and Gebremichael Gebremedhin Hailu. "Comparative effect of olive oil and flaxseed oil on drug induced hepatotoxicity in rats." *Food Science & Nutrition* 12, no. 11 (2024): 9673-9681. https://doi.org/10.1002/fsn3.4423Digital

Khurshid, Jamila, Zarlakhta Babar, Sajjad Ahmed, Muhammad Abdullah Butt, Umer Javeid, and Nida Khalil. "Beyond carbon footprints: unpacking the social dimensions of sustainability performance in emerging market firms." *Social Science Review Archives* 4, no. 1 (2026): 3386-3402. https://doi.org/10.70670/sra.v4i1.1902 \

Kim, D., Nam, J., Song, J. and Kim, S., 2013, May. Automatic patch generation learned from human-written patches. In 2013 35th international conference on software engineering (ICSE) (pp. 802-811). IEEE.

Le Goues, C., Nguyen, T., Forrest, S. and Weimer, W., 2011. Genprog: A generic method for automatic software repair. Ieee transactions on software engineering, 38(1), pp.54-72.

Li, R., Allal, L.B., Zi, Y., Muennighoff, N., Kocetkov, D., Mou, C., Marone, M., Akiki, C., Li, J., Chim, J. and Liu, Q., 2023. Starcoder: may the source be with you!. arXiv preprint arXiv:2305.06161.

Mahmood, Basit, Minahil Arif, Hafiz Muhammad Moiz Basit, Beenesh Nadeem, Ammarah Abdullah, and Muhammad Abdullah Butt. "Long-term knee joint loading alterations in athletes 5 years post-ACL reconstruction: A comparative gait analysis." *Pakistan Journal of Medical & Cardiological Review* 5, no. 2 (2026): 310-321. https://doi.org/10.66021/pakmcr822

Martinez, M. and Monperrus, M., 2019. Astor: Exploring the design space of generate-and-validate program repair beyond GenProg. Journal of Systems and Software, 151, pp.65-80.

Mehran Ali Tahir, Muhammad Abdullah Butt, Iza Almas, Hafiza Ariba Javaid, & Qaisar Sohail. (2026). ANALYTICAL AND NUMERICAL INVESTIGATION OF A NONLINEAR DIFFERENTIAL EQUATION ARISING IN PHYSICAL AND ENGINEERING SYSTEMS. *Spectrum of Engineering Sciences*, 4(3), 6683–6702. Retrieved from https://www.thesesjournal.com/index.php/1/article/view/3885 . https://doi.org/10.5281/zenodo.22827168

Monperrus, M., 2018. Automatic software repair: A bibliography. ACM Computing Surveys (CSUR), 51(1), pp.1-24.

Myers, G.J., Sandler, C. and Badgett, T., 2011. The art of software testing. John Wiley & Sons.

Nadeem, Muhammad Aqib, Sadia Sarwar, Chand Ali, Adeem Masih, Saleha Sohail, Kainat Saleem, Zarf E. Haider et al. "Comparative Analysis of Hepatitis C Virus (HCV) Screening Tests and Confirmatory PCR: Evaluation of Diagnostic Accuracy, Weak Positive Results, and Clinical Impact." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 5561-5572. https://doi.org/10.5281/zenodo.21682658

Naeem, Waleed, Muhammad Abdullah Butt, and Umer Javeid. "From entrepreneurship theory to startup execution: A simulation-based benchmark analysis of AI-enhanced venture decision systems in early-stage business performance." *Social Science Review Archives* 4, no. 1 (2026): 4065-4075. https://doi.org/10.70670/sra.v4i1.1998

Naeem, Waleed, Muhammad Abdullah Butt, and Umer Javeid. "Machine Learning-Based Fraud Detection Systems and Their Effectiveness in Reducing Cybersecurity Risks in Digital Banking." *Social Science Review Archives* 4, no. 2 (2026): 2553-2573. https://doi.org/10.70670/sra.v4i2.2344

Naeem, Waleed, Muhammad Abdullah Butt, and Umer Javeid. "The Influence of Artificial Intelligence Automation on Employee Productivity, Job Redesign, and Organizational Performance in Service Industries." *Social Science Review Archives* 4, no. 2 (2026): 2534-2552. https://doi.org/10.70670/sra.v4i2.2343

Naik, Aneela, Nayyar Fatima, Muhammad Hammad, Madiha Ahmad Khan, Laiba Shahadat, Nida Tehreem, Muhammad Adnan Jaffer, and Muhammad Abdullah Butt. "Antimicrobial Resistance in the Modern Era: Advances in Laboratory Detection, Emerging Mechanisms, and Diagnostic Challenges." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 5817-5839. https://doi.org/10.5281/zenodo.22958176

Nawaz, Hira, Nimra Shaukat, Ali Ahmad Bhatti, Moiza Noor, Savera Sayam, Tahira Yasmeen, Saira Fatima, Saira Sattar, and Muhammad Abdullah Butt. "Modulation of Gut Microbiota Through Functional Foods to Reduce Low-Grade Chronic Inflammation." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 2231-2246. https://doi.org/10.5281/zenodo.21744252

Nijkamp, E., Pang, B., Hayashi, H., Tu, L., Wang, H., Zhou, Y., Savarese, S. and Xiong, C., 2022. Codegen: An open large language model for code with multi-turn program synthesis. arXiv preprint arXiv:2203.13474.

Pacheco, C., Lahiri, S.K., Ernst, M.D. and Ball, T. (2007) 'Feedback-directed random test generation', Proceedings of the ACM International Conference on Software Engineering, pp. 75–84.

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B. and Karri, R., 2025. Asleep at the keyboard? assessing the security of github copilot's code contributions. Communications of the ACM, 68(2), pp.96-105.

Pressman, R.S. and Maxim, B.R., 2020. Software Engineering: A Practitioner's Approach. 9th edn. McGraw-Hill.

Raoof, Muhammad, Muhammad Junaid Ghafoor, Huzaifa Rauf, Muhammad Usman, and Muhammad Abdullah Butt. "COMPARATIVE EVALUATION OF HAND-CRAFTED AND CONVOLUTIONAL FEATURES FOR MULTICLASS LUNG CANCER CLASSIFICATION FROM CT IMAGES." *Spectrum of Engineering Sciences* 4, no. 3 (2026): 6420-6430. https://doi.org/10.5281/zenodo.22911816

Rashid, Mian Shahan, Zubaria Gull, Muhammad Abdullah Butt, Sawera Hayat, Shnshah E. Azam, Shazia Saeed, Muhammad Mudassar Bashir et al. "The role of functional probiotic yogurt consumption in medical weight loss: A GLP-1 friendly nutritional approach to metabolic health in UK adults." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 1623-1632. https://doi.org/10.5281/zenodo.22147937

Rehman, Anees Ur, Muhammad Abdullah Butt, Talha Riaz, and Qaisar Sohail. "Wellbeing-Oriented Human Resource Management and Workforce Sustainability in Healthcare Organisations: Comparative Evidence from Pakistan and Australia." *Journal for Current Sign* 4: 3422-3440.

Riaz, Muhammad, Muhammad Abdullah Butt, Qaisar Sohail, Talha Riaz, and Ambreen Saleem. "Dietary Habits, Nutritional Status, and Socioeconomic Factors Among University Students in Punjab, Pakistan: A Multicity Nutritional and Social Survey." *Social Science Review Archives* 4, no. 2 (2026): 2723-2742. https://doi.org/10.70670/sra.v4i2.2370

Riaz, Muhammad, Neelam Tariq, Madiha Iqbal, Muhammad Abdullah Butt, Qaisar Sohail, and Talha Riaz. "Migration, Displacement, and Health Inequities: A Comparative Study of Refugee Populations and Access to Healthcare Services." *Social Science Review Archives* 4, no. 2 (2026): 2704-2722. https://doi.org/10.70670/sra.v4i2.2369

Riaz, Saim, and Muhammad Abdullah Butt. "Nutritional and Molecular Determinants of Cancer Progression: Investigating the Role of Dietary Bioactives and Cellular Signaling Pathways." *Research Consortium Archive* 4, no. 1 (2026): 2321-2338. https://doi.org/10.5281/zenodo.22006246

Riaz, Talha, Areej Azhar, Zhijun Xia, Aliza Batool, Xianjiang Ye, Burhan Khalid, Muhammad Moeid Khan et al. "Advances in plant-based functional foods: Emerging trends, nutritional potential, and health implications." *Food Science & Applied Microbiology Reports* 5, no. 1 (2026): 1-17. https://doi.org/10.61363/fsamr.v5i1.332

Riaz, Talha, Syed Tahaa Munawar, Ahmad Din, Muhammad Abdullah Butt, Sawera Hayat, Areej Azhar, Rabiya Riaz et al. "Microbiological safety, adulteration, and heavy metal–associated health risks in raw cow and buffalo milk from Punjab, Pakistan." *Food Science & Applied Microbiology Reports* 5, no. 1 (2026): 18-28. https://doi.org/10.61363/fsamr.v5i1.336

Riaz, Talha, Xianjiang Ye, Te Yu, Zhijun Xia, Muhammad Moeid Khan, Xuan Yao, Rabiya Riaz et al. "Chitinase-primed ultrasound enhances Saccharomyces cerevisiae cell-wall deconstruction and yeast-protein remodelling in two complementary model systems." *Food Research International* (2026): 120543.

Richey, R.C. and Klein, J.D. (2007) Design and Development Research: Methods, Strategies, and Issues. Lawrence Erlbaum Associates.

Roziere, B., Gehring, J., Gloeckle, F., Sootla, S., Gat, I., Tan, X.E., Adi, Y., Liu, J., Sauvestre, R., Remez, T. and Rapin, J., 2023. Code llama: Open foundation models for code. arXiv preprint arXiv:2308.12950.

Saboor, Ahmad Abdul, Muhammad Abdullah Butt, and Anees ul Rehman. "Artificial Intelligence Adoption in B2B Healthcare Marketing: A Strategic Review and Framework for Australian Digital Health Services." *Research Consortium Archive* 4, no. 2 (2026): 4813-4832. https://doi.org/10.5281/zenodo.22798575

Sarwar, Sadia, Hashir Masood, Samra Zulfiqar, Muhammad Aqib Nadeem, Noor ul Ain, Hamda Raza, Kainat Saleem et al. "Interpretation Of Genexpert Results And Their Association With Sputum Smear Status In Suspected Pulmonary Tuberculosis Patients." *Spectrum of Engineering Sciences* 4, no. 3 (2026): 6393-6407. https://doi.org/10.5281/zenodo.22688820

Shafeeq, A., Shokat, T., Rizvi, R., Gohar, F., Riaz, R., Ansar, S., Khalid, B., Butt, M. A., Raza, A., Ahamed, M. S., Khan, M. M., & Riaz, T. (2026). Alleviation of adverse effects of salinity stress on cauliflower by foliar application of trehalose. Scientific Records, 3(2), 84–100. https://doi.org/10.62324/journal.sr/2026.069

Shah, Muhammad Saleem, and Muhammad Abdullah Butt. "Multi-Scale Computational Investigation of Plasma Instabilities in Next-Generation Fusion Reactors." *Spectrum of Engineering Sciences* 4, no. 3 (2026): 2257-2266.

Shazif, Nawal, Aneesa Ajmal, Hafsa Noor, Nimra Anees, Malahat Choudhary, Shafi Ullah, Moiza Noor, and Muhammad Abdullah Butt. "Association Between Gut Microbiome Diversity and Severe Acute Malnutrition in Children Under Five Years." *Social Science Review Archives* 4, no. 1 (2026): 6130-6146. https://doi.org/10.70670/sra.v4i1.2482

Sommerville, I., 2021. Software Engineering. 10th edn. Pearson.

Ullah, Saif, Muhammad Abdullah Butt, Qaisar Sohail, and Talha Riaz. "Multi-Omics Characterization of Fish Responses to Emerging Environmental Contaminants: Implications for Aquatic Toxicology and Sustainable Fisheries Management." *Pakistan Journal of Medical & Cardiological Review* 5, no. 2 (2026): 5666-5685. https://doi.org/10.5281/zenodo.20817609

Ullah, Saif, Muhammad Abdullah Butt, Qaisar Sohail, Talha Riaz, and Shazia Saeed. "Toxicological Evaluation of Microplastic and Nanoplastic Exposure on Growth Performance, Oxidative Stress Biomarkers, and Histopathological Alterations in Cultured Fish Species." *Social Science Review Archives* 4, no. 2 (2026): 2178-2198. https://doi.org/10.70670/sra.v4i2.2310

Vaithilingam, P., Zhang, T. and Glassman, E.L., 2022, April. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In Chi conference on human factors in computing systems extended abstracts (pp. 1-7).

Wieringa, R. (2014) Design Science Methodology for Information Systems and Software Engineering. Springer.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B. and Wesslén, A. (2012) Experimentation in Software Engineering. Springer.

Wong, W.E., Gao, R., Li, Y., Abreu, R. and Wotawa, F., 2016. A survey on software fault localization. IEEE Transactions on Software Engineering, 42(8), pp.707-740.

Yang, J., Jimenez, C., Wettig, A., Lieret, K., Yao, S., Narasimhan, K. and Press, O., 2024. Swe-agent: Agent-computer interfaces enable automated software engineering. Advances in Neural Information Processing Systems, 37, pp.50528-50652.

Zaidi, Zarish Fatima, and Muhammad Atif Randhawa. "Calcium and Vitamin D3-Fortified Chocolate Spread: Development, Properties, and Stability." *RADS Journal of Food Biosciences* 4, no. 2 (2025): 91-98.

Zaidi, Zarish Fatima, Shiza Fatima Zaidi, Sameen Ahsan, Dua Afzal Butt, and Sundus Ghayas Munir. "Nutritional Biomarkers as Predictors of Rehabilitation Success in Musculoskeletal Disorders: Bridging Nutrition Science and Physiotherapy." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 5705-5722.

Zeller, A., 2009. Why Programs Fail: A Guide to Systematic Debugging. 2nd edn. Morgan Kaufmann.

Zhang, S., Li, D., Zhao, M., Li, H. and Wong, W.E., 2024, July. Search-based automated program repair: a survey. In 2024 IEEE 24th International Conference on Software Quality, Reliability, and Security Companion (QRS-C) (pp. 447-456). IEEE.

IEER
Institute for Excellence in Education & Research