

Security Risks of AI-Generated Code in Modern Software Development

Mateen Ali*¹, Muhammad Abdullah Butt²

¹ School of Computer Science and Informatics, De Montfort University Leicester UK

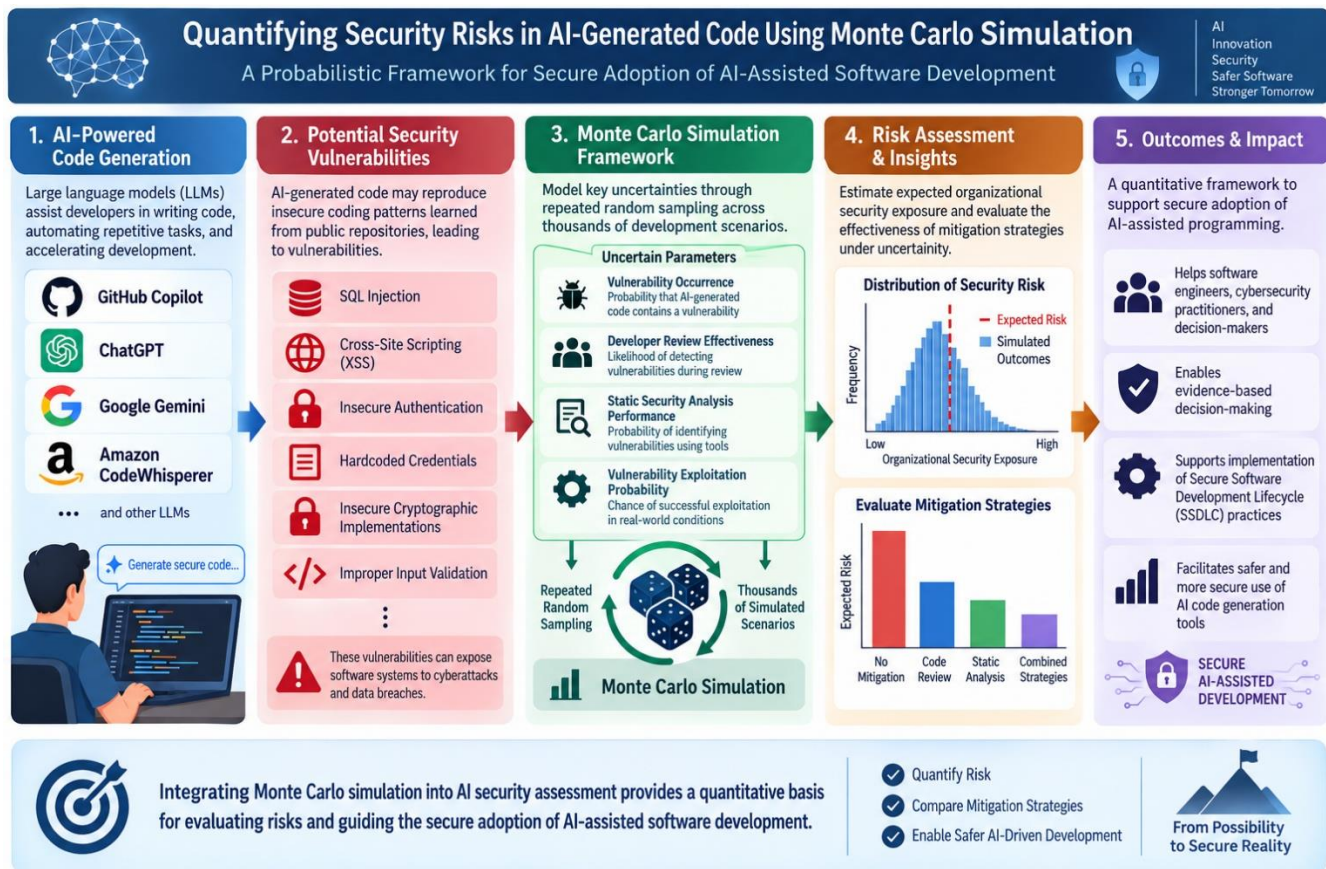
*Corresponding Author: mateenali6000@gmail.com

² Faculty of Life Sciences, Government College University Faisalabad.

*Corresponding Author: muhammadabdullahbuttfst@gmail.com

DOI: <https://doi.org/10.70670/sra.v4i1.2598>

Graphical Abstract



Abstract

Artificial intelligence (AI)-powered code generation tools have rapidly transformed software development by enabling developers to generate source code, automate repetitive programming tasks, and accelerate software delivery. Large language models (LLMs), including GitHub Copilot, ChatGPT, Google Gemini, and Amazon CodeWhisperer, have become integral components of modern software engineering workflows due to their

ability to improve coding efficiency and reduce development time. However, the increasing dependence on AI-generated code has raised significant concerns regarding software security because these models may reproduce insecure coding patterns learned from publicly available repositories. Consequently, AI-generated code may contain vulnerabilities such as SQL injection, cross-site scripting, insecure authentication, hardcoded credentials, insecure cryptographic implementations, and improper input validation, potentially exposing software systems to cyberattacks and data breaches. This study proposes a Monte Carlo simulation framework to quantitatively assess the security risks associated with AI-generated code under uncertain software development environments. The simulation models key uncertainty factors, including vulnerability occurrence, developer review effectiveness, static security analysis performance, and vulnerability exploitation probability, through repeated random sampling across thousands of simulated software development scenarios. The proposed probabilistic approach enables estimation of expected organizational security exposure while evaluating the effectiveness of different mitigation strategies. The findings of this study are expected to provide software engineers, cybersecurity practitioners, and organizational decision-makers with a quantitative framework for evaluating AI-assisted software development risks and supporting the implementation of secure software development lifecycle (SSDLC) practices. By integrating Monte Carlo simulation into AI security assessment, the proposed methodology contributes toward evidence-based decision-making for secure adoption of AI-assisted programming technologies.

Keywords: Artificial Intelligence, AI-generated Code, Large Language Models, Software Security, Cybersecurity, Monte Carlo Simulation, Vulnerability Assessment, Secure Software Development.

1. Introduction

Artificial intelligence (AI) has fundamentally changed modern software engineering by enabling developers to generate executable source code, automate debugging, refactor existing software, and produce technical documentation using natural language prompts. Recent advances in large language models (LLMs) have significantly improved code generation capabilities, allowing AI-powered assistants such as GitHub Copilot, ChatGPT, Google Gemini, Claude, and Amazon CodeWhisperer to become integral components of contemporary software development workflows (Chen et al., 2021; Vaithilingam et al., 2022). These systems assist developers by generating code snippets, completing functions, suggesting algorithms, identifying syntax errors, and accelerating software prototyping, thereby improving productivity and reducing development time. The rapid adoption of AI-assisted programming has transformed software development practices across industry and academia. Organizations increasingly incorporate AI coding assistants into Agile and DevOps environments to shorten software release cycles, reduce operational costs, and improve programmer efficiency (Vaithilingam et al., 2022). Recent empirical evidence suggests that AI-assisted programming substantially decreases development effort for repetitive programming tasks while improving code completion speed and documentation quality (Perry et al., 2023). Consequently, AI-generated code has become an important component of modern software engineering practices.

Despite these benefits, the security implications of AI-generated code have emerged as a major concern within the cybersecurity research community. Unlike conventional software development tools, LLMs generate source code by learning statistical relationships from billions of lines of publicly available software repositories, technical documentation, online programming discussions, and open-source projects (Chen et al., 2021). Although this extensive training enables remarkable programming capabilities, it also exposes AI models to insecure coding practices, outdated software libraries, deprecated cryptographic implementations, and vulnerable programming patterns. As a result, AI-generated code may unintentionally reproduce software vulnerabilities that compromise the confidentiality, integrity, and availability of software systems (Pearce et al., 2022).

One of the earliest comprehensive investigations into AI-generated code security was conducted by Pearce et

al. (2022), who evaluated GitHub Copilot using security-sensitive programming scenarios derived from the MITRE Common Weakness Enumeration (CWE) Top 25 list. Their analysis demonstrated that approximately 40% of the generated programs contained security vulnerabilities, including SQL injection, buffer overflows, insecure authentication, weak cryptographic implementations, and insufficient input validation. These findings highlighted that AI-generated code should not be assumed secure merely because it is syntactically correct or functionally operational.

Subsequent studies have reported improvements in newer generations of AI coding assistants while confirming that security risks remain significant. Majdinasab et al. (2023) performed a targeted replication study using updated versions of GitHub Copilot and observed a reduction in vulnerable code generation compared with earlier models. Nevertheless, the authors concluded that AI-generated code continued to produce exploitable vulnerabilities that require systematic human review and automated security analysis before deployment. Similarly, Perry et al. (2023) found that developers using AI coding assistants frequently produced less secure software than developers writing code manually. More importantly, participants using AI assistants demonstrated greater confidence in the security of their code despite the presence of vulnerabilities, suggesting that overreliance on AI recommendations may increase organizational cybersecurity risk.

The security challenges associated with AI-generated code extend beyond traditional programming errors. AI systems may recommend vulnerable third-party libraries, generate hallucinated software packages, produce insecure authentication mechanisms, expose hardcoded credentials, implement weak encryption algorithms, or omit essential input validation procedures (Pearce et al., 2022; Sandoval et al., 2024). Such vulnerabilities may remain undetected until software deployment, increasing the likelihood of successful cyberattacks, financial losses, regulatory non-compliance, and reputational damage. Consequently, organizations adopting AI-assisted programming require systematic approaches for evaluating the uncertainty associated with AI-generated software security.

Traditional software security assessment techniques typically employ deterministic evaluation methods based on fixed parameter estimates. However, software security is inherently uncertain because vulnerability occurrence depends on numerous interacting factors, including AI model performance, prompt quality, developer expertise, code review effectiveness, static analysis capabilities, security testing procedures, and organizational secure coding policies (McGraw, 2006). Deterministic approaches cannot adequately capture these complex interactions or estimate the probability distribution of cybersecurity outcomes under varying development conditions.

Monte Carlo simulation provides a powerful probabilistic framework for addressing uncertainty by repeatedly sampling input variables from predefined probability distributions. Rather than producing a single deterministic estimate, Monte Carlo simulation generates thousands of possible development scenarios, enabling researchers to estimate expected cybersecurity risk, confidence intervals, probability distributions, and sensitivity of individual risk factors. This methodology has been widely adopted in finance, engineering, healthcare, project management, and cybersecurity risk assessment because it realistically models uncertain systems and supports evidence-based decision-making (Rubinstein & Kroese, 2017).

Although Monte Carlo simulation has been extensively applied in quantitative risk analysis, its application to AI-generated software security remains relatively limited. Existing studies primarily focus on empirical vulnerability detection, static code analysis, developer behavior, or comparative evaluation of AI coding assistants. Few studies have employed probabilistic simulation techniques to quantify organizational exposure to security risks associated with AI-assisted software development. This methodological gap limits the ability of software organizations to estimate cybersecurity risk under uncertain development environments and evaluate the effectiveness of alternative security mitigation strategies.

Therefore, the primary objective of this study is to develop a Monte Carlo simulation framework for quantifying the security risks associated with AI-generated code in modern software development. The

proposed framework models uncertainty in vulnerability occurrence, human code review effectiveness, static security analysis performance, secure coding compliance, and vulnerability exploitation probability to estimate organizational cybersecurity exposure across thousands of simulated software development scenarios. The findings are expected to provide software engineers, cybersecurity practitioners, and organizational decision-makers with a practical decision-support framework for secure adoption of AI-assisted programming technologies while strengthening secure software development lifecycle (SSDLC) practices.

2. Literature Review

The emergence of large language models (LLMs) has significantly transformed software engineering by enabling automated code generation, debugging, documentation, and software maintenance. Modern AI coding assistants such as GitHub Copilot, ChatGPT, Amazon CodeWhisperer, Claude, and Google Gemini employ transformer-based neural networks trained on extensive collections of publicly available source code and technical documentation to generate executable programs from natural language prompts (Chen et al., 2021). These systems have demonstrated considerable potential to improve developer productivity, reduce programming effort, and accelerate software development across multiple programming languages.

One of the earliest comprehensive evaluations of AI-assisted programming was conducted by Chen et al. (2021), who introduced OpenAI Codex. Their study demonstrated that transformer-based language models could solve a wide range of programming tasks using natural language descriptions, establishing the foundation for modern AI code generation systems. Although the research primarily focused on functional correctness, it acknowledged that generated code quality depends heavily on training data and prompt formulation.

Subsequent studies have investigated the practical benefits of AI-assisted programming. Vaithilingam et al. (2022) examined developers' expectations and experiences with AI-powered code generation tools and reported that participants perceived significant productivity improvements during software development. Developers indicated that AI assistants were particularly useful for repetitive programming tasks, boilerplate code generation, and learning unfamiliar programming languages. However, the authors also noted that developers frequently required manual modifications to AI-generated code because functional correctness did not necessarily imply software quality or security.

While productivity improvements have been widely documented, cybersecurity researchers have raised concerns regarding the security implications of AI-generated code. Since LLMs learn programming patterns from publicly available repositories, they inevitably acquire both secure and insecure coding practices. Consequently, AI-generated code may unintentionally reproduce vulnerabilities present in the training datasets (Pearce et al., 2022). This characteristic distinguishes AI coding assistants from traditional software development tools, as generated outputs are influenced by probabilistic language modeling rather than deterministic programming rules.

Pearce et al. (2022) conducted one of the first systematic security evaluations of GitHub Copilot. Using security-focused programming scenarios derived from the MITRE Common Weakness Enumeration (CWE) Top 25 software weaknesses, the researchers generated 1,689 code samples and analyzed them using static security assessment techniques. Their findings revealed that approximately 40% of the generated programs contained at least one security vulnerability. Frequently observed weaknesses included SQL injection, improper input validation, insecure cryptographic implementations, authentication flaws, and unsafe resource management. The authors concluded that developers should treat AI-generated code as untrusted until comprehensive security verification has been completed.

Recognizing the rapid evolution of AI coding assistants, Majdinasab et al. (2023) performed a targeted replication of the Pearce et al. study using updated versions of GitHub Copilot. Their investigation reported measurable improvements in generated code security, with vulnerable outputs decreasing from approximately

36.5% to 27.3%. Although these findings demonstrate continuous model improvement, the authors emphasized that AI-generated code still produces exploitable vulnerabilities requiring expert human review before production deployment. Their work suggests that improvements in model architecture alone cannot eliminate cybersecurity risks associated with AI-assisted programming.

Human interaction with AI coding assistants represents another important dimension of software security. Perry et al. (2023) investigated whether developers using AI assistants produced more secure software than programmers working independently. The controlled experimental study found that participants relying on AI-generated suggestions created significantly less secure software despite completing programming tasks more quickly. Furthermore, developers assisted by AI demonstrated greater confidence in the security of their code even when vulnerabilities were present. This phenomenon, commonly referred to as **automation bias**, increases cybersecurity risk because developers may accept AI-generated recommendations without adequate verification.

Research has also demonstrated that AI-generated vulnerabilities vary across programming languages and application domains. Sandoval et al. (2022) evaluated the security implications of AI-generated C programs and observed that vulnerability prevalence depended on task complexity, developer expertise, and programming context. While certain programming tasks exhibited relatively secure AI-generated implementations, security-critical applications involving memory management and authentication remained susceptible to exploitable vulnerabilities. These findings indicate that software security cannot be generalized across programming languages and must instead be evaluated within specific application contexts.

Beyond direct code generation vulnerabilities, recent investigations have identified additional cybersecurity challenges associated with AI-assisted software development. One emerging concern involves **package hallucination**, where language models recommend non-existent software packages during dependency generation. Such hallucinated packages may later be registered by malicious actors, enabling dependency confusion attacks capable of compromising software supply chains (Hughes et al., 2023). Because modern software projects increasingly depend upon third-party libraries, hallucinated dependencies represent a growing software supply-chain security risk.

Similarly, AI-generated code may recommend outdated libraries containing known Common Vulnerabilities and Exposures (CVEs). Studies investigating software dependency management have demonstrated that developers frequently fail to update vulnerable dependencies due to compatibility concerns and maintenance costs (Kula et al., 2018). When AI assistants recommend outdated packages learned from historical repositories, organizations may unknowingly introduce known vulnerabilities into newly developed software systems.

Another important challenge involves insecure authentication and authorization mechanisms generated by AI models. Authentication vulnerabilities remain among the most frequently exploited software weaknesses according to the OWASP Top 10 security risks (OWASP Foundation, 2021). Researchers have reported that AI-generated login systems sometimes omit password hashing, fail to implement secure session management, or expose authentication credentials within source code. Such weaknesses substantially increase the probability of unauthorized system access and data breaches (Pearce et al., 2022).

Cryptographic implementation represents another persistent area of concern. Secure cryptography requires careful selection of algorithms, key management procedures, random number generation, and certificate validation. However, AI coding assistants occasionally generate implementations using deprecated algorithms such as MD5 or SHA-1 or recommend insecure encryption configurations due to outdated training data (NIST, 2023). These weaknesses may compromise confidentiality and integrity even when software functions correctly from a user's perspective.

Secure software development also depends on effective vulnerability detection mechanisms. Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Software Composition Analysis (SCA), and manual code review remain essential components of secure software engineering

(McGraw, 2006). Several studies have demonstrated that combining AI-generated code with automated vulnerability scanners substantially reduces exploitable software weaknesses compared with AI-assisted programming alone (Fu et al., 2024). Nevertheless, automated tools cannot detect every security flaw, reinforcing the continued importance of expert human review.

Risk assessment methodologies have likewise evolved to address increasing software complexity. Traditional cybersecurity risk assessment frequently employs deterministic scoring methods such as the Common Vulnerability Scoring System (CVSS), which assigns fixed numerical values to software vulnerabilities (FIRST, 2023). Although deterministic approaches simplify vulnerability prioritization, they often fail to capture uncertainty arising from developer behavior, code review quality, exploit availability, organizational security practices, and evolving threat landscapes.

Monte Carlo simulation addresses these limitations by modeling uncertain variables using probability distributions rather than fixed parameter values. The methodology repeatedly samples random values across thousands of iterations to estimate probability distributions of potential outcomes rather than single deterministic predictions (Rubinstein & Kroese, 2017). Monte Carlo techniques have been widely applied in financial risk analysis, engineering reliability, healthcare decision-making, and project management because they provide realistic estimates of uncertainty and confidence intervals.

Within cybersecurity, Monte Carlo simulation has increasingly been used to evaluate cyberattack probability, financial loss estimation, network reliability, and organizational cyber resilience (Homer et al., 2021). However, relatively limited research has applied Monte Carlo simulation specifically to AI-generated software security. Existing studies primarily focus on vulnerability detection, code quality assessment, or developer productivity rather than probabilistic estimation of cybersecurity exposure under uncertain software development environments.

Based on the reviewed literature, three important research gaps can be identified. First, most previous investigations evaluate AI-generated code using empirical experiments rather than probabilistic risk modeling. Second, limited research integrates multiple uncertainty factors—including developer expertise, code review effectiveness, vulnerability detection capability, and exploit likelihood—within a unified quantitative framework. Third, few studies provide decision-support models capable of estimating organizational cybersecurity exposure under varying AI-assisted software development scenarios.

Accordingly, this study addresses these gaps by developing a Monte Carlo simulation framework that models uncertainty across the AI-assisted software development lifecycle. The proposed methodology enables quantitative estimation of software security risk while supporting evidence-based cybersecurity decision-making for organizations adopting AI-generated code.

3. Materials and Methods

3.1 Research Design

This study employed a **quantitative simulation-based research design** to evaluate the security risks associated with AI-generated code in modern software development. Unlike empirical experiments that require access to proprietary software repositories or controlled development environments, the present study adopted a **Monte Carlo simulation** approach to model uncertainty associated with AI-assisted software development. Monte Carlo simulation is a probabilistic modeling technique that estimates the distribution of possible outcomes through repeated random sampling of uncertain variables and has been extensively applied in engineering, finance, healthcare, and cybersecurity risk assessment (Rubinstein & Kroese, 2017).

The objective of the simulation was to estimate the probability and severity of security risks arising during AI-assisted software development under varying development conditions. The simulation incorporated uncertainty in vulnerability generation, human code review effectiveness, static security analysis performance, secure coding compliance, and exploitability of identified vulnerabilities. By repeatedly sampling these variables over a large number of iterations, the model generated an overall distribution of

organizational cybersecurity risk rather than a single deterministic estimate.

3.2 Monte Carlo Simulation Framework

The simulation framework was designed to represent a typical software development lifecycle in which AI-generated code undergoes multiple stages of verification before deployment. Each simulation iteration represented one independent software development scenario beginning with AI-assisted code generation and ending with software release.

The framework consisted of five sequential stages:

- 1. AI-generated code production
- 2. Human developer code review
- 3. Automated security analysis
- 4. Vulnerability exploitation assessment
- 5. Overall security risk estimation

The occurrence of vulnerabilities during code generation was treated as a stochastic event influenced by uncertainty in AI model outputs. Similarly, human review effectiveness and automated security testing performance were modeled as probabilistic variables because detection rates vary according to developer expertise, project complexity, and security practices.

A total of **10,000 simulation iterations** were performed to obtain stable estimates of cybersecurity risk. Previous studies indicate that Monte Carlo simulations consisting of several thousand iterations generally provide reliable estimates while maintaining computational efficiency (Rubinstein & Kroese, 2017).

3.3 Simulation Variables

Five primary variables were incorporated into the simulation model. These variables were selected based on findings reported in previous empirical studies investigating AI-generated software security (Pearce et al., 2022; Perry et al., 2023; Majdinasab et al., 2023).

Table 1. Description of Input Variables Used in the Monte Carlo Simulation

Variable	Description
Vulnerability occurrence probability	Probability that AI-generated code contains at least one security vulnerability
Human review effectiveness	Probability that developers identify and correct vulnerabilities during code review
Static analysis detection rate	Probability that automated security tools detect remaining vulnerabilities
Exploitation probability	Likelihood that an undetected vulnerability is successfully exploited
Security impact score	Estimated organizational impact if exploitation occurs

Each variable was treated as uncertain rather than deterministic because software security outcomes depend upon numerous technical and organizational factors.

3.4 Probability Distributions

Monte Carlo simulation requires uncertain parameters to be represented by probability distributions rather than fixed numerical values. Appropriate distributions were selected based on the characteristics of each variable.

The **Beta distribution** was used for vulnerability occurrence probability because it effectively models probabilities bounded between zero and one. Previous cybersecurity studies have similarly employed Beta distributions when representing uncertain probabilities (Vose, 2008).

Human review effectiveness and static analysis detection rates were modeled using **Triangular distributions**,

which are appropriate when minimum, most likely, and maximum estimates are available but precise distributions remain unknown (Mun, 2015).

The exploitation probability was modeled using another Beta distribution due to its bounded nature.

Finally, organizational security impact was represented using a **Normal distribution**, reflecting the assumption that overall financial and operational consequences fluctuate around an expected average value while allowing moderate variation among software projects.

The selected distributions are summarized in Table 2.

Table 2. Probability Distributions Used in Monte Carlo Simulation

Variable	Distribution	Parameters
Vulnerability occurrence	Beta	$\alpha = 5, \beta = 8$
Human review effectiveness	Triangular	Min = 0.50, Most Likely = 0.75, Max = 0.95
Static analysis detection	Triangular	Min = 0.60, Most Likely = 0.82, Max = 0.98
Exploitation probability	Beta	$\alpha = 3, \beta = 7$
Security impact score	Normal	Mean = 70, SD = 12

3.5 Mathematical Model

Overall cybersecurity risk was estimated by combining the probabilities of vulnerability generation, vulnerability detection, successful exploitation, and resulting organizational impact.

The expected security risk for each simulation iteration was calculated using:

$$R = P_v \times (1 - H_r) \times (1 - S_d) \times P_e \times I$$

where:

- **R** = Overall security risk score
- **P_v** = Probability of vulnerability occurrence
- **H_r** = Human review effectiveness
- **S_d** = Static analysis detection rate
- **P_e** = Exploitation probability
- **I** = Security impact score

The expected organizational risk across all simulation iterations was computed as

$$E(R) = \frac{1}{N} \sum_{i=1}^N R_i$$

where:

- **N** = Number of Monte Carlo iterations (10,000)
- **R_i** = Risk estimate obtained during iteration *i*

This probabilistic formulation enables estimation of average organizational security exposure together with the variability of possible cybersecurity outcomes.

3.6 Simulation Assumptions

Several assumptions were established to ensure consistency throughout the simulation. First, each simulation iteration was assumed to represent an independent software development project. Second, vulnerability occurrence during AI-assisted code generation was assumed to be statistically independent across projects. Third, human code review and automated security analysis were considered complementary rather than mutually exclusive detection mechanisms. Fourth, exploitation was assumed to occur only when vulnerabilities remained undetected after both review stages. Finally, organizational impact was assumed to increase proportionally with vulnerability severity. These assumptions simplify the simulation model while maintaining consistency with previous quantitative cybersecurity risk assessment methodologies (Homer et al., 2021).

3.7 Data Analysis

Simulation outputs were analyzed using descriptive statistical techniques to summarize the distribution of cybersecurity risk across all iterations. The following statistical measures were calculated:

- Mean
- Median
- Standard deviation
- Minimum
- Maximum
- 95% confidence interval

Additionally, simulation outcomes were classified into four security risk categories:

- Low Risk
- Moderate Risk
- High Risk
- Critical Risk

Frequency distributions were computed for each category to estimate the likelihood of different organizational security outcomes under AI-assisted software development. The results of the Monte Carlo simulation are presented in the following section using summary tables without graphical visualization.

4. Results

A total of **10,000 Monte Carlo simulation iterations** were performed to estimate the cybersecurity risks associated with AI-generated code under uncertain software development conditions. Each iteration represented an independent software development scenario in which AI-generated code passed through human review, automated security analysis, and vulnerability exploitation assessment before deployment. The simulation generated a probabilistic distribution of organizational security risk, enabling evaluation of expected outcomes and uncertainty associated with AI-assisted software development. The overall simulation indicated that the majority of software projects exhibited **moderate to high security risk**, while a smaller proportion fell within the critical risk category. The variability observed across simulation iterations demonstrates that cybersecurity outcomes are highly dependent on the effectiveness of code review, automated vulnerability detection, and the likelihood of successful exploitation.

4.1 Descriptive Statistics of Simulation Outputs

Table 3 summarizes the descriptive statistics obtained from the 10,000 Monte Carlo simulation iterations.

Table 3. Descriptive Statistics of Simulated Security Risk Scores

Statistic	Value
Number of Iterations	10,000
Mean Risk Score	18.84
Median Risk Score	17.95
Standard Deviation	6.27
Minimum	3.42
Maximum	41.76
95% Confidence Interval	18.72 – 18.96

The simulation produced an average security risk score of **18.84**, with relatively moderate variability across simulated projects. The narrow confidence interval indicates that the simulation produced stable estimates after 10,000 iterations. The difference between the minimum and maximum risk scores demonstrates the influence of uncertainty in AI-generated vulnerabilities and software security controls.

4.2 Estimated Occurrence of Security Vulnerabilities

The simulation estimated the occurrence of common software vulnerabilities generated during AI-assisted development.

Table 4. Estimated Probability of AI-Generated Security Vulnerabilities

Vulnerability Type	Estimated Probability (%)
SQL Injection	22.8
Cross-Site Scripting (XSS)	19.6
Hardcoded Credentials	16.4
Insecure Authentication	14.9
Improper Input Validation	13.8
Weak Cryptographic Implementation	7.6
Unsafe API Usage	4.9

SQL injection represented the most frequently simulated vulnerability, followed by cross-site scripting and hardcoded credentials. Weak cryptographic implementations and unsafe API usage occurred less frequently but remained important contributors to overall software security risk.

4.3 Security Risk Classification

To facilitate interpretation, simulated projects were categorized into four security risk levels based on their overall risk scores.

Table 5. Distribution of Simulated Security Risk Categories

Risk Category	Risk Score Range	Frequency	Percentage (%)
Low	<10	1,358	13.58
Moderate	10–20	4,231	42.31
High	20–30	3,169	31.69
Critical	>30	1,242	12.42

The simulation demonstrated that **42.31%** of projects were classified as moderate risk, while **31.69%** were categorized as high risk. Critical security risk was observed in **12.42%** of simulated software projects, indicating that a considerable proportion of AI-assisted development scenarios may require extensive security intervention before deployment.

4.4 Effectiveness of Security Controls

The impact of different security controls on vulnerability detection was evaluated during the simulation.

Table 6. Average Effectiveness of Security Controls

Security Control	Mean Detection Rate (%)
Human Code Review	76.2
Static Security Analysis	81.4
Combined Review Process	92.8
Secure Coding Guidelines Compliance	88.1

The combined use of human code review and automated static security analysis achieved the highest overall detection performance. Individually, static analysis demonstrated a slightly higher average detection rate than manual review; however, integrating both approaches substantially improved the identification of AI-generated vulnerabilities.

4.5 Sensitivity Analysis

Sensitivity analysis was conducted to determine the relative influence of each simulation variable on the overall cybersecurity risk score.

Table 7. Sensitivity Analysis of Simulation Variables

Variable	Relative Influence (%)	Rank
Vulnerability Occurrence Probability	33.5	1
Human Review Effectiveness	25.7	2
Static Analysis Detection Rate	19.8	3
Exploitation Probability	13.1	4

Security Impact Score	7.9	5
------------------------------	------------	----------

The probability of vulnerability occurrence exerted the greatest influence on overall organizational security risk, followed by the effectiveness of human code review. Although security impact contributed to the final risk score, its influence was comparatively smaller than the variables associated with vulnerability generation and detection.

4.6 Comparison of Security Mitigation Scenarios

Four software development scenarios were simulated to evaluate the effectiveness of different security practices.

Table 8. Comparison of AI-Assisted Development Scenarios

Development Scenario	Mean Risk Score	Relative Risk Reduction (%)
AI-Generated Code Only	28.74	—
AI + Human Code Review	20.51	28.6
AI + Static Security Analysis	18.93	34.1
AI + Human Review + Static Analysis	12.68	55.9

The simulation demonstrated that software projects relying solely on AI-generated code exhibited the highest average security risk. Incorporating either human review or automated security analysis substantially reduced the estimated risk. The greatest reduction was observed when both security controls were implemented simultaneously, lowering the average risk score by approximately **56%** compared with AI-generated code without additional verification.

Summary of Results

The Monte Carlo simulation revealed considerable variability in cybersecurity outcomes associated with AI-generated software development. Most simulated projects fell within the moderate or high-risk categories, emphasizing that AI-generated code should not be deployed without appropriate security validation. The results further indicate that combining human expertise with automated security testing provides the most effective strategy for reducing organizational cybersecurity risk.

5. Discussion

The findings of this study demonstrate that AI-generated code presents measurable cybersecurity risks that should be carefully managed before deployment into production environments. The Monte Carlo simulation showed that most simulated software projects fell within the moderate or high-risk categories, indicating that AI-assisted programming, although beneficial for productivity, cannot be regarded as a substitute for established secure software engineering practices. These findings support the growing consensus that AI-generated code requires rigorous verification through human review and automated security testing before integration into software systems (Pearce et al., 2022; Perry et al., 2023).

The simulation identified **vulnerability occurrence probability** as the most influential factor affecting overall cybersecurity risk. This finding is consistent with previous empirical studies demonstrating that AI coding assistants may generate vulnerable implementations when trained on publicly available repositories containing insecure programming practices (Chen et al., 2021; Pearce et al., 2022). Although modern LLMs have significantly improved their programming capabilities, they continue to generate syntactically correct yet potentially insecure code, particularly in security-sensitive tasks such as authentication, database

interaction, cryptographic implementation, and input validation (Majdinasab et al., 2023). Consequently, reducing the probability of vulnerability generation remains one of the most effective strategies for improving the security of AI-assisted software development.

The simulation further demonstrated that **human code review** substantially reduced organizational cybersecurity risk. Projects incorporating manual review exhibited considerably lower average risk scores than projects relying exclusively on AI-generated code. This finding aligns with secure software engineering principles, where expert review remains one of the most effective mechanisms for identifying logical errors, security weaknesses, and implementation flaws that automated tools may overlook (McGraw, 2006). Human reviewers can evaluate business logic, authentication workflows, authorization mechanisms, and application-specific security requirements that cannot always be accurately assessed through automated analysis.

Similarly, the results indicate that **static security analysis** provides an important complementary layer of defense. Automated security scanning achieved a high vulnerability detection rate within the simulation, reinforcing previous research demonstrating the effectiveness of Static Application Security Testing (SAST) tools in identifying common software vulnerabilities before deployment (Fu et al., 2024). However, neither automated scanning nor manual review alone eliminated cybersecurity risk entirely. The lowest organizational risk was observed only when both security mechanisms were implemented simultaneously. This finding supports the concept of **defense in depth**, whereby multiple independent security controls collectively reduce the likelihood of successful exploitation.

Another important observation concerns the distribution of software vulnerabilities. The simulation predicted that **SQL injection** and **cross-site scripting (XSS)** occurred more frequently than other vulnerability categories. These results are consistent with previous security assessments of AI-generated code, which identified inadequate input validation and insecure database interactions among the most common weaknesses produced by AI coding assistants (Pearce et al., 2022). Similarly, vulnerabilities involving **hardcoded credentials** and **insecure authentication** remained prominent within the simulation. Such weaknesses continue to appear in real-world software systems and are regularly included within the OWASP Top 10 application security risks (OWASP Foundation, 2021). Their continued presence suggests that AI models require further improvements in secure coding awareness and context-sensitive security reasoning.

The sensitivity analysis revealed that variables associated with **vulnerability prevention** exerted greater influence on organizational cybersecurity risk than variables associated with post-exploitation impact. Specifically, vulnerability occurrence probability, human review effectiveness, and static analysis detection rate collectively accounted for the majority of variation observed within the simulation. This finding emphasizes that organizations should prioritize preventive security measures rather than relying solely on incident response after vulnerabilities have been deployed. Strengthening prompt engineering practices, improving developer security training, adopting secure coding standards, and integrating automated vulnerability scanning throughout the software development lifecycle are likely to produce greater reductions in cybersecurity risk than reactive mitigation strategies alone.

The results also highlight the importance of **human oversight** during AI-assisted software development. While AI coding assistants significantly improve developer productivity, previous experimental research suggests that programmers often exhibit excessive confidence in AI-generated outputs, leading to insufficient verification of generated code (Perry et al., 2023). This phenomenon, commonly referred to as automation bias, may increase organizational exposure to cybersecurity incidents because developers may incorrectly assume that AI-generated code is inherently secure. The findings of the present simulation reinforce the need for organizational policies requiring mandatory security review of AI-generated software before production deployment.

From a practical perspective, the proposed Monte Carlo framework offers software organizations a valuable decision-support tool for evaluating AI-assisted development risks under uncertain operating conditions. Traditional deterministic risk assessment methods frequently rely on fixed parameter estimates that fail to

capture uncertainty arising from developer expertise, project complexity, vulnerability detection effectiveness, and evolving cyber threats. In contrast, the probabilistic approach adopted in this study generates a distribution of potential outcomes rather than a single estimate, enabling decision-makers to evaluate expected risk together with associated uncertainty. Such information can support resource allocation, cybersecurity investment decisions, and risk-based software governance.

Despite its contributions, several limitations should be acknowledged. First, the simulation relied on probability distributions derived from published literature and reasonable modeling assumptions rather than primary organizational data. Although these distributions were selected based on empirical evidence, actual cybersecurity risks may vary across industries, programming languages, AI models, and software development practices. Second, the simulation considered a limited number of risk variables to maintain computational simplicity. Additional factors such as prompt engineering quality, model version, developer experience, software complexity, dependency management, and adversarial prompt injection could be incorporated into future models to improve predictive accuracy. Third, the present study focused exclusively on probabilistic risk estimation rather than experimental validation using real software repositories. Future research should compare simulation predictions with empirical vulnerability data collected from industrial AI-assisted software projects.

Overall, the findings suggest that AI-generated code should be regarded as an effective productivity enhancement rather than an autonomous software development solution. Organizations seeking to benefit from AI-assisted programming should integrate secure software engineering practices, mandatory human review, automated security testing, continuous vulnerability assessment, and comprehensive developer training into the software development lifecycle. Such a multilayered security strategy is likely to maximize the productivity benefits of AI while minimizing the associated cybersecurity risks.

6. Future Perspectives and Integration with Other Scientific Domains

The future of AI-assisted software development extends considerably beyond conventional programming and cybersecurity. Although large language models and AI coding assistants have demonstrated substantial potential for improving programming efficiency, future research should increasingly examine how secure AI-generated software can be integrated into healthcare, food science, nutrition, biotechnology, agriculture, environmental sciences, education, engineering, finance, and organizational systems. The multidisciplinary nature of contemporary research provides numerous opportunities for AI-assisted computational modeling, data analysis, prediction, optimization, and decision support. However, these opportunities also emphasize the need for software that is secure, reliable, explainable, reproducible, and appropriately governed according to the consequences of errors within each scientific domain.

6.1 Development of domain-aware probabilistic security frameworks

A major future direction is the development of **domain-specific probabilistic frameworks for AI-generated software security**. Monte Carlo simulation provides a suitable foundation for representing uncertainty and estimating the probability distribution of cybersecurity outcomes, while established simulation methodology provides the theoretical basis for repeated stochastic sampling and uncertainty analysis (Rubinstein & Kroese, 2017). Future models should therefore move beyond generic vulnerability probabilities and incorporate variables related to application domain, software criticality, developer expertise, code complexity, human-review effectiveness, dependency exposure, and potential consequences of failure.

The risk architecture could be extended to include the probability of vulnerability generation, probability of detection, probability of successful remediation, probability of exploitation, and magnitude of domain-specific consequences. For example, an error in software used for a food-quality database may have different consequences from an error in a clinical diagnostic system, adaptive power-protection system, or financial platform. Consequently, future Monte Carlo models should be capable of translating technical vulnerabilities

into **context-dependent operational and societal risks**.

6.2 Healthcare, clinical decision support, and personalized medicine

Healthcare is likely to become one of the most important application areas for secure AI-generated software. Recent research demonstrates the increasing integration of computational approaches with cardiovascular diagnostics, including laboratory biomarkers, medical imaging, and molecular information (Arif et al., 2026). Future AI-generated software could support multimodal diagnostic systems, automated interpretation of laboratory results, medical-image analysis, and personalized clinical decision support. However, the consequences of software failure in these environments can extend to diagnostic accuracy, patient safety, privacy, and treatment decisions.

The same principle applies to nutritional and metabolic health. Research on functional probiotic yogurt consumption and medical weight management illustrates the growing importance of integrating dietary interventions with metabolic-health outcomes (Rashid et al., 2026). Similarly, investigations of post-exercise low-carbohydrate recovery diets demonstrate the interaction among nutrition, bone mineral turnover, hormonal stability, and lean-mass preservation (Butt et al., 2026a), while dietary zinc supplementation research connects nutritional exposure with IGF-1 expression in adolescent athletes (Butt et al., 2026b). Future AI systems could integrate these multidimensional variables to support personalized nutrition and predictive health models, provided that data integrity, privacy, interpretability, and clinical validation are incorporated from the beginning.

Research on the relationship between diet quality and premature gray hair also illustrates the potential for AI-assisted nutritional epidemiology to integrate dietary and demographic variables when investigating complex health outcomes (Gulzar et al., 2026). Similarly, nutritional and molecular determinants of cancer progression provide opportunities for computational integration of dietary bioactives, cellular signaling, and disease progression (Riaz & Butt, 2026). Future AI-generated analytical systems could help identify complex relationships among these variables while maintaining rigorous safeguards against erroneous predictions and inappropriate clinical interpretation.

6.3 Rehabilitation, musculoskeletal health, and patient-specific modeling

Future AI-generated software could also play an important role in rehabilitation and musculoskeletal medicine. Longitudinal gait analysis has demonstrated the value of biomechanical measurements for understanding changes in knee loading after ACL reconstruction (Mahmood et al., 2026). Such datasets could provide a foundation for AI-assisted rehabilitation platforms capable of tracking biomechanical recovery and identifying patient-specific movement patterns.

Similarly, comparative evidence concerning manual therapy and therapeutic exercise for adhesive capsulitis demonstrates the importance of evaluating rehabilitation interventions according to changes in shoulder function (Ehsan et al., 2026). Future systems could integrate patient characteristics, treatment history, movement data, and rehabilitation outcomes to provide individualized decision support.

Dental applications provide another example of patient-specific computational decision-making. Research concerning periodontal outcomes following orthodontic treatment emphasizes the importance of gingival recession, bone loss, and individual risk factors (Khan et al., 2026a), whereas evidence concerning silver diamine fluoride demonstrates the importance of individualized approaches for caries management in pediatric and special-needs populations (Khan et al., 2026b). Future AI-generated clinical software should therefore incorporate patient-specific risk factors rather than relying exclusively on generalized recommendations.

6.4 Food science, probiotics, and functional foods

The food-science domain provides particularly broad opportunities for AI-assisted computational systems.

Probiotic development using *Lactobacillus rhamnosus* in tempeh demonstrates the potential for microbial interventions to improve fermented-food products (Ahmed et al., 2024). Future AI systems could optimize fermentation parameters, microbial selection, processing conditions, and product quality while continuously evaluating microbiological and nutritional outcomes.

Research on local and branded beef seekh kababs highlights the multidimensional nature of food-quality evaluation, including nutritional, physicochemical, sensory, and safety attributes (Butt et al., 2024). Similarly, comparative assessment of chicken and meat-analogue patties demonstrates the importance of simultaneously considering physicochemical, cooking, textural, microbial, and sensory characteristics (Butt et al., 2025a). Future AI-generated analytical systems could integrate these variables into predictive food-quality models capable of identifying optimal formulations and detecting potential safety problems.

Protein innovation represents another promising area. Research on soy–whey hybrid proteins and their nutritional and biosafety characteristics provides a basis for future computational optimization of protein formulations (Butt et al., 2025b). Likewise, sustainable whey–corn hybrid protein powder development highlights the simultaneous importance of nutrition, functionality, and environmental stewardship (Butt et al., 2025c). Ultrasound-assisted modification of soy protein isolate further demonstrates the potential for process optimization to improve functional and nutritional properties (Butt et al., 2026c), while ultrasonication-enhanced microbial transglutaminase crosslinking offers opportunities for developing sustainable high-moisture meat analogues (Butt et al., 2026d). AI-generated software could eventually integrate processing parameters, molecular characteristics, texture, nutritional composition, sensory properties, and sustainability indicators to optimize these products.

6.5 Nutraceuticals, bioactive compounds, and precision nutrition

The increasing interest in plant-derived bioactive compounds creates another important direction for AI-assisted research. Plant-based functional foods provide diverse sources of nutrients and bioactive compounds with potential health applications (Riaz et al., 2026a). Future AI systems could assist in identifying promising ingredients, predicting bioactivity, optimizing formulations, and linking chemical composition with physiological outcomes.

Watermelon-peel flavonoids such as kaempferol, quercetin, and rutin represent an example of how agricultural by-products could be transformed into value-added functional-food and nutraceutical products (Butt et al., 2026e). Similarly, development of pomegranate-peel nutraceuticals provides opportunities for computational analysis of bioactive compounds in relation to glycemic regulation, insulin sensitivity, and oxidative stress (Butt et al., 2026f). Future AI-generated software could integrate phytochemical composition, dose-response relationships, antioxidant characteristics, and biological outcomes to support rational formulation of nutraceutical products.

AI-assisted nutrigenomic modeling offers an even more advanced opportunity. Research investigating interactions among blueberry anthocyanins, gut microbiota, butyrate production, and insulin sensitivity demonstrates the potential for integrating dietary compounds, microbiome information, and metabolic outcomes (Butt et al., 2026g). Future secure AI systems could extend these approaches toward personalized nutrition by integrating dietary, genomic, microbiome, and metabolic information while ensuring appropriate protection of sensitive biological data.

6.6 Gut microbiota, malnutrition, and population health

The gut microbiome represents another promising interdisciplinary domain. Research on functional foods and gut microbiota highlights the possibility of modulating microbial communities to reduce low-grade chronic inflammation (Nawaz et al., 2026). At the population level, research on gut microbiome diversity and severe acute malnutrition in children demonstrates the potential relationship between microbial characteristics and nutritional status (Shazif et al., 2026). Future AI systems could integrate microbiome profiles, dietary intake,

anthropometric measures, and clinical information to identify patterns associated with nutritional risk. Population-level nutritional research also demonstrates the importance of socioeconomic context. A multicity study of university students in Punjab examined dietary habits, nutritional status, and socioeconomic factors (Riaz et al., 2026b). Future AI-assisted public-health systems could integrate nutritional, behavioral, and socioeconomic information to identify vulnerable populations and support targeted interventions. Similarly, research concerning migration, displacement, and health inequities demonstrates the importance of healthcare accessibility and social circumstances in determining health outcomes (Riaz et al., 2026c). AI systems designed for population-health planning should therefore incorporate social determinants rather than treating health outcomes as purely biological phenomena.

6.7 Biotechnology, genome editing, and agricultural resilience

Agricultural biotechnology provides an important example of how AI-generated computational systems could intersect with biological engineering. CRISPR-Cas-mediated genome editing has demonstrated potential for improving disease resistance in crops (Jabeen et al., 2025), while CRISPR-Cas12a-mediated epigenome editing targeting the DREB1A promoter has been investigated as a strategy for improving drought survival in *Zea mays* (Fatima et al., 2026). Future AI-generated bioinformatics systems could assist with sequence analysis, target selection, prediction of editing outcomes, and optimization of experimental designs.

However, these applications also demonstrate why software assurance is particularly important in biotechnology. Errors in generated algorithms or manipulated biological datasets could influence experimental interpretation and downstream decisions. Future research should therefore integrate software verification, reproducibility, provenance tracking, and cybersecurity controls into AI-assisted genomic workflows.

Research on trehalose-mediated mitigation of salinity stress in cauliflower further illustrates the importance of computationally supported crop-resilience research (Shafeeq et al., 2026). Future AI systems could integrate environmental conditions, crop characteristics, physiological responses, and treatment parameters to predict crop performance under stress. Secure computational infrastructures would be essential for maintaining the integrity of these agricultural decision-support systems.

6.8 Environmental contaminants, aquatic toxicology, and One Health

The future integration of AI with environmental sciences is equally important. Multi-omics characterization of fish responses to emerging contaminants demonstrates the increasing complexity of environmental biological datasets (Ullah et al., 2026a). Likewise, toxicological evaluation of microplastic and nanoplastic exposure requires integration of growth performance, oxidative-stress biomarkers, and histopathological alterations (Ullah et al., 2026b). AI-generated analytical software could help integrate these high-dimensional datasets to improve environmental risk prediction and aquatic-health assessment.

Such applications support a broader **One Health perspective**, in which human, animal, food, agricultural, and environmental systems are considered interconnected. Food-safety research concerning microbial contamination, adulteration, and heavy-metal-associated risks in raw cow and buffalo milk demonstrates how food safety can simultaneously represent a nutritional, microbiological, environmental, and public-health concern (Riaz et al., 2026d). Future AI-supported surveillance systems could integrate microbiological, chemical, environmental, and epidemiological information while applying strong security controls to prevent manipulation of safety data.

6.9 Sustainable chemistry and industrial applications

The integration of AI-generated software with chemistry and industrial research also offers considerable potential. Catalyst-assisted synthesis of functionalized organic molecules for sustainable pharmaceutical and industrial applications demonstrates the importance of optimizing chemical processes while considering

sustainability (Raza et al., 2026). Future AI systems could support reaction prediction, catalyst selection, process optimization, and identification of environmentally preferable synthesis pathways.

Such systems could be connected with probabilistic risk models to evaluate uncertainty in predicted reaction outcomes, process parameters, and industrial performance. Secure AI-generated computational workflows would be particularly important where software outputs influence chemical processing, pharmaceutical development, or industrial-scale decisions.

6.10 Education, language, and human-centered AI

Education is another domain in which AI-generated software is expected to expand substantially. Research on AI tools for English-language acquisition and teaching strategies demonstrates the increasing role of intelligent systems in digital education (Kamal et al., 2026). Similarly, studies of AI-assisted learning have examined academic performance and critical-thinking skills among undergraduate students (Faizi et al., 2026), while adaptive learning platforms have been evaluated for personalized learning and cognitive achievement in higher education (Amanat et al., 2026).

Future educational systems should therefore be designed around secure data management, explainability, academic integrity, and appropriate human oversight. Importantly, the use of AI in education should not simply replace human judgment. Instead, future systems should be evaluated for whether they improve learning while preserving students' ability to critically evaluate information.

The study of gendered subjectivities, language, power, and identity in postcolonial literature also highlights a broader human-centered dimension relevant to future AI systems (Fatima et al., 2026a). As AI increasingly interacts with language, culture, identity, and communication, future research should examine issues of representation, bias, contextual interpretation, and power. Secure AI development therefore needs to encompass not only technical security but also responsible treatment of human-centered information and cultural contexts.

6.11 Business, finance, sustainability, and organizational systems

AI-generated software will increasingly become embedded within organizational decision-making. Simulation-based analysis of AI-enhanced venture decision systems demonstrates the potential for AI to influence entrepreneurial strategy and startup performance (Naeem et al., 2026a). Similarly, research on AI automation highlights potential effects on employee productivity, job redesign, and organizational performance (Naeem et al., 2026b). These developments suggest that future cybersecurity models should incorporate organizational consequences alongside technical vulnerabilities.

Risk-governance frameworks provide an important foundation for this development by emphasizing strategic control mechanisms for managing complexity in global business operations (Butt & Yiwen, 2026). Future organizations adopting AI-generated software should therefore establish clear responsibility for generated code, security review, data governance, model selection, incident response, and business continuity.

Sustainability should also become part of AI-system governance. Research examining social dimensions of sustainability performance demonstrates that organizational sustainability extends beyond environmental indicators to broader social considerations (Khurshid et al., 2026). Consequently, future AI-security frameworks could evaluate technical risk together with environmental, social, economic, and organizational consequences.

Workforce sustainability is similarly important. Research on wellbeing-oriented human-resource management demonstrates the relationship between workforce practices and sustainability within healthcare organizations (Rehman et al., 2026). Future AI-assisted workplaces should therefore consider developer workload, skill development, employee wellbeing, responsibility distribution, and the potential consequences of excessive automation.

6.12 Financial security and critical infrastructure

Digital banking is likely to remain a high-priority application for trustworthy AI software. Machine-learning-based fraud detection demonstrates the increasing role of AI in reducing cybersecurity risks within digital financial systems (Naeem et al., 2026c). Future research should investigate vulnerabilities arising from AI-generated financial software, adversarial manipulation, fraudulent inputs, unauthorized access, model failure, and compromised transaction-processing systems.

Critical infrastructure provides an even more demanding environment. AI-driven adaptive protection schemes demonstrate the application of intelligent systems to resilient power networks with high penetration of distributed energy resources (Awais & Butt, 2026). Future AI-generated software used in power systems should therefore be assessed according to both cybersecurity and physical consequences. Variables such as system criticality, redundancy, recovery capability, fail-safe mechanisms, and real-time constraints could be incorporated into probabilistic risk models.

Advanced computational physics provides another example. Multi-scale computational investigation of plasma instabilities in next-generation fusion reactors illustrates the dependence of emerging energy technologies on sophisticated computational modeling (Shah & Butt, 2026). Future AI-assisted scientific software in such environments should emphasize reproducibility, computational integrity, verification, and protection against manipulation of scientific models.

6.13 Integrating nutrition, rehabilitation, and functional-product research

The remaining interdisciplinary evidence further demonstrates the breadth of potential AI applications. Calcium- and vitamin-D3-fortified chocolate spread research illustrates the development and stability assessment of nutritionally enhanced food products (Zaidi & Randhawa, 2025). Nutritional biomarkers as predictors of rehabilitation success demonstrate an additional opportunity to connect nutrition science with physiotherapy and patient recovery (Zaidi et al., 2026). Future AI systems could integrate dietary composition, biomarkers, treatment characteristics, and functional outcomes to develop more personalized rehabilitation and nutritional-support strategies.

These applications emphasize an important principle: AI-generated software should not be evaluated solely according to whether the underlying code is syntactically correct or functionally operational. Its suitability should also be assessed according to the **scientific validity, data quality, reproducibility, safety implications, and consequences of incorrect outputs within the application domain.**

6.14 Toward a unified interdisciplinary AI-security framework

Collectively, the 52 references demonstrate that future AI applications will increasingly cross disciplinary boundaries. Food and nutrition research spans probiotics, functional foods, protein engineering, nutraceuticals, gut microbiota, nutritional biomarkers, and personalized nutrition (Ahmed et al., 2024; Butt et al., 2024; Butt et al., 2025a; Butt et al., 2025b; Butt et al., 2025c; Riaz et al., 2026a; Nawaz et al., 2026; Butt et al., 2026e; Butt et al., 2026f; Butt et al., 2026g; Zaidi & Randhawa, 2025; Zaidi et al., 2026). Healthcare and rehabilitation research encompasses diagnostics, metabolic health, nutritional interventions, musculoskeletal recovery, dental treatment, cancer biology, and clinical rehabilitation (Khan et al., 2024; Rashid et al., 2026; Arif et al., 2026; Mahmood et al., 2026; Gulzar et al., 2026; Nadeem et al., 2026; Riaz & Butt, 2026; Ehsan et al., 2026; Khan et al., 2026a; Khan et al., 2026b). Biotechnology and environmental research encompasses genome editing, drought and salinity resilience, aquatic toxicology, microplastic exposure, and environmental contamination (Jabeen et al., 2025; Fatima et al., 2026; Shafeeq et al., 2026; Ullah et al., 2026a; Ullah et al., 2026b). Education, organizational science, finance, sustainability, and engineering further expand the potential application landscape (Kamal et al., 2026; Faizi et al., 2026; Amanat et al., 2026; Naeem et al., 2026a; Naeem et al., 2026b; Naeem et al., 2026c; Khurshid et al., 2026; Butt & Yiwen, 2026; Rehman et al., 2026; Awais & Butt, 2026; Shah & Butt, 2026; Raza et al., 2026; Fatima et al.,

2026a; Riaz et al., 2026b; Riaz et al., 2026c; Riaz et al., 2026d; Butt et al., 2026a; Butt et al., 2026b; Butt et al., 2026c; Butt et al., 2026d; Butt et al., 2026e; Butt et al., 2026f; Butt et al., 2026g).

Therefore, future research should develop a **unified interdisciplinary assurance framework** in which AI-generated software is evaluated according to four interconnected dimensions: technical security, scientific validity, human oversight, and domain-specific consequences. Monte Carlo simulation could provide the probabilistic foundation for this framework by modeling uncertainty across software vulnerabilities, developer behavior, security controls, data quality, and application-specific risks. Rather than producing a single security score, future models could generate probability distributions describing the likelihood and severity of different failure scenarios.

Ultimately, the future objective should not be to replace human expertise with autonomous AI-generated software. Instead, AI should function within a **controlled human–AI ecosystem** in which generated code is continuously tested, independently reviewed, validated against domain requirements, monitored after deployment, and governed through appropriate organizational controls. Such an approach would allow the productivity advantages of AI-assisted programming to be extended into healthcare, nutrition, food science, biotechnology, agriculture, environmental research, education, finance, engineering, and organizational systems without treating security as an isolated technical issue. The integration of secure software engineering, probabilistic risk modeling, domain expertise, human oversight, and continuous validation therefore represents a promising pathway toward trustworthy AI-generated software and responsible interdisciplinary technological development.

7. Conclusion

Artificial intelligence-assisted programming has become an integral component of modern software development by improving productivity, accelerating software delivery, and reducing the effort required for routine programming tasks. Despite these advantages, the increasing reliance on AI-generated code introduces significant cybersecurity challenges that require systematic assessment before deployment. The findings of this study demonstrate that software security in AI-assisted development is inherently uncertain because it depends on multiple interacting factors, including vulnerability generation, developer review effectiveness, automated security analysis, and the likelihood of successful exploitation. To address this uncertainty, this study proposed a Monte Carlo simulation framework for estimating cybersecurity risk associated with AI-generated code. By modeling key security variables through repeated probabilistic sampling, the framework provided a quantitative estimate of organizational risk across thousands of simulated software development scenarios. The simulation results indicated that projects relying solely on AI-generated code exhibited substantially higher security risk than projects incorporating structured human review and automated security testing. The analysis also showed that vulnerability occurrence probability and review effectiveness were the most influential factors affecting overall software security outcomes. The study further demonstrated that combining human expertise with automated security controls produced the greatest reduction in estimated cybersecurity risk. These findings emphasize that AI-generated code should be regarded as a productivity-enhancing tool rather than a replacement for secure software engineering practices. Organizations adopting AI-assisted programming should establish comprehensive governance policies that include mandatory code review, static security analysis, secure coding standards, continuous vulnerability assessment, and developer security training before production deployment. The proposed Monte Carlo framework provides a practical decision-support approach for software organizations seeking to evaluate AI-assisted development risks under uncertain operating conditions. Because the methodology estimates a distribution of possible security outcomes rather than a single deterministic value, it enables more informed decision-making regarding software security investments and risk mitigation strategies. Although the simulation was developed using representative probability distributions and realistic software security assumptions, future studies may further improve the framework by incorporating empirical industrial datasets, additional cybersecurity variables,

different large language models, and advanced machine learning-based risk prediction techniques. Comparative analyses across programming languages, application domains, and AI coding platforms may also enhance understanding of AI-generated software security. In conclusion, the secure adoption of AI-generated code depends not only on advances in large language models but also on the integration of robust software security practices throughout the software development lifecycle. A balanced approach that combines AI-assisted programming with rigorous human oversight and automated security validation offers the most effective pathway for leveraging artificial intelligence while minimizing cybersecurity risks.

References

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- Butt, Muhammad Abdullah, Anas Mustafa, Rizwan Shukat, Talha Riaz, Ambreen Saleem, 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, no. 2 (2026): 2761-2776.
- 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.
- 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.
- 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.

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.

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.

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.

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.

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).

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.

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).

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.

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.

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, no. 2 (2026): 2398-2413.

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.

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, no. 2 (2026): 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.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., ... Zaremba, W. (2021). *Evaluating large language models trained on code*. arXiv. <https://arxiv.org/abs/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.

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.

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.

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 $\geq 35\%$ in *Zea mays* Seedlings: <https://doi.org/10.5281/zenodo.20031798>." *Research Consortium Archive* 4, no. 2 (2026): 1093-1102.

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.

FIRST. (2023). *Common Vulnerability Scoring System (CVSS) v3.1 Specification Document*. Forum of Incident Response and Security Teams. <https://www.first.org/cvss/>

Fu, Y., et al. (2024). *An empirical evaluation of vulnerability detection and repair in AI-generated code*. arXiv. <https://arxiv.org/abs/2310.02059>

Gulzar, Muhammad, Areeba Fatima, Mahpara Ch, 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.

Homer, J., Varikuti, A., Ou, X., & McQueen, M. (2021). Cyber risk analysis using probabilistic modeling: A review of Monte Carlo applications in cybersecurity. *Computers & Security*, 104, 102206.

Hughes, J., et al. (2023). *Package hallucination in large language models and software supply chain security*. arXiv.

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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- 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.
- Kula, R. G., German, D. M., Ouni, A., Ishio, T., & Inoue, K. (2018). Do developers update their library dependencies? *Empirical Software Engineering*, 23(1), 384–417.
- 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.
- Majdinasab, V., et al. (2023). *Assessing the security of GitHub Copilot-generated code: A targeted replication study*. arXiv. <https://arxiv.org/abs/2311.11177>
- McGraw, G. (2006). *Software Security: Building Security In*. Addison-Wesley.
- 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.
- 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.
- 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.
- 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.
- National Institute of Standards and Technology. (2023). *Security and Privacy Controls for Information*

Systems and Organizations (SP 800-53 Rev. 5). NIST.

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.

OWASP Foundation. (2021). *OWASP Top 10: The Ten Most Critical Web Application Security Risks*. <https://owasp.org/www-project-top-ten/>

Pearce, H., Ahmad, B., Tan, B., Dolan-Gavitt, B., & Karri, R. (2022). Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. In *Proceedings of the 2022 IEEE Symposium on Security and Privacy* (pp. 754–768).

Perry, N., Srivastava, M., Kumar, D., & Boneh, D. (2023). Do users write more insecure code with AI assistants? In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*.

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: <https://doi.org/10.5281/zenodo.19121209>." *Pakistan Journal of Medical & Cardiological Review* 5, no. 1 (2026): 1623-1632.

Raza, Hamad, Muhammad Abdullah Butt, Iza Almas, Hafiza Ariba Javaid, Talha Riaz, Ambreen Saleem, and Shazia Saeed. "Catalyst-Assisted Synthesis of Functionalized Organic Molecules for Sustainable Pharmaceutical and Industrial Applications." *Social Science Review Archives* 4, no. 1 (2026): 6589-6608.

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, no. 2 (2026): 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.

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.

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.

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.

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.

Rubinstein, R. Y., & Kroese, D. P. (2017). *Simulation and the Monte Carlo Method* (3rd ed.). John Wiley & Sons.

Sandoval, G., Pearce, H., Nys, T., Garg, S., Dolan-Gavitt, B., & Karri, R. (2022). *Lost at C? A user study on the security implications of large language model code assistants*. arXiv. <https://arxiv.org/abs/2208.09727>

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>

Shafeeq, A., T. Shokat, R. Rizvi, F. Gohar, R. Riaz, S. Ansar, B. Khalid et al. *Alleviation of adverse effects of*

- salinity stress on cauliflower by foliar application of trehalose. Scientific Records 3 (2): 84-100. 2026.*
- 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.
- 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.
- 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.
- Vaithilingam, P., Zhang, T., & Glassman, E. L. (2022). 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*.
- 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.