# Secure Engineering of Autonomous AI Agents: A Threat-Driven Development Framework

Tanvir Ahmed , Samiul Hasan , Ahammed Shorif , Ansarul Hoque , Shadman Sajid , Md. Badiuzzaman Biplob *

Posted Date: 14 August 2025

doi: 10.20944/preprints202508.1003.v1

Keywords: generative AI; threat model; AI agents; cybersecurity; attack vectors; security framework

 doi:10.20944/preprints202508.1003.v1



*Article*

# Secure Engineering of Autonomous AI Agents: A Threat-Driven Development Framework

**Tanvir Ahmed, Samiul Hasan, Ahammed Shorif, Ansarul Hoque, Shadman Sajid and Md. Badiuzzaman Biplob***

Department of CSE, International Islamic University Chittagong, Kumira, Chattogram, Bangladesh
* Correspondence: biplob.cse45@iiuc.ac.bd

**Abstract**

The integration of generative AI( Gen- AI) agents within business settings presents unique security challenges that differ from those of traditional systems. These agents extend beyond introductory LLMs, flaunting their ability to reason, retain information, and operate autonomously. This exploration introduces a comprehensive trouble model specifically adapted for Gen-AI agents, highlighting the new challenges associated with their independence, enduring memory access, advanced logic, and integration with tools. The study identifies nine significant pitfalls, grouped into five crucial categories: functional prosecution vulnerabilities, concession of trust boundaries, vulnerabilities within the cognitive armature, temporal continuity pitfalls, and governance endurance. Real-world issues, such as detainments in exploitability, cross-system spread, side movement, and subtle thing misalignment, are difficult to spot using current fabrics and conventional styles. To address these challenges, this study proposes two supplementary frameworks. The advanced trouble Framework for Autonomous AI Agents (ATFAA) categorizes pitfalls material to agents, while SHIELD offers practical threat mitigation strategies to reduce organizational exposure. While focusing on earlier AI security and LLM exploration, this study focuses on what distinguishes these agents and underscores the significance of these features. Eventually, Study 1 argues for a new security perspective for GenAI agents. Without reassessing our threat models and defenses to incorporate their specific infrastructures and actions, we threaten transubstantiating an important new tool into a substantial liability for enterprises.

**Keywords:** generative AI; threat model; AI agents; cybersecurity; attack vectors; security framework

## 1. Introduction

Generative artificial intelligence( GenAI) agents are emerging as a new form of artificial technology. Unlike conventional systems, they integrate planning features, constant memory access, third-party/internal tool integration, and large language models( LLMs) Shu et al. (2024). These agents not only induce responses but also proactively interact with systems, formulate opinions, and operate autonomously in diverse commercial environments, often with minimal human intervention AI (2023).

Their growing autonomy distinguishes them, particularly in terms of security challenges. GenAI agents can cross organizational boundaries, execute variable API calls, and handle enterprise data, occasionally without unequivocal user input Domkundwar et al. (2024). They are dynamic, adaptive, and deeply integrated into the operational processes. Standard security protocols may not sufficiently mitigate the risks posed by these agents. Agentic architecture, which combines reasoning components, memory systems, language interfaces, and external tools, creates a much larger and more intricate attack surface than most current frameworks are designed to manage Isabirye (2024). Frameworks such as the NIST AI Risk Management Framework AI (2023), MITRE ATLAS OWASP (2023), OWASP Top 10 for LLMs Jedrzejewski et al. (2025), and CSA MAESTRO Alliance (2025) are useful, although

 doi:10.20944/preprints202508.1003.v1



they often treat LLMs as separate components or offer high-level risk guidance. They frequently fail to consider the new security features that appear when long-term memory access, autonomy, and dynamic tool usage are combined. The purpose of this study is to close this gap. To handle the particular risks posed by GenAI agents, it presents the Advanced Threat Framework for Autonomous AI Agents (ATFAA) along with a corresponding protection model called SHIELD. The contributions to this effort include the following:

- Examination of GenAI agent architectures, emphasizing the security problems arising from tool use, autonomy, reasoning, and memory.
- A list of nine main threats that target these agentic capabilities.
- A review of pertinent attack avenues, including untested exploitation methods.
- Identifying risks to the STRIDE framework and developing customized mitigation techniques for SHIELD.

This threat model offers a strong foundation for developing security solutions tailored to the operational behavior of autonomous agents in a rapidly growing field. Without specific limitations, what appears to be a game-changing technology might easily become a significant problem for the company. Jedrzejewski et al. (2025).

## 2. Literature Review

The Internet and its related services are now used by society in many different fields and have become an essential part of the infrastructure supporting modern life. The increasing dependence on digital technologies emphasizes the critical function of cybersecurity, a sector that impacts many other fields. Cybersecurity has changed from an optional security measure to a vital cornerstone as digital systems become increasingly integrated into daily life and commercial operations. It serves as the foundation for protection in other areas, guaranteeing that data is private, safe, and accessible Pawlicki et al. (2024).

AI plays a key role in cybersecurity by enhancing the interpretability and transparency of the machine learning models. This enables security professionals to more accurately assess and identify dangers, weaknesses, or hostile attacks, bolstering and enhancing cyber-defense systems. Significant developments and enduring difficulties in this field are reflected in the literature on AI-driven cybersecurity for coding secure software. This section examines previous studies, emphasizing their significant contributions, approaches, and knowledge gaps. The following four major themes served as the framework for this review.

### *2.1. AI Applications in Cybersecurity*

AI has completely transformed cybersecurity by automating intelligence activities in a timely, accurate, and thorough manner to counter threats, artificial intelligence (AI) has completely transformed cybersecurity. Threat detection and forecasting are two important uses of AI in cybersecurity, wherein AI systems search through massive databases for warning signs of possible security threats. Algorithms assist firms preventingent system breaches by enabling machines to evaluate historical data and predict future vulnerabilities Al-Mhiqani et al. (2024). Furthermore, AI-powered intrusion detection systems (IDS) facilitate the real-time monitoring of network traffic in real time in order to identify unusual trends and malicious activity; cutting-edge methods, such as support vector machines (SVMs) and deep learning, improve the precision of identifying zero-day attacks Yeoh et al. (2023). By using natural language processing (NLP) and machine learning (ML) techniques to evaluate message content and verify URL and sender credibility, artificial intelligence (AI) plays a critical role in thwarting phishing and social engineering Nanda et al. (2024). Convolutional Neural Networks (CNN) provide high detection accuracy for various malware types, and AI-driven systems conduct comprehensive malware analysis using both static and dynamic methods Vouvoutsis et al. (2025). AI-based endpoint detection and response (EDR) systems monitor device activity to stop unwanted access and possible dangers, which improves the identification of advanced persistent threats (APTs) for targeted data monitoring

 doi:10.20944/preprints202508.1003.v1



for endpoint protection Admass et al. (2024). Additionally, AI automates incident response procedures, which greatly accelerates the mitigation of possible threats to the system. Security Orchestration Automation Response (SOAR) platforms employ AI to efficiently handle threats Kaur et al. (2023). AI also improves cryptography protocols by refining algorithms to support secure communication. Finally, by creating risk assessments for focused assessments, encouraging targeted remedial activities, and using predictive models to anticipate possible cyber-criminal exploits, AI helps organizations manage vulnerabilities Wang et al. (2024).

### *2.2. Cybersecurity risks and vulnerabilities: impact on software coding*

The results of software coding can alter how a system would normally secure its security framework due to the substantial influence of cybersecurity threats and current vulnerabilities. Unpatched code flaws serve as attack vectors, giving unauthorized parties the ability to compromise data and interfere with operations of the system. Poor secure authentication procedures and insufficient input validation make systems vulnerable to attacks such as SQL injection and cross-site scripting Khan et al. (2024).Secure coding is essential because of the rise in sophisticated attacks, such as zero-day vulnerabilities. Because these zero-day attacks target unidentified system faults while they are still vulnerable, developers must constantly manage these vulnerabilities Itodo and Ozer (2024).As outdated cryptographic standards and lax encryption methods present security risks and could expose private information to unauthorized parties, strong encryption techniques are crucial Hasan et al. (2021). According to developmental standards, integrating third-party libraries with development frameworks poses potential risks that could compromise project success. The significance of careful dependency management is highlighted by the existence of active vulnerabilities and hazardous code obligations in software components. During the software development process, developers must implement defenses against integrity assaults on models caused by AI and machine learning. Approaches that include regular security audits, established secure coding techniques, and developer education regarding emerging cyber threats are necessary for the development of secure software. Research shows that firms can defend against possible risks and create more robust systems by integrating cybersecurity considerations into every phase of software development Khan et al. (2022).

### *2.3. Secure Software Coding Practices*

The confidentiality, availability, and integrity of software systems are safeguarded when software is programmed using secure coding principles. These procedures help increase confidence in digital networks by preventing exploitative assaults and safeguarding private information. By restricting user and system rights to the minimal levels required for their work responsibilities, developers uphold security using the least-privilege principle Chang et al. (2018). To reduce data exposure to unauthorized breaches, an application handling critical user data must restrict access to this data to authorized program processes only Admass et al. (2024).Input validation and sanitization are essential components of secure coding. User-supplied inputs are the primary point of entry for attackers, particularly when they are not verified properly. To lower these risks, developers must employ output-escaping techniques, parameterized queries, and thorough validation checks Khan and Khan (2018). Developers should use prepared statements rather than direct SQL query concatenation because secure programming necessitates that input parameters and actual statements be separated. To identify defects early, software deployment necessitates a dual strategic approach that combines code reviews with static analysis. Peer reviews, in which multiple team members evaluate the code, provide organizations with various perspectives on compliance with quality and security standards. SonarQube and Veracode are code analysis tools that keep an eye 4 on security threats by alerting developers to issues like buffer overflows and unsafe cryptography usage Manjunath and Baunach (2024). Compared to conventional post-deployment detection methods, post-sectional vulnerability rectification costs are reduced when computerized tools are integrated within the SDLC.

It is also essential to include secure libraries and frameworks. Secure authentication APIs and OpenSSL cryptographic interfaces are proven frameworks that assist developers in avoiding introduc-

 doi:10.20944/preprints202508.1003.v1



tioncing of unsafe code into their systems Hasan et al. (2021). Because outdated dependencies might pose security risks, developers are responsible for updating libraries. Finally, thorough monitoring and logging provide insights into security issues and application behavior. Data anonymization and steps to ensure tamper-proof log file management are both part of secure logging practices Patel et al. (2024).Teams may react swiftly to cybersecurity risks by using monitoring tools such as Splunk and ELK Stack, which alert them to unusual activity Nanda et al. (2024). By incorporating robust cybersecurity policies throughout the Systems Development Life Cycle framework, organizations can build robust information software systems. The pace at which AI technology is incorporated into cybersecurity procedures is modest. For instance, Gurtu and Lim (2025), demonstrated that automatic secure code analysis tools perform only modestly because of their poor ability to comprehend contextual settings, leading to inaccurate results and recognition errors. The accurate performance and flexible usefulness of these tools can be enhanced using AI-based reinforcement learning approaches.

### *2.4. Integration of AI Frameworks with Maturity Models*

To properly match AI deployment to an organization's readiness and strategic goals, AI frameworks must be in line with maturity models. AI frameworks provide the technical foundation for task automation, improving decision-making, and increasing predictive accuracy is provided by AI frameworks comprise tools, methodologies, and technologies for developing and implementing AI solutions. Expert instruments assess how well companies handle certain aspects, such as strategy and technology, across specific phases of development. Organizations can move beyond user-reliant AI applications by combining AI frameworks with maturity models to create an alignment that maximizes technology adoption and reduces risks. For example, the Capability Maturity Model Integration (CMMI) addresses specific AI-related concerns, such as data integrity, ethical considerations, and model coordination requirements Gurtu and Lim (2025).In addition to working with maturity models to guide technological investments and specify the deployment of suitable solutions depending on an organization's readiness level, AI frameworks assist organizations in methodically identifying operational skill deficiencies. Inexperienced organizations frequently face knowledge gaps and data isolation. However, the use of AI tools (such as PyTorch or TensorFlow) requires workforce development and strategic planning.More established businesses can incorporate AI with advanced tactics, such as autonomous systems and generative AI, for intricate innovation and decision-making. Additionally, this integration improves compliance and governance Ilyas et al. (2024).Organizations benefit from maturity models that integrate proven best practices to meet public and regulatory expectations. Explainability features and fairness mechanisms (such as AI Fairness 360 or LIME) integrated into AI frameworks enforce a certain degree of rigor before deployment. The return on investment increases when investments in AI 5 are explicitly connected to quantifiable business results. According to McKinsey research, businesses may launch AI initiatives more quickly when AI frameworks and maturity models are applied together than when they are used separately. The results indicate that when adopting AI initiatives, firms that use this synergy obtain a 30% boost in efficiency compared to those that only align with strategy. Organizations may advance the adoption of transformational AI while incorporating sustainability and scalability into the process by utilizing the convergence of AI frameworks and maturity models. Consequently, combining AI frameworks with assessments of organizational preparedness is advancing AI deployment by defining moral means of achieving noteworthy results in various industries. Large Language Models (LLMs) and Artificial Intelligence (AI) have recently advanced to the point where AI-based code generation tools are a viable option for software development Zhou et al. (2025). Large Language Models (LLMs) are a kind of deep learning-based natural language processing (NLP) method that can automatically learn language grammar, semantics, and pragmatics while producing a vast array of information.LLMs have demonstrated remarkable NLP capabilities with their extensive training datasets and wide range of parameters.often reaching or surpassing human-like competence in tasks such as sentiment analysis and text translation. LLM-powered AI code creation systems that have been thoroughly trained on code snippets have recently become more popular (as mentioned in AI-augmented development in Gartner Trends 2024). These AI tools can generate answers that are

 doi:10.20944/preprints202508.1003.v1



more sophisticated than those of inexperienced programmers for straightforward and moderately complex coding problems. Kuhail et al. (2024). Large language model (LLM)-based generative artificial intelligence (GenAI) tools, such as ChatGPT and GitHub Copilot, which can write code, have the potential to revolutionize the software development industry.

However, owing to its sophisticated use of predictive AI techniques on continuous learning with the ISM, the proposed "AI-driven Cybersecurity Framework for Software Development Based on the ANN-ISM Paradigm" surpasses conventional cybersecurity maturity models, such as the NIST framework or CMMI, in this regard. Consequently, this model outperforms the current framework for the following reasons:

- *Proactive threat identification and prevention*: The ANN-ISM paradigm's primary strength is its ability to forecast using Artificial Neural Networks (ANN). An ANN learns from each new piece of historical or real-time data and anticipates potential dangers before they become problems. Compared to traditional approaches, it has the advantage of early intervention and prevention because it proactively detects new cyber security threats.

  The NIST Cybersecurity Framework and Capability Maturity Model Integration (CMMI) are excellent for risk management, compliance, and process improvement, but what is reactive? In contrast, the framework emphasizes organized methods for addressing cybersecurity threats and weaknesses. However, they do not have the real-time predictive capabilities necessary to proactively mitigate the unknown and dynamic risks at play.

- *Adaptability to emerging threats*: Your model's ANN component enables continuous learning in the ANN-ISM Paradigm. The framework thus combines ISM and AI threat detection to generate real-time alerts and countermeasures. As fresh data are collected, this system adjusts to the new attack signals and changes to keep up with the speed of cyberthreats. Security policies are further improved by integrating the ISM framework, which increases the system's adaptability and flexibility in changing contexts.

  The CMMI and NIST approaches have static frameworks that are Even while they offer good suggestions for future cybersecurity best practices, they are unable to respond automatically to novel threats or environmental changes without human oversight. For instance, the NIST requires constant human oversight to adapt controls and processes to newly obtained threat intelligence, even though it depends on a risk-based management approach and associated best practices.

- *Real-time threat detection and response*: Our model detects the same cybersecurity problem in real-time and responds to it automatically by taking the necessary action immediately. The framework thus combines ISM and AI threat detection to generate real-time alerts and countermeasures.

  Real-time threat detection and response are not supported by the NIST and CMMI frameworks, although they facilitate the creation of detailed security processes. Although the NIST emphasizes monitoring for continual development, it lacks the capability to respond quickly to threats, as exemplified by the ANN-ISM model. Similarly, CMMI is more suited to organizational capability and process maturity than to immediate cybersecurity incident response.

- Flexibility and Scalability: When neural networks and ISM are combined (ANN-ISM paradigm), the framework is incredibly scalable. Without compromising its functionality, it can be developed over time to accommodate ever-increasing data volumes and security requirements. With the amount of data, machine learning is used in the framework to enhance the system and detect intricate and unknown dangers more accurately.

  The CMMI and NIST frameworks are both excellent tools for offering an organized method of approaching the concept of cybersecurity maturity. They are typically less adaptable to resolving novel, intricate, or expanding issues. The security procedure may be enhanced, and a baseline established with the aid of these models. However, they do not scale effectively by nature to accommodate growing data, complexity, and threats without manual updates or modifications.

 doi:10.20944/preprints202508.1003.v1



- *It facilitates the management and continuous improvement of automated process*: The model's ISM component incorporates the ANN-ISM Paradigm, a continuous improvement cycle, into the framework. Through the use of machine learning and security management procedures, the system continuously assesses its operations and makes the necessary adjustments in response. Consequently, the system can use new information and shifts in the threat landscape to continuously improve its cybersecurity protocols.

  The CMMI Framework focuses more on continuous process improvement and emphasizes organizational procedures and capability maturity. It doesn't take into 7 account the fact that data-driven learning is automated. The NIST framework provides guidelines for improved security postures. Nevertheless, it lacks an automated upgrading procedure that would enable constant security posture improvement and relies on sporadic manual upgrades.

- *Cost-effectiveness over time*: The initial expenditure may be higher due to the blend of AI and continuous learning capabilities in the ANN-ISM technique. However, by reducing incident response expenses, improving the security posture, and eliminating manual intervention, this becomes less costly over time. The cost of cybersecurity management is low because it does not include the cost of stopping attacks before they occur.

  More substantial operational costs are associated with frameworks such as the NIST and CMMI frameworks, which use manual procedures, frequent upgrades, and frequent process monitoring and alignment for improvement. Although these procedures are useful for managing cybersecurity, they are not autonomous threat detection systems or constantly changing without human assistance.

- *Holistic security approach*: The ANN-ISM paradigm offers a comprehensive perspective on cybersecurity by combining structured security management with AI-based predictive analytics. This guarantees that every aspect of cybersecurity, from policy administration to threat detection, functions in unison to provide a comprehensive solution that addresses both managerial and technical aspects of cybersecurity.

  Although both the CMMI and NIST Cybersecurity Frameworks offer helpful advice for organizations to strengthen their cybersecurity practices, their positions are more dispersed. While CMMI covers process maturity to a considerable extent without fully including predictive technology (CTIS) and continuous learning of security architecture, NIST deals with risk management and control setups.

The Cybersecurity Framework for Software Development Based on the ANN—ANNISM paradigm offers a more dynamic, adaptable, and predictive approach than the conventional NIST and CMMI models. The NIST and CMMI frameworks offer a useful foundation for managing and enhancing security procedures over time; however, because they necessitate manual intervention, they are reactive and expensive when dealing with novel and unexpected threats. In exchange for the loss of many of the less efficient tools of human bureaucracy, the AI-powered ANN-ISM paradigm increases cybersecurity's scalability to changing surroundings, automation, and real-timeness.

## 3. Methodology

Methodology The research methodology of this study is intended to guarantee the methodical and exacting procedures needed to build a strong security mitigation model. The five main stages of this process are the Systematic Literature Review, Questionnaire Survey, Expert Panel Review, Artificial Neural Networks Analysis, and Interpretive Structural Modeling (ISM) are the five main stages 8 of this process. The overall objective of creating a combined AI-Driven Cybersecurity Mitigation Framework for Secure Software Coding: An ANN-ISM Framework is achieved through an iterative approach that is aided by each phase. The graphical research structure of this study is shown in Figure 1.

 doi:10.20944/preprints202508.1003.v1



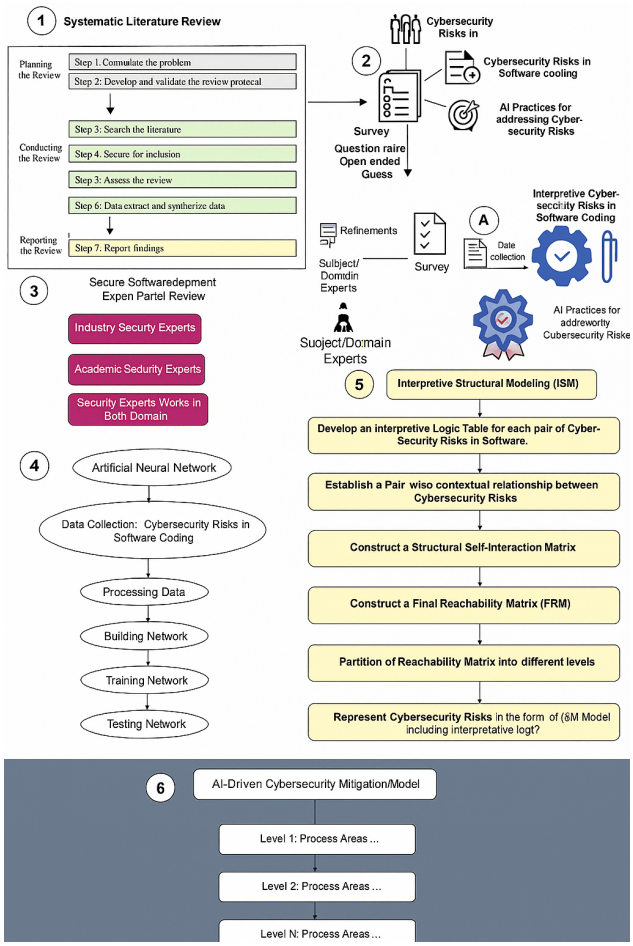


**Figure 1**

### *3.1. Phase 1: Systematic Literature Review (SLR)*

The goal of this study was to create a thorough grasp of the scientific literature on secure software coding with an emphasis on cybersecurity dangers and the methods that may be used to reduce them. The report offers insightful opinions from both industry professionals and scholarly experts. Scholars are actively looking for a thorough resource that offers a wide viewpoint on the latest developments in secure software development. At the same time, practitioners are keen to learn about new academic trends and technologies that can be applied effectively in practical settings. A systematic literature review (SLR) technique was used to apply a well-established guideline.Reference: Khan et al. (2024) Kitchenham et al. (2009).Under the direction of the co-authors, the lead author developed a rigorous review methodology, carried out comprehensive searches, vetted papers, and carried out data extraction while being closely supervised by the second 9 author, who specializes in SLRs. Every author took an active part in the data analysis process. As the main author prepared the analytical report and arranged the data into tables and graphs, the other authors participated in discussions, offered comments, and offered ongoing feedback.

 doi:10.20944/preprints202508.1003.v1



### 3.2. Search Strategy

| Digital libraries | Search string findings | Initial selection | Final selection |
|---|---|---|---|
| IEEE Xplore | 98 | 28 | 19 |
| ScienceDirect | 150 | 33 | 18 |
| ACM | 135 | 39 | 15 |
| Wiley Online Library | 60 | 10 | 8 |
| SpringerLink | 230 | 50 | 16 |
| Google Scholar | 2021 | 60 | 14 |
| Total | 2694 | 220 | 90 |

### 3.3. Data Extraction

After thoroughly reviewing each original study, we collected data straight from the publication. Both qualitative and quantitative data were included in this: (i) cybersecurity risks in software coding and (ii) AI strategies for mitigating cybersecurity risks were included in the qualitative data. The sample size, or the number of participants, data sources, and dataset lengths were all included in the quantitative data.

### 3.4. The Findings

Section 4 of this publication organizes and presents the SLR's comprehensive findings. As a result, these precise and methodical procedures will fully examine cybersecurity risks and the AI-based countermeasures used in software development.

## 4. Findings and Conversations

This section provides a summary of the results from the hybrid study approach, which included ANN, ISM expert panel review, empirical survey, and SLR The following subsections provide more detail on the answers to the research questions presented in Section 3.

Potential Cybersecurity Risks and Vulnerabilities in Software Coding

Important insights are revealed by examining relevant literature and industry research on security threats and vulnerabilities in software development processes. These flaws pose serious security concerns, possible monetary losses, and reputational damages if left unchecked. Seeing weaknesses from two angles provides both theoretical insights and concrete realities, as does academic research and real-world implementation. These kinds of literature evaluations encourage paradigm shifts and go over earlier case studies to improve the fundamental knowledge of software vulnerabilities. In the meantime, industry surveys offer a glimpse of the current threat landscape, bridging the gap between research and practice with useful information and data derived from real-world situations. This dual approach guarantees that policymakers, cybersecurity specialists, and software developers are knowledgeable about theoretical concerns and have workable solutions that are suited to current challenges.

Additionally, by breaking down vulnerabilities, we may focus on critical problems like poor authentication practices, dangerous coding methods, and insufficient encryption, which may lead to targeted action. Organizations can develop more secure software by identifying these risks early on and implementing preventative measures. Organizations must maintain regulatory compliance, build end-user confidence, and fortify their software systems in an increasingly dynamic digital world.

Using linear regression, we analyzed and contrasted software security coding hazards found in surveys and scholarly papers. We used a statistical correlation metric to measure the degree to which the survey results (software security coding risks) align with the literature review.The existence and strength of a relationship between two variables can be established with the help of this technique. This link is represented by the correlation coefficient, which ranges from [-1] to 1. While 0 denotes no association, [+1] denotes a perfect positive correlation, [-1] a perfect negative correlation. The Pearson

 doi:10.20944/preprints202508.1003.v1



correlation coefficient, the most prevalent kind of correlation, can be computed using the method below:

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \sum(y_i - \overline{y})^2}} \tag{1}$$

where xi and yi represent individual values of the two variables (Literature Review and Industry Survey), and $\overline{x}$ and $\overline{y}$ are their respective means. The mean impact percentage of literature is 12.13, while that of the industry survey is 8.47. With standard deviations of 25.33 for the industry survey and 29.05 for the literature, the total product of deviations is 638.07.When these numbers are used instead, the Pearson correlation coefficient is

$$r = \frac{638.07}{29.05 \times 25.33} \approx 0.867 \tag{2}$$

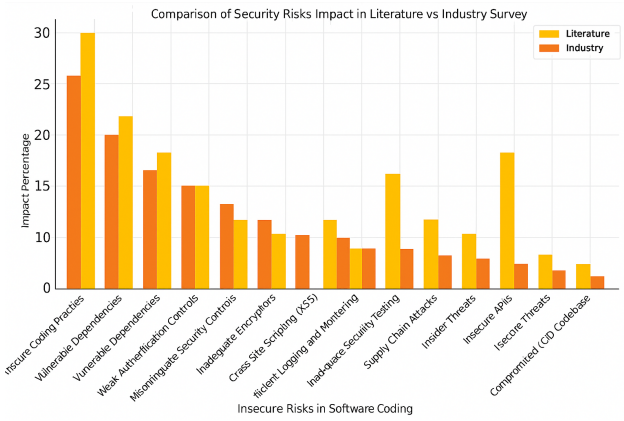


**Figure 2.** Software security coding risks as determined by surveys and literature reviews are compared.

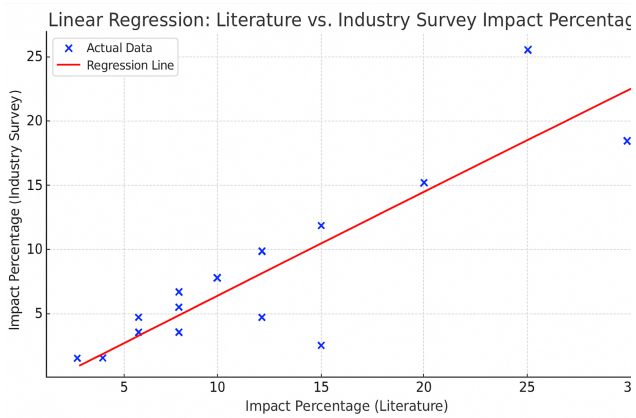


**Figure 3.** Software security coding risks detected by surveys and literature reviews are linearly regressed.

This result indicates a strong positive linear relationship between literature review findings and industry survey results, suggesting significant alignment between academic research and real-world cybersecurity risks.

## 5. Development of AI-Driven Cybersecurity Mitigation Model for Secure Software Coding: Using ANN-ISM Approach

The frameworks of SAMM Jaatun et al. (2015), BSIMM, and SCCCMM are the foundation of the suggested AI-driven cybersecurity mitigation approach for secure software coding. Using these models as a basis, five levels—each comprising different process areas—were modified to produce the

 doi:10.20944/preprints202508.1003.v1



suggested model. the overall process used to create the suggested model. The AI-driven cybersecurity mitigation approach for secure software coding was developed using the following procedures:

Data collection:

- ANN Data: We combine data from a survey of academic sources plus real-world studies to create a strong database to train the ANN system. We normalize and prepare our data to make it consistent and exact in its measurement.
- ISM Data: Through online interviews with professionals, we performed focus groups and surveys to learn how cybersecurity threats impact software coding.

Model building

- Training of the ANN: The model's ability to process qualitative data is essential to the trained ANN. In order to forecast cybersecurity risk situations that are advantageous to each software coding system, the neural network system makes use of input data relationships.
- Constructing ISM: Qualitative data becomes the foundation for designing an ISM chart that depicts all security risk influences on software coding security.

Model integration

- Hybrid Framework: Our group develops a single, cohesive strategy that blends ISM and ANN results. While ANN aids in threat prediction and secure setting identification, ISM demonstrates how various cybersecurity hazards interact with their fundamental elements.
- Model Validation: Our integrated model is put through a number of tests using a variety of datasets and security professionals to demonstrate that it can identify and address coding security issues in software.
- Implementation: By combining ANN forecasts with ISM analysis, the validated model creates comprehensive security protection techniques for coding projects, thereby preventing software coding issues.

With five stages of increasing complexity to address cybersecurity threats, Figure 4 illustrates our AI-driven strategy for software coding system protection utilizing ANNISM. Software coding protection is the responsibility of specific process domains at each level. The following subsections present the breakdown analysis and significance of each level:

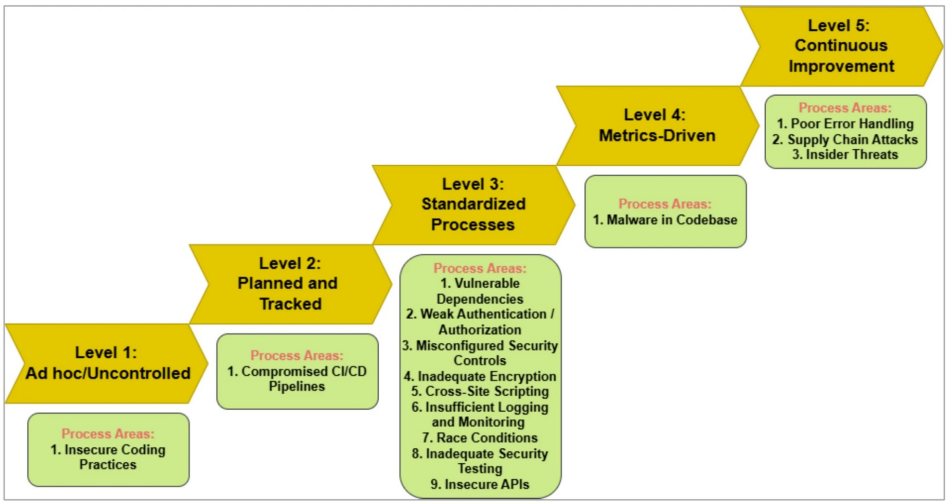


**Figure 4.** AI-driven cybersecurity mitigation model for secure software coding.

Our ANNISM-based AI-driven model for software coding system protection, which offers five tiers of increasing complexity to address cybersecurity threats, is depicted in Figure 4. The process areas in charge of safeguarding the software coding are distinct for each level, with different levels representing increasing maturity and security sophistication. The following subsection describe more explanation of the figure:

 doi:10.20944/preprints202508.1003.v1



*5.1. Levels Overview*

- Level-1: Ad hoc/Uncontrolled
  - Median Score:3
  - Appraisal: Advanced
  - Analysis: The software development organization received an Advanced appraisal, demonstrating the successful implementation of fundamental security measures, even though it falls into an early-stage category. AI technologies are used by 13 organizations to identify coding security flaws and prioritize taking the necessary corrective action. Because the strategies lack defined principles and effective distribution methods, they rely on informal tactics.
- Level-2: Planned and Tracked
  - Median Score: 3
  - Appraisal: Advanced
  - Analysis: At this level, the secure software development businesses showed that their software security measures were planned and monitored. Organizations at the advanced level concentrate on allowing AI systems to identify anomalous system behavior and dangers. Installing security measures lays the groundwork for continued increased development phases.
- Level-3: Standardized Processes
  - Median Score: 3
  - Appraisal: Advanced
  - Analysis: The core focus is keeping all business systems uniform across all units. This phase shows that our organization uses AI correctly through established industry techniques:
    - Dependency Scanning
    - Vulnerability management
    - Secure Frameworks: This consistency ensures scalability and fosters long-term security resilience.
- Level-4: Metrics Driven
  - Median Score- 3
  - Appraisal- Advanced
  - Analysis: Measurable criteria help the association ameliorate, but an appraisal reveals performance pretensions that need adaptation:
    - AI is not being used enough for incident response and sophisticated monitoring.
    - Lack of established marks to measure and optimize AI-driven security sweeps. fastening on enriching criteria and using data perceptivity can elevate this position to advanced maturity.
- Level-5: Nonstop enhancement
  - Median Score- 3
  - Appraisal: Advanced
  - Analysis: The company exhibits early appreciation of nonstop enhancement as it begins to use performance pointers, but it also has difficulties when using AI advancements on a large scale. Choosing realistic path conditioning is difficult for the association since performance measures punctuate performance weaknesses:
    - Limited AI relinquishment in dynamic trouble modeling or real-time trouble responses.
    - There's a straightforward procedure to ameliorate security measures. Cutting-edge AI tools and training will help the company turn its core moxie into comprehensive security advancements.

Crucial compliances:

 doi:10.20944/preprints202508.1003.v1



- Advanced Security at original situations. The association's high appraisals in the first three situations indicate strong foundational and standardized practices.
- Decline in After situations. The major drop in conditions at the enhancement and Understanding situations shows that our association needs to strengthen its investment in security criteria while streamlining its systems continuously.
- AI as a Differentiator Beforehand-level success pointers show how well AI tools support secure programming while collecting trouble data and constantly covering.

Recommendations for enhancement:

- Refine Metrics and KPIs( Level 4): apply measurable security criteria for assessing AI effectiveness, similar as:
  - Time to descry respond to pitfalls.
  - The number of vulnerabilities renovated through AI robotization.
- Expand nonstop enhancement Efforts( Level 5):
  - Incorporate adaptive AI systems for real-time monitoring and predictive trouble modeling.
  - Establishing a feedback medium to learn from once security incidents and acclimate AI-driven processes consequently.
- Enhance Training and mindfulness: Investment in training inventors to align their practices with AI tools, especially at situations 4 and 5.

| Levels | **Five orders of AI-driven cybersecurity mitigation model for secure software rendering** | Median | **Appraisal of software development association** |
|---|---|---|---|
| Level 1 | Ad hoc/ unbridled | 3 | Advanced |
| Level 2 | Planned and tracked | 3 | Advanced |
| Level 3 | Formalized processes | 3 | Advanced |
| Level 4 | Metrics driven | 2 | Enhancement |
| Level 5 | Nonstop Enhancement | 1 | Understanding |

## 6. Conclusions

By combining ANN and ISM to produce an AI-driven result that identifies security excrescencies and reduces cybersecurity pitfalls, this study enhances how well software businesses manage security during software development processes. Beforehand in the software development lifecycle, AI has successfully set up vulnerabilities, preventing pitfalls before they have a chance to affect the finished product. The ANN-ISM frame guards against XSS assaults, identifies law injection excrescencies, tracks security pitfalls, and stops buffer overflow situations in real time. Adaptive cybersecurity protection across devel- opment platforms is achieved by combining security protocols with machine- learning technologies. The analysis demonstrates why, rather than being a voluntary compo- nent, security ought to be incorporated into every phase of software development. By putting cybersecurity measures into place beforehand on, we may produce software systems that are more secure and avoid spending time and money patching vulnerabilities after they are discovered. Although the ANN- ISM armature offers an implicit approach, more study is needed to enhance its performance and operation. Its commerce with DevOps and nonstop deployment technologies throughout the development lifecycle should be delved into unborn systems.

Future exploration should concentrate on:

- Probing the possibility of combining ANN with different AI fabrics, similar underpinning literacy and inheritable algorithms, in order to ameliorate cybersecurity features could be the focus of future exploration.
- Operations across different disciplines: The ANN-ISM approach might be applicable to other areas, including pall computing protection, IoT security, and cybersecurity fabrics grounded on Blockchain technology.
- Perpetration and confirmation in practice unborn work needs to acclimatize and estimate this frame in practice, integrated into software development workflows.

 doi:10.20944/preprints202508.1003.v1



Our findings introduce a new AI-driven approach to secure software development, which has a substantial influence on professionals in advanced education and industry. By automating trouble discovery, lowering the threat of unauthorized access, and enhancing vulnerability detection through ANN-grounded literacy, the system improves cybersecurity. Its cross-domain connection includes mobile operation development, pall security, and the Internet of Things. It enhances scalability and cost effectiveness while promoting secure coding styles. Relinquishment may also have an impact on regulation- ulatory fabrics and cybersecurity regulations. Although the results show how AI technology can ameliorate cybersecurity, it's important to fete its limitations, which include issues with data scarcity and quality, model complexity, generalization- tion to colorful software surroundings, performance outflow, rigidity to changing pitfalls, and confirmation/ testing difficulties.To further increase software defense in our increasingly digital terrain, unborn exploration should concentrate on perfecting the frame's rigidity to varied security surrounds and broadening its operation across multitudinous scripts.

## References


Admass, Wasyihun Sema, Yirga Yayeh Munaye, and Abebe Abeshu Diro. 2024. Cyber security: State of the art, challenges and future directions. *Cyber Security and Applications* 2, 100031.

AI, NIST. 2023. Artificial intelligence risk management framework (ai rmf 1.0). *https://nvlpubs.nist.gov/nistpubs/ai/nist.ai*, 100–1.

Al-Mhiqani, Mohammed Nasser, Tariq Alsboui, Taher Al-Shehari, Karrar hameed Abdulkareem, Rabiah Ahmad, and Mazin Abed Mohammed. 2024. Insider threat detection in cyber-physical systems: a systematic literature review. *Computers and Electrical Engineering 119*, 109489.

Alliance, Cloud Security. 2025. *"Agentic AI threat modeling framework: MAESTRO,"*. Ph. D. thesis, OWASP. CSA blog: https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro.

Chang, Younghoon, Siew Fan Wong, Christian Fernando Libaque-Saenz, and Hwansoo Lee. 2018. The role of privacy policy on consumers' perceived privacy. *Government Information Quarterly 35*(3), 445–459.

Domkundwar, Ishaan, Ishaan Bhola, Riddhik Kochhar, et al. 2024. Safeguarding ai agents: Developing and analyzing safety architectures. *arXivpreprintarXiv:2409.03793*.

Gurtu, Anurag and Damien Lim. 2025. Use of artificial intelligence (ai) in cybersecurity. In *Computer and information security handbook*, pp. 1617–1624. Elsevier.

Hasan, Mohammad Kamrul, Muhammad Shafiq, Shayla Islam, Bishwajeet Pandey, Yousef A Baker El-Ebiary, Nazmus Shaker Nafi, R Ciro Rodriguez, and Doris Esenarro Vargas. 2021. Lightweight cryptographic algorithms for guessing attack protection in complex internet of things applications. *Complexity 2021*(1), 5540296.

Ilyas, Muhammad, Siffat Ullah Khan, Habib Ullah Khan, and Nasir Rashid. 2024. Software integration model: An assessment tool for global software development vendors. *Journal of Software: Evolution and Process 36*(4), e2540.

Isabirye, Edward. 2024. Securing the ai supply chain: Mitigating vulnerabilities in ai model development and deployment. *World Journal of Advanced Research and Reviews 22*(2), 2336–2346.

Itodo, Cornelius and Murat Ozer. 2024. Multivocal literature review on zero-trust security implementation. *Computers & Security*, 103827.

Jaatun, Martin Gilje, Daniela S Cruzes, Karin Bernsmed, Inger Anne Tøndel, and Lillian Røstad. 2015. Software security maturity in public organisations. In *International Conference on Information Security*, pp. 120–138. Springer.

Jedrzejewski, Felix Viktor, Davide Fucci, and Oleksandr Adamov. 2025. Threat modeling of large language model-integrated applications. *arXivpreprintarXiv:2504.18369*.

Kaur, Ramanpreet, Dušan Gabrijelčič, and Tomaž Klobučar. 2023. Artificial intelligence for cybersecurity: Literature review and future research directions. *Information Fusion 97*, 101804.

Khan, Rafiq Ahmad and Siffat Ullah Khan. 2018. A preliminary structure of software security assurance model. In *Proceedings of the 13th International Conference on Global Software Engineering*, pp. 137–140.

Khan, Rafiq Ahmad, Siffat Ullah Khan, Muhammad Azeem Akbar, and Musaad Alzahrani. 2024. Security risks of global software development life cycle: Industry practitioner's perspective. *Journal of Software: Evolution and Process 36*(3), e2521.

 doi:10.20944/preprints202508.1003.v1



Khan, Rafiq Ahmad, Siffat Ullah Khan, Habib Ullah Khan, and Muhammad Ilyas. 2022. Systematic literature review on security risks and its practices in secure software development. *ieee Access 10*, 5456–5481.

Kitchenham, Barbara, O Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. 2009. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology 51*(1), 7–15.

Kuhail, Mohammad Amin, Sujith Samuel Mathew, Ashraf Khalil, Jose Berengueres, and Syed Jawad Hussain Shah. 2024. "will i be replaced?" assessing chatgpt's effect on software development and programmer perceptions of ai tools. *Science of Computer Programming 235*, 103111.

Manjunath, Vignesh and Marcel Baunach. 2024. A framework for static analysis and verification of low-level rtos code. *Journal of Systems Architecture 154*, 103220.

Nanda, Manika, Mala Saraswat, and Pankaj Kumar Sharma. 2024. Enhancing cybersecurity: A review and comparative analysis of convolutional neural network approaches for detecting url-based phishing attacks. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 100533.

OWASP, Top. 2023. Owasp top 10 for large language model applications.

Patel, Soham, Kailas Patil, and Prawit Chumchu. 2024. Bhramari: Bug driven highly reusable automated model for automated test bed generation and integration. *Software Impacts 21*, 100687.

Pawlicki, Marek, Aleksandra Pawlicka, Rafał Kozik, and Michał Choraś. 2024. Advanced insights through systematic analysis: Mapping future research directions and opportunities for xai in deep learning and artificial intelligence used in cybersecurity. *Neurocomputing*, 127759.

Shu, Raphael, Nilaksh Das, Michelle Yuan, Monica Sunkara, and Yi Zhang. 2024. Towards effective genai multi-agent collaboration: Design and evaluation for enterprise applications.

Vouvoutsis, Vasilis, Fran Casino, and Constantinos Patsakis. 2025. Beyond the sandbox: Leveraging symbolic execution for evasive malware classification. *Computers & Security 149*, 104193.

Wang, Pingyan, Shaoying Liu, Ai Liu, and Wen Jiang. 2024. Detecting security vulnerabilities with vulnerability nets. *Journal of Systems and Software 208*, 111902.

Yeoh, William, Marina Liu, Malcolm Shore, and Frank Jiang. 2023. Zero trust cybersecurity: Critical success factors and a maturity assessment framework. *Computers & Security 133*, 103412.

Zhou, Xiyu, Peng Liang, Beiqi Zhang, Zengyang Li, Aakash Ahmad, Mojtaba Shahin, and Muhammad Waseem. 2025. Exploring the problems, their causes and solutions of ai pair programming: A study on github and stack overflow. *Journal of Systems and Software 219*, 112204.