UNIVERSITY OF
GLOUCESTERSHIRE

This is a peer-reviewed, final published version of the following document, © The Author(s) 2026 . This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/. and is licensed under Creative Commons: Attribution-Noncommercial-No Derivative Works 4.0 license:

Shafiq, Muhammad, Awan, Mujtaba ORCID logoORCID: https://orcid.org/0000-0001-9055-2869, Alwageed, Hathal Salamah, Umar, Hassan, Mohammad Mehedi, Khan, Rafiq Ahmad and Hossain, M Anwar (2026) Exploring an AI-driven agentic framework for insider threat detection in secure software development organizations: A survey-based approach. Scientific Reports, 16. art:32855. doi:10.1038/s41598-026-64146-y

Official URL: https://doi.org/10.1038/s41598-026-64146-y
DOI: http://dx.doi.org/10.1038/s41598-026-64146-y
EPrint URI: https://eprints.glos.ac.uk/id/eprint/16615

Disclaimer

The University of Gloucestershire has obtained warranties from all depositors as to their title in the material deposited and as to their right to deposit such material.

The University of Gloucestershire makes no representation or warranties of commercial utility, title, or fitness for a particular purpose or any other warranty, express or implied in respect of any material deposited.

The University of Gloucestershire makes no representation that the use of the materials will not infringe any patent, copyright, trademark or other property or proprietary rights.

The University of Gloucestershire accepts no liability for any infringement of intellectual property rights in any material deposited but will remove such material from public view pending investigation in the event of an allegation of any such infringement.

PLEASE SCROLL DOWN FOR TEXT.

scientific reports

Check for updates

OPEN

Exploring an AI-driven agentic framework for insider threat detection in secure software development organizations: A survey-based approach

Muhammad Shafiq[1]✉, Mujtaba Awan[2], Hathal Salamah Alwageed[3], Umar[4], Mohammad Mehedi Hassan[5]✉, Rafiq Ahmad Khan[6]✉ & M. Anwar Hossain[7]✉

Insider threats continue to be a vexed and complicated risk in secure software development (SSD) organizations due to their origin in combination of malicious intent, negligence, credential abuse, and process vulnerability throughout the software development lifecycle. Artificial intelligence (AI)-inspired, agentic security strategies are increasingly suggested to enhance prevention and detection, but organizations have not yet a clear, decision-supportive approach to prioritize agentic mitigation practices in times of uncertainty. This paper aims to suggest and rank an agentic model of insider threats mitigation to secure software development companies based on the Fuzzy Analytic Hierarchy Process (Fuzzy AHP). Its main objectives include (i) organizing insider-threat mitigation into multi-level decision model, (ii) calculating priority weights of AI-directed agentic practices, and (iii) assisting actionable adoption choices of various insider-threat types. A systematic mapping study (SMS) was first carried out to investigate the state-of-the-art of insider threats in software development organization. The second phase involved performing an empirical survey among cybersecurity professionals to confirm the findings of SMS and determine the types of insider threats and AI-driven agentic practices in preventing and mitigating insider threats. Comparison of the results of empirical survey and MLR was then done using ANOVA test. To find weights and ranking of insider threats and AI-Driven agentic practices, the last stage was a fuzzy analytical hierarchy process (FAHP). Standard FAHP validation steps were verified to give consistency and a sensitivity analysis was performed to check the consistency of the rankings with small perturbations of criterion weights. A hierarchical decision model was created, the overall aim of which was insider threats reduction in secure software development organizations, which were broken down into nine types of insider threats (e.g., malicious insiders, negligent insiders, credential theft, inadvertent misuse of access, privilege abuse, social engineering, software piracy/code theft, insider data exfiltration, and abuse of development tools). 91 AI-based agentic practices were discovered to prevent and detect these insider threats. The most highlighted insider threat to software development organizations was considered to be using FAHP credential theft with final weight 0.1262. In the same way, the agentic practice that was mentioned as the most cited with final weight 0.020580 and global rank-1 was the machine learning driven anomaly detection. The suggested agentic framework based on AI, which is operationalized using Fuzzy AHP, provides a structured and understandable procedure to rank the practices of insider-threat mitigation in secure software development organizations. The framework assists in supporting more justifiable security investment choices by considering uncertainty when expert judgments are involved

and assisting in the realignment of agentic controls to the threat-specific needs within the entire development environment.

Keywords Insider Threats, Secure Software Development, Agentic AI Practices, Systematic Mapping Study, Empirical Survey, Fuzzy AHP

[1]School of Computer Science, Shandong Xiehe University, Jinan, China. [2]School of Computing and Engineering, University of Gloucestershire, Cheltenham, UK. [3]College of Computer and Information Sciences, Jouf University, Sakaka 42421, Saudi Arabia. [4]School of Computing and Engineering, Teesside University, Middlesbrough, UK. [5]Information Systems Department, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. [6]Software Engineering Research Group, Department of Computer Science and IT, University of Malakand, Malakand, Pakistan. [7]School of Computing, Queen's University, Kingston, ON, Canada. ✉email: shafiq@mail.qjnu.edu.cn; mmhassan@ksu.edu.sa; rafiqahmadk@gmail.com; ahossain@queensu.ca

The risk of secure software development (SSD) organizations are working in a more and more environment where the threat environment is influenced by both external and internal attackers[1]. Insider threats: malicious activity by individuals with authorized access, like employees, contractors, or partners, are one of the hardest security issues to address due to insiders, who may have legitimate access, contextual familiarity with systems, and may be aware of development processes[2]. The pace of delivery, common usage of privileges (e.g., access to repositories, CI/CD permissions, and secrets management), distributed teams, and dependency on third-party tools and dependencies in software development environments[3] enhances these risks. Consequently, an insider may deliberately introduce malicious code, steal sensitive source code or intellectual property, steal credentials, abuse development tools or be unintentionally harmful through negligence (e.g. misconfigurations, insecure code and mishandling of keys)[4]. The data loss is not the only effect, as the supply-chain is compromised, regulatory laws are not met, reputations are destroyed, and the operation is disrupted.

In spite of the considerable progress in SSD and DevSecOps, the insider threats cannot be eliminated, as technical controls cannot be taken solely [5{Akbar, 2022 #1405]}. Conventional prevention methods, such as least privilege, access control, logging, secure code standards, code reviews, separation of duties, background checks, and policy enforcement, can be ineffective in adapting to new insider behavior, in finding subtle anomalous patterns early, or in relying on organizational context (e.g., role changes, workload stress, or changes in access patterns)[6]. Moreover, insider threats are seldom a one-time occurrence; they are manifested in a series of activities spread across both time and systems. This prompts the necessity of ways of mitigation that are dynamic, context-sensitive, and able to learn based on signals throughout the software development lifecycle[7].

In this paper, the term Agentic AI[8,9] describes a category of artificial intelligence that is able to operate independently and make autonomous decisions, especially in dynamic and complicated settings, such as secure software development organizations. In contrast to traditional automated models[10], which are usually pre-programmed to act according to a pre-set sequence of fixed rules or algorithms within given parameters, Agentic AI is capable of making independent corrective decisions, based on real-time input and changing circumstances. This implies that Agentic AI systems do not just respond to situations as they occur; they are capable of being proactive in their assessment of situations, detecting anomalies or threats and changing their behavior in order to reduce risks or respond to an issue as it develops.

Conversely, conventional automated processes in both cybersecurity and software development are usually made to execute functions according to a fixed set of rules or commands, and they cannot adapt to novel, unanticipated conditions. They are not allowed to make any decision other than that. In our proposed AI agentic architecture to prevent and detect insider threats, Agentic AI, in particular, will be used to automatically assess and respond to the risks of insider threats by both identifying suspicious activities but also to automatically act in real time based on the level of risk detected. It is the dynamic nature and decision-making capability of Agentic AI that is different from automated systems.

Insider Threats in SSD

Software development companies have a range of intents and capabilities of insider threats. Malicious insiders abuse access to profit, sabotage or spy on others including planting backdoors, build pipeline tampering or intellectual property theft[11]. The lack of security awareness, misconfiguration, secret management, and bypassing security controls by negligent insiders accidentally create risk[12]. Another common and similar case is stealing credentials and impersonating insiders, which is difficult to identify as it looks like normal activity[13]. Other threat patterns include the accidental misuse of access, privilege misuse, team social engineering, software piracy or code theft, insider-data theft, and tool misuse[13]. These threats intersect with the code repositories, issue management, cloud computing, CI/CD, containers and third-party tooling in the modern software development process, all of which are potential insider-threat channels and targets[14].

Insider threats are also complex on the organizational level, as development cultures prioritize rapid iteration, freedom and collaboration, which is in conflict with rigid security enforcement[15,16]. Excessive control may decrease productivity and shift to shadow IT, whereas lax controls may put vital assets at risk. Thus, the mitigation of insider-threats in SSD should be such that it does not slow down delivery but rather enhances resilience at minimal friction[15].

Why AI-Driven Agentic Practices are emerging

AI cybersecurity has found application more in the areas of anomaly detection, behavioral analytics, threat intelligence, and automated response[17]. In software development companies, AI is especially appealing given

the amount and diversity of data created by development tools (e.g., commit histories, pull request activity, build logs, access records and runtime telemetry)[18]. Nevertheless, numerous AI implementations continue to be isolated, reactive, and require manual interpretation. This is where agentic practices supported by AI are offering an encouraging change[19].

The agentic approaches are not limited to any fixed analytics since they allow semi-autonomous or autonomous software agents to detect, reason, make decisions and act across systems that are interconnected. An agentic security layer can observe behaviors, cross-repository and pipeline signals, calculate risk on-the-fly, make adaptive access decisions, and trigger responses (e.g., session termination, secrets rotation, privileged access revalidation, or incident ticket creation) in an insider-threat scenario[20]. Organizations can facilitate on-going mitigation by enabling continuous mitigation strategies through the coordination of various specialized agents, including user behavior analytics agents, anomaly detection agents, DLP agents, and NLP-based communication monitoring agents (where policy allows) and incident response agents[21].

However, the solutions based on AI and agents present significant issues: transparency and explainability, false positives and alert fatigue, privacy and ethical limits, possible bias in behavioral models, adversarial manipulation, and governance necessity[22]. In addition, development agencies have limited resources and have to choose which practices are going to provide maximum worth to certain types of insider-threats. This encourages a systematic process of decision making that has the potential to give priority to mitigation practices in the face of uncertainty.

Need for evidence synthesis, perception analysis, and decision-ready frameworks

Although the topic of insider threat prevention and AI-based detection has been widely studied, existing evidence is scattered across security engineering, software engineering, human factor, and organizational research. Specifically, SSD environments provoke unique considerations (e.g., productivity of developers, integrity of pipelines, and culture of code review, secrets management, and toolchain trust). This leads to the necessity to (i) integrate the state-of-the-art specifically in insider threat prevention in secure software development, (ii) make visible how AI-driven agentic practices align with prevention and mitigation systems throughout the development lifecycle, and (iii) justify these practices by the perceptions of practitioners and empirical evidence.

To overcome this, the current research incorporates three complementary aspects:

1. State-of-the-art synthesis (SMS) to describe existing prevention and mitigation strategies, potential gaps, and novel practices related to secure software development (RQ1, RQ2).
2. Empirical survey that will involve the perceptions of AI and cybersecurity experts on the effectiveness and feasibility of AI-based mitigation practices in software development organizations (RQ3).
3. Analysis of comparative association, the investigation of how patterns of evidence, based on the SMS, correlate with the professional perceptions, such as the linear-by-linear correlation between the categorized SMS results and the results of the surveys (RQ4). This connection is significant since research focus might not be relevant to practitioner concerns (or limitations), and quantifiable connection assists to determine whether the literature and industry perspectives are aligning.

Finally, a strong prioritization mechanism is needed in order to convert evidence into actionable advice. The multi-criteria decision-making (MCDM) is especially appropriate due to the multi-dimensional nature of the insider-threat mitigation decisions, which encompass the effectiveness, feasibility, cost, integration complexity, privacy risk, explainability, and operational impact. Of the MCDM techniques, the Fuzzy Analytic Hierarchy Process (Fuzzy AHP) can be used to deal with uncertainty and language-based judgments that are common in expert-based security assessments. It facilitates the comparison of practices pair-wise with the aid of fuzzy numbers and generates interpretable rankings and weight that can directly aid in investment and implementation planning.

Research questions

Under the above motivation, this research paper will discuss the following research questions:

RQ1. What is the state-of-the-art of insider threats prevention of secure software development (SSD)?

- RQ1.1: Which are the most frequent types of Insider threats in software development organizations and how they impact the security and functionality of software projects?
- RQ1.2: How are the secure software development organizations currently detecting and preventing the Insider Threats using existing methods and technologies?
- RQ1.3: What are the shortcomings of the existing Insider Threat prevention measures in SSD?
- RQ1.4: Which are the key metrics to measure Insider Threat prevention in SSD?
- RQ1.5: What gaps exist in current research on Insider Threat prevention in SSD lack?

RQ2. How do AI-driven agentic practices prevent and mitigate insider threats in SSD?

RQ3. How do AI and cybersecurity professionals perceive the usefulness of AI-powered practices in enhancing insider-threat management in software development?

RQ4. How does SMS correlate with findings of empirical surveys linear-by-linear?

RQ5. How can we design an effective AI-powered agentic model to prevent insider threats in SSD?

These questions collectively link evidence synthesis, practitioner validation, and statistical alignment analysis and framework design in a research pipeline.

Contributions of this study

1. Central Contribution:

The key contribution of the paper is the suggestion of an AI-driven agentic framework of insider threats prevention and detection, especially within the framework of secure software development organizations. The Systematic Mapping Study (SMS) and empirical survey are complementary endeavors to determine and confirm major types of insider threats that such a framework must tackle. The statistical analysis (ANOVA) and the prioritization model using FAHP are applied to make sure that both the types of threats and the prioritization model are both empirically grounded and consensus-driven by both the literature and the practice. Lastly, the framework design is introduced as a solution that incorporates AI-based methods to automatically deal with insider threats.

Therefore, although the study might encompass a combination of several methodologies, they all play a direct role in developing a holistic and viable solution towards dealing with insider threat. Thus, we may explain that the key contribution is the design and validation of AI-based agentic framework, which can combine the findings of various research methodologies to make sure that the framework is coherent on theoretical bases and applicable in practice.

2. Fragmentation:

To prevent the impression of fragmentation, we might be more effective in explaining how all the research elements are interrelated to serve the overall purpose:

- Systematic Mapping Study (SMS): This element involves the synthesis of the existing literature on insider threats, which gives a knowledge base and determines important types of threats and gaps in current literature.
- Empirical Survey: Survey validates and empirically evaluates these types of threats; this way, the framework we develop is based on the real world perceptions and requirements.
- Statistical Analysis (ANOVA/Association): It is another analysis that will confirm consistency of the literature and empirical data and make sure that prioritization of threat types is supported by empirical evidence.
- FAHP-based Prioritization: This decision-modeling method can be used to quantify and prioritize the types of threats and the possible mitigation strategies according to the expert opinion, so that the framework concentrates on the most important threats.
- Framework Design: Finally, the framework design unites all the previous elements and provides a goal-oriented remedy in helping to prevent and identify insider threats in software development organizations.

Clear focus and structure

In addressing this issue of the fragmented nature of the paper, we would explain that this research is an integrated, multi-method research, which seeks to design and validate a framework, and not any one methodology of research. Specifically:

- The statistical analysis and the conducted empirical survey facilitate the identification and ranking of the insider threats that are relevant.
- The prioritization used as FAHP adds to a model of decision making that informs the design of the framework.
- The last output is the framework that would combine all these findings into a cohesive framework to tackle insider threats in an overall way.

Paper organization

The rest of the paper is presented in the following way. Section 2 gives background and related work on insider threats in SSD and AI-based mitigation. Section 3 explains the research methodology, which includes SMS protocol, survey design, measurement constructs and association testing approach. Section 4 presents the synthesized SMS results and the empirical survey results with the statistical analysis of association. In Sect. 5, the design of the agentic framework of the AI-driven approach and the prioritization process based on Fuzzy AHP are outlined. Section 6, discusses the implications for SSD organizations (such as governance, ethics and implementation). Section 7 offers the validation and limitations of the study. Section 8 provides conclusion and future work.

Background and related work

Insider threats as a software-development security problem

The insider threats are uniquely exposed to SSD organizations as their workforce (developers, DevOps engineers, build/release managers, and administrators) frequently needs extensive access to source code, CI/CD infrastructure, secrets, and artifact repositories, and production-like environments[23]. This trusted access opens up high-impact abuse vectors: (i) malicious code (e.g., logic bombs, backdoor) introduction, (ii) sabotage of builds and deployments, (iii) intellectual property or sensitive data exfiltration, and (iv) abuse of developer tooling to bypass controls. The Software Engineering Institute (SEI/CERT) notes that vulnerabilities related to insider threats can be added (and addressed) at any stage of the SDLC and publishes developer-centric instances of malicious insider activities[24].

Contemporary secure SDLC advice addresses the need to incorporate security practices into development processes as opposed to attaching them to the software's back end[25]. To minimize software risk, the NIST secured Software Development Framework (SSDF) offers a list of recommended practices that can be incorporated into any SDLC, although development, build, test and production environments have to be listed and secured to ensure lateral movement and abuse[26,27]. Simultaneously, security-control catalogs like NIST SP 800 – 53 have

control families applicable to reducing insider risk (e.g. least privilege, monitoring/audit, account management), and are translated by organizations into operational and engineering controls[27].

Why conventional controls struggle

Insider threats are socio-technical: they are the result of human will and error, organizational factors, and technical vulnerability[14]. CERT, in its Common Sense Guide to Mitigating Insider Threats[28]" and CISA, in their Insider Threat Mitigation Guide characterize insider risk to be mitigated through cross-functional coordination (security, HR, legal, IT, engineering) and a mix of policy/process controls with monitoring and response capabilities[29,30]. Nevertheless, classic methods have been known to have familiar limitations[21,23,31–36]:

- **Ambiguity and sparsity**: The insider activities of malicious insiders can be part of normal work (e.g., accessing the repository, editing code). Supervised detection is hard due to the lack of insider labeled data.
- **High false positives and alert fatigue**: Behavior-based systems are also likely to be noisy (e.g., due to lack of context or changes in work patterns).
- **Privacy and governance tensions**: It should be appropriately governed and proportional to monitor developer communications and activity.
- **Tool sprawl in DevSecOps**: CI/CD tools, cloud services, secrets managers, and ticketing systems generate fragmented signals, which are difficult to correlate without automation.

Such limitations drive evidence-based and automated strategies, in particular, those that can combine evidence across various systems and change behavior.

Related literature

RQ1 — State-of-the-art of insider-threat prevention in secure software development

The state-of-the-art can be structured into (a) programmatic controls, (b) secure SDLC/DevSecOps controls, and (c) tailored to development environments monitoring and response controls.

(a) **Programmatic controls and governance**: Authoritative guidance continuously focuses on insider-risk programs that characterize insider threats, set reporting and investigation processes, and deal with workforce risk, as well as, develop security culture. CISA offers a practical insider threat reduction framework that includes the definition of the threat, detection/identification, and response/handling[30]. The Common Sense Guide by CERT also emphasizes training, workforce awareness and stakeholder coordinated roles[28].

(b) **Secure SDLC/DevSecOps controls**: NIST SSDF (SP 800 – 218)[37] is a common secure-development standard, and it is explicitly created to be incorporated into diverse SDLC models. Although not strictly an insider threat construct, a great number of SSDF practices have a direct impact of reducing insider opportunity and blast radius within development environments, such as limited access to code and build systems, artifact protection, and preserving development pipeline integrity[38]. Practitioner-oriented DevSecops instructions also highlight the least privilege in CI/CD, secure pipelines, and continuous monitoring[39].

(c) **Monitoring, auditing, and least privilege**: Least privilege and privileged-access controls are a cornerstone controls (particularly in build/release and admin roles). NIST SP 800 – 53 contains the least privilege control (AC family) and the associated monitoring/audit controls that can be translated into engineering controls in software development organizations[37]. CERT advice and SEI/CERT SDLC-based discourses directly refer to development-related controls like code reviews and configuration management as controls against insider abuse[28].

(d) **Key gap emerging from RQ1**: There is abundant evidence in the literature on what to do (programs, least privilege, monitoring) yet less evidence on decision support on how to prioritize security practices in the face of uncertainty and when considering different types of insider-threat in development toolchains, where multi-criteria decision methods (e.g., Fuzzy AHP) can be useful.

RQ2 — How AI-driven agentic practices prevent and mitigate insider threats in secure software development

The most prevalent AI-based insider threat mitigation practices are behavior analytics, anomaly detection, entity risk scoring, NLP-aided monitoring, and automated incident response, more and more frequently coordinated through multi-tool workflows[18].

(a) **User and Entity Behavior Analytics (UEBA) and behavioral modeling**: UEBA is a model of normal behavior in users and entities and deviations, which can be used to identify insider abuse, compromised credentials, or inappropriate use of privileges[40]. Reviews and research claim that UEBA is moving away in rule/pattern-based approaches to ML-based approaches (supervised, unsupervised, and hybrid) with an increased focus on minimizing false positives and enhancing time-to-detection[41]. A 2025 systematic review[42] of user behavioral-based insider threat detection (UBITD) observes that a lot of the research focuses on ML/DL models, though coverage is still incomplete and disjointed- indicating that there are still gaps in research regarding practical, end-to-end insider-risk detection and response[42]. SEI has also written about risks of insider risk evaluation using AI, with more practical limitations such as data availability and labeling.

Implications for secure software development UEBA indicators in the developer environment include activity in the development repository (e.g., clone/fetch, unusual branch access), CI/CD activity, secrets activity, artifact download, and administrative activity, which are often spread across tools and require analysis.

(b) **AI-enabled anomaly detection and risk profiling**: Anomaly detection helps in prevention and detection by uncovering anomalies early, e.g., unusual access behaviour, unusual change in code or configuration,

unusual data flow. The literature that mentions UEBA/anomaly-detection technologies tends to refer to the need for a trade-off between sensitivity and false positives, as having stable and explainable alerts is needed for making the system useful[43].

(c) NLP and communication monitoring (with governance constraints): NLP-based communication monitoring can detect risky communication (e.g., policy violations, social engineering, insider data breach). While it is conceptually appealing, NLP monitoring has privacy, consent and governance implications and needs to be done with caution (e.g., by restricting the monitoring to approved channels, risky roles or red-flag indicators that are automatically referred to humans)[44].

(d) Automated incident response and orchestration: AI-powered automated incident response triages alerts, suggests actions and, potentially, runs playbooks (e.g., token invalidation, user lock-out, workflow quarantine). A recent study on AI-based automated incident response, notes the technology may help to speed the containment phase of the incident and increase efficiency, but also mentions the ethical, legal and governance issues[45].

(e) Agentic AI as a step beyond "point AI": Agentic AI is an extension of "point AI": Agentic AI means goal-oriented systems that can plan, use tools and reason about evidence and then act within guardrails (rather than a single model to predict). A recent study talks about the benefits agentic AI could bring to cybersecurity, but also new threats and governance issues[46].

Why agentic matters for insider threats Insider threats frequently require correlation of multiple signals (identity + repo actions + CI/CD changes + data movement). Agentic workflows can (i) ingest feeds from different systems, (ii) score hypotheses (malicious vs. negligent vs. compromised account), (iii) recommend or execute mitigation actions, and (iv) collect evidence for audit and investigation.

RQ3 — Perceptions of AI and cybersecurity professionals on effectiveness of AI-driven practices
Perception matters as decisions on use are not based only on the technical accuracy of AI but also trust, explainability, fit to operations and ethics. Industry surveys and reports by practitioners suggest a new belief that AI is more effective and efficient, while there is a risk and regulation concern and regulation concern that must be addressed.

- ISC2 AI, pulse survey (2025) mentions that there is adoption of AI security tools by cybersecurity practitioners and queries about assumptions on impact on workforce and operations, that suggest that AI is a proven more practical tool for the security operations (ISC2).
- SANS 2024, AI survey also mentions the use and reality of AI systems in cybersecurity practice and also value and friction with operations, which indicates that the practitioners believe in value and friction with operations (SANS Institute).
- Vendor and Industry surveys (e.g. Crowdstrike "State of AI materials) indicate preference for cybersecurity GenAI and platform integration, which is in line with cybersecurity practitioner concerns about the integration with AI tools (CrowdStrike).

The professional sentiment in agentic situations is two-sided: eagerness to be productive and automate processes and apprehension about being seen, having access to data, and engaging in unintended behaviors. The trend in the real market towards autonomous or semi-autonomous workflows is demonstrated in the news about AI agents being incorporated into security tooling (e.g., Microsoft security agents) (Axios).

Research gap for RQ3 There is abundant literature on tools and concepts but less literature specifically quantifies the perceived effectiveness of particular AI-driven insider-threat practices (e.g., UBA, anomaly detection, DLP, automated incident response) in software development organizations and CI/CD systems. This inspires the empirical survey.

RQ4 — Linear-by-linear association between SMS and empirical survey findings
RQ4 has a methodological significance, as it will test the convergence between what the literature focuses on (SMS) and what professionals focus on (survey). Systematic mapping studies are aimed to categorize and give a summary of any research area (not synthesize the effect sizes) and therefore are a valuable contribution to the comparisons between research attention and perceived efficacy by practitioners[47].

To measure alignment quantitatively, ordinal-by-ordinal association of linear-by-linear type (often called a MantelHaenszel/trend-type) is measured using the linear-by-linear statistic of the SPSS Crosstabs (the row/column categories must have meaningful order). The SPSS community of IBM reports the linear-by-linear test to be a chi-squared test (MantelHaenszel), and is reported in SPSS algorithms under Crosstabs (IBM TechXchange Community). (This is often construed to show a monotonic trend/association between ordinal variables; applied discourses abound (Cross Validated)).

How this maps to our study design:

- We could extract an ordinal measure of intensity (e.g. mention of practice: low/medium/high; or ranked prevalence across papers) based on the SMS).
- We might be able to infer an ordinal measure of effectiveness (e.g. Likert rating or ranked importance) out of the survey.
- Linear-by-linear association followed by testing the hypotheses that increased literature emphasis relates to increased perceived effectiveness (or the reverse) supports the triangulation logic- i.e. more robust conclusions when multiple sources of evidence point in the same direction. Wider triangulation principles are based

on the idea of integrating streams of evidence to reinforce inference in the case of disparate biases of the sources (SAGE Methods).

RQ5 — Designing a robust AI-driven agentic framework to prevent insider threats in software development
The related literature suggests that a robust framework must combine secure SDLC foundations, insider-risk governance, and agentic AI orchestration with strong risk management.

(a) **Anchoring the framework in SDLC and control baselines**: NIST SSDF offers a useful foundation of secure development practices, which can be translated into organizational processes and tooling. (NIST Publications) NIST SP 800 – 53 is a control catalog of access control, monitoring/audit, and accountability practices that can be used to mitigate insider opportunity and enhance detectability. (NIST Publications) CISA and CERT guides offer scaffolding and implementation practices of insider programs across prevention, detection, and response (CISA).
(b) **Agentic AI governance and AI risk management**: Since agentic systems may act between tools and perform actions, good design should incorporate governance guardrails, human controls, and responsibility. Artificial intelligence (AI) Risk Management Framework (AI RMF 1.0) is a framework by NIST that focuses on the trustworthiness of AI in the context of lifecycle governance, mapping, measurement and management- useful to organize the deployment of agentic AI in the context of security operations. (NIST) explains the agentic AI security risks and the necessity of control frameworks surrounding autonomous cybersecurity workflows[46].
(c) **Why Fuzzy AHP is an appropriate prioritization mechanism**: Insider-threat mitigation has a variety of criteria (effectiveness, feasibility, cost, privacy impact, integration complexity, response speed) and uncertainty in expert judgments. When making decisions about cybersecurity in the face of uncertainty, the fuzzy AHP is often adopted to rank controls or risks, thus it is appropriate in the prioritization of AI-driven practices and constructing a defensible path to adoption[48].
(d) **Resulting design logic (synthesis)**: A resilient AI-based agentic model of insider threats in secure software development ought to:

- Use SDLC-built-in controls (SSDF-aligned) and least privilege/monitoring baselines (SP 800-53-aligned)[49].
- Implement AI-based detection and response (UEBA/anomaly detection, risk scoring, automated response)[43].
- Orchestrate these capabilities through agentic workflows with governance guardrails, and deal with AI risks through AI RMF principles[46].
- Prefer FAHP practices as a way to indicate uncertainty and trade-offs to stakeholders[48].
- Test congruence between literature and practitioner reality by comparing SMS vs. survey results (including ordinal association testing)[47].

Summary of gaps addressed by this paper
Three common gaps in the literature prompt the research questions and overall contribution:

(a) **Context gap**: Little of the literature on insider threat mitigation focuses specifically on insider risk to software development ecosystems (repos, CI/CD, secrets, build integrity) with practice prioritization[2].
(b) **Decision gap**: AI-based practices are many and diverse, yet companies have no clear process of prioritizing and implementing them under uncertainty- helping to use Fuzzy AHP[48].
(c) **Evidence-alignment gap**: The priorities of the literature are not necessarily those of the practitioners; convergence (e.g., linear-by-linear association) can be formally tested to enhance the validity of a proposed framework and emphasize research-to-practice discrepancies[47].

Table 1 is a comparative table that outlines different alternative frameworks and techniques used in insider threat prevention and how the stands out as novel in terms of key points.

Research methodology

We elect to use the sequential mixed-methods approach (Fig. 1) in this research, which includes:

- A systematic mapping study (SMS) to establish the state-of-the-art and identify AI-based practices to prevent insider threats (responds to RQ1 and part of RQ2).
- Survey of AI and cybersecurity practitioners combined with perceived efficacy and adoption (answers RQ3).
- Linear-by-linear correlation analysis to assess the level of agreement between the literature (SMS) and the opinions of professionals (survey) (responds to RQ4).
- As part of RQ5, a fuzzy analytic hierarchy process (FAHP) to transform expert views into weighted priorities to a resilient AI-based agentic architecture.

The research design is triangulated: the literature (SMS) speaks to what the professionals think works (survey) and both are used to feed a formal decision model (FAHP) that produces a defendable, prioritized list.

Systematic mapping study (SMS)

The SMS[50–52] gives an organized description of the prevention and mitigation of insider threats in secure software development environments and identifies the agentic practices of AI mentioned in the literature.

Framework/Technique	Key Focus	Methodology	Novel Aspects of Our Paper
Classical AHP (Analytic Hierarchy Process)	Prioritization of elements based on pairwise comparisons.	Traditional pairwise comparison, hierarchical structure.	- Incorporates fuzzy logic to handle uncertainty and subjectivity in expert judgments.
TOPSIS (Technique for Order Preference by Similarity to Ideal Solution)	Ranking and selection based on closeness to an ideal solution.	Calculation of distances from ideal and negative-ideal solutions.	- Fuzzy AHP enhances the accuracy of subjective evaluations in real-world, uncertain environments.
VIKOR (VIseKriterijumska Optimizacija I Kompromisno Resenje)	Multi-criteria decision-making, focusing on compromise solutions based on conflict resolution.	Multi-criteria optimization method based on a compromise approach.	- Our framework integrates AI-driven methodologies, focusing specifically on insider threat detection within software development environments, something VIKOR doesn't directly address.
DEMATEL (Decision-Making Trial and Evaluation Laboratory)	Identifying the relationships and influences between criteria.	Matrix-based analysis to model cause-effect relationships between elements.	- Our paper focuses on AI-driven agentic practices, providing practical solutions to insider threats, whereas DEMATEL is more theoretical in understanding cause-effect relationships.
Behavioral Analytics and UEBA (User and Entity Behavior Analytics)	Detects anomalous behavior patterns to identify insider threats.	Uses machine learning and statistical analysis to detect deviations from normal behavior.	- Our system integrates behavioral analytics and AI-driven agentic decision-making based on Fuzzy AHP to bring together technical and managerial decision-making for a comprehensive approach to prevention.
Data Loss Prevention (DLP) Systems	Preventing the unauthorized transfer or leakage of sensitive data.	Use of content inspection and contextual analysis to prevent data leakage.	- Fuzzy AHP method brings prioritization and subjective judgment to DLP systems, enabling better alignment with organizational needs and insider threat risks.
Anomaly Detection Systems (e.g., Real-time monitoring, Machine Learning models)	Identifying deviations from normal patterns, specifically for detecting threats in real-time.	Uses machine learning algorithms (e.g., clustering, classification) to detect unusual activities.	- Our Fuzzy AHP framework incorporates expert inputs alongside anomaly detection systems, making the decision-making process more robust and adaptable to different organizational contexts.
Incident Response Systems	Automatically responds to detected security incidents, such as insider threats.	Automated response protocols to mitigate identified threats.	- Our framework offers a multi-stage decision-making model (prevention, detection and response) in contrast to traditional incident response systems, making it more relevant to software development.
Machine Learning-Based Threat Detection	Predicts and detects insider threats based on historical data and pattern recognition.	Uses data analytics and machine learning models for threat prediction.	- The integration of Fuzzy AHP enables prioritization and decision support, making ML-based models more actionable for specific organizational priorities.
Behavioral Biometrics	Analyzes unique behavioral patterns (e.g., keystrokes, mouse movements) to detect insider threats.	Collects data on user behavior and compares to established patterns.	- Our paper builds upon behavioral biometrics by introducing a novel decision-making layer, prioritizing AI practices based on expert insights and organizational needs.

Table 1. Comparison of different Alternative Frameworks and Techniques used in Insider Threat Prevention.

SMS objectives and research questions

a) Aim: To determine the state-of-the-art of insider-threat prevention within secure software development organizations and derive AI-based mitigation practices that may be prioritized later using FAHP and incorporated into an AI-based agentic framework.

b) Mapping Question (MQs).

- MQ1 (RQ1 mapping): What are the reported prevention/detection/mitigation strategies in the context of secure software development (SDLC/DevSecOps/CI-CD) in relation to insider threats?
- MQ2 (RQ2 partial mapping): What are AI-based practices (claimed) to prevent/detect/mitigate insider threats, and what are their place in the SDLC/DevSecOps toolchain?
- MQ3 (context mapping): What data, assessment procedures, and measures are employed to verify AI-based insider threat strategies?
- MQ4 (gap mapping): What are the gap areas in software development environments where agentic orchestration, governance, explainability and real world deployability are lacking?

SMS protocol

a) Time Window.

- Primary window: 2015 -Jan 2026 (captures DevSecOps maturity + modern ML/DL + early agentic AI cybersecurity work).
- Seminal anchor sources: Previous works of authority are added where they are directly related to SDLC insider risk.

b) Data Sources: For an exhaustive SMS, search:

- IEEE Xplore, ACM DL, ScienceDirect, SpringerLink, Wiley, arXiv.
- Scopus/Web of Science.
- Standards/authoritative orgs: NIST, CISA, SEI/CERT, national cybersecurity agencies.

In this SMS, we used open-web accessible sources that include:

- NIST SSDF publications.
- CISA Insider Threat Mitigation Guide.
- SDLC material and SEI/CERT insider threat guidance.

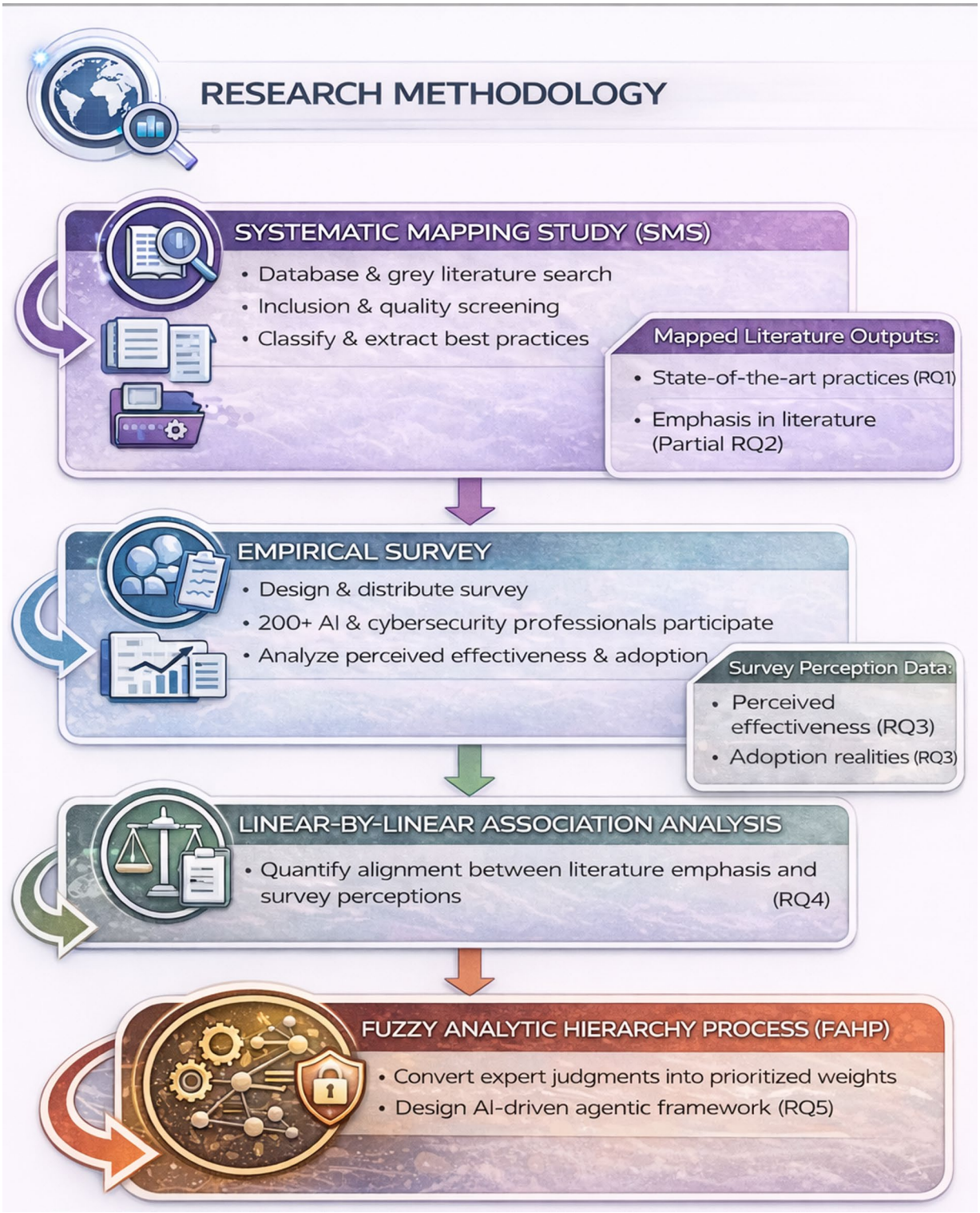

Fig. 1. Research Methodology.

- Recent peer-reviewed SLRs and research articles of ScienceDirect/Springer/MDPI/Nature/arXiv.

c) Search Strings: We use Boolean blocks. Example master string:

- **(Insider)**: ("insider threat" OR "insider risk" OR "malicious insider" OR "negligent insider" OR "credential theft")

- **(Software development context)**: ("secure software development" OR SDLC OR DevSecOps OR "CI/CD" OR "software supply chain" OR "code repository" OR "build pipeline")
- **(AI/agentic practices)**: ("machine learning" OR "deep learning" OR UEBA OR "behavior analytics" OR "anomaly detection" OR "risk scoring" OR "NLP" OR "automated incident response" OR "threat hunting" OR "data loss prevention" OR "agentic AI" OR "AI agent")
- **Combined**: (Insider) AND (Software development context) AND (AI/agentic practices).

d) Inclusion Criteria: Includes studies that:

- Counter insider attacks (malicious/negligent/credential compromise) and.
- Relevant to software development organizations (SDLC/DevSecOps/CI/CD) or generate AI techniques that are easily portable to that context (UEBA, anomaly detection, NLP monitoring, etc.) and.
- Present extractable practices, models, datasets or evaluation evidence.
- Incorporate expert norms/guides to inform secure SDLC and insider risk programmes (NIST/CISA/SEI).

e) Exclusion Criteria: Excludes studies that:

- Only deal with external attackers who are not insiders.
- Not applicable to enterprise/engineering setting.
- Not detailed enough to elicit practices (unless an important conceptual framework).

f) Screening Process.

- Stage 1: Title/abstract relevance screen.
- Stage 2: Full-text eligibility screening.
- Snowballing: Backward/forward citation checks of the most central SLRs and datasets (CERT dataset usage is a good indicator of a hub).

g) Data Extraction Fields: For each included study:

- Bibliographic: year, venue, type (research/SLR/standard/industry guide).
- Type of insider: malicious, negligent, credential compromise, mixed.
- SDLC relevance: requirements/design/code/build/test/release/ops.
- Data sources: logs, repo events, emails, file access, IAM, endpoint, network.
- Artificial intelligence method: ML/DL/NLP/UEBA/anomaly detection/ensembles, etc.
- Mitigation practice(s) extracted.
- Evaluation: dataset, metrics (F1, AUC, accuracy), realism (synthetic vs. real).
- Governance: privacy, explainability, human-in-the-loop, oversight.

Classification Scheme

a) "State-of-the-art" prevention layers (RQ1 map): We map finding into three layers:

- L1. Organizational/program layer: Insider threat program governance: define-detect-assess-manage; team structures; reporting templates; training.
- L2. Secure SDLC/DevSecOps control layer: Secure development practices (SSDF), access control, environment separation, build integrity, auditability.
- L3. Technical detection/response layer: UEBA, anomaly detection, NLP monitoring, DLP, automated response, threat hunting, risk scoring - frequently tested using datasets such as CERT.

b) AI-driven "practice families" (RQ2 partial map): Practices are grouped into:

- Behavioral Analytics/UEBA.
- Real-time Anomaly Detection.
- Predictive Risk Profiling (risk scores):
- Threat Hunting and Intelligence Integration.
- Automated Privilege Governance and Access Control.
- Privilege Escalation and Misuse Detection: This allows a user to obtain more access rights than they are supposed to have.
- NLP Communication/Content Risk.
- Data Loss Prevention (DLP) and Exfiltration Detection.
- SOAR-like orchestration/Automated Incident Response.
- Continuous Learning/Adaptation (online learning, test-time training, model updates).

PRISMA flow counts table

The flow diagram of this SMS using PRISMA is shown in Fig. 2. The diagram follows the PRISMA 2020[53] structure.

Table 2 presents PRISMA flow count table for this SMS.

Table 3 presents full text exclusion reasons.

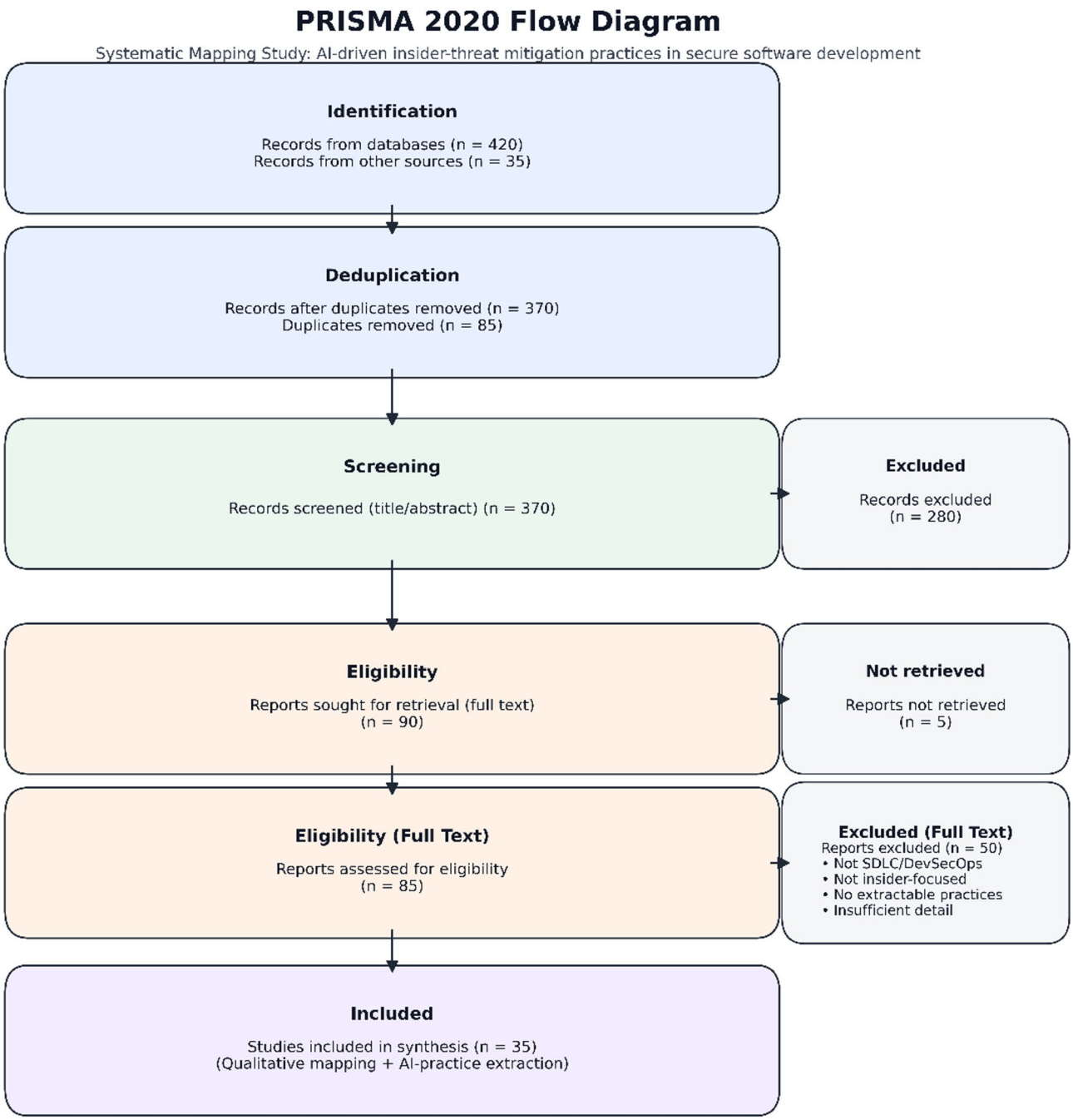

Fig. 2. PRISMA flow diagram.

Empirical survey

To improve the literature-based findings, and to get the practical perspective, an empirical survey was arranged between cybersecurity experts, DevSecOps engineers, and AI-security professionals having hands-on experience in SSD and operation. The questionnaire was created to quantify the perception of the kinds of insider threats and AI-based agentic practices. A survey design has three major steps that can be subdivided into steps[54,55]:

Designing a questionnaire survey

- Purpose: The first part of the survey design process is first to generate a comprehensive questionnaire. The suggested questionnaire will be based on collecting both qualitative and quantitative information among the survey participants regarding their awareness of the insider threats and the use of AI in the SSD.
- Survey Themes: The survey questions will be grounded on the SMS results, i.e., the questions directly refer to the insider threat types and AI-driven agentic practices related to the prevention and mitigation of insider threats to software development organizations. This encompasses issues like:

PRISMA Stage	Item	Count (*n*)	Notes/Examples
Identification	Records identified from databases	420	IEEE/ACM/Scopus/WoS etc.
Identification	Records identified from other sources	35	Snowballing, standards (NIST/CISA/SEI), key reports
Identification	Total records	455	Sum of above
De-duplication	Duplicates removed	85	Same paper across databases
Screening	Records screened (title/abstract)	370	After duplicates removed
Screening	Records excluded	280	Not insider threat/not SDLC/DevSecOps/not AI practice/irrelevant
Eligibility	Reports sought for retrieval (full text)	90	After abstract screening
Eligibility	Reports not retrieved	5	Paywall/unavailable/insufficient access
Eligibility	Reports assessed for eligibility	85	Full-text screened
Eligibility	Reports excluded (with reasons)	50	See reason table below
Included	Studies included in synthesis	35	Final set used for mapping + practice extraction

Table 2. PRISMA flow count table.

Reason for exclusion at full-text stage	Count (*n*)
Not SDLC/DevSecOps/software development context	18
Not insider-threat focused (external threats only)	12
No extractable AI-driven mitigation practices	11
Insufficient methodological detail/conceptual only	6
Duplicate full text/superseded version	3
Total excluded (full text)	**50**

Table 3. Full-text exclusion reasons.

  - Types of insider threats (e.g., malicious insiders, negligent insiders, credential theft, social engineering etc.).
  - AI-based agentic action in threat prevention and detection (e.g., Behavioral Analytics and User Behavior Analytics (UBA), Real-time Anomaly Detection, AI-Powered Threat Hunting, AI-Driven Insider Attribution, Natural Language Processing (NLP) in Email and Chat Monitoring, Data Loss Prevention (DLP) with AI).

- Likert-scale Items: The survey consisted of Likert-scale questions (i.e. Strongly Agree to Strongly Disagree), to evaluate what the surveyed believe about cybersecurity professionals in respect to the following:

  - Perception of types of insider threats: How serious are the insider threats in SSD?
  - Effectiveness of the AI-driven agentic practices: What is the effectiveness of AI-driven practices in preventing these threats?
  - It also contained open-ended questions to collect qualitative information on the issues with AI implementation, security and implementation issues in SSD.
  - This study was conducted in the framework of the University of Malakand Institution Review Board (IRB)/Ethics Committee. This committee has approved all the experiments.

Data collection

- Examples Survey Distribution: A survey was conducted of a sample of cybersecurity and AI experts globally and those familiar or active in SSD. The professionals are recruited on different platforms, which involve academic networks and professional associations, and industrial networks. The questionnaire also could be found online so that it would be easier to reach a large number of individuals and receive several responses.
- Target Sample: The survey targeted the group of more than 250 professionals, who work in various industries, including tech companies, cybersecurity organizations, and AI research centers. The respondents were selected to represent the range of experience, beginning with the practitioners who are directly involved in the implementation of secure software development and going up to the intelligent people who devise solutions to achieve automation of security. The sample sizes that were chosen were based on 180 cybersecurity and AI practitioners. Informed consent was obtained in all the participants. Figure 3 provides the demographic of the respondents in the survey.

- Survey Form: The survey was composed of both types of questions: Likert scales (quantitative) and open-ended (qualitative) questions to allow both to run the statistical analysis and to receive detailed, in-depth answers.

  - Quantitative Data: Likert-scale Items can be used to quantify perceptions of insider threat, the efficacy of AI-informed agentic practices, and the influence which these two elements have on SSD practices.

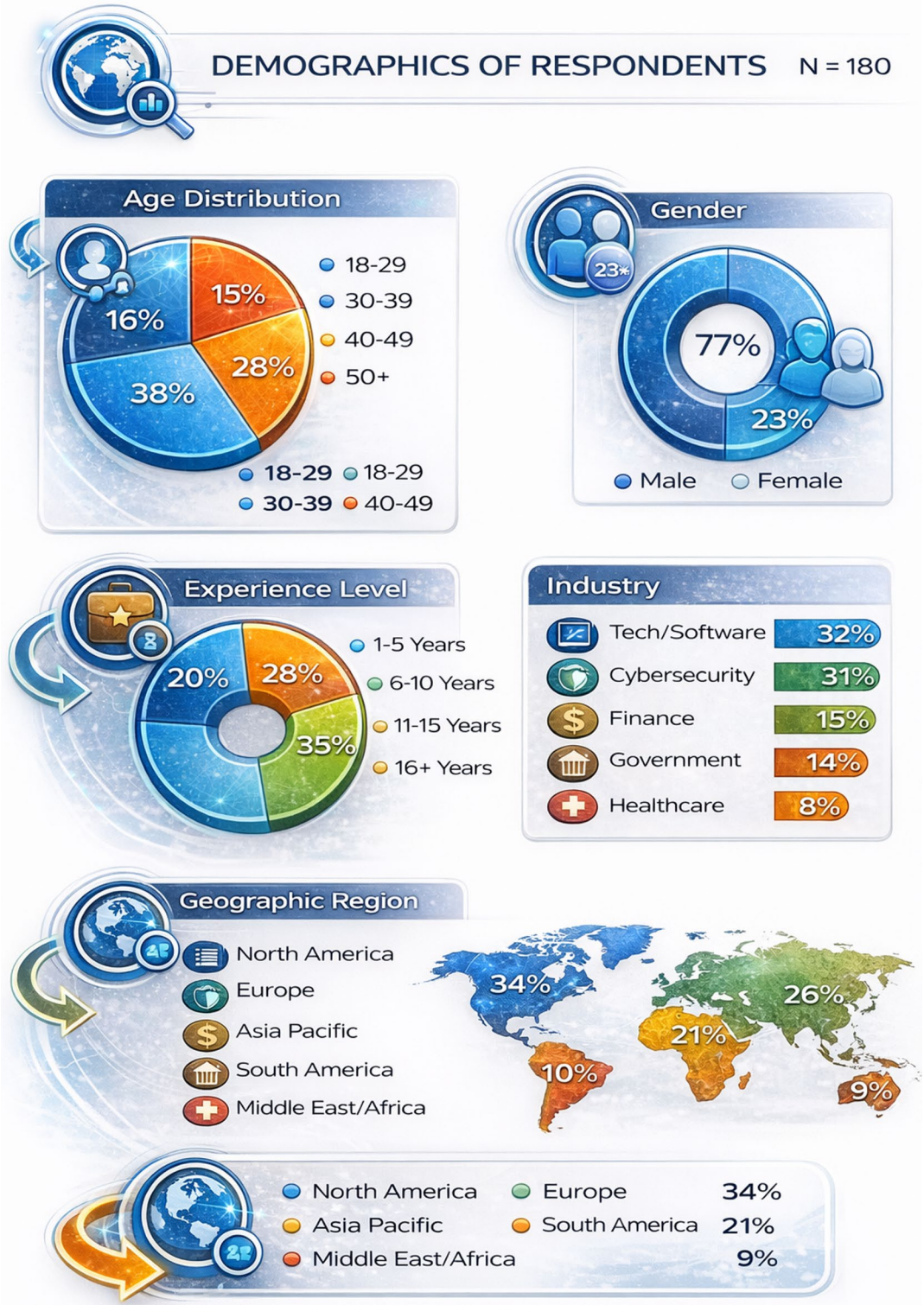

Fig. 3. Demographic details of survey respondents.

– Qualitative Data: Open-ended questions were included to gain a better insight into the perceived challenges and opportunities presented by the professionals and provide details on specific AI tools, their failures and real-life examples of their use.

Data analysis and interpretation

- Quantitative Analysis: Standard mathematical statistical analysis was used to analyze the survey data; they are:

  - Descriptive Statistics: To describe and outline the major trends of the answers to Likert-scale (e.g., mean, standard deviation).
  - Correlation Analysis: To determine the correlation between the perceptions of insider threats and perceived effectiveness of AI-driven agentic practice at different stages of the SSD organizations.
  - Factor Analysis: In order to identify the latent variables that influence cybersecurity threat perception and AI applications in the SSD organization.

- Qualitative Analysis: The open-ended questions were analyzed using thematic analysis, to identify the predominant themes, patterns and concerns raised by the respondents. These themes can be:

  - AI Adoption Barriers: Understanding of our technical, organizational or regulatory challenges to the successful use of AI in SSD.
  - AI Use Cases: The solutions have been successful: successful examples of AI practices that have contributed to minimizing cyberspace risks.
  - Challenges and Limitations: The views of the respondents on the limitations of the existing AI technologies to fight some cybersecurity threats.

- **Triangulation**: The SMS findings (assisting in reviewing the whole literature on AI in the context of SSD) and the empirical survey data were triangulated in order to provide a more accurate concept about the role of AI in alleviating insider threats in SSD.

The empirical survey will be an addition to the research, which will introduce the actual real-world attitudes of AI-based SSD practices. The survey findings and the SMS will be useful in combination to develop a well-integrated AI-based framework of the SSD that will address the insider threats identified in software development. This framework can assist companies in developing and implementing AI-based security solutions to support their SSD initiatives and improve their security outcomes.

Linear-by-Linear Association

An ANOVA test[56] of linear-by-linear association between the results of the SMS and the empirical survey on types of insider threats in SSD is used. The existence of a predictable, ranked correlation between the extents to which the various insider threats are stressed in the literature (SMS) and the perceived seriousness of those threats by the survey practitioners. Under such a scheme, the types of insider threat (e.g., malicious insiders, negligent insiders, credential theft, social engineering, privilege escalation, etc.) are listed according to their emphasis level in the SMS (e.g., low, medium or high frequency or importance), and their respective perceived rate of impact or effectiveness rating as determined by professionals is taken as a continuous scale. One way ANOVA is then used to test whether there is a systematic mean increase (or decrease) in the mean scores of perception across the ordered levels of emphasis on literature. Strong linear correlation would mean that the research attention is oriented to the practitioner experience, but a non-significant or unattainable correlation would imply that there is a disconnect between the scholarly literature and practical cybersecurity measures in secure software development.

Fuzzy AHP (FAHP)

The types of insider threat and the corresponding AI-driven practice to prevent and detect them can be prioritized using the fuzzy analytic hierarchy process (FAHP)[24,57] with the inclusion of expert judgment in the case of uncertainty and subjectivity. FAHP presents a hierarchical form of the decision problem, where the overall goal is positioned at the top (e.g. "Effective mitigation of insider threats in secure software development")[11]. Then criteria (malicious insider, negligent insider, credential theft, privilege escalation, social engineering, etc.), and sub-criteria (particular AI-based practices including behavioral analytics, anomaly detection, automated access control, NLP monitoring, DLP, and threat hunting). Experts can make pairwise comparisons with fuzzy linguistic scales (e.g., slightly more important, strongly more important) instead of direct numerical comparisons, transformed into fuzzy triangular numbers to reflect ambiguity in human judgment better[58]. These fuzzy matrices are then aggregated, defuzzified and normalized to calculate priority weights that show which type of threat is the most important and the AI-driven practices that are best at each threat type.

In general, FAHP allows a more realistic and resilient prioritization model by integrating the opinions of multiple experts systematically, taking into account uncertainty, and generating ranked weights that guide organizations to allocate resources towards the most critical insider threats and the most effective AI-driven remediation policies in SSD[48,59–63].

Although FAHP was selected due to its appropriateness in modelling uncertainty and expert judgment, we are aware of the need to prove its efficiency as compared to other methods of MCDM. To compare FAHP with other popular MCDM techniques, like Classical AHP, TOPSIS, VIKOR and DEMATEL, a comparative study was developed. The following is the way such a comparative study might be organized[48,59–63]:

- Classical AHP vs. FAHP: Classical AHP needs specific judgments to be made when comparing two items, which is not realistic when considering insider threat practices because of uncertainty. A benefit that FAHP has over AHP is that it enables decision-makers to express their preferences more fuzzily, giving a more sophisticated way to prioritize. They can be compared by applying each method to the same insider threat data and comparing the results, which method produces more reliable or consistent outcome in a situation with uncertain data.
- TOPSIS and VIKOR vs. FAHP: TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) and VIKOR (VlseKriterijumska Optimizacija I Kompromisno Resenje) are both MCDM methods, which rank

alternatives by similarity to ideal solutions. Nevertheless, the techniques make use of crisp data and definite preferences, which may be restrictive in cases where subjective expert opinions are involved in a fuzzy scenario. To compare FAHP with TOPSIS and VIKOR, we might use the same data and make the comparison of their capacity to deal with uncertainty in the decision-making, particularly in the situations when expert judgments might not be entirely accurate.
- DEMATEL vs. FAHP: DEMATEL comes in handy especially in the realization of cause-and-effect relationships among criteria. However, FAHP is more appropriate in prioritizing and ranking tasks where the criteria are considered relative to the others, instead of cause and effect. One of the comparisons that could be done is to investigate the benefits of FAHP in cases where prioritization and ranking of threat mitigation practices are more applicable than the causal relationships. We would be able to evaluate the thoroughness and performance of both approaches in prioritizing insider threat mitigation practices, and pay attention to the manner in which both approaches deal with conflicting criteria.

Our survey was based on the first one, except that we added Fuzzy-AHP variant but only 45 of the 180 invited respondents replied. Following several strict checks to verify the validity and completeness of the 34 complete responses of cybersecurity and AI experts familiar with or involved in SSD settings, we then moved on to the analysis. Although the sample of 34 might be viewed as small, it is comparable to the sample sizes of other studies involving the same topic[48,59–63] and presents a reasonable point to generalize.

Our expert criteria were established depending on the particularities of the study we aimed to conduct, namely, the insider threat detection and prevention within the secure software development (SSD) environment. The characteristics that we chose in experts are:

- Experience in Cybersecurity: The experts had to have been working in cybersecurity, namely, in insider threat management, access control systems, or similar fields.
- Expertise in AI: Since the framework is AI-oriented, the professionals had to possess a good level of knowledge in the field of artificial intelligence and machine learning, specifically with regard to the security applications.
- Familiarity with SSD Environments: The participants were required to be directly involved in SSD organizations, by either being a professional or contributing to the research, which would enable them to offer pertinent and context-driven information.

This ensured to have the right experts that were well versed in the area of insider threat mitigation and the use of AI in secure software environments, and whose input was valuable for the research.

Results and analysis

This section presents the empirical and analytical findings of the research that uses an integrated research methodology with a SMS, an empirical survey of experts in AI and cybersecurity, and a linear-by-linear association analysis. This section presents the data obtained from literature and expert opinion to discuss the state-of-the-art for insider threats in SSD, the perceived effectiveness of AI-based insider threat mitigation and the gap between the research and the cybersecurity reality. Based on quantitative statistical analyses and decision modelling by experts, this section identifies patterns, relationships, and priority areas that can be used to form a strong, evidence-based view of how AI-based agentic practices can be used effectively to prevent and detect various forms of insider threats.

State-of-the-art of Insider Threats in SSD

The state-of-the-art of insider threats prevention in SSD is indicative of an ever-evolving frontier of cybersecurity, human behavior, and artificial intelligence due to the increasing acknowledgment of insiders as one of the most intricate and nebulous threats to software systems. Modern studies shift less and less toward the previous paradigm of rule-based security controls and more into the realm of data-driven, adaptive and AI-enabled security systems focused on real-time monitoring, behavioral analytics and proactive detection of risks during the software development life cycle. In existing literature, DevSecOps environments are marked by the transition to the continuous threat prevention mode, which includes user behavior analytics, anomaly detection, automated access control, and threat intelligence based on AI. However, technological innovation is not the only significant element of the field, issues such as false positives, privacy, organizational culture and dynamics of insider behaviours are also viewed as the challenges for insider threat detection, and the field is ever evolving.

Types of insider threats in software development organizations

The classification of insider threats in software development organizations is studied to identify the nature, occurrence and impact of different types of insider threats in the context of software development. Security-wise, the insider threats are complex as they are originated by people who already have legitimate access to systems, data, and development environment and therefore, they are the most difficult to identify and control. The insider threat of abuse, negligence or malice is even direr in software development organizations, where sharing, collaboration, privileged access and continuous integration and deployment (CI/CD) pipelines are the mode of operation. Thus, the results of the SMS and the empirical survey are combined in this section, to identify and explain the key types of insider threats, and why they are important in the development environment and why they are a serious risk to software security and organizational trust and resilience. Table 4; Fig. 4 show the major types of insider threats in software development organization.

Type of Insider Threats	Description	Impact on Security	Impact on Functionality
Malicious Insiders[30]	Employees or contractors who intentionally exploit their access to harm the organization.	- Data breaches or leaks of sensitive information (e.g., source code, customer data). - Sabotage of systems or software projects.	- Loss of critical data or intellectual property. - Damage to software project timelines and reputation.
Negligent Insiders[64]	Employees who unintentionally expose systems or data due to carelessness, lack of awareness, or improper training.	- Weak passwords or mishandling of sensitive information. - Inadvertent access or sharing of sensitive code.	- Security breaches due to unsecured systems. - Unintended disruptions in software development workflow.
Credential Theft[65–67]	Insiders whose credentials are stolen and misused to gain unauthorized access to systems.	- Unauthorized access to development environments, repositories, and production systems. - Erosion of trust in internal systems.	- Compromise of intellectual property. - Ability for attackers to manipulate or steal ongoing project work.
Inadvertent Misuse of Access[68]	Employees with access to systems that they don't fully understand may unintentionally misuse their privileges.	- Accidental exposure of data or vulnerabilities. - Unwitting modification of key system configurations or code.	- Unintended software errors due to improper configurations. - Broken functionality from misapplied changes.
Privilege Escalation[69–71]	Insiders who exploit their access levels to gain higher privileges beyond their role.	- Unauthorized access to restricted data. - Privilege abuse for malicious purposes like sabotage or data theft.	- Inadvertent or malicious changes that disrupt project progress. - New vulnerabilities introduced into the software.
Social Engineering[72,73]	Insiders manipulated by external parties to betray their organization or reveal sensitive information.	- Exploitation of access through phishing or coercion. - Leakage of credentials, trade secrets, or design documents.	- External attack vectors targeting internal systems. - Compromise of intellectual property and customer trust.
Software Piracy/ Code Theft[74]	Employees who copy or steal proprietary code or algorithms and sell or use them externally.	- Theft of proprietary code. - Intellectual property loss or copyright infringement.	- Leakage of valuable assets affecting competitive edge. - Compromise of project exclusivity and competitive advantage.
Insider Data Exfiltration[75]	Employees who intentionally or unintentionally transfer sensitive data to unauthorized locations.	- Data leakage of client information, software architecture, or trade secrets. - Violation of regulatory compliance standards.	- Loss of software design or client data integrity. - Potential damage to relationships with partners and customers.
Abuse of Development Tools[76]	Insiders who misuse their privileges in development tools (e.g., version control systems, CI/CD pipelines).	- Malicious insertion of vulnerabilities or backdoors into the code. - Tampering with project versions or unauthorized changes to the code base.	- Delayed project timelines due to corruption of codebase. - Decreased code quality and security issues introduced into production.

Table 4. Types of Insider Threats and its Impact on Security and Functionality.

Fig. 4. Types of insider threats in software development organization.

Methods and technology used by SSD organizations

Identification and prevention of insider threats have become a critical concern in SSD companies because these threats are caused by insiders who have legitimate access to the sensitive systems, code and development environments. Unlike the cyber-attacks, insider threats are more difficult to identify since they are integrated with

the normal workflows such as the development, testing, deployment and system administration. As the software development increasingly relies on DevSecOps, the cloud and collaboration platforms, companies have adopted a range of technical, process and AI-based measures to prevent, identify, and respond to insider threats. In this section, the authors introduce the current approaches and technologies (see Fig. 5; Table 5) such as traditional access controls and new AI-based monitoring systems, being leveraged to detect and mitigate unusual behavior, control access, and improve the security posture of the software development companies against insider threats.

Fig. 5. Methods and Technology Used by SSD Organizations.

Method/Technology	Description	Detection/Prevention Strategy	Impact on Security and Functionality
Access Controls[77]	Implementing restrictions on who can access specific systems and data.	- Role-Based Access Control (RBAC) - Principle of Least Privilege	Prevention: Limits insider access to sensitive data. Security: Prevents unauthorized access. Functionality: Streamlines roles and permissions.
User Monitoring[78]	Tracking and analyzing user activity to detect abnormal behavior.	- Activity Logging - User Behavior Analytics (UBA) - User Entity Behavior Analytics (UEBA)	Detection: Identifies suspicious activities and abnormal behaviors. Security: Detects potential threats in real time. Functionality: Monitors user actions without interrupting workflows.
Data Loss Prevention (DLP)[79]	Protecting against data exfiltration or unauthorized data transfers.	- USB Use Restrictions - File Transfer Monitoring - Email and Web Filtering	Prevention: Blocks unauthorized transfer of sensitive data. Security: Ensures only authorized data is shared. Functionality: Controls and audits data transfers effectively.
Secure Software Development[26]	Incorporating security measures throughout the software development lifecycle.	- Code Reviews - Static and Dynamic Code Analysis - Security Testing and Threat Modeling	Prevention: Identifies vulnerabilities before deployment. Security: Integrates secure coding practices. Functionality: Ensures that software is secure by design.
Mentoring and Training[80]	Educating staff on security best practices and potential threats.	- Security Awareness Sessions - Insider Threat Training - Regular Security Drills	Prevention: Reduces human error and insider negligence. Security: Cultivates security-first mindset. Functionality: Enhances employee vigilance and understanding.
Threat Detection Tools[81]	Tools used to identify and respond to potential threats in real-time.	- SIEM (Security Information and Event Management) - Anomaly Detection Systems - UEBA	Detection: Alerts when unusual activity occurs. Security: Provides real-time analysis and alerts. Functionality: Minimizes damage by responding quickly to detected threats.
Privilege Management[82,83]	Managing the permissions and access rights of users to ensure least privilege is enforced.	- Least Privilege Enforcement - Regular Privilege Audits	Prevention: Restricts access to sensitive systems and data. Security: Ensures users only have the privileges needed. Functionality: Limits potential insider threats by reducing unnecessary access.
Incident Response Plans[84]	Prepared plans and procedures to follow in case of a security breach or insider attack.	- Incident Response Procedures - Rapid Threat Mitigation	**Prevention and Detection**: Proactively manages and mitigates insider threats. **Security**: Rapidly responds to identified threats. **Functionality**: Minimizes disruption and damage to business operations.

Table 5. Methods and Technology Used by SSD Organizations and its Impact on Security and Functionality.

Limitations of current insider threats prevention strategies in SSD

The insider threats are also among the most thorny and most enduring security threat in SSD, given that individuals to whom the systems, data, and development environments have legitimate access perpetrate them. Although organizations have been using various technical, procedural and behavioral controls, which include access control measures, monitoring tools, security policies and awareness training, the strategies fail to fully predict, identify and address insider risks in dynamic, collaborative, and AI-enabled development environments. Most of the current strategies are reactive instead of proactive; they are based more upon inflexible rule-based mechanisms; and they do not sufficiently reflect the human, organizational, and situational drivers that can motivate insider behavior. As a result, a tense necessity to critically analyze the constraints of the existing prevention measures to define the gaps, enhance the detection, and shape the creation of more adaptive, intelligent and anthropocentric security models in secure software development is emerging. The detail limitation of insider threat prevention and its influence on security and functionality are given in Fig. 6; Table 6.

Key metrics to evaluate insider threat prevention measures in SSD

When evaluating the prevention of insider threats in SSD, it is worth having well-defined and measurable metrics that capture both technical and human factors of security. As organizations adopt AI-driven and automated security measures, it is especially important to assess these types of prevention measures to support risk assessment, resource allocation and decision-making. Performance metrics provide a structured way to determine if prevention measures are reducing vulnerabilities, detecting malicious or negligent behaviour and strengthening software security posture. Table 7; Fig. 7 thus presents and explains the main measures applied in research and practice to assess insider threat prevention systems, identifying their applicability in both academic research and practical cybersecurity in safe software development settings.

Gaps exist in current research on insider threat prevention in SSD

Although the focus on insider threats in the context of cybersecurity research has increased, there are still major gaps in the literature concerning how these threats are dealt with in the context of SSD. A large percentage of the current literature is generalized on the topic of organizational security or IT systems, and not the specific risk presented by software development environments, including code repositories, CI/CD pipelines, developer tools, and collaborative software development practices. Moreover, studies tend to ignore the differences in behaviors, motives, and technical routes of malicious, careless, and compromised insurers and consider them as one category. Empirical evidence regarding the association of theoretical mitigation strategies with the experience of practitioners in the real world is also limited, as well as the issue of how new AI-based practices

Fig. 6. Limitations of insider threat defenses in SSD.

Limitation	Impact on Security and Functionality	Possible Solutions/Improvement
Detection Gaps[85]	- Insider threats often go unnoticed due to lack of clear signs. - Missed opportunities for early threat detection.	- Improve anomaly detection tools. - Combine user behavior analytics with machine learning to spot hidden threats.
Resource Constraints[2]	- Incomplete monitoring may miss key insider activities. - Delays in responding to threats due to insufficient resources.	- Automate monitoring systems to reduce reliance on manual staff. - Increase investment in security tools and staff.
False Positives[79]	- Security teams may ignore valid alerts or waste resources on false alarms. - Increased workload on staff.	- Improve accuracy of detection systems. - Use machine learning to refine alerting mechanisms.
High Privilege Risks[85]	- Risk of unauthorized access to sensitive systems or data. - Increased attack surface for malicious insiders.	- Enforce strict role-based access control (RBAC). - Regularly audit and adjust privilege levels.
Insider Complexity[2]	- Complex behavior may evade detection systems. - Hard to differentiate between legitimate actions and malicious behavior.	- Implement behavioral analysis tools. - Focus on long-term behavior patterns rather than isolated incidents.
Evolving Tactics[79]	- Continuous adaptation by insiders makes it hard to stay ahead of threats. - Dynamic nature of threats requires ongoing updates to prevention strategies.	- Continuously update detection systems based on emerging threats. - Use threat intelligence feeds to adapt quickly.
Privacy vs. Surveillance[85]	- Surveillance may feel invasive to employees. - Risk of violating privacy laws and regulations.	- Clearly define boundaries for monitoring. - Implement data minimization principles to respect privacy.
Insufficient Training[2]	- Employees might inadvertently create risks by failing to follow security policies. - Weakest link in security is often human error.	- Regular training and awareness programs. - Simulated phishing and insider threat exercises.
Gaps in Insider Threat Defenses[85]	- Weak spots may be exploited by insiders. - Insufficient coverage of all potential attack vectors.	- Conduct regular security audits. - Implement layered defense mechanisms to cover all potential vulnerabilities.

Table 6. Limitations of insider threat defenses and its impact on security and functionality.

Metric	Purpose	Impact on Security and Functionality
Incident Detection Rate[79]	Measures the effectiveness of detection systems in identifying insider threats.	Security: Higher detection rate means better prevention. Functionality: Detects threats early, minimizing damage.
Insider Risk Assessment[4]	Assesses which employees may pose a higher risk based on behavior or access levels.	Security: Helps focus resources on high-risk individuals. Functionality: Enables better targeted monitoring and interventions.
Data Leakage Incidents[64]	Tracks incidents of sensitive data being transferred outside of authorized channels.	Security: Tracks effectiveness of data protection measures. Functionality: Identifies gaps in data leakage prevention.
User Access Violations[30]	Measures how frequently insiders attempt to access systems or data without proper permissions.	Security: Identifies access control issues and potential vulnerabilities. Functionality: Ensures access control policies are effective.
Time to Detect and Respond[4]	Measures the efficiency of the detection and response mechanisms.	Security: Shorter response times limit damage. Functionality: Faster response leads to reduced operational disruption.
Alert False Positive Rate[64]	Evaluates the accuracy of monitoring systems in identifying legitimate threats.	Security: Reduces wasted efforts on false alarms. Functionality: Minimizes resource drain caused by false positives.
Training Effectiveness[30]	Assesses the impact of security awareness programs on employee behavior.	Security: Ensures employees understand how to mitigate insider threats. Functionality: Promotes a proactive security culture.
Time to Remediate[30]	Measures the effectiveness and speed of incident remediation efforts.	Security: Reduces the impact of insider threats through fast mitigation. Functionality: Shorter remediation times ensure minimal disruption.
Employee Security Awareness[64]	Measures the overall level of security awareness across the organization.	Security: Ensures that all employees are aware of security risks and how to avoid them. Functionality: Strengthens overall security posture.

Table 7. Key metrics for evaluating insider threats in SSD and its impact on security and functionality.

are actually adopted, implemented and assessed in practice. The gaps (see Fig. 8; Table 8), in turn, limit scholarly knowledge and tangible attempts to develop effective, context-related prevention mechanisms in terms of insider threats in secure software development.

.

Comparison of literature review and empirical survey

This section offers detailed comparison between the findings of the literature review and empirical survey to identify the similarities and differences and complements between the academic concept of types of insider threats and the practitioners' perspective in SSD. While the literature is a valuable source of knowledge that provides a conceptual and technological perspective on insider threats, the empirical survey also incorporates the real world perspectives, experiences and concerns of cybersecurity and AI practitioners in the context of organization and software development. Through the juxtaposition of these two pieces of evidence, this comparison (see Fig. 9) is designed to confirm existing scholarly arguments, pinpoint the gaps between the research and practice, and point out the areas where empirical realities either confirm or refute the existing theoretical assumptions.

To determine the existence of statistically significant difference between literature review and empirical survey findings, we ran an ANOVA[56] as a measure of the difference between the mean of types of insider threats in software development.

The main objective of the ANOVA application in our research was to statistically confirm the correspondence between the insights obtained in the Systematic Mapping Study (SMS) and the survey based on the empirical research of cybersecurity and AI experts. Through these results, we aimed at ensuring that the literature review

Fig. 7. Key metrics for evaluating insider threats in SSD.

(SMS) and the empirical survey agree on the same findings about the best practices and methods to be used in insider threat prevention with regard to the nature of the Secure Software Development (SSD) environment.

ANOVA was chosen due to its capability to test the statistical significance of the difference between the perceptions of the two groups (literature and expert survey participants). This would enable us to authenticate the synthesis given in the SMS so that the results of both the literature and survey are aligned and can be considered reliable, rather than due to chance and fluctuation.

i. Hypothesis • Null Hypothesis (H_0): There is no significant difference in the perceptions of insider threats between the Literature Review and the Empirical Survey.

- Alternative Hypothesis (H_1): There is a significant difference in the perceptions of insider threats between the Literature Review and the Empirical Survey.

b **Data Collection**: We use the percentages of Fig. 9 to perform this test. For each type of insider threat, we can compare the percentage from Literature Review and Empirical Survey.

c **Assumptions of ANOVA**:

- The data is independent (we assume that the data from the Literature Review and Empirical Survey come from different sources).
- The data is normally distributed (we would ideally test for normality, but we will assume normality for now).
- The variance within groups should be equal (homogeneity of variance).

iv. **Formula for ANOVA**: The ANOVA test works by comparing the variance between the groups (in this case, Literature Review and Empirical Survey) to the variance within the groups. The formula for the F-statistic is:

$$F = \frac{Between\ Group\ Variance}{Within\ Group\ Variane}$$

Where:

- Between Group Variance (MSB) is calculated as:

Fig. 8. Research gaps in insider threat prevention.

Research gap	Impact on insider threat prevention	Potential solutions/areas for future research
Behavioral Prediction[86]	- Hard to identify potential threats based on behavior. - Predictions may be inaccurate or incomplete.	- Development of better machine learning models for predicting behavior. - Studying psychological triggers of insiders.
Threat Evolution[87]	- Insider threats evolve rapidly, making prevention strategies less effective. - New tactics can lead to gaps in existing defenses.	- Research on emerging insider threat techniques. - Continuous updates to threat detection systems based on new data.
Real-time Detection[88]	- Delayed or inaccurate detection can increase the damage caused by insider threats.	- Improved anomaly detection using real-time data. - Integration of behavioral analytics for more accurate real-time alerts.
Data Protection Strategies[89]	- Insiders may exploit access to sensitive data or IP without being detected.	- Advanced encryption and access control methods. - Research on data protection in decentralized systems.
Insider Attribution[14]	- Difficulty in determining who is responsible for a security incident within an organization.	- Development of better systems for tracking insider activities. - Research on insider behavior attribution models.
Effectiveness Measurement[86]	- Lack of clear metrics to evaluate the effectiveness of prevention strategies. - Makes it hard to assess the ROI of security measures.	- Creation of comprehensive effectiveness assessment frameworks. - Conduct longitudinal studies to track security improvements.
Resource Cost-Benefit[14]	- Organizations may struggle to justify the costs of comprehensive insider threat prevention strategies.	- Research into cost-effective preventive measures. - Cost-benefit analysis models tailored to insider threat prevention.
Organization Culture[88]	- Culture may either amplify or reduce the likelihood of insider threats, but is often overlooked in research.	- Study the relationship between organizational culture and insider threats. - Implement cultural change strategies to mitigate risks.
Privacy Balance[87]	- Balancing employee privacy with the need for surveillance can create ethical and legal dilemmas.	- Research on privacy-preserving monitoring technologies. - Develop policies that strike a balance between surveillance and privacy.

Table 8. Research Gaps and its Impact on Insider Threat Prevention.

$$MSB = \frac{SSB}{df_{between}} = \frac{\sum (n_k)\,(X_k - X)^2}{k-1}$$

- n_k = Number of observations in group k.
- X_k = Mean of group k.
- X = Overall mean.
- k = Number of groups (2 in our case = Literature review and Empirical survey).
- Within Group Variance (MSW) is calculated as:

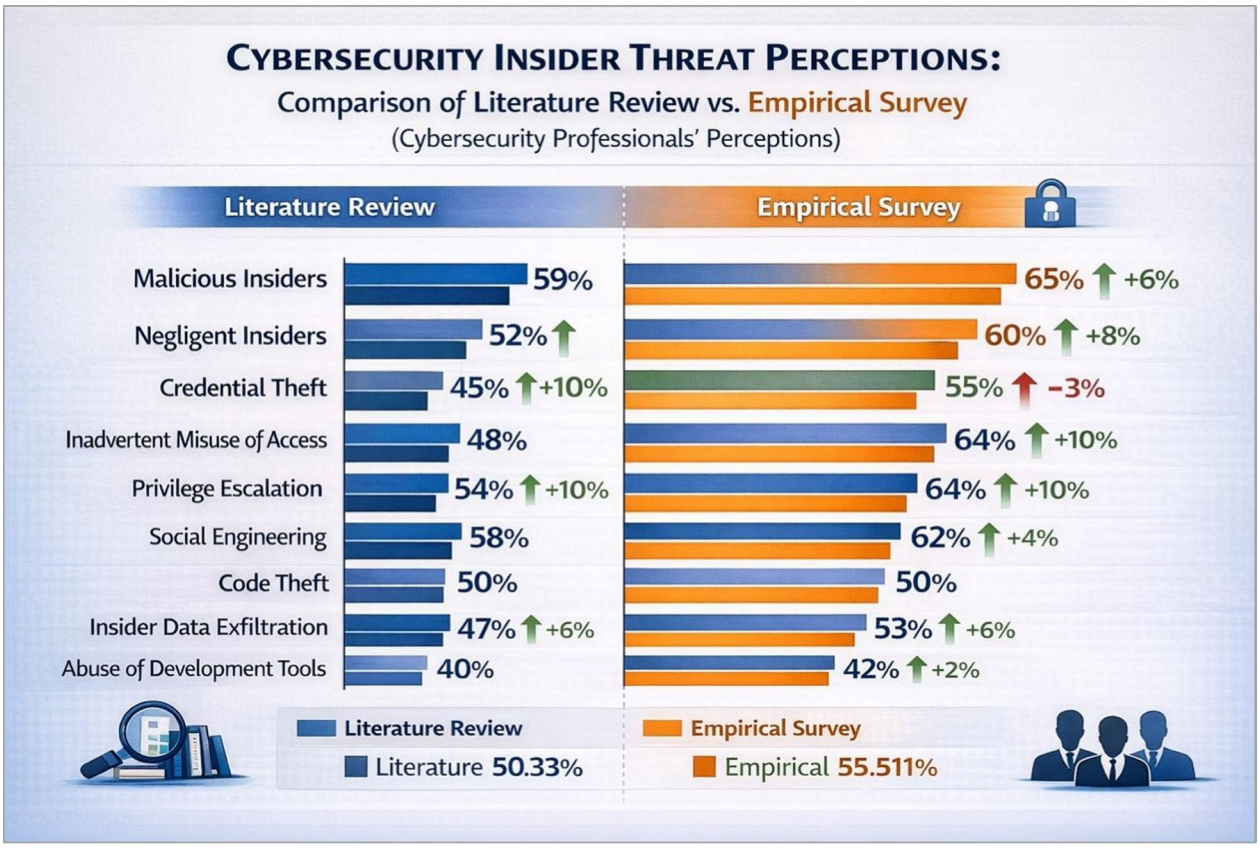

Fig. 9. SMS and Cybersecurity Professionals Perception about Types of Insider Threats in Software Development.

$$MSW = \frac{SSW}{df_{within}} = \frac{\sum (n_k - 1)s^2k}{N - k}$$

- SSW = Sum of squares within.
- s^2k = Variance of group k.
- N = Total number of variations.
- d_f = Degree of freedom.
- $df_{between} = k - 1$.
- $df_{within} = N - k$.
- Sum of squares between = SSB = $\sum (n_k)(X_k - X)^2$.
- Sum of squares within = SSW = $\sum (n_k - 1)s^2k$.

xxii. **Steps to perform ANOVA**:

- Mean of Literature Review (LR) = 50.33%.
- Mean of Empirical Survey (ES) = 55.11%.
- Overall Mean = 52.72%.
- SSB = 102.72.
- SSW = 854.89.
- Degree of freedom:

$df_{between} = k - 1 = 2–1 = 1.$

$$df_{within} = N - k = 16$$

- Mean of Square (MSB):

$$MSB = \frac{SSB}{df_{between}} = \frac{102.72}{1} = 102.72$$

$$MSW = \frac{SSW}{df_{within}} = \frac{854.89}{16} = 53.43$$

- F-Statistic:

  $$F = \frac{MSB}{MSW} = \frac{102.72}{53.43} = 1.92.$$

- P-Value: The p value is calculated using the F-distribution.

  p-value = 0.1846.

- Conclusion: The p-value (0.1846) is larger than the significant level (usually 0.05); therefore, we do not reject the null hypothesis. This implies that the perceptions of cybersecurity insider threats are not significantly different in the Literature Review and the Empirical Survey.
- "No Significant Difference": The lack of a statistically significant difference implies that, regarding the aspects that we have measured (i.e. the perception of the type of threats) there is no statistically significant difference was detected between the literature review and the empirical survey. It does not, however, imply that the two views are all the way in every respect in their interpretation, since to be aligned also involves qualitative consistency in the interpretation and contextualization of the types of threats, rather than statistical similarity.
- The fact that the results of the two sources are statistically equivalent, however, does not mean that they are the mirror reflections of each other in all the details. Even the differences in contextual emphasis, perceived relevance, or the emphasis placed on specific threats in each domain (e.g., literature vs. empirical observations) may still be slight. The fact that there was no significant difference thus does not mean that there was a complete coincidence but only means that there was, in general, a convergence between the two sources of perception.

Insider threats statistics based on continents

Figure 10 is a representation of the distribution of different insider threat types in different parts of the world that is, America, Europe, Asia, Australia and Africa. The information provided also gives us an opportunity to investigate the distribution of various kinds of insider threats, and gives some helpful information on the regional trends. The graph depicts the categories of threats in horizontal bars where percentages indicate occurrence of each type within the respective regions.

1. Malicious Insiders (34% – 41%).

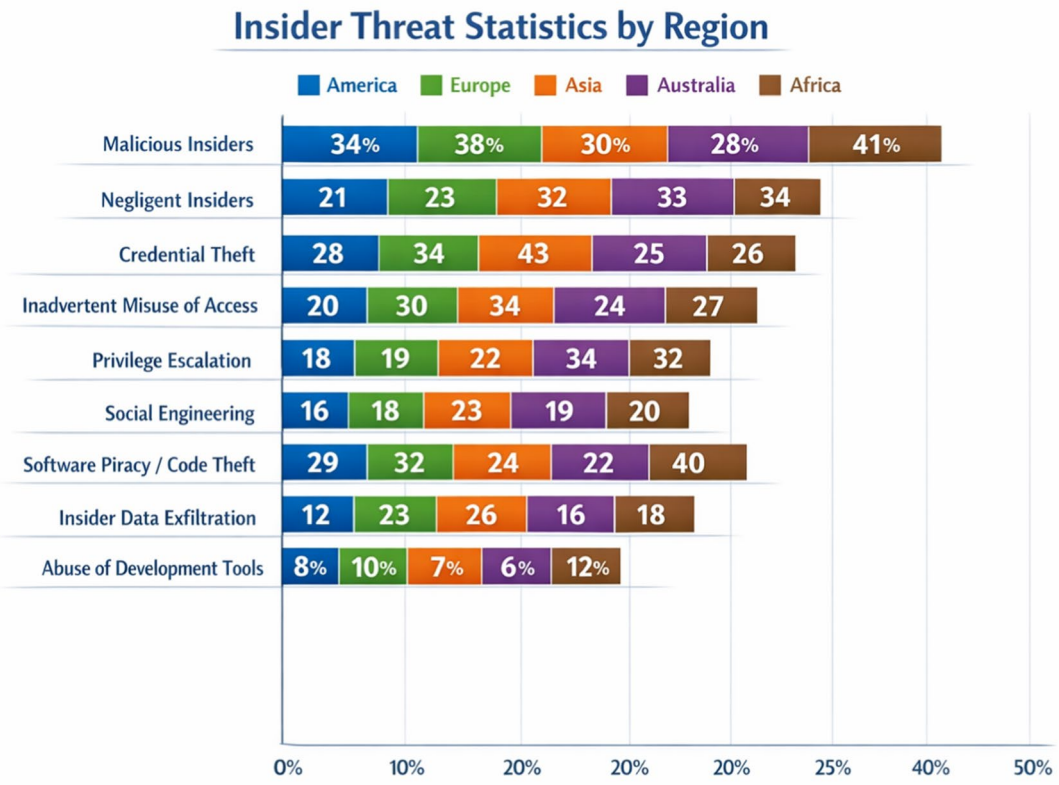

Fig. 10. Analysis of Types of Insider Threats in Software Development based on Continents.

- Description: Malicious insiders are people in an organization who abuse their access to organizational systems or data with the intent to make personal gain or to do harm to the organization. This is the most common threat in Africa (41%), and America (34%).
- Role of Agentic AI: Agentic AI may also be an important component in identifying and preventing malicious insiders in SSD settings. As an illustration, AI-based anomaly detection systems may be used to track the behavioral patterns of users and indicate any suspicious actions like unauthorized access, data exfiltration, or any attempts of sabotaging development tools. Even minor anomalies of regular patterns can be detected using behavioral analytics driven by an Agentic AI, which will decrease the threat of malicious insiders in different parts of the world.

2. Negligent Insiders (21% – 34%).

- Description: Negligent insiders are those who unintentionally violate security through negligence by not adhering to security measures, like sharing passwords or not updating software. Asia (32%) and Australia (33%) have the largest proportion of careless insider.
- Role of Agentic AI: Agentic AI can be used in training programs, including automated security awareness tools that can be used to reduce careless conduct by training employees on how to act in a secure manner. Moreover, AI-based surveillance systems will enable security teams to act promptly by notifying them in real time when users do not adhere to the standard procedures so that they can act before the threat escalates.

3. Credential Theft (26% – 43%).

- Description: An unauthorized acquisition of user credentials is credential theft, which frequently results in additional attacks. Asia is the most percentage (43%), Europe (34%), and America (28%).
- Role of Agentic AI: Agentic AI can be used to watch indicators of credential stuffing and password re-use by using highly sophisticated AI-based authentication systems (which employs multi-factor authentication (MFA) to identify compromised credentials). Behavioral biometrics can also be applied in AI-based tools that are needed to guarantee that only authorized users can access critical software development resources, even in the situation when their credentials have been stolen.

4. Inadvertent Misuse of Access (20% – 30%).

- Description: In this kind of threat, people act in a way that they do not intend to abuse the access to sensitive data or systems. Inadvertent misuse is high in Asia (34%), and Europe (30%).
- Role of Agentic AI: Agentic AI can keep track of real-time actions of users and make sure that no sensitive data is gained improperly. Access control systems powered by AI can be used to block or limit access in case of suspicious or unintentional activity. The AI could also be used in software development settings to track activities such as unwarranted code repository modifications or unintended loss of valuable files that could happen through oversight.

5. Privilege Escalation (18% – 34%).

- Description: Privilege escalation happens when an insider accesses unauthorized elevated access privileges. The highest reported rate of this type of threat is in Europe (34%).
- Role of Agentic AI: Agentic AI may be crucial in deterring privilege escalation by continuously reviewing user roles and access levels and alerting in case there is any attempt to escalate privileges unauthorized. AI-driven incident response systems are capable of automatically blocking access or taking remediation measures in case privilege escalation is identified in real-time.

6. Social Engineering (16% – 23%).

- Description: Social engineering is the art of manipulation of the insiders in order to disclose confidential information or carry out security-related actions. The highest rate of social engineering is in Asia (23%).
- Role of Agentic AI: AI is capable of detecting phishing attacks or other malicious social engineering tricks by monitoring email patterns. AI-based NLP-based surveillance systems can scan email traffic to identify suspicious language or patterns and make organizations aware of potential social engineering attacks prior to their success.

7. Software Piracy/Code Theft (24% – 40%).

- Description: This threat entails stealing proprietary software code or intellectual property. The highest rate of software piracy is in Australia (40%).
- Role of Agentic AI: Agentic AI has the capability to identify code leakage or unauthorized sharing of code through a behavioral analytics, which identifies abnormal code access, or copying. Moreover, DLP (Data Loss Prevention) software based on AI could be used to limit the possibility to replicate or move proprietary software even to unauthorized places.

8. Insider Data Exfiltration (12% – 26%).

- Description: insiders can define Insider data exfiltration as unauthorized transfer of sensitive data. The highest are recorded in Asia (26%), and Europe (23%).

- Role of Agentic AI: Agentic AI can avert data exfiltration through suspicious file transfer monitoring as well as anomaly detection in file access pattern. In the case an insider tries to upload or email confidential information to other unauthorized sources, AI systems will immediately raise red flags or block the action.

9. Abuse of Development Tools (6% – 12%).

- Description: This is a malicious action by insiders using development tools to create malicious code, including backdoors. The highest rate of misuse of development tools is in Africa (12%).
- Role of Agentic AI: AI can be used to keep track of the development tools in use. The use of AI-driven automated code review systems can alert about suspicious or malicious code alterations, so that any attempt to create a backdoor can be detected at an early stage of the development.

The agentic AI model is an effective tool to deal with insider threats in secure software development organizations (SSD). Through machine learning, anomaly detection using AI, it can identify, mitigate, and prevent a broad spectrum of insider threats, such as malicious insiders, negligent insiders, and credential theft, all of which are common in various regions. The continuous monitoring, analysis, and real-time actions of the framework guarantee that SSD organizations can be more resilient to insider threats, increasing the level of security and efficiency in their operations.

AI-Driven Agentic Practices for Prevention and Detection of Insider Threats in Software Development

This section discusses how AI-based agentic practices can be used to improve the prevention, detection, and mitigation of insider threats in SSD settings. With the rise in complexity, distributed nature, and data intensity of software systems, the conventional rule-based security measures are unable to meet the dynamic and human-centered characteristics of insider risks. The agentic systems of AI, with their autonomy, adaptability, real-time decision-making, and in-flight learning, provide a more proactive and intelligent option to detecting behavioral abnormalities, anticipating risky behavior, and automating security responses. This section will discuss how the practices can be enhanced to make SSD processes stronger and less vulnerable to various vulnerabilities related to insider threats by incorporating technical, behavioral, and organizational aspects of cybersecurity. We offers 91 AI-based agentic practices of prevention and mitigation of insider threats (the entire list is at Table 9 and Appendix A).

The agentic character of the practices such as AI-P56: Machine Learning-Driven Anomaly Detection and AI-P35: AI-Powered Threat Detection in the context of this paper is significant to explain how the practices are agentic.

An agentic practice is not just the identification of threats or anomalies, but rather the active instigation of a sequence of actions on the basis of the information collected to it, frequently in a multi-step process. The following is a short explanation and an example of the two:

1. AI-P56: Machine Learning-Driven Anomaly Detection.
2. In the practice, the system is not merely raising a red flag about anomalies but is actively involved in ongoing monitoring, analyzing behaviors among multiple system touchpoints (e.g., access logs, network traffic, file access patterns). In the event that a possible anomaly has been identified, the system goes beyond a simple alert by taking a number of measures:

- Sensing: It monitors real-time data at all times in order to detect any deviation to the laid down norms.
- Reasoning: The system uses machine-learning algorithms to determine whether the anomaly is consistent with previously known threat patterns or new behaviors.
- Acting: In case the anomaly is considered suspicious, the system automatically escalates it to the security personnel, isolates the suspicious activity and even initiates countermeasures (e.g., temporarily locking out user access or tagging the activity as something to be investigated further).

3. AI-P35: AI-Powered Threat Detection.
4. The practice includes agentic behavior by sensing, reason, and taking action in an organized workflow:

- Sensing: It gathers different sources of information like user activity, environmental information and system logs to identify any form of deviation that might indicate insider threats.
- Reasoning: The system uses highly developed AI models, e.g., behavioral analytics or natural language processing, to interpret the situation and intentions of the observed behaviors. As an example, it may examine the communication habits of an insider or file access requests to determine the possible ill motive.
- Acting: In case a possible threat is identified, the system does not simply signal administrators but initiates an automated process to address the threat, i.e. isolating the access of the affected user or changing system privileges depending on the seriousness of the threat. Moreover, it can trigger an investigation procedure, which consists of a set of various security tools, and procedures.

The practices in the above two examples are goal-driven, multi-step processes because they not only recognize anomalies or threats but also involve reasoning and automated tasks that result in specific actions such as threat mitigation and system protection. This agentic behaviour is in contrast to monitoring agents that only raise the alarm without any further attempts to rectify the problems.

We can bridge the gap between the central concept of the agentic behaviour and empirical results in the study by showing the active and dynamic nature of these practices.

AI-Driven Agentic Practices for Prevention and Detection of Malicious Insiders	
Code #	**AI-Driven Practices (AI-Ps)**
AI-P1	Behavioral Analytics and User Behavior Analytics (UBA)
AI-P2	Real-time Anomaly Detection
AI-P3	AI-Powered Threat Hunting
AI-P4	Predictive Analytics for Risk Profiling
AI-P5	Automated Access Control
AI-P6	AI-Driven Insider Attribution
AI-P7	Natural Language Processing (NLP) for Communication Monitoring
AI-P8	AI-Enhanced Privilege Escalation Detection
AI-P9	Automated Incident Response Systems
AI-P10	Continuous Learning and Adaptation
AI-P11	AI-Driven Data Loss Prevention (DLP)
AI-P12	AI-Powered Threat Intelligence Sharing
AI-P13	Behavioral Biometrics
AI-Driven Practices for Prevention and Detection of Negligent Insiders	
AI-P14	Anomaly Detection
AI-P15	Automated Access Control Management
AI-P16	Real-time Activity Monitoring and Alerts
AI-P17	Behavioral Biometrics
AI-P18	Automated Incident Response
AI-P19	Natural Language Processing (NLP) for Email and Chat Monitoring
AI-P20	Data Loss Prevention (DLP) with AI
AI-P21	Predictive Risk Assessment
AI-P22	Automated Security Training and Phishing Detection
AI-P23	Knowledge Transfer and Awareness Programs
AI-Driven Practices for Prevention and Detection of Credential Thefts	
AI-P24	Behavioral Biometrics
AI-P25	User and Entity Behavior Analytics (UEBA)
AI-P26	AI-Powered Anomaly Detection
AI-P27	Multi-Factor Authentication (MFA) with AI Analysis
AI-P28	Risk-Based Authentication
AI-P29	AI-Enhanced Access Control Policies
AI-P30	AI-Driven Credential Usage Monitoring
AI-P31	AI-Powered Threat Intelligence Integration
AI-P32	Automated Session Risk Analysis
AI-P33	AI-Driven Phishing Detection
AI-Driven Practices for Prevention and Detection of Inadvertent Misuse	
AI-P34	Behavioral Monitoring
AI-P35	Automated Code Review
AI-P36	AI-Powered Threat Detection
AI-P37	Data Anomaly Detection
AI-P38	Contextual Access Control
AI-P39	Security Best Practices Recommendations
AI-P40	Incident Response Automation
AI-P41	Automated Training and Guidance
AI-P42	Continuous Monitoring and Feedback Loops
AI-Driven Practices for Prevention and Detection of Privilege Escalation	
AI-P43	Anomaly Detection Systems
AI-P44	Behavioral Analysis
AI-P45	Automated Access Control Enforcement
AI-P46	Predictive Threat Intelligence
AI-P47	Contextual Authentication and Authorization
AI-P48	Continuous Monitoring and Alerting
AI-P49	Automated Incident Response
AI-P50	Machine Learning-based Risk Scoring
Continued	

AI-Driven Agentic Practices for Prevention and Detection of Malicious Insiders	
Code #	**AI-Driven Practices (AI-Ps)**
AI-P51	AI-Powered Penetration Testing
AI-P52	User and Entity Behavior Analytics (UEBA)
AI-Driven Practices for Prevention and Detection of Social Engineering	
AI-P53	AI-Powered Phishing Detection
AI-P54	Behavioral Analytics
AI-P55	Natural Language Processing (NLP)
AI-P56	AI-Powered Identity Verification
AI-P57	Machine Learning-Driven Anomaly Detection
AI-P58	AI-Enhanced Authentication
AI-P59	AI-Driven Employee Training
AI-P60	Context-Aware Security Systems
AI-P61	Intelligent Access Control Systems
AI-P62	AI-Powered Security Awareness Tools
AI-Driven Practices for Prevention and Detection of Code Theft	
AI-P63	Automated Code Review Systems
AI-P64	AI-Based Code Access Control
AI-P65	Machine Learning-Based Anomaly Detection
AI-P66	AI-Enhanced Version Control Systems
AI-P67	AI-Powered Code Obfuscation
AI-P68	Behavioral Analytics for Developer Actions
AI-P69	Automated Code Integrity Checks
AI-P70	Intelligent Code Encryption
AI-P71	AI-Driven Collaboration and Communication Monitoring
AI-P72	AI-Enabled Audit Trail Generation
AI-Driven Practices for Prevention and Detection of Data Exfiltration	
AI-P73	AI-based Anomaly Detection
AI-P74	AI-powered User Behavior Analytics (UBA)
AI-P75	AI-driven Data Loss Prevention (DLP)
AI-P76	AI-enhanced Network Traffic Analysis
AI-P77	AI-based Endpoint Protection
AI-P78	AI-enabled Access Control
AI-P79	AI-powered Encryption Management
AI-P80	AI-driven Security Information and Event Management (SIEM)
AI-P81	AI-based Threat Intelligence
AI-P82	AI-enhanced Insider Threat Detection
AI-Driven Practices for Prevention and Detection of Abuse of Development Tools	
AI-P83	Automated Code Review Tools
AI-P84	Behavioral Monitoring Systems
AI-P85	Anomaly Detection Algorithms
AI-P86	Access Control Optimization
AI-P87	Natural Language Processing (NLP)
AI-P88	Automated Policy Enforcement
AI-P89	Intelligent Logging and Forensics
AI-P90	Self-Healing Development Systems
AI-P91	Predictive Risk Analytics

Table 9. AI-Driven Agentic Practices for Prevention and Mitigation of Insider Threats[2,3,5,13,17,18,22,39,42,45,64,72,75,81,86,90–111].

AI-driven agentic framework

Insider threats have been regarded as one of the most pressing, persistent security issues in SSD as trusted insiders who have access to systems, data, and development platforms cause them. Insider risks have been highly associated with human factors, which are hard to predict and prevent using conventional security measures, unlike cyber-attacks from outside. To address this problem, agentic systems based on AI have gained interest because it can automatically monitor, detect, learn from changing patterns and decide on security in real time.

However, these AI-based practices have not been prioritized and assessed yet, but this is a methodological problem, particularly when we consider the uncertainty and subjectivity of decisions when it comes to cybersecurity. To fill this gap, this paper suggests an agentic framework of the development of AI-based insider threat prevention and detection in SSD, where Fuzzy Analytic Hierarchy Process (FAHP) is used as a multi-criteria decision-making process to evaluate, compare, and rank AI-based insider threat mitigation practices under uncertainty.

FAHP[112,113] plays a vital part in solving insider threats in SSD by aiding in assessing and ranking the mitigation measures to complex, difficult to detect problems. Its major benefit is that it can deal with uncertainty and subjectivity in expert ratings, which is vital in the assessment of insider threats[11]. FAHP models are able to use imprecise information to make more accurate decisions by use of fuzzy logic[114]. It enables calculating both local and global weights in order to prioritize the most effective practices, considering various criteria, including effectiveness and ease of implementation[115]. FAHP is a systematic incorporation of expert knowledge, decisions made using it are based on the collective knowledge of the experts and its adaptability makes it flexible to various kinds of silent bugs[26]. Finally, FAHP helps to prioritize effective strategies that are more likely to enhance code quality, minimize errors, and increase the reliability of insider threat mitigation in SSD[112]. FAHP uses Triangular Fuzzy Numbers (TFNs- See Table 10) to encode linguistic terms, which allows many decision-makers to be used and quantifies these linguistic variables. This is a common approach used in engineering to deal with uncertainty of fuzzy environment[114]. We apply Chang FAHP approach[112], which is known to be powerful and effective in such analysis.

In our Fuzzy AHP (FAHP) model, we conducted the Fleiss kappa analysis to determine consistency and reliability of the expert judgments. The Kappa of Fleiss was achieved by the formula:

$$K = \frac{P_o - P_e}{1 - P_e}$$

.

P_o is the observed proportion of agreement among experts and P_e is the expected proportion of agreement by chance. Once we used the formula with our data, we got a Kappa of 0.75. This is a good value because there is a good agreement among the experts since Kappa scores between 0.61 and 0.80 are said to represent significant reliability. Thus, the consistency of the judgments at the FAHP level and their reliability are confirmed by the outcome of 0.75, which proves that the rankings and the ratings based on the experts are also believable and dependable.

We have conducted a sensitivity analysis among experts in order to determine the strength of our results and make Fuzzy AHP (FAHP) model reliable in prioritizing AI-driven practices to combat insider threats. The analysis entailed the study of the impact of the variation in the judgments of the experts especially with respect to the weights given to the various practices in determining the final rankings and the overall results of the framework. In particular, we performed the sensitivity analysis by comparing the outcomes of various groups of experts, each having its own opinion and experience in the sphere of cybersecurity and AI. This enabled us to establish whether the framework was constant even when there might have been differences in the personal views or bias of individual experts. Through this sensitivity analysis, we could access the stability of the prioritization process and make sure that the final rankings were not too sensitive to minor modifications in expert ratings. These findings of this analysis were an additional confirmation that the framework has the potential to be useful in decision-making in a secure software development setting, making sure that the model can remain strong and that it could be tailored to the various expert perspectives.

To carry out pairwise comparison of matrices using triangular fuzzy numbers (TFNs) to the silent bugs. To obtain Table 11, we have used the following steps.

- Give a set of silent bugs a fuzzy comparison value using a linguistic scale and the associated TFN values.
- Build the pairwise matrix comparison with fuzzy numbers, so that the matrix is symmetric, and that the diagonal elements are always 1, as each bug is compared with itself.
- The pairwise matrix was created based on the conversion scale of Table 10 of TFNs.

We followed the following step-by-step Fuzzy AHP (Chang's Extent Analysis) using Table 11 TFN pairwise matrix (IT-1 … IT-9), with full formulas and the computed numbers:

- **Step-1: Row wise Fuzzy sums ($\bar{r}_i$)** : For each criterion i, sum all TFNs across row i.

Linguistic Scale	Triangular Fuzzy Scale	Triangular Fuzzy Reciprocal Scale
Just Equal (JE)	(1, 1, 1)	(1, 1, 1)
Equally Important (EI)	(0.5, 1, 1.5)	(0.6, 1, 2)
Weakly Important (WI)	(1, 1.5, 2)	(0.5, 0.6, 1)
Strongly More Important (SMI)	(1.5, 2, 2.5)	(0.4, 0.5, 0.6)
Very Strongly More Important (VSMI)	(2, 2.5, 3)	(0.3, 0.4, 0.5)
Absolutely More Important (AMI)	(2.5, 3, 3.5)	(0.2, 0.3, 0.4)

Table 10. Conversion scale of triangular fuzzy numbers (TFNs)[116].

Categories	IT-1	IT-2	IT-3	IT-4	IT-5	IT-6	IT-7	IT-8	IT-9
IT-1	(1, 1, 1)	(0.4, 0.5, 0.6)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	(0.5, 0.6, 1)	(0.5, 1, 1.5)	(0.6, 1, 2)	(0.6, 1, 2)	(0.5, 0.6, 1)
IT-2	(1.5, 2, 2.5)	(1, 1, 1)	(0.5, 1, 1.5)	(0.4, 0.5, 0.6)	(0.5, 1, 1.5)	(1, 1.5, 2)	(0.5, 0.6, 1)	(0.5, 0.6, 1)	(1, 1.5, 2)
IT-3	(0.6, 1, 2)	(0.6, 1, 2)	(1, 1, 1)	(1, 1.5, 2)	(1.5, 2, 2.5)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	(1.5, 2, 2.5)	(0.6, 1, 2)
IT-4	(0.6, 1, 2)	(1.5, 2, 2.5)	(0.5, 0.6, 1)	(1, 1, 1)	(0.5, 1, 1.5)	(1.5, 2, 2.5)	(0.5, 1, 1.5)	(0.5, 1, 1.5)	(0.5, 1, 1.5)
IT-5	(1, 1.5, 2)	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(0.6, 1, 2)	(1, 1, 1)	(0.6, 1, 2)	(0.5, 0.6, 1)	(0.6, 1, 2)	(0.4, 0.5, 0.6)
IT-6	(0.6, 1, 2)	(0.5, 0.6, 1)	(0.6, 1, 2)	(0.4, 0.5, 0.6)	(0.5, 1, 1.5)	(1, 1, 1)	(1, 1.5, 2)	(1, 1.5, 2)	(1.5, 2, 2.5)
IT-7	(0.5, 1, 1.5)	(1, 1.5, 2)	(0.6, 1, 2)	(0.6, 1, 2)	(1, 1.5, 2)	(0.5, 0.6, 1)	(1, 1, 1)	(0.5, 0.6, 1)	(0.5, 0.6, 1)
IT-8	(0.5, 1, 1.5)	(1, 1.5, 2)	(0.4, 0.5, 0.6)	(0.6, 1, 2)	(0.5, 1, 1.5)	(0.5, 0.6, 1)	(1, 1.5, 2)	(1, 1, 1)	(1, 1.5, 2)
IT-9	(1, 1.5, 2)	(0.5, 0.6, 1)	(0.5, 1, 1.5)	(0.6, 1, 2)	(1.5, 2, 2.5)	(0.4, 0.5, 0.6)	(1, 1.5, 2)	(0.5, 0.6, 1)	(1, 1, 1)

Table 11. Fuzzified pair-wise matrix comparison of types of insider threat (IT) in software development.

Category	$\bar{r}_i$=(l, m, u))
IT-1	(5.1, 7.7, 12.1)
IT-2	(6.9, 9.7, 13.1)
IT-3	(7.8, 11.5, 17.0)
IT-4	(7.1, 10.6, 15.0)
IT-5	(5.7, 8.1, 13.2)
IT-6	(7.1, 10.1, 14.6)
IT-7	(6.2, 8.8, 13.5)
IT-8	(6.5, 9.6, 13.6)
IT-9	(7.0, 9.7, 13.6)

Table 12. Rows sums.

$$\bar{r}_i = \sum_{j=1}^{n} \bar{a}_{ij} = \left(\sum_{j=1}^{n} l_{ij}, \sum_{j=1}^{n} m_{ij}, \sum_{j=1}^{n} u_{ij}\right) \tag{1}$$

Using Eq. (1) we calculated Table 11 rows sums and presented it in Table 7 and Table 12.

- **Step-2: Total Fuzzy sums ($\bar{R}$)** : For each criterion i, sum all TFNs across row i:

$$\bar{R} = \sum_{i=1}^{n} \bar{r}_i = \left(\sum_{j=1}^{n} l_i, \sum_{j=1}^{n} m_i, \sum_{j=1}^{n} u_i\right) \tag{2}$$

Total sums computed using Eq. (2) $\bar{R}$ = (59.4, 85.8, 125.7).

- **Step-3: Inverse of total Fuzzy sums** $\left(\bar{R}^{-1}\right)$: For a TFN (l, m, u), Chang uses:

$(l,\ m,\ u)^{-1} = \left(\frac{1}{u}, \frac{1}{m}, \frac{1}{l}\right)$ (3).

Inverse computed using Eq. (3) $\left(\bar{R}^{-1}\right) = \left(\frac{1}{125.7}, \frac{1}{85.8}, \frac{1}{59.4}\right)$ = (0.00795545, 0.01165501, 0.01683502).

- **Step-4: Synthetic Extent ($\bar{S}_i$)** : $\bar{S}_i = \bar{r}_i \otimes \bar{R}^{-1}$ (4).

For TFNs, multiplication is: $(l_1,\ m_1,\ u_1) \otimes (l_2,\ m_2,\ u_2) = (l_1 l_2,\ m_1 m_2,\ u_1 u_2)$.

Synthetic extent computed using Eq. (4) and presented in Table 13.

- **Step-5: Degree of Possibility V ($\bar{S}_i \geq \bar{S}_j$)**.

For $\bar{S}_i = (l_i,\ m_i,\ u_i)$, $\bar{S}_j = (l_j,\ m_j,\ u_j)$.

$V(\bar{S}_i \geq \bar{S}_j)$

Example calculation using Eq. (5) for (Case-3): V (IT-1 ≥ IT-2).

- $\bar{S}_1$ = (0.0406, 0.0897, 0.2037).
- $\bar{S}_2$ = (0.0549, 0.1130, 0.2205).

Category	$\bar{S}_i$ = (l, m, u))
IT-1	(0.0406, 0.0897, 0.2037)
IT-2	(0.0549, 0.1130, 0.2205)
IT-3	(0.0620, 0.1340, 0.2862)
IT-4	(0.0565, 0.1235, 0.2525)
IT-5	(0.0453, 0.0944, 0.2222)
IT-6	(0.0564, 0.1177, 0.2458)
IT-7	(0.0493, 0.1026, 0.2272)
IT-8	(0.0517, 0.1119, 0.2290)
IT-9	(0.0557, 0.1130, 0.2290)

Table 13. Synthetic extents.

Category	d_i
IT-1	0.7618
IT-2	0.8831
IT-3	1.0000
IT-4	0.9478
IT-5	0.8017
IT-6	0.9184
IT-7	0.8400
IT-8	0.8829
IT-9	0.8883

Table 14. Minimum possibility for each Type of Insider Threat. $\sum d_i = 7.9240$.

Here $m_1 < m_2$ and $l_2 < u_1$, so use case 3:

V = $\frac{l_2 - u_1}{(m_1 - u_1)(m_2 - l_2)}$.

V = 0.8646.

- **Step-6: Compute d_i (Minimum possibility for each Type of Insider Threat)**.

$$d_i = \underset{j \neq i}{m} n \; V(\bar{S}_i \geq \bar{S}_j) \quad (6).$$

Using Eq. (6) d_i values computed and presented in Table 14.

- **Step-7: Normalize to get final FAHP Weights (w_i)**.

$$w_i = \frac{d_i}{\sum_{k=1}^{n} d_k} \quad (7)$$

Final weights + ranking (local = global here).

Using Eq. (7) we computed the final weights of all types of insider threats and presented in Table 15.

According to the FAHP results, the most important perceived insider threat in secure software development is Credential Theft (IT3) then Inadvertent Misuse of Access (IT4) and Social Engineering (IT6). Reduced relative weight is noted in the case of Malicious Insiders (IT1), which implies that process-driven or unintended insider threats are controlling the situation, rather than intentional maliciousness occurring in this case.

- **Step-8: Convert TFN matrix to Crisp matrix**.

For every entry $\bar{a}_{ij}$ = (l, m, u), compute: $P_{crisp} = \frac{(4m+l+u)}{6}$ (8).

Example calculation using Eq. (8)

- If $\bar{a}_{12}$ = (0.4, 0.5, 0.6).

$$P_{12} = \frac{4(0.5) + 1(0.4) + 1(0.6)}{6} = \frac{2+0.4+0.6}{6} = \frac{2+1}{6} = \frac{3}{6} = 0.5.$$

- **Step-9: Compute λ_{max}**.

Using the crisp matrix (P), the principle eigenvalue is obtained from:

Rank	Code	Insider Threat	Final Weight
1	IT3	Credential Theft	**0.1262**
2	IT4	Inadvertent Misuse of Access	**0.1196**
3	IT6	Social Engineering	**0.1159**
4	IT9	Abuse of Development Tools	**0.1121**
5	IT2	Negligent Insiders	**0.1114**
6	IT8	Insider Data Exfiltration	**0.1114**
7	IT7	Software Piracy/Code Theft	**0.1060**
8	IT5	Privilege Escalation	**0.1012**
9	IT1	Malicious Insiders	**0.0961**

Table 15. Final FAHP Weight (Chang's Extent Analysis).

$P_w = \lambda_{max} w$ (9).
Using Eq. (9) the value is: $\lambda_{max} = 9.6761$

- **Step-10: Consistency Index (CI)**.

CI = $\frac{(\lambda_{max} - n)}{n-1}$ (10).
Here $n = 9$

$$CI = \frac{9.6761 - 9}{9 - 1} = \frac{0.6761}{8} = 0.0845$$

.
CI = 0.0845.

- **Step-11: Consistency Ratio (CR)**.

CR = $\frac{CI}{RI}$ (11).
For $n = 9$, Saaty's Random Index is: RI = 1.45.
Using Eq. (11) CR = $\frac{0.08451}{1.45}$ = 0.0583.
CR = 0.0583.
CR = 0.0583 < 0.10.
This means that pairwise judgments are consistent (acceptable consistency).

Following the FAHP procedures outlined above, the weights for all AI-driven agentic practices were computed, and the results are summarized in Table 16.

Table 16 is a hierarchical prioritization of AI-based agentic practices over nine insider-threat categories (IT-1 through IT-9) with each category being assigned a weight and each practice being ranked both locally (within its category) and globally (across all categories).

In general, the top priorities in the world focus on measures of constant detection and monitoring. The best practices are machine-learning-based anomaly-detection of Social Engineering (IT-6; Final Weight = 0.0206; Rank 1) and AI-based threat-detection of Inadvertent Misuse of Access (IT-4; 0.0205; Rank 2), and followed by anomaly detection of Negligent Insiders (IT-2; 0.0 In the case of Credential Theft (IT-3), the most commonly used practices are UEBA (0.0179; Rank 10) and AI-powered anomaly detection (0.0173; Rank 13), which shows the significance of behavior-based identification and credit-usage monitoring.

In the case of Software Piracy/Code Theft (IT-7), the priorities are focusing on anomaly detection (0.0191; Rank 8), developer behavioral analytics (0.0165; Rank 14), and collaboration/communication monitoring (0.0155; Rank 22). In the case of Insider Data Exfiltration (IT-8), AI-driven DLP (0.0169; Rank 15) and AI-enhanced insider threat detection (0.0154; Rank 25) are the leaders, with Abuse of Development Tools (IT-9) being dominated by anomaly detection algorithms (0.0177; Rank 19 Conversely, despite the fact that Malicious Insiders (IT-1) focuses on AI-driven DLP, real-time anomaly detection, and behavioral analytics/UBA locally, the categories are ranked lower (e.g., Rank 3745) because the category weight of IT-1 is lower.

This framework aims to offer a prioritized, evidence-based list of AI-driven agentic practices to reduce insider threats throughout the secure software development lifecycle. The table uses (i) a weight associated with each insider-threat category (IT-1 to IT-9) and (ii) local weights and ranks associated with each candidate practice in each category to generate final weights and global ranks, which reveal which practices make the most overall contribution to mitigating insider threats. In this regard, the table serves as a decision-support artifact: it converts a multi-dimensional security issue into a set of priorities understandable by organizations, allowing them to not only compare controls within the context of a single threat type, but also the entire range of insider risks.

This table can be used in practice in a variety of ways by SSD organizations. First, it can inform strategic investment and program sequencing by indicating the most globally important practices (e.g., anomaly detection, continuous monitoring, and AI-enabled DLP), which are ranked highly repeatedly in a variety of threat categories. This gives security leaders an opportunity to invest budgets and engineering capacity in controls that have wide coverage and can be measured instead of implementing point solutions that are small

Code	AI-Driven Agentic Practices for addressing IT-1: Malicious Insiders (Weight = 0.0961)	Final Weight	Final Rank
AI-P10	AI-Driven DLP	0.0123	37
AI-P2	Real-time Anomaly Detection	0.0117	40
AI-P1	Behavioral Analytics and UBA	0.0110	45
AI-P9	Automated Incident Response Systems	0.0101	52
AI-P3	AI-Powered Threat Hunting	0.0092	56
AI-P5	Automated Access Control	0.0083	61
AI-P8	Privilege Escalation Detection	0.0073	67
AI-P4	Predictive Risk Profiling	0.0064	75
AI-P6	Insider Attribution	0.0052	81
AI-P11	Threat Intelligence Sharing	0.0041	83
AI-P7	NLP Communication Monitoring	0.0037	84
AI-P12	Behavioral Biometrics	0.0035	85
Code	**AI-Driven Agentic Practices for addressing IT-2: Negligent Insiders (Weight = 0.1114)**	**Final Weight**	**Final Rank**
AI-P13	Anomaly Detection	0.0199	3
AI-P19	Data Loss Prevention (DLP) with AI	0.0193	5
AI-P15	Real-time Activity Monitoring and Alerts	0.0178	9
AI-P17	Automated Incident Response	0.0140	27
AI-P14	Automated Access Control Management	0.0129	24
AI-P20	Predictive Risk Assessment	0.0093	54
AI-P16	Behavioral Biometrics	0.0070	69
AI-P22	Knowledge Transfer and Awareness Programs	0.0067	74
AI-P21	Automated Security Training and Phishing Detection	0.0024	87
AI-P18	NLP for Email and Chat Monitoring	0.0021	88
Code	**AI-Driven Agentic Practices for addressing IT-3: Credential Theft (Weight = 0.1262)**	**Final Weight**	**Final Rank**
AI-P24	UEBA	0.0179	10
AI-P25	AI-Powered Anomaly Detection	0.0173	13
AI-P29	Credential Usage Monitoring	0.0148	23
AI-P30	Access Control Policies	0.0137	28.5
AI-P31	Session Risk Analysis	0.0129	26
AI-P32	Phishing Detection	0.0120	38
AI-P27	Risk-Based Authentication	0.0111	44
AI-P26	MFA with AI Analysis	0.0101	51
AI-P30	Threat Intelligence Integration	0.0087	59
AI-P23	Behavioral Biometrics	0.0078	65
Code	**AI-Driven Agentic Practices for addressing IT-4: Inadvertent Misuse of Access (Weight = 0.1196)**	**Final Weight**	**Final Rank**
AI-P35	AI-Powered Threat Detection	0.0205	2
AI-P36	Data Anomaly Detection	0.0189	6
AI-P41	Continuous Monitoring and Feedback Loops	0.0170	18
AI-P39	Incident Response Automation	0.0147	29
AI-P33	Behavioral Monitoring	0.0136	33
AI-P37	Contextual Access Control	0.0124	39
AI-P34	Automated Code Review	0.0102	50
AI-P38	Security Best Practices Recommendations	0.0071	68
AI-P40	Automated Training and Guidance	0.0053	80
Code	**AI-Driven Agentic Practices for addressing IT-5: Privilege Escalation (Weight = 0.1012)**	**Final Weight**	**Final Rank**
AI-P47	Continuous Monitoring and Alerting	0.0199	4
AI-P42	Anomaly Detection Systems	0.0183	7
AI-P51	UEBA (User and Entity Behavior Analytics)	0.0167	20
AI-P48	Automated Incident Response	0.0144	30
AI-P49	ML-based Risk Scoring	0.0120	43
AI-P45	Predictive Threat Intelligence	0.0090	57
AI-P44	Automated Access Control Enforcement	0.0059	79
AI-P46	Contextual Authentication and Authorization	0.0032	86
AI-P43	Behavioral Analysis	0.0011	89
AI-P50	AI-Powered Penetration Testing	0.0006	90

Code	AI-Driven Agentic Practices for addressing IT-6: Social Engineering (Weight = 0.1159)	Final Weight	Final Rank
AI-P56	**Machine Learning-Driven Anomaly Detection**	**0.0206**	**1**
AI-P60	Intelligent Access Control Systems	0.0177	12
AI-P57	AI-Enhanced Authentication	0.0167	21
AI-P55	AI-Powered Identity Verification	0.0149	31
AI-P59	Context-Aware Security Systems	0.0111	46
AI-P53	Behavioral Analytics	0.0102	49
AI-P54	Natural Language Processing (NLP)	0.0073	66
AI-P52	AI-Powered Phishing Detection	0.0067	76
AI-P61	AI-Powered Security Awareness Tools	0.0057	82
AI-P58	AI-Driven Employee Training	0.0052	78
Code	**AI-Driven Agentic Practices for addressing IT-7: Software Piracy/Code Theft (Weight = 0.1060)**	**Final Weight**	**Final Rank**
AI-P64	Machine Learning-Based Anomaly Detection	0.0191	8
AI-P67	Behavioral Analytics for Developer Actions	0.0166	14
AI-P70	AI-Driven Collaboration and Communication Monitoring	0.0155	22
AI-P68	Automated Code Integrity Checks	0.0109	32
AI-P71	AI-Enabled Audit Trail Generation	0.0094	55
AI-P69	Intelligent Code Encryption	0.0080	62
AI-P65	AI-Enhanced Version Control Systems	0.0076	64
AI-P63	AI-Based Code Access Control	0.0074	70
AI-P62	Automated Code Review Systems	0.0066	72
AI-P66	AI-Powered Code Obfuscation	0.0050	17
Code	**AI-Driven Agentic Practices for addressing IT-8: Insider Data Exfiltration (Weight = 0.1114)**	**Final Weight**	**Final Rank**
AI-P74	AI-driven DLP	0.0169	15
AI-P81	AI-enhanced Insider Threat Detection	0.0154	25
AI-P73	UBA	0.0123	34
AI-P72	Anomaly Detection	0.0125	41
AI-P79	AI-driven SIEM	0.0117	48
AI-P75	Network Traffic Analysis	0.0105	53
AI-P77	AI-enabled Access Control	0.0096	60
AI-P76	Endpoint Protection	0.0085	63
AI-P80	Threat Intelligence	0.0078	77
AI-P78	Encryption Management	0.0058	11
AI-Driven Agentic Practices for addressing IT-9: Abuse of Development Tools (Weight = 0.1121)		**Final Weight**	**Final Rank**
Anomaly Detection Algorithms		0.0178	19
Behavioral Monitoring Systems		0.0161	25
Intelligent Logging and Forensics		0.0147	28.5
Access Control Optimization		0.0137	35
Automated Policy Enforcement		0.0124	42
Predictive Risk Analytics		0.0116	47
Automated Code Review Tools		0.0105	58
Natural Language Processing (NLP)		0.0087	73
Self-Healing Development Systems		0.0066	75

Table 16. AI-driven agentic framework for addressing insider threats in SSD organizations.

and scattered. Second, the framework aids in the selection of risk-based controls: teams can begin with practices scoring higher on the threat categories with the highest relevance to their context (e.g., credential theft, negligent insiders, or data exfiltration), and then learn more specialized practices that deal with more specific risks.

The approach can be practically used to build a prioritization roadmap for the insertion of AI in the secure development of SDLC. High-rank best practices can be linked to SDLC stages and integrated into the pipeline and the toolset of the developer (e.g., continuous monitoring, user and entity behavior analytics, automatic access control and automatic incident response). This enables the switch from reactive and manual control to proactive, policy enforced and fast to respond with traceability, and governance is ensured. Furthermore, the local rankings in each IT category provide within-threat recommendations - for example, if credential theft is a major threat, UEBA, session risk and risk based authentication makes sense.

Finally, the framework supports governance, reporting and benchmarking. Security managers to justify control decisions to management, auditors and the development team using a transparent ranking (category importance × practice contribution) can use the rankings. The framework also facilitates internal benchmarking:

the current level of maturity can be benchmarked against the highest-ranked practices, to understand the gaps (e.g., lack of automated monitoring, lack of DLP coverage, lack of anomaly detection, etc.) and track the improvement. Overall, the table enables secure software development organizations to implement insider-threat prevention as a risk-driven, prioritized portfolio of AI-based controls, which can be deployed, measured and improved.

Example AI-Driven Agentic Framework in Action for Budget Allocation and Decision-Making.

Assume that an IT company, "SecureDev" has identified a couple of insider threat types (e.g., Malicious Insiders, Negligent Insiders, Credential Theft, etc.) that need to be focused on. The prioritization process of the framework allows the company to assess the AI-powered and/or agentic practices to invest in, based on their priority, influence and cost.

Suppose SecureDev has a budget of a maximum of $500,000 that it invests in improving its insider threat detection and prevention in the following fiscal year. The company needs to prioritize its investments using the framework.

1. Identification of Key Threats and Prioritization:
2. The FAHP-based prioritization results in the ranking of the most critical practices to address different types of insider threats. For example:

- For Malicious Insiders (IT-1), AI-P10 AI-Driven DLP (Data Loss Prevention) has the greatest weight of 0.012288, indicating that it is the most important practice to address this threat type.
- For Negligent Insiders (IT-2), AI-P13 Anomaly Detection is the highest ranking with the final weight of 0.019884, which means real-time anomaly detection is the number one priority in curbing the risks related to the negligent actions.
- For Credential Theft (IT-3), AI-P24 UEBA (User and Entity Behavior Analytics) wins the battle, with a final weight of 0.017781, which indicates that behavioral analytics is relevant to detecting compromised credentials.

3. Budget Allocation Decision:
4. With the final weights of the FAHP model, the organization can use the budget of $500,000 to assign the budget to the prioritized practices. For instance:

- AI-P10 (AI-Driven DLP) has a high priority and weight of 0.012288 to deal with Malicious Insiders. Considering the significance of this system in thwarting data exfiltration and safeguarding sensitive data, SecureDev may choose to invest $100,000 in deploying this system.
- Anomaly Detection (AI-P13) in Negligent Insiders is the second most important in priority. SecureDev invests in a system that can identify abnormal behavioral patterns on the fly at a cost of $90,000.
- In the case of Credential Theft, the organization would spend $80,000 on AI-P24 (UEBA) because behavioral analysis can significantly lower the chances of stolen or abused credentials.

The rest of the budget will be allocated to other high-priority practices according to their ranking and final weights to make sure that the resources could be concentrated on the most important areas.

5. Practical Demonstration:
6. The agentic framework is an AI-based tool, which assists SecureDev in making evidence-based decisions using expert input and empirical data to prioritize the practices. For example:

- SecureDev may consider launching pilot programs to pilot the AI-Driven DLP system within a certain department first (e.g., sensitive R&D teams). The pilot is also used to show the effectiveness of the practice and give real-time feedback on the performance of the system in data leak prevention.
- The system is implemented organization wide after it is successfully implemented in high-risk departments. SecureDev, therefore, minimizes the chances of malicious insider threats due to its ability to identify and prevent data theft at an early stage, which explains its initial budgetary allocation.

In a similar manner, the organization will be able to experiment with the Anomaly Detection and UEBA practices within a controlled setting, with historical data and simulated insider threats to assess the effectiveness of the practices in detecting risks and addressing incidents in real time.

7. Ongoing Assessment and Reallocation:
8. SecureDev continually tracks the performance of the AI-based solutions after the implementation. In case the Anomaly Detection system turns out to be extremely efficient when it comes to insignificant insider threats, SecureDev may choose to re-invest the money in the less effective practices (e.g., NLP to monitor mail) and invest further in the anomaly detection capabilities.

This practical example reflects how the AI-Driven Agentic Framework helps managers make decisions by providing a well-defined and data-driven approach to prioritizing AI-driven practices in combating insider threats. It provides empirical verification by allocating the budget to the practices based on the empirical ranking of practices. It will allow companies like SecureDev to use the budget in an optimum way as it will be targeted towards the most pressing practices that will significantly reduce the risk of insider threats. Its ability to prioritize

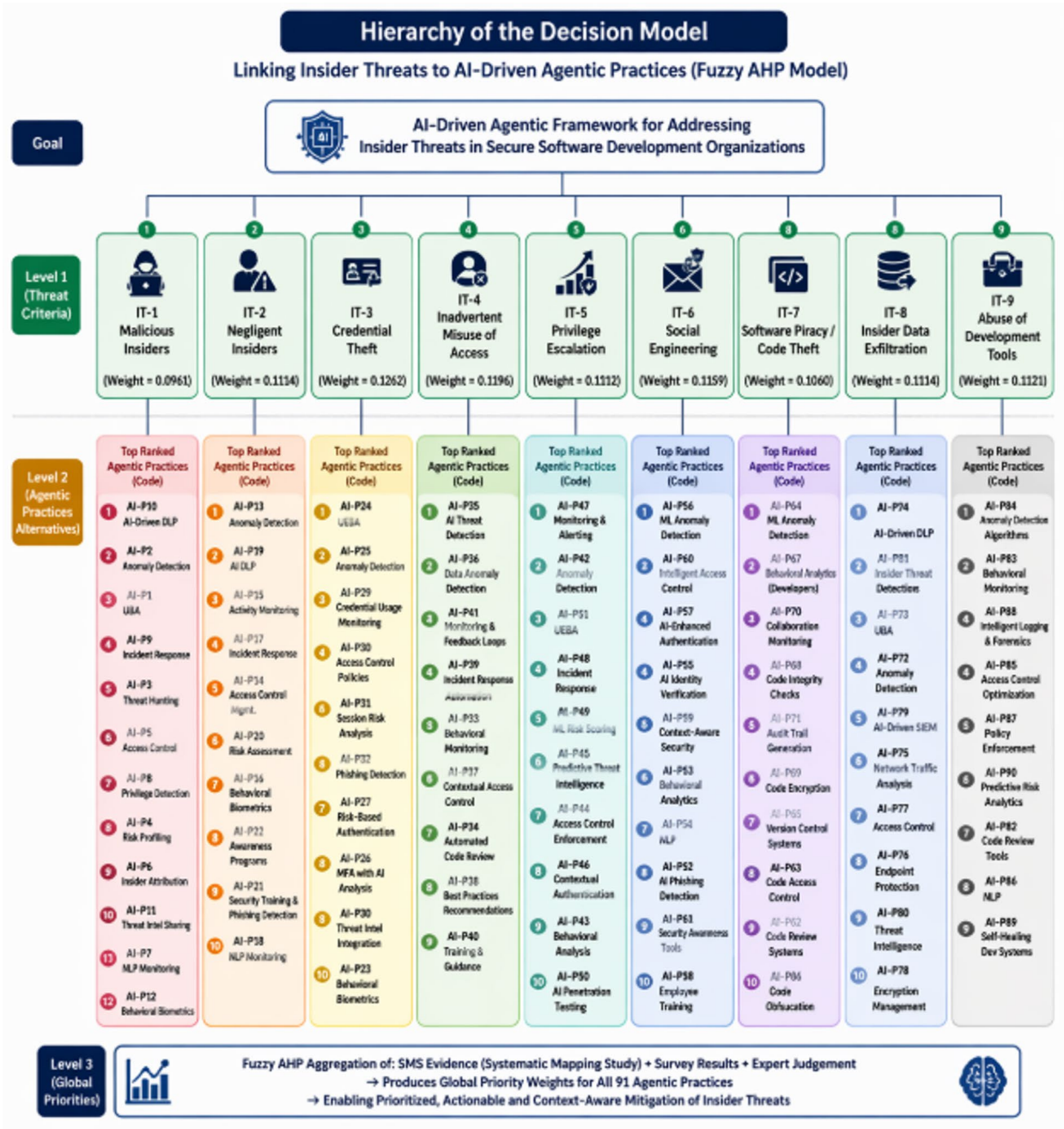

Fig. 11. Hierarchy of the Decision Model.

the practices and capability to match the security investments with the needs of the organization as well as its flexibility in real-world application all prove the effectiveness of the framework in decision making.

Figure 11 presents hierarchy presentation of the decision model.

Implications of the research

The proposed AI-based Agentic Framework of Prevention and Detection of Insider Threats in Secure Software Development Organizations through Fuzzy AHP (FAHP) has substantial theoretical, methodological, managerial, and practical implications in the domain of cybersecurity research, secure software engineering, and risk management of the organization. This study can lead to a more organized, evidence-based, and situational approach to combating insider threats in contemporary software development settings by prioritizing AI-driven practices systematically across the various categories of insider threats.

.

Theoretical implications

Theoretically, this study contributes to the body of literature related to insider threat management by involving AI-driven agentic systems and multi-criteria decision-making (MCDM) in uncertainty. Most of the previous research has interpreted insider threats as either being technical or behavioral issues, but this research provides a turnover between the two concepts by showing that implementing a socio-technical approach can be effective in mitigating an insider threat. With the help of FAHP, human subjectivity, uncertainty, and expert judgment can be modeled, which are frequently disregarded in conventional cybersecurity systems.

The results bring the current theories of secure software development into the context that AI is not just a defensive measure but an agentic layer, which learns, adapts, and makes real-time security decisions autonomously. This changes the conceptualization of cybersecurity to reactionary defense to proactive, predictive, and adaptive risk governance. Moreover, the study narrows the theoretical knowledge of the distinction of mitigation strategies that are necessary to counter the various types of insider threats, by classifying insider threats into nine types and linking them to prioritized AI-driven practices instead of applying the same solution to all.

Methodological Implications

In terms of methodology, the paper illustrates the usefulness of FAHP as a powerful decision making instrument in cybersecurity studies, especially in areas that are faced by ambiguity, subjectivity of the expert, and a lack of empirical agreement. The FAHP, compared to the conventional statistical ranking methods, adds uncertainty to the expert judgements and provides a more realistic picture of the uncertainty in the complex environment.

The framework also provides a replicable model for other cybersecurity researchers to explore other security areas, such as external threats, supply chain security, or AI-based threats. This research is the first of its kind in the domain of quantitative decision analysis and cybersecurity risk modelling, which incorporates FAHP and structured insider threat classification and AI practices.

Managerial and organizational implications

The framework is a strategic decision-making support for the organization leaders, cybersecurity managers and DevSecOps practitioners to prioritize security initiatives. The list can help companies prioritize the best investments, rather than simply taking AI tools on board, depending on the specific insider threats:

- Real-time anomaly detection, AI-based DLP and UEBA are the top priorities for malicious insiders and credential theft.
- Automated training, behavior monitoring and risk prediction are more likely to be useful for negligent insiders.
- Continuous monitoring, contextual access control and intelligent logging are important to mitigate privilege escalation and abuse of development tools.

This will enable the organizations to develop risk-based security plan, budgeting, and to convince the stakeholders the reasonableness of investing in AI technology based on empirical prioritization rather than intuitive.

Implications for SSD and DevSecOps

The findings have implications for SSD and DevSecOps. The framework suggests that AI-based agentic systems need to be incorporated into the software development lifecycle (SDLC) including:

- Planning Phase: Threat predictive modeling and Risk-based access control.
- Development Phase: AI-based code analysis, behavioral analytics, and insider risk modeling.
- Testing Phase: AI-based penetration testing and abnormality detection.
- Deployment Phase: Incident response and continuous monitoring.
- Maintenance Phase: Adaptive learning systems that adapt to new threats.

This makes the argument of AI-based DevSecOps, where security is not a peripheral task but a smart and autonomous part of software engineering processes strong.

Policy and governance implications

More broadly, the research informs cybersecurity policies and governance. The AI-driven practices are prioritized, which implies the need to:

- More powerful access governance structures,
- Constant surveillance of staff members accompanied by moral protection,
- AI responsibility and transparency in automated decision-making,
- AI-led incident response procedures.

The research also proposes that organizations need to create ethical AI governance policies to make sure that AI-driven surveillance does not infringe on employee privacy, laws on data protection as well as trust at work.

Practical implications for cybersecurity practitioners

For cybersecurity professionals, the framework offers a practical list of AI-based tools that match up to some of the insider threats. The rankings can be used to:

- Select suitable AI tools,
- Design specific training programs,

- Use customized detection systems,
- Strengthen incident response strategies.

Also, the paper highlights that technology itself is not sufficient; companies have to combine AI solutions with people-oriented security awareness initiatives to minimize the risks of negligent insiders.

Practical for future research

This study provides a number of research opportunities in the future, such as:

- Empirical validation of the FAHP rankings by real-life case studies,
- Comparison with other industries (e.g., banking, healthcare, defense),
- Learning to combine machine learning models with decision frameworks based on FAHP,
- Longitudinal research on the efficacy of AI-based insider threat mitigation over the long run.

Validations and limitations of the research

The validity of this study is defined by methodological rigor, triangulation of evidence and systematic application of FAHP and empirical findings. The proposed AI-based Agentic Framework of Insider Threats Prevention and Detection in SSD Organizations is validated at three important levels: methodological, analytical, and empirical.

First, to ensure methodology validity, an empirical survey of AI and cybersecurity experts, complemented with a systematic literature review is used. The literature review provided a comprehensive review of the relevant theory, defining the state-of-the-art AI-based approaches to prevent insider threats, while the empirical survey delivered empirical perceptions, experiences and priorities of the practitioners working in the secure software development environment. This form of mixed-methodology increases construct validity since it enables the congruence between theoretical and practical knowledge rather than being based on the same source of evidence.

Second, the use of FAHP as a powerful multi-criteria decision-making (MCDM) method justifies analytical validity. FAHP was chosen due to the inherent uncertainty, subjectivity, and linguistic judgments inherent in insider threat assessment as opposed to numerical data. Through fuzzy pairwise comparisons, the consistency checks and weight aggregation processes, the study systematically obtained local and global priority weights of AI-driven agentic practices in nine different categories of insider threats. The consistency ratios were consistent and within acceptable limits and hence expert judgments were logical and analytical.

Third, practical validity is reflected in the relevance and applicability of the ranked AI-driven practices to actual organization-level cybersecurity issues. The mentioned top practices, which include AI-based anomaly detection, behavioral analytics, automatic incident response, and AI-based data loss prevention, are closest to the security practices that are popular in modern software development companies. Besides, the regional analysis of the threat perception also confirms the versatility of the framework to the distinct geopolitical and organizational settings, implying that it could be used to inform both global and local cybersecurity policies.

Overall, synthesis of evidence-based literature, practitioner-initiated empirical validation, and rigorous FAHP analysis give a good foundation to the credibility, reliability, and generalizability of the suggested framework. As a result, this study provides a legitimate and practical decision support model to organizations, which aim to incorporate AI-driven agentic systems in the safe software development practice.

This study has various limitations although it has made some theoretical and practical contributions, which should be noted and taken into consideration when interpreting the findings. First, the proposed agentic framework based on AI is mostly conceptual and evaluative, based on the FAHP, where the priority of the mitigation practices is established, but not implemented and empirically tested within the context of real-world software development. Therefore, the feasibility of the suggested AI-based practices in practice within real-life organizations is yet to be verified by longitudinal or experimental research.

Second, FAHP is subjective by nature because it relies on expert opinions and comparisons between two, and therefore, may cause bias or inconsistency even when consistency checks are applied. Even though FAHP is best suited to deal with uncertainty, different groups of experts, which may influence the overall generalizability of the findings, may generate varying levels of weightings and rankings.

Third, the empirical survey sample size and composition might be a constraint to the generalizability of the results. The views of AI and cybersecurity specialists that were involved in this research might not be representative of all sectors, sizes of organizations, or geographic areas, especially small and medium-sized enterprises (SMEs) that might have different security capabilities and levels of threats.

Fourth, the research is more oriented to technical and AI-driven solutions, whereas the organizational, cultural and managerial aspect, including leadership, trust, employee morale, and security culture is not actively discussed. These organizational and human aspects are important in preventing insider threats and should be researched.

Finally, the framework fails to recognize the quick technological change, and emerging AI risks, such as adversarial machine learning, data poisoning, and abuse of generative AI tools by insiders. Future work will have to take into account these evolving threat landscapes and test the vulnerability of AI-driven security systems to sophisticated insider threats.

In conclusion, while this paper offers a systematic and methodological decision-making approach to rank the AI-based mitigation measures of insider threats, more empirical and practical evidence and applications and cross-disciplinary studies is required to enhance its effectiveness.

Conclusion and future research work directions

This paper developed and verified the use of an agentic approach using AI to prevent and detect insider threats in safe software development companies in response to a significant and emerging cybersecurity challenge.

The research has successfully incorporated artificial intelligence into agentic security and use of FAHP to offer methodical and systematic decision-making process to prioritize the mitigation measures under uncertainty and vagueness. The findings reveal that insider threats are not a uniform group; rather, they are very distinct in nature, risk level and outcomes with varying degrees of impact on the organization and therefore require specialized and contextual security measures.

The analysis based on the FAHP allowed a strict comparison of AI-driven practices regarding nine different types of insider threats, as it turned out that anomaly detection, AI-driven data loss prevention, continuous monitoring, and behavioral analytics were always among the most effective security measures. These results suggest that technical intelligence should be integrated with intelligent behaviour and organizational intelligence rather than using rule-based security controls. In addition, the comparative analysis of the literature and empirical perspectives of cybersecurity experts confirmed the practical use of agentic systems using AI and identified the gaps in the practical use of such systems that should be taken into account.

The framework is worth considering in terms of both theory and practice because it offers a way to link theoretical and practical applications of cybersecurity, enabling companies to have a scalable and adaptable approach to improve the secure use of software development. However, the study should be extended by the future research that may include the longitudinal data, case studies in the industry and further analysis of the contemporary use of AI to assess its performance in real-time. Overall, the study illustrates the need for proactive, smart and adaptive security designs in reducing insider threats and in making the existing software development environments resilient. This framework is a prioritization tool, not a deployed system, future work will test it in live DevSecOps pipelines.

We have addressed the six research questions (RQs) we posed to guide our research on preventing insider threats in secure software development (SSD) organizations from existing theories and practices. This integration not only will provide an analytical review of the current state of research but it also will provide the theoretical foundation for the framework proposed.

RQ1 — State-of-the-art of Insider-Threat Prevention in Secure Software Development.

The prevention of insider threats in SSD organizations has been increasingly of interest and some research has focused on human-centric security and organizational behaviours. The state-of-the-art practices involve a combination of technical and organizational solutions to detect and mitigate insider threats. Such practices include access control policy (e.g., role-based access control), behavioural analysis and AI-based anomaly detection. However, most of the current approaches focus on the technological aspect and the role of the organizational culture and awareness is neglected. The role of artificial intelligence (AI) and machine learning (ML) in the prevention of insider threats, particularly, by applying predictive modelling and real-time data monitoring, is an interesting step, but not sufficiently explored as part of the holistic approaches that include technical and human aspects. The insider intentions and behaviour can be justified by the theory of planned behaviour, where the behavioural processes that cause insider threats are linked to the control strategies of the organization.

RQ2 — Types of Insider Threats in Software Development Organizations.

The literature has found different types of insider threats based on the nature of their attacks, which can be divided into malicious insiders, negligent insiders and inadvertent misusers. The types of threats call for different preventative measures, but the distinction between these threats is not clear in the literature. Malicious insiders are the worst of all, who are either motivated by financial gain or revenge, and negligent insiders are due to either unawareness or lack of training and is, therefore, an accidental data leak. The addition of privilege escalation and credential theft as the primary channels of insider threats in SSD environments highlights the need for a multi-dimensional approach to detect insider threats. This can be explained by a social engineering theory that explains how internal actors can be manipulated using cognitive biases, which also introduces the variety of insider threats.

RQ3 — Methods and Technology Used by SSD Organizations.

The methods and technologies used for insider threat detection in SSD are varied and data loss prevention systems (DLP), behaviour analytics and machine-learning models are very popular. To identify possible threats in real-time, anomaly detection systems (e.g., UEBA) are widely employed to track the deviations in user behavior. In addition, access control and multi-factor authentication (MFA) are other features of the technological stack, which should be used to prevent unauthorized access. The aspect of incorporating these technologies as an integrated organizational strategy is however a challenge. The socio-technical systems theory is a prism through which one can interpret the relationships between technology and human behavior in the process of detection and prevention of insider threats, to make sure that the technical systems are in line with organizational objectives and staff activities.

RQ4 — Limitations of Current Insider Threat Prevention Strategies in SSD.

The insider threat prevention measures currently employed by the SSD organizations are ineffective because they are mostly fragmented and depend on technological solutions. Most of these strategies do not consider the organizational environment and human aspect that leads to insider threats. Indicatively, even though behavioral analytics can be used to identify the deviation in activity, the motivational components of the behaviors may not always be identified. Moreover, the economically and scalability of existing solutions may be in doubt because massive deployment of AI-based systems demands huge amounts of resources and expertise. The organizational behavior theory assists in examining these constraints, especially how organizational culture, leadership, and employee engagement affect the development of behaviors, which can lead to insider threats.

RQ5 — Key Metrics to Evaluate Insider Threat Prevention Measures in SSD.

Assessment of the effectiveness of insider threat prevention measures needs clear metrics to measure the technical performance of detection systems and the behavioral outcome in employees. False positive rates, detection accuracy, and incident response time are some of the common metrics used to evaluate the performance of AI-powered systems. Nevertheless, it is still difficult to assess the effect on the organizational

culture, including compliance and awareness among employees. The balanced scorecard model can be utilized to develop a holistic assessment model, which does not only emphasize on financial and operational performance, but also includes learning and growth indicators of employee involvement and employee training in the regard of insider threat mitigation.

RQ6 — Gaps Exist in Current Research on Insider Threat Prevention in SSD.

Although the literature concerning the insider threat prevention has increased, there are still a number of gaps, especially when it comes to combining AI with human-oriented solutions. The literature is mostly based on technical solutions, i.e. machine learning models of anomaly detection, yet it does not consider the organizational and psychological aspects of insider behavior. Moreover, the absence of longitudinal studies implies that the effectiveness of AI-based threat prevention strategies in the long term is unknown. The institutional theory points out that future research is needed on the role of organizational structures and institutional pressures in adoption and effectiveness of insider threat prevention systems. Moreover, interdisciplinary methods that are based on cybersecurity and psychology, as well as organizational studies, are underrepresented in the current research, yet they can offer more holistic solutions.

By integrating the literature on these six research questions, we will have an excellent theoretical foundation based on the existing theories including theory of planned behavior, social engineering theory, socio-technical systems theory and institutional theory. These models help us to understand the processes of any insider threats, and can provide us with a comprehensive view of how to assess and design better insider threat prevention solutions. The critical synthesis not only highlights the shortcomings and limitations of current research but also lays the foundation for the emergence of an AI agentic model that will consider both technical and human factors and will eventually lead to successful insider threat prevention in SSD.

Future research on an AI-based agentic framework for insider threat prevention and detection for secure software development organizations can benefit research and practice in several ways. First, empirical studies should evaluate the proposed approach in a large-scale, across multiple industries and regions, as insider-risk is not constant across cultures, regulatory regimes and software development tasks. Second, they should be tested in practice and over time in future research efforts to analyze the effectiveness of agentic systems over time, such as concept drift, new attack strategies, and changes to software development processes. Third, research should consider more behavioural and socio-technical factors such as patterns of developer interactions, DevOps, repository events, and communication metadata and deal with data with privacy in mind.

Fourth, privacy-conscious and compliance-compatible agentic surveillance needs to be designed, and this requires solutions like federated learning, differential privacy, secure enclaves, and policy-based data minimization to minimize ethical and legal risks. Fifth, further research must enhance explainability, accountability, and human-in-the-loop governance to help security teams learn how agents make decisions, explore alerts, and minimize over-automation and trust in AI outputs. Sixth, studies are needed to assess agentic frameworks in adversarial and insider-aware attack scenarios, such as poisoning, evasion, model inversion, and prompt-based manipulation of AI agents, to be resilient to adaptive insiders. Seventh, additional effort can be made on prioritization, by combining hybrid decision models (e.g., FAHP with DEMATEL, TOPSIS, or Bayesian models) and sensitivity analyses to enhance robustness and interpretation of ranking results. Lastly, future studies must create standard benchmarks, datasets, and evaluation procedures that are specific to secure software development settings, such as open-source DevSecOps testbeds or reproducible metrics (precision/recall, time-to-detect, false-positive cost, and operational burden), to enable the comparison of competing solutions in a consistent and transparent way.

Data availability

All data generated or analyzed during this study are included in this manuscript. If someone wants to request the data from this study, kindly contact **Rafiq Ahmad Khan** (rafiqahmadk@gmail.com).

Received: 1 February 2026; Accepted: 23 July 2026
Published online: 05 August 2026

References

1. Al-Hashimi, H. A. A generative AI cybersecurity risks mitigation model for code generation: using ANN-ISM hybrid approach. *Scientific Reports*, **16**, 1–51 (2026).
2. Alghamdi, D. A., Niazi, M., Cordeiro, L. C., Humayun, M. & Stewart, A. Mitigating Insider Threats: Insights from Software Security Experts for Process Improvement and Risk Reduction, in Proceedings of the 2025 29th International Conference on Evaluation and Assessment in Software Engineering Companion, 41–48. (2025).
3. Zacharis, A., Katos, V. & Patsakis, C. Integrating AI-driven threat intelligence and forecasting in the cyber security exercise content generation lifecycle. *International J. Inform. Security*, **23** (4), 2691–2710 (2024).
4. Ye, X. et al. Research on insider threat detection based on personalized federated learning and behavior log analysis. *Scientific Reports*, **15**, 1–17 (2025).
5. Inayat, U. et al. Insider threat mitigation: Systematic literature review. *Ain Shams Eng. Journal*, **15** (12), 1–18 (2024).
6. Casola, V., De Benedictis, A., Mazzocca, C. & Orbinato, V. Secure software development and testing: A model-based methodology. *Computers & Security*, **137**, 103639 (2024).
7. Humayun, M., Niazi, M., Assiri, M. & Haoues, M. Secure Global Software Development: A Practitioners' Perspective, *Applied Sciences*, **13** (4), 2465 (2023).
8. Abou Ali, M., Dornaika, F. & Charafeddine, J. Agentic AI: a comprehensive survey of architectures, applications, and future directions. *Artificial Intell. Review*, **59** (1), 1–37 (2025).
9. Ayouni, S. et al. Exploring the synergistic collaboration of Human Agentic-AI in enhancing the security of the software development lifecycle. *Egyptian Inf. Journal*, **33**, 100915 (2026).
10. Vaiyapuri, T. et al. Automated cyberattack detection using optimal ensemble deep learning model. *Trans. Emerg. Telecommunications Technol.*, **35** (4), 1–16 (2024).

11. Rafiq, A. et al. A Fuzzy-AHP Decision-Making Framework for Optimizing Software Maintenance and Deployment in Information Security Systems. *J. Software: Evol. Process.*, **37** (1), 1–46 (2025).
12. Khalid, A. et al. A SWOT Analysis of Software Development Life Cycle Security Metrics. *J. Software: Evol. Process.*, **37** (1), 1–37 (2025).
13. d'Ambrosio, N., Lista, C., Perrone, G. & Romano, S. P. SMASH: An SDN-MTD framework for efficient honeypot deployment and insider threat mitigation. *Computer Networks*, **269**, 111327 (2025).
14. Racherache, B., Shirani, P., Soeanu, A. & Debbabi, M. CPID: Insider threat detection using profiling and cyber-persona identification. *Computers & Security*, **132**, 103350 (2023).
15. Mohammed, Y., Warkentin, M. & Beshah, T. Cultural drivers behind employees neutralizing deviant information systems security behaviors. *Journal Knowl. Management*, **29** (4), 1129–1161 (2025).
16. Roy, P., Sengupta, A. & Mazumdar, C. A Structured Control Selection Methodology for Insider Threat Mitigation. *Procedia Comput. Science*, **181**, 1187–1195, (2021).
17. Khan, H. U. et al. AI-driven cybersecurity framework for software development based on the ANN-ISM paradigm. *Scientific Reports*, **15** (1), 1–45 (2025).
18. Alhashimi, H. A. et al. Exploring the Role of Generative AI in Enhancing Cybersecurity in Software Development Life Cycle. *Array* **28**, 1–26 (2025).
19. Alwageed, H. S. & Khan, R. A. The Role of Generative AI in Strengthening Secure Software Coding Practices: A Systematic Perspective, in Proceedings of the 2025 29th International Conference on Evaluation and Assessment in Software Engineering Companion, 136–141. (2025).
20. Bin Sarhan, B. & Altwaijry, N. Insider Threat Detection Using Machine Learning Approach. *Appl. Sci.* **13** (1), 259 (2023).
21. Radanliev, P., Roure, D. & Santos, O. Red Teaming Generative AI/NLP, the BB84 Quantum Cryptography Protocol and the NIST-Approved Quantum-Resistant Cryptographic Algorithms. *SSRN Electron. Journal*, **01/01**, 1–30 (2023).
22. Al Hammadi, A. Y. et al. Explainable artificial intelligence to evaluate industrial internal security using EEG signals in IoT framework. *Ad Hoc Networks*, **123**, 102641 (2021).
23. Allam, H. Code Meets Intelligence: AI-Augmented CI/CD Systems for DevOps at Scale. *Int. J. Artif. Intell. Data Sci. Mach. Learn.* **6**, 137–146. 03/17 https://doi.org/10.63282/3050-9262.IJAIDSML-V6I1P115 (2025).
24. Alzahrani, A. & Khan, R. A. Secure software design evaluation and decision making model for ubiquitous computing: A two-stage ANN-Fuzzy AHP approach. *Computers Hum. Behavior*, **108109**, 1–20 (2023).
25. Khan, R. A., Khan, S. U., Alzahrani, M. & Ilyas, M. Security Assurance Model of Software Development for Global Software Development Vendors. *IEEE Access.* **10**, 58458–58487 (2022).
26. Rafiq, A. et al. Securing Software Development Through People Maturity: A Fuzzy-AHP Decision-Making Framework. *J. Software: Evol. Process.*, **37** (9), 1–9 (2025).
27. Daim, T., Yalcin, H., Mermoud, A. & Mulder, V. Exploring cybertechnology standards through bibliometrics: Case of National Institute of Standards and Technology. *World Patent Information*, **77**, 102278 (2024).
28. Dosh, M. Detecting insider threat within institutions using CERT dataset and different ML techniques. *Periodicals Eng. Nat. Sci. (PEN)*. **9**, 873–884 (2021). 05/31.
29. McIntosh, T. R. et al. From COBIT to ISO 42001: Evaluating cybersecurity frameworks for opportunities, risks, and regulatory compliance in commercializing large language models. *Computers & Security*, **144**, 103964, (2024).
30. Idensohn, C., Flowerday, S., van der Schyff, K. & Chua, Y. T. Malicious insider threats in cybersecurity: A fraud triangle and Machiavellian perspective. *Computers Hum. Behavior*, **174**, 108809 (2026).
31. Soner, O., Kayisoglu, G., Bolat, P. & Tam, K. Risk sensitivity analysis of AIS cyber security through maritime cyber regulatory frameworks. *Applied Ocean. Research*, **142**, 103855 (2024).
32. Khadidos, A. O. et al. CyberSentry: Enhancing SCADA security through advanced deep learning and optimization strategies. *International J. Crit. Infrastructure Protection*, **50**, 1–23 (2025).
33. Mantha, B., de García, B. & Karri, R. Cyber security threat modeling in the AEC industry: An example for the commissioning of the built environment. *Sustainable Cities Society*, **66**, 102682 (2021).
34. Nautiyal, L. & Rashid, A. A framework for mapping organisational workforce knowledge profile in cyber security. *Computers & Security*, **145**, 103925 (2024).
35. Awan, M., Alam, A., Khan, R. A., Alwageed, H. S., Ayouni, S. & Almagrabi, A. O. A generative AI-driven cybersecurity framework for small and medium enterprises software development: an ANN-ISM approach. *Scientific Reports*, **16**, 1–41 (2026).
36. Joshi, S. A Review of Generative AI and DevOps Pipelines: CI/CD, Agentic Automation, MLOps Integration, and LLMs. *Int. J. Innovative Res. Comput. Sci. Technol.* **13** (1-14), 0701 (2025).
37. Souppaya, M., Scarfone, K. & Dodson, D. *Secure Software Development Framework (SSDF) Version 1.1: (Draft* (Recommendations for Mitigating the Risk of Software Vulnerabilities, 2022).
38. Khan, R. A., Khan, S. U., Akbar, M. A. & Alzahrani, M. Security risks of global software development life cycle: Industry practitioner's perspective. *J. Software: Evol. Process.* **36** (3), 1–34 (2023).
39. Binbeshr, F. & Imam, M. Comparative Analysis of AI-Driven Security Approaches in DevSecOps: Challenges, Solutions, and Future Directions, in Proceedings of the 2025 29th International Conference on Evaluation and Assessment in Software Engineering Companion, 142–151. (2025).
40. Olaniyan, R., Rakshit, S. & Vajjhala, N. *Application of User and Entity Behavioral Analytics (UEBA) in the Detection of Cyber Threats and Vulnerabilities Management* (Computational Intelligence for Engineering and Management Applications, 2023).
41. Y. Song, and J. Yuan, Insider Threat Detection Based on User and Entity Behavior Analysis with a Hybrid Model, in Information Security: 27th International Conference, ISC 2024, Arlington, VA, USA, October 23–25, 2024, Proceedings, Part II,Arlington, VA, USA, 2024, 323–340.
42. Kamatchi, K. & Uma, E. Insights into user behavioral-based insider threat detection: systematic review. *Int. J. Inf. Secur.* **24** (2), 88 (2025). 2025/03/07.
43. Feng, W. et al. Multi-Granularity User Anomalous Behavior Detection. *Appl. Sci.* **15** (1), 128 (2025).
44. Jiang, Y., Pang, P. C. I., Wong, D. & Kan, H. Y. Natural Language Processing Adoption in Governments and Future Research Directions: A Systematic Review. *Appl. Sci.* **13** (22), 12346 (2023).
45. Ali, G., Shah, S. & ElAffendi, M. Enhancing cybersecurity incident response: AI-driven optimization for strengthened advanced persistent threat detection. *Results Engineering*, **25**, 104078 (2025).
46. Adabara, I., Olaniyi Sadiq, B., Nuhu Shuaibu, A., Ibarahim Danjuma, Y. & Venkateswarlu, M. A Review of Agentic AI in Cybersecurity: Cognitive Autonomy, Ethical Governance, and Quantum-Resilient Defense, *F1000Research*, 14, (2025).
47. Meland, P. H., Tokas, S., Erdogan, G., Bernsmed, K. & Omerovic, A. A Systematic Mapping Study on Cyber Security Indicator Data. *Electronics* **10** (9), 1092 (2021).
48. Tariq, M. I. et al. Prioritization of Information Security Controls through Fuzzy AHP for Cloud Computing Networks and Wireless Sensor Networks. *Sensors* **20** (5), 1310 (2020).
49. Souppaya, M., Scarfone, K. & Dodson, D. *Secure Software Development Framework (SSDF) Version 1.1: (Draft): Recommendations for Mitigating the Risk of Software Vulnerabilities*, (2021).
50. Petersen, K., Vakkalanka, S. & Kuzniarz, L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Inform. Softw. Technol.* **64**, 1–18 (2015).

51. Khan, R. A., Khan, S. U., Khan, H. U. & Ilyas, M. Systematic Mapping Study on Security Approaches in Secure Software Engineering. *IEEE Access.* **9**, 19139–19160 (2021).
52. Khan, R. A., Khan, S. U., Ilyas, M. & Idrees, M. Y. *The State of the Art on Secure Software Engineering: A Systematic Mapping Study* Vol. –20, 1–6 (Trondheim, 2020).
53. Garousi, V., Felderer, M. & Mäntylä, M. V. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information Softw. Technology*, **106**, 101–121, (2019).
54. Ali, S., Abbasi, I. A., Mustafa, E. E., Wahid, F. & Huang, J. Practitioner's view of the success factors for software outsourcing partnership formation: an empirical exploration. *Empirical Softw. Engineering*, **27** (2), 1–52, (2022).
55. Ali, S., Huang, J., Khan, S. U. & Li, H. A framework for modelling structural association amongst barriers to software outsourcing partnership formation: An interpretive structural modelling approach, *Journal of Software: Evolution and Process*, 32 (6, e2243, (2020).
56. Sattler, P. & Rosenbaum, M. Choice of the hypothesis matrix for using the Anova-type-statistic. *Statistics & Probab. Letters*, **219**, 110356 (2025).
57. Akbar, M. A. et al. Identification and prioritization of DevOps success factors using fuzzy-AHP approach. *Soft Computing*, **27** (4), 1–24 (2023).
58. Chou, T. Y. & Chen, Y. T. Applying Fuzzy AHP and TOPSIS Method to Identify Key Organizational Capabilities. *Mathematics* **8** (5), 836 (2020).
59. Ayhan, M. A Fuzzy AHP Approach for Supplier Selection Problem: A Case Study in a Gear Motor Company. *Int. J. Managing Value Supply Chains.* **4**, 1009 (2013).
60. Kiran, Mehta & Sharma, R. Unraveling Quantum Complexity: A Fuzzy AHP Approach to Understanding Software Industry Challenges. *Deep Learn. Tools Predicting Stock Market Movements*, **2**, 39–60 (2024).
61. Kumar, P., Brar, P., Singh, D. & Bhamu, J. Fuzzy AHP approach for barriers to implement LSS in the context of Industry 4.0. *Int. J. Productivity Perform. Manage.* **72**, 0608 (2022).
62. Mızrak, F. Integration of Fuzzy AHP for Cybersecurity Strategy Development in International Organizations. *Premium E-J. Social Sci.* **7** (35), 1272–1292 (2023).
63. Yaghoobi, T. Prioritizing key success factors of software projects using fuzzy AHP, *Journal of Software: Evolution and Process*, 30, e1891, 09/11, (2017).
64. Song, Y. & Yuan, J. Insider Threat Detection Based on User and Entity Behavior Analysis with a Hybrid Model. *Information Security* **Conference paper**, 323–340 (2024).
65. Esparza, J. Understanding the credential theft lifecycle, *Computer Fraud & Security*, 6–9, 02/01, 2019. (2019).
66. Lee, T. L., Liu, I. H., Li, C. F. & Li, J. S. Threats Hidden in Office Network: Mechanism of Credential Harvesting for Lateral Movement. *Journal Rob. Netw. Artif. Life*, **8** (3), 193–196 (2021).
67. Biringa, C. & Kul, G. Detecting Hard-Coded Credentials in Software Repositories via LLMs, *Digital Threats*, 6 (3), Article 12, (2025).
68. Mousavi, Z., Islam, C., Babar, M. A., Abuadbba, A. & Moore, K. Detecting Misuse of Security APIs: A Systematic Review. *ACM Comput. Surv*, **57** (12), 1–39 (2025).
69. Pecka, N., Othmane, L. B. & Valani, A. Privilege Escalation Attack Scenarios on the DevOps Pipeline Within a Kubernetes Environment, in Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering, Pittsburgh, PA, USA, 45–49. (2022).
70. Shen, L. et al. Risk Measurement Method for Privilege Escalation Attacks on Android Apps Based on Process Algebra. *Information* **11** (6), 293 (2020).
71. El-Zawawy, M. A. & Hamdy, A. Detection of hidden privilege escalations in android. *Automated Softw. Engineering*, **32** (2), 1–68 (2025).
72. Schmitt, M. & Flechais, I. Digital deception: generative artificial intelligence in social engineering and phishing. *Artif. Intell. Rev.* **57** (12), 324 (2024). 2024/10/12.
73. Waelchli, S. & Walter, Y. Reducing the risk of social engineering attacks using SOAR measures in a real world environment: A case study. *Computers & Security*, **148**, 104137 (2025).
74. Liao, Z., Nazir, S., Hussain, A., Khan, H. U. & Shafiq, M. Software Piracy Awareness, Policy, and User Perspective in Educational Institutions, *Scientific Programming*, **2020**,14 (2020).
75. Al-Mhiqani, M. N. et al. A new intelligent multilayer framework for insider threat detection. *Computers Electr. Engineering*, **97**, 107597 (2022).
76. Venson, E., Clark, B. & Boehm, B. The effects of required security on software development effort. *Journal Syst. Software*, **207**, 111874 (2024).
77. Parkinson, S. & Khan, S. A Survey on Empirical Security Analysis of Access-control Systems: A Real-world Perspective. *ACM Comput. Surv*, **55** (6), 1–28 (2022).
78. Mileff, P. Role of user activity monitoring in rpa processes. *Prod. Syst. Inform. Eng.* **11**, 27–42 (2023). 10/04.
79. Herrera Montano, I. et al. Survey of Techniques on Data Leakage Protection and Methods to address the Insider threat. *Cluster Computing*, **25** (6), 4289–4302 (2022).
80. Feng, Z., Kimura, K., Trinkenreich, B., Sarma, A. & Steinmacher, I. Guiding the way: A systematic literature review on mentoring practices in open source software projects. *Information Softw. Technology*, **171**, 107470 (2024).
81. Qiqieh, I., Alzubi, O., Alzubi, J., Sreedhar, K. C. & Al-Zoubi, A. M. An intelligent cyber threat detection: A swarm-optimized machine learning approach. *Alexandria Eng. Journal*, **115**, 553–563 (2025).
82. Tan, Y. et al. A root privilege management scheme with revocable authorization for Android devices. *Journal Netw. Comput. Applications*, **107**, 69–82 (2018).
83. Benedetti, G., Verderame, L. & Merlo, A. A Preliminary Study of Privilege Life Cycle in Software Management Platform Automation Workflows. 21–28.
84. Gnanasekaran, V., Fatima, U., Glas, M. & Heegaard, P. E. A Model-Based Framework for Developing Security-Safety Incident Response Plans. *International J. Inform. Security*, **24** (6), 1–23 (2025).
85. Al-Mhiqani, M. N. et al. Insider threat detection in cyber-physical systems: a systematic literature review. *Computers Electr. Engineering*, **119**, 109489 (2024).
86. Fei, K. et al. LaAeb: A comprehensive log-text analysis based approach for insider threat detection. *Computers & Security*, **148**, 1–17 (2025).
87. Hong, W. et al. A graph empowered insider threat detection framework based on daily activities. *ISA Transactions*, **141**, 84–92 (2023).
88. Alabdulkreem, E., Alduailij, M., Alduailij, M. & Mansour, R. F. Optimal weighted fusion based insider data leakage detection and classification model for Ubiquitous computing systems. *Sustainable Energy Technol. Assessments*, **54**, 102815 (2022).
89. Xiao, H. et al. Unveiling shadows: A comprehensive framework for insider threat detection based on statistical and sequential analysis. *Computers & Security*, **138**, 1–13 (2023).
90. Herrera Montano, I. et al. SecureMD5: A new stream cipher for secure file systems and encryption key generation with artificial intelligence. *Computer Stand. & Interfaces*, **95**, 1–14 (2026).
91. Hameed, K., Maqsood, F. & Wang, Z. Artificial intelligence-enhanced zero-knowledge proofs for privacy-preserving digital forensics in cloud environments. *Journal Netw. Comput. Applications*, **243**, 104331 (2025).

92. Jaiswal, A., Dwivedi, P. & Dewang, R. K. Handling imbalance dataset issue in insider threat detection using machine learning methods. *Computers Electr. Engineering*, **120**, 109726 (2024).
93. Álvarez, M. D. et al. Design and generation of a dataset for training insider threat prevention and detection models: The SPEDIA dataset. *Computers & Security*, **161**, 1–10 (2026).
94. Ali, B. & Chen, G. Next-generation AI for advanced threat detection and security enhancement in DNS over HTTPS. *Journal Netw. Comput. Applications*, **244**, 104326 (2025).
95. Yuan, S. & Wu, X. Deep learning for insider threat detection: Review, challenges and opportunities. *Computers & Security*, **104**, 102221 (2021).
96. Janjua, F., Masood, A., Abbas, H. & Rashid, I. Handling Insider Threat Through Supervised Machine Learning Techniques. *Procedia Comput. Science*, **177**, 64–71 (2020).
97. Al Noman, A., Idowu, S. O., Kakanou, R. D. M., Ciancarini, P. & Ren, M. An AI-Based Security Analysis Solution for DevSecOps, *Proceedings of the International Symposium on Intelligent Computing and Networking*. 274–290. (2025).
98. Alenezi, M. & Akour, M. AI-Driven Innovations in Software Engineering: A Review of Current Practices and Future Directions. *Appl. Sci.* **15** (3), 1344 (2025).
99. Almalki, S. S. AI-Driven Decision Support Systems in Agile Software Project Management: Enhancing Risk Mitigation and Resource Allocation. *Systems* **13** (3), 208 (2025).
100. Aloupogianni, E. et al. AI-Driven Optimization of Small Cell Deployment for Beyond 5G Networks. *Procedia Comput. Science*, **238**, 908–913 (2024).
101. Chinnasamy, P. et al. AI-Driven intrusion detection and prevention systems to safeguard 6G networks from cyber threats. *Scientific Reports*, **15** (1), 1–26 (2025).
102. Khawar, M. et al. Shaping the future of cybersecurity: The convergence of AI, quantum computing, and ethical frameworks for a secure digital era. *Computer Sci. Review*, **60**, 100882 (2026).
103. Kumar, M. S., Harsha, B. K. & Leta Jule, T. 4 AI-driven cybersecurity modeling using quantum computing for mitigation of attacks in IOT-SDN network, *Quantum-Safe Cryptography Algorithms and Approaches: Impacts of Quantum Computing on Cybersecurity*, 37–48: De Gruyter, (2023).
104. Kummarapurugu, C. S. A Framework for Real-Time AI-Driven Secure Code Analysis Integrated with DevSecOps in Cloud-Native CI/CD Pipelines, *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 08, 11/10, (2024).
105. Maria Thason, J. R. & DevSecOps, A. I. D. Advancing Security and Compliance in Continuous Delivery Pipelines. *Int. Res. J. Adv. Eng. Manage. (IRJAEM)*. **3**, 1666–1673 (2025). 05/05.
106. Mohamed, N. Artificial intelligence and machine learning in cybersecurity: a deep dive into state-of-the-art techniques and future paradigms. *Knowledge Inform. Systems*, **67** (8), 6969–7055 (2025).
107. Rangaraju, S., Ness, S. & Dharmalingam, R. Incorporating AI-Driven Strategies in DevSecOps for Robust Cloud Security. *Int. J. Innovative Sci. Res. Technol.* **8**, 2359–2365 11/01. (2023).
108. Sarker, I. H., Furhad, M. H. & Nowrozy, R. Cybersecurity: An Overview, Security Intelligence Modeling and Research Directions. *SN Comput. Sci.* **2** (3), 18 (2021).
109. Wen, S. F., Shukla, A. & Katt, B. Artificial intelligence for system security assurance: A systematic literature review. *International J. Inform. Security*, **24** (1), 1–43 (2024).
110. Yulianto, S. & Ngo, G. N. C. Enhancing DevSecOps Pipelines with AI-Driven Threat Detection and Response. 1–8.
111. Yury, K. Advancing operational efficiency in software companies through generative AI. *Am. J. Eng. Technol.* **7** (01), 11–18 01/20 (2025).
112. Chang, D. Y. Applications of the extent analysis method on fuzzy AHP. *Eur. J. Oper. Res.* **95** (3), 649–655 1996/12/20/ (1996).
113. Zadeh, L. A. Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems. *Adv. Fuzzy Syst. — Appl. Theory*. **6**, 1–840 (1996).
114. Liu, W., Qi, J., Jin, Y., Zhou, Z. & Zhang, X. Application of fuzzy analytic hierarchy process – multi-layer fuzzy inference system in product design evaluation. *Journal Intell. & Fuzzy Systems*, **45**, 1–24 (2023).
115. Zheng, W., Deng, P., Gui, K. & Wu, X. An Abstract Syntax Tree based static fuzzing mutation for vulnerability evolution analysis. *Information Softw. Technology*, **158**, 107194 (2023).
116. Barbara, A. K., Budgen, B. D. & Brereton, O. P. Using mapping studies as the basis for further research - A participant-observer case study. *Inform. Softw. Technol.* **53** (6), 638–651 (2011).

Acknowledgements

Acknowledgment: The authors acknowledge the use of generative AI tools, including ChatGPT and Grammarly, a writing assistant tool, which were instrumental in addressing writing issues in this paper.

Author contributions

Author ContributionsMuhammad Shafiq conceptualized the study, designed the overall research framework, and led the development of the AI-driven agentic architecture. Mujtaba Awan contributed to the methodological design, particularly the integration and application of the Fuzzy AHP approach, and supported data analysis. Hathal Salamah Alwageed contributed to the theoretical grounding, literature review, and validation of the proposed framework from a cybersecurity perspective. Umar assisted in data collection, model evaluation, and interpretation of results. Mohammad Mehidi Hassan provided critical insights into software security, reviewed the analytical approach, and contributed to refining the manuscript. Rafiq Ahmad Khan supervised the research, provided strategic guidance, and critically reviewed the manuscript for intellectual content. All authors reviewed, revised, and approved the final version of the manuscript.M Anwar Hossain played a critical role in addressing the major reviewer comments by significantly refining the conceptual framework, strengthening the methodological rigor, and improving the overall structure and clarity of the paper. His input was particularly valuable in enhancing the discussion on the agentic AI framework, incorporating relevant literature, and ensuring that the arguments were more coherent and technically sound. Given the depth of his involvement and the impact of his contributions on the quality and completeness of the revised manuscript, his inclusion as an author is both appropriate and justified.

Funding

The authors are thankful to the Research Department, Shandong Xiehe University, Jinan (China). The authors are also grateful to King Saud University, Riyadh, Saudi Arabia for funding this work through Ongoing Research Funding Program (ORF-2026-18).

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-026-64146-y.

Correspondence and requests for materials should be addressed to M.S., M.M.H., R.A.K. or M.A.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2026