AI-Enhanced Secure Software Engineering: A Focus on Explainable AI (XAI) Techniques

Holley Hudson
S22009650
North Wales Management School
Wrexham Glyndŵr University
Wales, UK
S22009650@mail.glyndwr.ac.uk

***Abstract*—The integration of Artificial Intelligence (AI) in secure software engineering is transforming traditional practices by automating critical tasks such as vulnerability detection and code analysis. However, concerns over the opacity of AI-driven decisions hinder trust and widespread adoption, particularly in security sensitive environments. This research investigates how Explainable AI (XAI) can enhance transparency, trust, and compliance in AI-powered secure coding practices, while addressing emerging cyber threats.**

The study explores the limitations of traditional secure coding methods, such as manual code reviews and static analysis tools, in handling advanced and large-scale vulnerabilities. AI tools, including GitHub CoPilot and automated vulnerability scanners, offer enhanced detection capabilities but introduce challenges related to integration and transparency. XAI techniques, such as SHAP and LIME, are critical for providing explanations for AI-driven decision, ensuring compliance with security standards like ISO/IEC 2701 and regulatory frameworks such as GDPR.

Through qualitative and quantitative analysis, this research highlights the effectiveness of XAI in improving trustworthiness and transparency in secure coding. However, significant challenges remain, including integration into Continuous Integration/Continuous Deployment (CI/CD) pipelines and overcoming technical and ethical barriers. This dissertation offers recommendations for implementing AI and XAI tools in secure software development while maintaining compliance with industry standards and addressing emerging security threats.

Keywords—Explainable AI, secure software engineering, vulnerability detection, CI/CD, transparency, security standards, AI compliance

I. Introduction

The integration of Artificial Intelligence (AI) in secure software development represents a significant change, automating labour-intensive processes, such as manual code reviews, enhancing threat detection, and providing complex vulnerability analysis. While traditional methods like manual code reviews have been foundational in ensuring secure coding, they often struggle with scalability, accuracy, and the ability to adapt to complex and emerging threats. However, the integration of complex AI tools like GitHub Copilot and automated scanning technologies often leads to concerns regarding transparency and trust, especially when developers are unable to fully understand AI-generated decision. This is where AI tools, combined with Explainable AI (XAI) techniques, come into play. Explainable AI (XAI) addresses this gap by ensuring AI-driven decisions are transparent, understandable, and interpretable. This enhances trust, promotes adoption, and supports alignment with security standards and legal frameworks. Moreover, integrating AI-enhanced practices into secure coding requires careful consideration of compliance with established regulatory frameworks such as ISO/IEC 27001, National Institute of Standards and Technology (NIST), and General Data Protection Regulation (GDPR), all while maintaining the effectiveness and legal compliance of security solutions [1] [2] [3].

This research aims to evaluate how AI tools, particularly those using XAI, enhance secure software development practices. It will explore the limitations of traditional methods, the integration challenges of AI into Continuous Integration/ Continuous Deployment (CI/CD) pipelines, and the legal and regulatory implications of using AI in secure software engineering. The study also seeks to provide recommendations for the effective implementation of AI-enhanced secure coding standards, with XAI at the core of these advancements.

A. Background

AI tools like GitHub CoPilot and advanced automated scanning technologies, such as Static Application Security Testing (SAST) and Dynamic Application Security Testing (DAST), have shown potential in enhancing secure coding practices, especially when used together [4] [5] [6] [7]. These tools offer real-time assistance in identifying vulnerabilities, which traditional methods, like manual code reviews, struggle to address efficiently at scale.

While manual code reviews have been the foundation of secure software engineering, they are increasingly insufficient against modern and complicated threats [5] [8]. For instance, an empirical study on two large open-source projects (OpenSSL and PHP) revealed unresolved or unacknowledged security-related coding weaknesses due to developer disagreement [8]. The study also found that certain critical vulnerabilities were overlooked, suggesting that human error, especially within large-scale software systems, plays a role in security lapses. Another study demonstrated that peer code reviews, while effective to an extent, leave many vulnerabilities undetected, highlighting a gap between the intended outcomes of these reviews and their real-world effectiveness [9]. This suggests that manual reviews alone cannot scale to meet the demands of modern software security. AI-powered tools, such as GitHub Copilot, provide fast and context-aware code suggestions. However, they also come with risks, where lack of human oversight may introduce new vulnerabilities within these AI-generated code snippets [6] [10] [11]. While these tools show superior

accuracy in controlled settings, their practical performance in real world environments often falls short [6]. The challenges lie not only in their effectiveness but also in their integrations into existing CI/CD pipelines and the lack of transparency in AI-driven decisions. This is where XAI plays a crucial role. Developers often find it difficult to trust AI decisions because AI models can be opaque and complex. XAI techniques like Shapley Additive Explanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), make it possible for developers to understand and trust AI-generated outputs, enhancing transparency and facilitating broader adoption of AI in secure software engineering [12] [13] [14]. Despite these advancements, challenges remain in integration XAI tools with existing security practices and ensuring compliance with industry standards, a gap this research seeks to address.

B. Research Aims and Objectives

This research aims to evaluate how the integration of AI tools, particularly XAI techniques, enhances secure coding practices within secure software engineering and aligns with current security standards and regulatory frameworks. The evaluation of these aims is supported by identification of research objectives.

- To explore the limitations of traditional secure coding practices in addressing emerging and complex cyber threats.
- To assess the effectiveness of AI tools, such as GitHub Copilot and automated scanning technologies, in improving secure coding practices.
- Will investigate how XAI enhances transparency and trust in AI-driven decisions within secure software development.
- Will analyse how XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines, and the associated regulatory considerations.
- Aims to identify the primary challenges in integrating XAI tools into secure coding processes and to propose solutions to overcome these challenges.

II. RESEARCH HYPOTHESIS AND QUESTIONS

The integration of XAI tools into secure coding practices within secure software engineering improves the transparency and trustworthiness of AI-driven decisions. However, technical professionals perceive significant challenges related to integrating these tools into existing CI/CD pipelines and ensuring alignment with current security standards and regulatory frameworks.

The following questions have been designed to investigate the hypothesis:

Main Research Question:

- How can the integration of AI tools, particularly Explainable AI (XAI), enhance secure coding practices and align with security standards?

Sub-questions:

- What are the limitations of traditional secure coding practices in addressing emerging and complex cyber threats?
- How effective are AI tools, such as GitHub Copilot and automated scanning technologies, in improving secure coding practices, and how does XAI enhance transparency and trust in AI-driven decisions?
- How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines, and what are the regulatory considerations?
- What are the primary challenges in integrating XAI tools into secure coding processes, and what solutions can help overcome these challenges?

III. LITERATURE REVIEW

A. Overview of Secure Coding Practices

Secure coding is a fundamental aspect of software development, aimed at mitigating the risk of cyber threats by ensuring software resistance to unguarded vulnerabilities. It involves best practices, guidelines, and tools that developers use to identify, assess, and mitigate potential security risks at code level. These practices are essential to application security, as code vulnerabilities can provide exposure points, compromising data, services, or unauthorised access to sensitive information. Adhering to frameworks that standardise security throughout development, such as OWASP(Open Web Application Security Project) [15], Top Ten, and MITRE's CWE (Common Weakness Enumeration) [16], provide developers with structured approaches to address vulnerabilities, like Structured Query Language injection (SQLi) to cross-site scripting (XSS). Secure coding extends beyond writing functional code, it requires anticipating attack vectors and implementing protective measures. As software becomes increasingly complex, secure coding must evolve to address modern architectures like microservices, and third-party libraries, where vulnerabilities in one component can impact the entire system. Secure coding practices must be robust in detecting traditional issues and adaptable to new and evolving threats.

B. Traditional Methods

Historically, secure coding has relied on manual code reviews and rule-based static analysis tools. While manual reviews allow deep insights into specific code parts for flaws or weaknesses, the process is time consuming, and difficult to scale for large projects or dynamic codebases. This process is highly dependent on the expertise and attention of the reviewer and is often subjective, leading to inconsistency in the identification of security issues.

Rule-based static analysis tools automate the code scanning process to identify vulnerabilities based on predefined rules [3] [5] .These tools are effective in detecting certain issues, such as hardcoded credentials, SQL injections, or buffer overflows. However, these tools are limited by outdated rule sets as new vulnerabilities emerge [3] [8] [9]

[13] [14] [17]. Furthermore, these tools often produce false positives, where non-issues as vulnerabilities are flagged [18], leading to alert fatigue among developers, or fails to identify vulnerabilities, leaving the application exposed to attacks. Traditional methods also struggle with adaptability to modern development cycles, often vulnerabilities before they reach production. This has led to a growing need for real-time, automated approaches that continuously monitor and detect security issues.

C. Emerging Cyber Threats

The cyber threat landscape has evolved in recent years, with attackers increasingly targeting software vulnerabilities as entry points into systems. These threats are characterised by their sophistication, persistence, and ability to evade traditional detection mechanisms.

Advanced Persistent Threats (APTs), for instance, are long-term, targeted attacks that aim to establish and maintain unauthorised access to a network over an extended period. APTs often exploit zero-day vulnerabilities, which are unknown and unpatched flaws in software that are ripe for corruption [19]. Additionally, buffer overflow attacks remain a prevalent threat. Excess data is sent to a buffer causing it to overwrite adjacent memory spaces. This attack can give the attacker control over the system, allowing for arbitrary code execution or crashing the application [20]. Buffer overflows are still a significant concern, particularly where secure coding practices have not been rigorously applied [20].

SQL injection (SQLi) remains a leading threat, especially in legacy systems that lack proper validation [15] [21]. Attackers insert of manipulate SQL queries to execute unintended commands, compromising data confidentiality, integrity, or availability [15].

Zero-day vulnerabilities, which are unknown flaws with no available patch, are highly dangerous. Attackers race to exploit these vulnerabilities before the software vendors can issue fixes, often leading to widespread damage [19]. In environments relying on traditional static analysis and manual reviews, zero-day vulnerabilities can remain undetected until it's too late [22].

Supply chain attacks, where attackers exploit third-party software components and libraries by embedding malicious code that can later be circulated into the software product, pose another risk [23]. Such attacks, exemplified by high-profile incidents like SolarWinds [24], reveal the critical gaps in traditional secure coding practices when examining dependencies and external components.

The complexity and adaptability of these emerging threats emphasise the need for enhanced secure coding practices that are proactive and utilise advanced tools and techniques to surpass limitations of traditional approaches.

D. Limitations of Traditional Secure Coding Practices

One major challenge with traditional secure coding methods face is lack of scalability [3] [13] [14] [17]. Modern development environments, like those using agile methodologies and CI/CD (Continuous Integration/Continuous Deployment) pipelines, produce vast amounts of code at an unprecedented pace. Reviewing thousands of lines of code daily is time consuming and prone to human error or fatigue, becoming unmanageable [7] [25]. As development teams grow larger and more distributed, especially in enterprises with global operations, the volume of code review far exceeds the capacity of manual processes [10] [25].

Traditional methods also struggle in fast paced CI/CD environments, emphasising continuous integration and code delivery multiple times daily, demands automated checks and real-time feedback [10] [26]. Manual code reviews, cannot keep up with this pace, creating delays and increasing the likelihood that vulnerabilities will be introduced without adequate inspection [10] [25]. Automated dynamic security testing techniques, like Web Application Security Testing (WAST) and Security API Scanning (SAS), can integrate into CI/CD pipelines to address this issue and improve scalability by offering real-time vulnerability detection during the continuous deployment process [7]. The inability to scale these methods across expansive and dynamic codebases leaves organisations exposed to undetected security risks, especially as their software systems become more complex and interconnected [27]. In contrast, modern secure coding practices are relying on more scalable, automated solutions. These approaches use AI-driven tools to automatically review large codebases, helping developers identify vulnerabilities across projects without manual intervention [27]. However, while these tools are promising, the transition away from traditional methods remains slow due to the perceived reliability of manual reviews [27].

E. AI Tools in Secure Coding

Artificial Intelligence (AI) has become transformative in the field of secure coding, with tools like GitHub Copilot, DeepCode, and automated vulnerability scanners representing significant advancements [3] [28]. These tools primarily leverage Machine Learning (ML) to process large volumes of code, identifying patterns that suggest potential security vulnerabilities [5] [11]. Their key value lies in automating labor-intensive tasks traditionally managed by manual reviews and static analysis. This improves both efficiency and focus on addressing more complex security challenges [10].

GitHub Copilot, powered by OpenAI's Codex, is an AI-powered code completion tool that suggests entire code snippets based on the developer's input. While initially aimed at boosting productivity, it offers contextually relevant suggestions aligned to secure coding practices, based on its training data [4] [5]. However, Copilot lacks a specific security focus, unlike cools like DeepCode [5] [6].

DeepCode goes further by not only generating code but also identifying and fixing security flaws in the code.

DeepCode's ability to leverage large language models (LLMs) has made it a leader in program repair, improving both accuracy and efficiency in detecting and resolving complex vulnerabilities [29].

Automated vulnerability scanners, like SonarQube or Snyk, assess the codebase against known vulnerabilities in real-time, scanning for potential security flaws across libraries, frameworks, and application code [5] [6] [29] [30]. Synk integrates AI for dynamic detection, while SonarQube offers a more static analysis-driven approach [10] [30].

By using AI's predictive capabilities, these tools can flag potential vulnerabilities early in the development lifecycle, aligning with shift-left security practices to address security issues sooner [31]. As organisations increasingly adopt agile and DevSecOps practices, AI-driven are proving invaluable for large scale, rapidly evolving codebases [32] [33] [34] [35].

F. Effectivenss of AI Tools in Improving Secure Coding

AI tools have revamped secure coding by accelerating vulnerability detection and reducing time spent on identifying potential issues. Unlike traditional static analysis tools, which rely on predefined rules, AI-driven tools dynamically learn from historical data, adapting to evolving threat vectors [11]. These tools can analyse large codebases faster than it would take human developers, thereby keeping up with the fast-paced nature of modern software development [36] [37].

Automating defect prediction and code reviews significantly enhances the quality of security outcomes by minimising human error [38]. Developers may accidentally overlook unremarkable yet dangerous coding patterns due to time constraints, fatigue, or a lack of security expertise [8]. AI tools mitigate this risk by constantly scanning for issues with a higher precision and consistency that humans cannot replicate [37]. AI models learn from massive repositories of historical vulnerabilities, identifying subtle patterns that traditional methods might miss, such as logical errors that could lead to buffer overflows, injection attacks or insecure authentication mechanisms [5] [34] [37].

Several case studies highlight AI tool effectiveness. For example, Pearce, et al. [6], found that although Copilot generated correct code, approximately 40% of suggestions were vulnerable based on MITRE's Top 25 CWE [4] [6]. However, when used with secure coding libraries like 'bcrypt' for password hashing, Copilot produced secure, non-vulnerable code [6]. Similarly, Berabi et al. [29] showed how DeepCode AI Fix leveraged large language models (LLMs) to achieve an 80% success rate in removing security defects, outperforming traditional methods. DeepCode excels at identifying and repairing complex, non-trivial security vulnerabilities and provide real-time feedback on fixing semantic bugs. This highlights the potential of AI tools in automating secure coding practices and enhancing scalability and accuracy of vulnerability detection [36].

AI's ability to consider vulnerabilities is another crucial advantage. Instead of a simple "vulnerable/not vulnerable" assessments, many AI tools offer detailed explanations of why certain code snippets are questionable, offering insights into how to fix them [29] [6]. This reduces the mental load on developers and allows them to address security flaws more effectively, ensuring the code is robust and secure by design [11].

G. Explainable AI (XAI) and Secure Coding

A significant concern with AI-driven security tools is the "black-box" issue of lack of transparency in how AI models make decisions. This opacity can undermine trust in AI-generated outputs, especially in security-critical environments where understanding the reasoning behind decisions is essential. This is where XAI plays a vital role in making AI tools more transparent, interpretable, and therefore more trustworthy for developers [3] [14].

XAI techniques such as LIME and SHAP explain how AI models detect vulnerabilities by breaking down the decision-making process and explaining the importance of various features that led to an outcome [14] [39]. For example, SHAP assigns importance scores to code, indicating which parts contributed most to the model's classification of a vulnerability. This transparency allows better understanding of AI-generated outputs and verifies their accuracy [29] [39].

The adoption of XAI enhances human-AI collaboration by enabling developers to challenge, verify, and refine the AI's output, fostering trust in AI systems, which is crucial for adoption in security-critical environments [4] [14]. Additionally, XAI ensures compliance with security standards like GDPR [1] or ISO/IEC 27001 [14] [39] [40]. XAI offers the necessary transparency and accountability in decision making to demonstrate the responsible and compliant use of AI tools for clear audit trails and explanations for security related decisions [4].

H. Aligning XAI-Enhanced Secure Coding with Security Standards

Security standards are critical for ensuring that software development processes align with best practices for mitigating threats. ISO/IEC 27001 and the NIST Cybersecurity Framework [2] provide comprehensive guidelines to protect information assets, manage security risks, and establish a culture of security.

ISO/IEC 27001 is internationally recognised information security management and often required where sensitive data is handled. It offers a framework for establishing, implementing, maintaining, and continually improving an Information Security Management System (ISMS) [13] [40]. The standard outlines controls for managing cybersecurity risks, ranging from technical vulnerabilities in software to broader governance issues [5].

NIST [2] offers a structured approach to identifying, protecting, detecting, responding to, and recovering from cybersecurity threats [36]. It emphasises secure development practices, including vulnerability management and real-time threat detection, which are core principles of secure coding. These standards protect against external threats while ensuring organisations maintain cyber hygiene throughout the software development lifecycle (SDLC) [41].

I. How XAI-Enhanced Practices Meet Standards

XAI-enhanced secure coding aligns especially well with security standards by offering transparency, accountability, and traceability in AI-driven security processes. ISO/IEC 27001 and NIST emphasises the need for comprehensive documentation and accountability in handling security risks [2] [14] [40]. In traditional secure coding environments, compliance with these standards involves manual reviews, audits, and documentation of security measures. However, in an AI-enhanced secure coding, XAI becomes critical in demonstrating how decisions are made and ensuring they meet regulatory and security requirements [41] [42]. For example, AI-driven vulnerability detection tools can generate accurate predictions about potential issues, but without explainability, can be difficult for developers or auditors to understand why certain vulnerabilities were flagged [34] [43]. This opacity could lead to mistrust or non-compliance with security standards that demand a clear audit trail. XAI tools like LIME and SHAP, however, provide actionable insights into how AI models arrive at their conclusions [14]. By offering feature importance scores and visual explanations of model behaviour, these tools bridge the gap between "black-box" AI systems and the strict documentation requirements set by security standards [7]. Moreover, XAI-enhanced secure coding practices promote better adherence to SDLC processes. In environments governed by standards such as ISO/IEC 27001, the SDLC is integral to maintaining continuous security vigilance from the planning stages through to deployment and maintenance [13] [17]. XAI ensures that AI tools provide clear explanations of security decisions during each stage of the SDLC. This can significantly improve the accountability of developers and security teams, ensuring they understand and can justify AI-driven decisions [34].

Case studies have shown that AI-based systems, when combined with XAI, have successfully met ISO and NIST standards while delivering robust and interpretable security solutions [5] [36]. For example, in a study involving large-scale enterprise applications, the integration of XAI into automated vulnerability detection tools not only improved the speed and accuracy of threat detection but also provided necessary transparency to pass compliance checks [34] [44]. This ability to generate compliant reports that outlined the reasoning behind AI-driven decisions is essential for organisations operating under strict regulatory frameworks [42].

J. Regulatory Considerations

Adopting AI tools in secure coding brings regulatory challenges, particularly concerning data privacy and ethical AI governance [4]. The use of AI systems in security critical tasks raises concerns about AI's "black-box" nature, where decisions are made without any explanation [3] [14]. This becomes an issue when complying with data protection regulations like GDPR [7] [14]. GDPR mandates that organisations not only protect personal data but explain how decisions affecting individuals are made, particularly with automated systems [45]. The regulation addresses automated decision making, stating that individuals have right to request an explanation of decision made about them. In a secure coding context, AI-driven tools that flag security issues or suggest security related changes must provide explainability of their decisions in compliance with GDPR's "right to explanation" [14]. XAI-enhanced AI tools can assist in compliance with requirements by offering clear, interpretable explanations of how AI identifies vulnerabilities or suggest secure coding practices [11] [34]. XAI provides transparency, ensuring that the logic behind automated security decisions can be understood, scrutinised, and adjusted as needed [34] [39]. For organisations bound by GDPR or similar privacy regulations, the use of XAI enables them to provide explanations that align with ethical AI principles and regulatory expectations [43].

Beyond data privacy, the European Union's AI Act [46], which is expected to set a global precedent for regulating AI systems, also emphasises accountability, transparency, and human oversight [36] [29]. This will likely mandate that AI systems used in security-sensitive sectors must be interpretable and explainable, making XAI a critical tool for compliance [41]. Organisations that use AI systems without explainability may face regulatory penalties for failing to meet these standards [7]. Additionally, XAI supports organisations in conducting compliance audits by providing transparent decision-making processes. This enhances responses to regulatory inquiries, demonstrating the responsible use of AI tools and security vulnerabilities are being addressed that align with legal and regulatory frameworks [11] [14].

K. Challenges and Solutions in Integrating XAI

Several challenges delay the integration of XAI into secure coding. These include technical barriers, such as the complexity of integrating XAI into existing codebases and development pipelines, and cultural resistance to adopting AI-based tools in traditional software development environments. Moreover, there is often a trade-off between transparency and performance, where increasing explainability may reduce AI model efficiency.

Solutions to these challenges include the adoption of best practices for integrating XAI into secure development workflows. Organisations can also address resistance to AI by providing training and regulatory transparency, supporting a deeper understanding of AI tools. Future advancements in

XAI, such as improving interpretability without sacrificing performance, are essential for widespread adoption.

L. Key Findings

The literature reveals that while traditional secure coding practices are foundational, they face significant limitations in detecting emerging cyber threats. For instance, Charoenwet et al. [8] found that coding weaknesses often remain unfixed due to incomplete reviews and disagreements among developers, highlighting the shortcomings of traditional code review processes. Similarly, Bosu et al. [9] identified that vulnerabilities frequently slip through peer reviews, especially when introduced by less experienced developers.

AI-enhanced tools, particularly those incorporating XAI, offer promising solutions by automating vulnerability detection and improving transparency. Berabi et al. [29] demonstrated that their DeepCode AI Fix system could effectively fix security vulnerabilities, outperforming models like GPT-4. Chmioelowski et al. [11] showed that XAI models could match black-box model accuracy while providing valuable explanations, aiding in developer trust and understanding. Furthermore, Bilgin et al. [37] illustrated how AI can enhance software security beyond traditional methods through their work on machine learning for vulnerability prediction. However, challenges remain in integrating XAI into secure development workflows. Mohammadkhani et al. [3] highlighted a lack of research in applying XAI to generation-based software engineering tasks, indicating gaps in current methodologies. Tantihamthavorn et al. [42] emphasised that the opacity of AI/ML models can hinder developer trust, a critical factor for adoption. Moreover, Shi et al. [4] suggested that while AI tools are advancing, their explainability is not keeping pace, posing integration challenges.

M. Future Research and Industry Relevance

Future research should focus on refining XAI techniques to address scalability and performance challenges, ensuring that AI-driven secure coding solutions are both effective and interpretable. Additionally, there is a need for more studies exploring how XAI can be seamlessly integrated into CI/CD pipelines. For developers, cybersecurity teams, and regulatory bodies, AI and XAI represent the future of secure software development. Adopting these technologies can help organisations stay ahead of evolving cyber threats while maintaining compliance with security standards.

IV. Methodology

A. Introduction

This methodology was designed to align with the research objectives, addressing the complex integration of AI, particularly XAI, into secure software engineering. A mixed-methods approach, combining both qualitative and quantitative methods, provided a refined understanding of interactions between AI tools, secure coding practices, ethical considerations, and compliance with security standards. This approach allowed for an in-depth exploration of individual experiences alongside general trends.

B. Qualitative Exploration of AI and XAI in Secure Software Engineering

The study began with a qualitative approach to explore the dynamics between AI technologies, especially XAI, and secure software practices. This phase enabled participants to share detailed experiences, offering insight into XAI's role in secure software engineering. The research used diverse resources such as scholarly articles, industry white papers, and semi-structured interviews and questionnaires with professionals, to understand AI integration in secure coding, identifying challenges and opportunities in using XAI. Key themes included AI tool effectiveness, XAI's role in transparency and trust, and ethical considerations in AI-driven decisions.

C. Establishing Context and Identifying Core Themes

Building upon findings from the literature review and initial data, the study identified core themes relevant to the research questions. Thematic analysis was employed to systematically capture patterns within participant's responses, focusing on themes that directly addressed the main research questions. Grounding the study in these key themes ensured a focused approached to addressing the main research questions and sub-questions.

D. Data Collection Methods

1. Semi-structured Interviews and Participant Recruitment

Semi-structured interviews were conducted with three professionals and all seven respondents answered the questionnaire in lieu of interviews, including the three who completed audio interviews. They were selected based on their ability to provide rich and relevant data concerning AI and secure software engineering [47]. This approach aligns with purposive sampling, a non-probability sampling method that intentionally targets individuals based on knowledge and experience, as required within a niche focus, like XAI. Despite recruitment challenges, participants held senior roles like Technical Lead and Head of Product Delivery, providing insights on XAI's challenges and ethical considerations in secure coding. Unfortunately, one interview was part-recorded, however the respondent compensated with an expanded questionnaire response.

The interviews and questionnaires included both open-ended and closed-ended questions (Appendix F), providing quantitative data on variables such as years of experience, familiarity with AI tools, satisfaction levels, and concerns regarding ethical implications. This comprehensive approach to data collection was essential for framing the subsequent stages of the study, which focused on empirical data analysis to validate and expand upon these initial insights. Each interview was conducted individually, either in person or via

video conferencing platforms, depending on the participant's location and preference. With the consent of the participants, the interviews were audio recorded and subsequently transcribed verbatim to ensure accuracy in capturing their perspectives.

1. Questionnaires

To ensure sufficient data collection and accommodate participants who were uncomfortable with audio-recorded interviews, the interview questions was adapted into a questionnaire format (Appendix F) and distributed electronically, using an online survey platform, to all seven participants. Reminders were sent to encourage timely responses, and a reasonable time frame was provided. This approach ensured consistency across the data set. Roles among participants varied, including Software Engineer, Data Analyst, Head of ICT, Web Developer, and others. Their experience ranged from 1-3 years to over ten years. Open-ended questions allowed for detailed qualitative responses, while close-ended questions provided quantitative data on variables like experience, familiarity with AI, satisfaction, and ethical concerns, rated on a Likert scale.

E. Data Analysis Methods

1. Qualitative Analysis Methods

Thematic analysis was applied to qualitative data from interviews and open-ended questionnaire responses, following Braun and Clarke's approach [48]. This process involved coding, theme development, and refinement, ensuring that the themes accurately represented participants' perspectives. Significant statements and phrases relevant to the research questions were highlighted (Appendix D). Codes were assigned to these segments to represent key concepts and ideas. Codes were then grouped into potential themes based on similarities and relationships (As shown in Appendix E). Each theme was clearly defined and named to capture the essence of the data it represented. This systematic approach ensured that the analysis remained rigorous and that the resulting themes were grounded in the data.

2. Quantitative Data Analysis

Quantitative data from close-ended questions were analysed using descriptive and inferential statistics to summarise participants' experiences and perceptions. Responses were organised into a database for systematic analysis, with numeric variables used for years of experience, a Likert scale to measure familiarity with AI tools, and a scale from 1 (not satisfied) to 5 (extremely satisfied) for satisfaction levels, and ethical concern ratings scaled as 1 (not concerned) to 5 (extremely concerned). Descriptive statistics, including mean, median, and mode, summarised satisfaction and ethical concerns, while response patterns across categories were analysed for trends. Pearson's coefficient was used to examine correlations between experience, satisfaction with AI tools, and ethical concerns, as it is suitable for linear relationships between continuous variables [49]. This analysis was conducted in Python, using libraries such as Pandas and NumPy for data manipulations and statistical calculations (Appendix C).

F. Data Entry and Software Tools

Microsoft Excel was used for organising qualitative and quantitative data, facilitating identification of codes and themes relevant to research questions. Due to the small sample size, visualisations were not used. Instead, the data is presented in a table identifying variables with mean, median, mode, and correlations with ethical concerns. By combining Excel and Python, the study ensured effective data handling across qualitative and quantitative measures. Appendix C).

G. Thematic Insights from Annotated Literature

Annotated references (Appendix A) contributed to thematic analysis by highlighting gaps and questions in existing literature, which informed the study's framework and ensured depth in addressing the research topic [47] [50] [51].

H. Integration of Qualitative and Quantitative Findings

The mixed-methods approach enabled a comprehensive understanding of the research problem by integrating qualitative and quantitative data. Findings from both data sets were compared to identify similarities, differences, and supporting data. This cross-checking was employed to enhance the credibility and validity of findings.

I. Ethical Considerations

Ethical standards were rigorously followed throughout the research process to ensure the integrity of the study. Protection of participants and were preceded by an Ethics document (Appendix B). Participants were fully informed of purpose of the study, nature of their involvement, their data use, and rights, including the right to withdraw without penalty and consent was obtained prior to participation. Confidentiality and anonymity were maintained at all stages of the process with personal identifiers removed from transcripts and datasets. Any potentially identifying information was anonymised in the reporting to maintain confidentiality. All data were securely stored on password-protected devices and encrypted storage solutions, accessible only to the researcher. Data handling procedures complied with relevant data protection regulations and institutional guidelines. These procedures ensured secure data management and respect of participant privacy.

J. Validity and Reliability

Several strategies were utilised to enhance the validity and reliability of the research findings. Comparing insights across different data and perspectives, i.e. interviews and questionnaires, strengthened the credibility of the study [47]. A detailed audit trail documenting all steps of data collection, analysis, and interpretation was maintained, allowing for replication of the study and providing a clear account of the research process. Reflexivity was practiced by the researcher

to acknowledge and mitigate potential biases, ensuring that findings were based on participants' perspectives rather than influenced by the researcher's preconceptions.

K. Limitations of the Study

The research acknowledges certain limitations that may affect the interpretation and transferability of the findings. The small sample size (n=7) limits the extent to which the findings can be generalised to the broader population. While participants held diverse roles and experience levels, they may not fully represent all viewpoints within the industry, particularly given the wide range of conditions and challenges that professionals face in secure software engineering. Participants' responses may be influenced by self-reporting bias, where individuals present themselves in a favourable light or provide socially desirable answers. Additionally, the adjustment from conducting solely interviews to including questionnaires may have impacted the depth of qualitative data collected, as open-ended written responses may be less detailed than verbal interviews.

L. Methodology Conclusion

The integration of qualitative and quantitative data enriched the understanding of AI and XAI in secure software engineering, providing refined observations into participants' experiences and perspectives. This mixed-methods approach enhanced the validity of the research findings and highlighted key themes that will guide future exploration in this field.

V. Findings

A. Introduction

This section presents the findings of the study, focusing on how the integration of AI tools, particularly XAI, can enhance secure coding practices and align with security standards. These findings are organised to address the main research question; "How can the integration of AI tools, particularly XAI, enhance secure coding practices and align with security standards?". Thoroughly exploring this question requires integrating both quantitative and qualitative data, structured around the sub-questions: "What are the limitations of traditional secure coding practices in addressing emerging and complex cyber-threats?", "How effective are AI tools, such as GitHub Copilot and automated scanning technologies, in improving secure coding practices, and how does XAI enhance transparency and trust in AI-driven decisions?", "How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines, and what are the regulatory considerations?", and "What are the primary challenges in integrating XAI tools into secure coding processes, and what solutions can help overcome these challenges?".

By directly addressing these questions, this study aims to provide a comprehensive understanding of the potential depth and richness of the qualitative data collected. Written responses in the questionnaire may lack the detail and tone often captured in verbal interviews, potentially limiting insights into participants' personal experiences and perceptions. The shift may have also impacted the comparability between the two data collection methods, reducing the reliability of qualitative findings.

The study touched upon ethical concerns related to AI use in secure software development. However, the limited sample size and the specific professional backgrounds of participants may have restricted the span of the ethical issues explored. A larger, more diverse sample could offer a deeper exploration of AI ethics across different organisational and cultural contexts.

As AI technology evolves rapidly, the findings of this research are grounded in the tools and techniques available at the time of the study. Any subsequent advancements in AI or XAI technologies may alter the applicability of the findings. Future studies would benefit from continuous re-evaluation as both technology and industry standards evolve.

While the focus on AI-enhanced secure software engineering is broad, the findings may be more relevant to certain sectors or organisational types. Different industries may face unique challenges that are not captured in this research, affecting the transferability of the conclusions or other contexts.

B. Quantitative Analysis Findings

The quantitative data were collected through a survey administered to software developers and security professionals, yielding responses from seven participants. The survey assessed participants satisfaction with AI tools, their ethical concerns, and years of experience. The data provided insights into the general perceptions and attitudes toward AI tools in secure coding practices.

C. Summary of Quantitative Data

Table I summarises the key statistical measures of the variables studied.

TABLE I. *Quantitative Summary*

Variable	***Mean***	***Median***	***Mode***	***Correlations with Ethical Concerns***
Satisfaction with AI Tools (1-5)	*3.29*	*3.00*	*3*	*-0.679*
Ethical Concerns (1-5)	*3.71*	*4.00*	*4*	*-*
Years of Experience (Numeric)	*6.00*	*6.00*	*5*	*0.286*

Participants reported a moderate level of satisfaction with AI tools (mean = 3.29), suggesting that while AI tools are beneficial, reservations persist about their use in secure

coding practices. Ethical concerns were relatively high (mean = 3.71), indicating significant apprehension regarding the ethical implications of AI, including issues of bias and accountability.

D. Correlation Analysis

A negative correlation (-0.679) was observed between satisfaction with AI tools and ethical concerns, implying that participants with higher ethical concerns tended to report lower satisfaction with AI tools, Additionally, a slight positive correlation (+0.286) was found between years of experience and ethical concerns, suggesting that more experienced professionals may have heightened ethical concerns, although the relationship is not strong.

Given the small sample size (n=7), it is important to consider the statistical significance of these correlations. With such a limited dataset, the correlations may not be statistically significant and should be interpreted with caution. As such, the findings may not be generalisable to the broader population.

E. Interpretation in Relation to Research Questions

The quantitative findings suggest that while AI tools have the potential to enhance secure coding practices, ethical concerns remain a significant barrier to their full acceptance and satisfaction among professionals. The negative correlations between satisfaction and ethical concerns underscores the importance of addressing ethical issues to improve user perception of AI tools. The slight positive correlations between years of experience and ethical concerns may indicate that seasoned professionals are more aware of the ethical complexities associated with AI integrations.

These findings relate to the main research question by highlighting the necessity of integrating XAI to improve transparency and trustworthiness in AI-driven decisions, thereby potentially alleviating ethical concerns and enhancing satisfaction.

F. Qualitative Analysis Findings

The qualitative data were derived from semi-structured interviews and/or questionnaires with the same seven participants. This aimed to provide deeper insights into their experiences and perceptions regarding AI and XAI tools in secure coding practices. A thematic analysis was conducted to identify key themes relevant to the research questions revealing seven overarching themes:

- Effectiveness of AI Tools in Secure Coding
- Limitations of Traditional Secure Coding Practices
- Challenges in Integrating AI Tools into Secure Coding Processes
- Explainable AI(XAI) for Transparency, Trust, and Compliance
- Ethical Considerations in Using AI Tools for Secure Coding
- Trust and Overreliance Issues with AI Tools
- Future Trends and Recommendations for AI in Secure Software Engineering

Detailed discussions of each theme are presented in the next section. The implications of these findings in relation to existing literature are discussed in the subsequent section.

G. Summary of Qualitative Data

Table II summarises the key statistical measures of the variables studied.

TABLE II. *QUALITATIVE SUMMARY*

Themes and Frequency Analysis		
Theme	***Frequency (out of 7)***	***Key Insights***
AI Tool Effectiveness	7/7	*AI tools significantly improve productivity and security, offering real-time assistance in code suggestions and vulnerability detection.*
Limitations of Traditional Practices	3/7	*Participants express frustration with traditional tools, which often stall productivity and may not adequately detect vulnerabilities.*
Integration Challenges	6/7	*Integration of AI tools into existing CI/CD pipelines presents significant barriers, including technical debt and compatibility issues with legacy systems.*
XAI	6/7	*XAI techniques are critical for enhancing trust, compliance, and transparency, especially in regulated industries, by providing explanations of AI-driven decisions.*
Ethical Considerations	6/7	*Strong concerns exist about bias, accountability, and the need for ethical guidelines in the use of AI tools, particularly in decision-making processes*
Trust and Overreliance Issues with AI	6/7	*Scepticism toward AI tools remain high, with participants expressing the necessity of human oversight to*
Future Trends and Recommendations	7/7	*Participants anticipate increased adoption of XAI and regulatory focus on ethical AI governance, emphasising the need for improved explainability and integration with existing workflows*

H. Key Themes and Research Question Relevance

1. Effectiveness of AI Tools

All seven participants highlighted the effectiveness of AI tools in improving productivity and security. These tools, such as ChatGPT and GitHub Copilot, were noted for automating security checks and providing real-time assistance in secure coding practices. One participant illustrated the practical benefits and stated, *"Very effective at providing suggestions to problems faced with extracting and manipulating data. Particularly with designing and implementing data flows."* Another added, *"Chat GPT 90% effective, I'm probably not providing enough info for perfect answer every time. 90% for CoPilot code line auto completion, it's no always right, but most of the time it is."*

These comments illustrate the practical benefits of AI tools in improving code quality and security, addressing the second sub-question by demonstrating the effectiveness of AI tools in enhancing secure coding practices.

2. Limitations of Traditional Practices

Three out of seven participants expressed frustration with traditional secure coding practices. They emphasised that traditional secure coding practices such as manual code reviews and static analysis tools, are often insufficient in addressing emerging and complex cyber threats. One participant remarked, *"Some of these tools like Sonar... may not be as skilled enough... that it doesn't reject or raise a flag."* and *"Automated scanning technologies like Sonar, SAST, and DAST... but sometimes these tools can stop productivity.*

Additionally, a participant expressed, *"They are useful; however, they often lead to a more manual examination of code, as the AI often leaves you with as many questions as answers. Indeed, often left with the feeling that it would sometimes be quicker to just check everything yourself. Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"*

These insights highlight the limitations of traditional tools in keeping up with the demands of modern secure software engineering. This theme addresses the first sub-question by identifying the shortcomings of traditional secure coding practices and underscoring the need for more advanced solutions capable of coping with modern cyber threats.

3. Integration Challenges

Six participants identified significant challenges to integrating AI tools into secure coding processes. Issues cited, such as technical debt, compatibility with legacy systems, performance bottlenecks, and need for training. The integration of AI into existing CI/CD pipelines emerged as a common challenge, with participants citing difficulties in compatibility and performance.

One participant stated, *"The main challenges include compatibility with legacy systems and performance bottlenecks. We addressed these by gradually phasing in AI tools and optimising the pipeline for faster execution times."*. This participant also stated, *"You can't just deploy AI and expect it to do something for you. You have to give it explicit instructions."* Additionally, a participant elaborated, *"... One major issue is ensuring these tools work smoothly with the existing setup. I've tackled this by choosing AI tools that are compatible with our CI/CD platforms and have good API support. Another challenge is the extra computational power needed for AI operations, which I manage by leveraging scalable cloud services. Balancing speed and thoroughness can be tricky too, so I fine-tune the AI tools to focus on essential security checks without slowing down the deployment process too much."*

Notably, one participant mentioned, *"Due to the nature of security ramifications of our data, implementation of AI in our data life cycle is strictly prohibited."*

This theme addresses the fourth sub-question by identifying the primary challenges in integration and highlighting areas that require attention to facilitate successful adoption.

4. Explainable AI (XAI)

Six participants emphasised the importance of XAI in improving trust and transparency in AI-driven decisions. XAI was considered essential in regulated industries, where understanding AI decisions is not only beneficial but also legally required. A participant stated, *"... Transparency is crucial, especially in security-focused environments where decisions need to be auditable and understandable by both developers and stakeholders. XAI was critical in a recent project where we needed to explain AI-driven decisions to non-technical stakeholders."* and *"XAI allowed us to trace the logic behind an AI-based intrusion detection system's decisions, making it easier to fine-tune the system and avoid false alarms. I foresee XAI playing a significant role in regulatory compliance, where explaining AI decisions will be a legal requirement."*

Another participant noted, *"The ability for AI to recognize the potential weaknesses of its own outputs, providing alternative solutions for different use cases. This would make decision-making more transparent and allow developers to make informed decisions."*

This theme directly relates to the main research question and the second sub-question by highlighting how XAI enhances transparency and trust in AI-driven decision, making AI tools more acceptable to users.

5. Ethical Considerations

Ethical considerations, such as biases in AI models and data quality issues, were significant concerns among participants. There was caution against overreliance on AI tools without human oversight. Participants stressed the necessity of human validation to ensure reliability and accountability. One participant remarked, *"A well-known concern in AI systems is their potential to reflect and amplify biases present in their training data. When used in testing, a*

biased AI could lead to uneven results. Ensuring diverse and representative training data is essential to avoid these biases in the software being tested." and *"Data Privacy needs to be tightened. AI is being used increasingly in Software Development which brings into question how data is scanned and used. Better guidelines for Ethics, establishing unambiguous guidelines for moral AI development and application is essential to ensuring that technology advances society rather than undermines it."*

Additionally, a participant observed, *"I can see that if there are specific biases present in the training data, then these biases will be replicated in the output."*

This theme underscores the importance of addressing ethical concerns and maintain human involvement in the secure coding process, reinforcing the need for a balanced approach between automation and human expertise.

6. Trust and Overreliance Issues

Six participants expressed concerns about overreliance on AI tools, stressing the need for human oversight. While AI tools were seen as helpful, participants were reluctant to place complete trust in them, highlighting current scepticism about AI's decision-making processes. One participant remarked, *"You still have to review the AI's work…No one will trust it completely anytime soon,"* and
"… The Global Business is currently defining Governance for Microsoft Copilot which will be the only AI tool [omitted]adopts in the near future. The business is interested to see what Copilot can bring and from its usage and testing will define whether we look into different areas moving forward."

Another participant shared, *"At the moment, I think people are trustier now with a lot of things within their life, even though it's only a 5-year-old. They tend to think, 'Oh, look what's been invented! Oh, the Internet's always right…This is gonna save me so much time,' and they believe it."*

This theme emphasises the necessity of balancing AI automation with human oversight, addressing concerns about trust and overreliance on AI tools.

7. Future Trends and Recommendations

All seven participants discussed anticipated future trends in AI adoption and XAI integration. They highlighted the potential of AI in enhancing threat detection and automating code remediation. Participants emphasised the importance of regulatory compliance, ethical governance, and seamless integration of AI tools into DevSecOps workflows.

One participant stated, *"The integration of AI and secure software engineering is expected to evolve significantly in the coming years, driven by advancements in AI technologies and the increasing complexity of cybersecurity challenges. ... AI will increasingly be used to predict and prevent security threats before they occur...analysing patterns and behaviours in real-time, AI can anticipate potential vulnerabilities or attack vectors and suggest pre-emptive measures."* and *"AI will become an integral part of DevSecOps, automating security checks at every stage of the software development lifecycle...will include AI-driven static and dynamic code analysis, automated threat modelling, and continuous monitoring….will play a larger role in incident response, helping security teams detect, analyse, and respond to security incidents more quickly and accurately. AI-powered tools will...automate the identification of threats, prioritize incidents, and even initiate automated responses"* and *"AI tools that assist in writing secure code will become more sophisticated, providing developers with real-time suggestions and corrections as they code. These tools will leverage machine learning models trained on vast datasets of secure and insecure code examples."* Additionally, this participant stated *"To better support secure software engineering practices, AI tools can be improved or enhanced with...Context-Aware Security Recommendations…Real-Time Secure Coding Assistance…Adaptive Learning from Feedback Loops…Integration with Threat Intelligence Feeds…Automated Threat Modelling and Risk Assessment"*

Another participant commented, *"I foresee that as training data becomes flooded with AI-generated content, then the outputs will trend towards the mean, reducing and stifling innovation."*

Additionally, a participant mentioned *"... Incorporating better XAI features to ensure security-related AI decisions can be fully understood and trusted... AI tools should offer better support for legacy codebases...Tools should include features for ethical use, such as bias detection."*

This theme addresses the fourth sub-question by exploring future trends and recommendations, highlighting the potential advancements in AI and XAI integrations, and emphasising the importance of regulatory compliance and ethical governance.

I. Synthesis of Findings

The qualitative findings reveal an involved relationship between the effectiveness of AI tools and the challenges of integrating them into secure coding practices. While AI tools are recognised for their ability to improve productivity and security, significant concerns remain regarding ethical considerations, integration challenges, and overreliance on AI without human oversight. The importance of XAI in enhancing transparency and trust is underscored, particularly in the context of regulatory compliance and ethical governanance.

VI. Discussion

The focus of these findings evaluated how the integration of AI tools, particularly XAI techniques, enhances secure coding practices and aligns with current security standards and regulatory frameworks.

The identified themes were interpreted in the context of existing literature on AI and secure software engineering [3] [4] [48]. This integration helped to identify how the findings align with, extend, or challenge current research, highlighting gaps and emerging questions that require further exploration. By placing the findings within the academic discussion, the

study reinforces its contribution to the discussion on AI in secure software engineering.

The integration of AI tools, particularly XAI, enhances secure coding practices by automating security checks, providing real-time assistance, and reducing vulnerabilities [4] [5] [29]. XAI plays a crucial role in improving transparency and trust in AI-driven decisions, making AI tools more acceptable to users and aligning practices with security standards [12] [14]. However, significant challenges exist, including ethical concerns, integration difficulties, and trust issues, which must be addressed to fully realise the benefits of AI integration.

Traditional secure coding practices are limited in their ability to address emerging and complex cyber threats due to their time-consuming nature and inability to scale effectively [5] [8] [9]. Manual code reviews and static analysis tools may not detect complex vulnerabilities, highlighting the need for more advanced solutions [29] [44]. Studies have shown that vulnerabilities often remain unfixed due to limitations of manual review [8] [9] underscoring the need for AI-enhanced methods.

AI tools are effective in improving coding practices by enhancing productivity and reducing vulnerabilities [4] [11]. For instance, AI-driven systems like VulDeePecker have demonstrated the ability to detect vulnerabilities with higher accuracy than traditional methods [18]. XAI enhances transparency and trust by providing explanations for AI-driven decisions, increasing developer confidence in using AI tools [12] [13] [14]. This transparency is crucial for developers to understand and trust the recommendations provided by AI systems [11].

XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines by providing the necessary transparency and auditability required for compliance [1] [2] [40]. The explainability of AI decisions facilitates adherence to legal and industry requirements, such as the Data Protection Act 2018 and the EU AI Act, which emphasise transparency and accountability in AI systems [1] [46]. Regulatory considerations are addressed through the explainability of AI decisions, encouraging adherence to legal and industry requirements [46].

The primary challenges in integrating XAI tools include technical debt, compatibility issues with legacy systems, performance bottlenecks, and the need for training [10] [32] [35]. Integration difficulties are often exacerbated by the complexity of existing CI/CD pipelines and the rapid evolution of AI technologies [25]. Solutions involve phased implementation strategies, ensuring compatibility with existing CI/CD pipelines, comprehensive training programs, and organisational support to promote adoption [25] [26]. Emphasising socio-technical approach can also aid in addressing human factors associated with AI integration [35].

Looking ahead, the future of XAI in secure software engineering is likely to focus on improving scalability, performance, and seamless integration with development pipelines [12] [42]. As security threats evolve, AI-driven tools will increasingly rely on XAI techniques to provide real-time insights into AI decisions, making them more accessible and understandable to developers [13] [29]. Additionally, emerging regulations, such as the European Union's AI Act, are likely to mandate higher levels of transparency, further driving the development of XAI tools that meet regulatory compliance while maintaining high performance [46].

Notably, one of the interview participants highlighted the growing demand for context-aware XAI tools, which could integrate with threat intelligence systems to provide more relevant and actionable recommendations for secure coding. The ability to provide clear, explainable, and actionable insights in real-time will be critical for XAI to reach its full potential in enhancing both security and trust [11] [39]. This would provide context-specific explanations and would help developers make more informed decisions and ensure security measures are appropriate to specific scenarios, supporting the need for personalised and adaptive AI solutions in secure software engineering [43].

VII. CONCLUSION

The study reveals that the integration of AI tools, particularly XAI, hold significant potential for enhancing secure coding practices and aligning with security standards. AI tools improve productivity and reduce vulnerabilities, while XAI enhances transparency and trust in AI-driven decisions. However, ethical concerns, integration challenges, and trust issues pose significant barriers to their full acceptance and effectiveness [13] [43].

Addressing these challenges requires a multifaceted approach. Ethical concerns can be mitigated by developing robust ethical guidelines, ensuring diverse and representative training data, and maintaining human oversight to validate AI outputs. Integration challenges can be overcome through phased implementation, selecting AI tools compatible with existing systems, and providing comprehensive training for technical staff. Enhancing explainability and ensuring ethical AI governance are essential for aligning AI tools with security standards and regulatory requirement.

By acknowledging and addressing these issues, organisations can better leverage AI technologies to improve security outcomes while maintaining compliance and trustworthiness. The findings contribute to the broader understanding of how AI and XAI tools can be effectively integrated into secure software engineering practices, offering practical insights for professionals and researchers in the field.

REFERENCES

[1] GOV.UK, "Data Protection Act 2018," legistlation.gov.uk, 2024. [Online]. Available: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted. [Accessed 2024 September 2024].

[2] National Institute of Standards And Technology, "The NIST Cybersecurity Framework(CSF) 2.0," US Dept of Commerce, 2024.

[3] A. H. e. a. Mohammadkhani, “A Systematic Literature Review of Explainable AI for Software Engineering,” arXiv, 2023.

[4] Y. Shi, N. Sakib, H. Shahriar, D. Lo, H. Chi and K. Qian, “AI-Assisted Security: A Step towards Reimagining Software Development for a Safer Future,” in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, Torino.

[5] D. R. Chittibala, “Advancements in automated code scanning techniques for detecting security vulnerabilities in open source software.,” *International Journal of Computing and Engineering*, vol. 5, no. 2, pp. 16-25. doi: https://doi.org/10.47941/ijce.1737, 21 March 2024.

[6] H. Pearce and e. al., “Asleep at the Keyboard? Assessing the Security of GitHub Copilot’s Code Contributions,” in *2022 IEEE Symposium on Security and Privacy (SP)*, San Francisco, 2022.

[7] T. Rangnau, R. v. Buijtenen, F. Fransen and F. Turkmen, “Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines,” in *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*, Eindhoven, 2020.

[8] W. Charoenwet, P. Thongtanunam, V.-T. Pham and C. Treude, “Towardeffective secure code reviews: an empirical study of security-related coding weaknesses,” *Empirical Software Engineering*, vol. 29, no. 88, pp. 1-47. https://doi.org/10.1007/s10664-024-10496-y, 8 June 2024.

[9] A. Bosu and e. al., “Identifying the characteristics of vulnerable code changes: an empirical study,” in *FSE 2014: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*, New York, 2014.

[10] F. Zampetti and e. al., “Continuous Integration and Delivery Practices for Cyber-Physical Systems: An Interview-Based Study,” *ACM Transactions on Software Engineering and Methodology,*, no. 73, pp. 1-44. doi: https://doi.org/10.1145/3571854, 26 April 2023.

[11] L. Chmioelowski, M. Kucharzak and R. Burduk, “Application of Explainable Artificial Intelligence in Software Bug Classification,” *IAPGOŚ*, vol. 13, no. 1, pp. 14-17. doi: http://doi.org/10.35784/iapgos.3396, 2023.

[12] F. Charmet and e. al., “Explainable artificial intelligence for cybersecurity: a literature survey,” *Annals of Telecommunications*, vol. 77, pp. 789-812. doi: https://doi.org/10.1007/s12243-022-00926-7, 26 October 2022.

[13] A. B. Arrieta and e. al., “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82-115. doi: https://doi.org/10.1016/j.inffus.2019.12.012, June 2020.

[14] A. Adadi and M. Berrada, “Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),” *IEEE Access*, vol. 6, pp. 52138-52160. doi: 10.1109/ACCESS.2018.2870052, 16 September 2018.

[15] OWASP, “OWASP Top Ten,” September 2024. [Online]. Available: https://owasp.org/www-project-top-ten/. [Accessed 26 September 2024].

[16] Mitre, “CWE List Version 4.15,” Mitre, 29 February 2024. [Online]. Available: https://cwe.mitre.org/data/index.html. [Accessed 26 September 2024].

[17] B. R. Maddireddy and B. R. Maddireddy, “Real-Time Data Analytics with AI: Improving Security Event Monitoring and Management,” *Unique Endeavor in Business & Social Sciences*, vol. 1, no. 2, pp. 47-62. Available: https://unbss.com/index.php/unbss/article/view/42 [Accessed: 4 July 2024], 6 June 2022.

[18] Z. Li, D. Zou, S. Xu, X. Ou, H. Jine, S. Wang, Z. Deng and Y. Zhong, “VulDeePecker: A Deep Learning-Based System for Vulnerability Detection,” arxiv, 2018.

[19] D. Dave, G. Sawhney, P. Aggarwal, N. Silswal and D. Khut, “The New Frontier of Cybersecurity: Emerging Threats and Innovations,” in *2023 29th International Conference on Telecommunications (ICT),*, Toba, Indonesia, 2023.

[20] T. Fahmawi, A. Nabot, I. Jebreen and A. Al-Qerem, “Exploring Code Vulnerabilities through Code Reviews: An Exploring Code Vulnerabilities through Code Reviews: An Empirical Study on OpenStack Nova Empirical Study on OpenStack Nova,” *Journal of Statistics Applications & Probability*, vol. 13, no. 2 | Article 10, pp. 681-689. doi: http://dx.doi.org/10.18576/isl/130208, 1 March 2024.

[21] P. Ogini, D. E. Taylor and D. N. Nwiabu, “A Deep Learning Approach for The Detection of Structured Query Language Injection Vulnerability,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 11, no. 5, pp. 211-217. doi: https://doi.org/10.30534/ijatcse/2022/051152022 , 6 October 2022.

[22] I. P. Zengeni and M. F. B. Zolkipli, “Zero-Day Exploits and Vulnerability Management,” *Borneo International Journal*, vol. 7, no. 3, pp. 26-33. [Online]. Available: https://majmuah.com/journal/index.php/bij/article/view/648/329. [Accessed: 26th September 2024]., 1 September 2024.

[23] B. Hammi, S. Zeadally and J. Nebhen, “Security Threats, Countermeasures, and Challenges of Digital Supply Chains,” *ACM Digital Library*, vol. 55, no. 14s | Article No 316, pp. 1-40. doi: https://doi.org/10.1145/3588999, 17 July 2023.

[24] S. Oladimeji and S. M. Kerner, “SolarWinds hack explained: Everything you need to know,” TechTarget, 3 November 2023. [Online]. Available: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know. [Accessed 26 September 2024].

[25] A. Houerbi, R. G. Chavan, D. E. Rzig and F. Hassan, “Empirical Analysis on CI/CD Pipeline Evolution in Machine Learning Projects,” arXiv, 2024.

[26] V. K. Thatikonda, “Beyond the Buzz: A Journey Through CI/CD Principles and Best Practices,” *European Journal of Theoretical and Applied Sciences*, vol. 1, no. 5, pp. 334-340. [Online]. Available: doi: https://doi.org/10.59324/ejtas.2023.1(5).24 [Accessed: 25th September 2024]., 2023.

[27] T. W. Thomas, “The intersection of static analysis and security code reviews: A collaborative model,” *International Journal of Engineering in Computer Science*, vol. 5, no. 2, pp. 6-12. doi: https://doi.org/10.33545/26633582.2023.v5.i2a.93 , 22 June 2023.

[28] N. Pakovskie, “DeepCode: Revolutionizing Code Review with AI-Powered Bug Detection,” 12 November 2023. [Online]. Available: https://www.geekpedia.com/deepcode-ai-code-review-bug-detection/. [Accessed 25 September 2024].

[29] B. Berabi, G. Sivanrupan, A. Gronskiy, V. Chibotaru, V. Raychev and M. Vechev, “DeepCode AI Fix: Fixing

Security Vulnerabilities with Large Language Models," arXiv, 2024.

[30] V. Bhutani, F. G. Toosi and J. Buckley, "Analysing the Analysers: An Investigation of Source Code Analysis Tools," *Applied Computer Systems,* vol. 29, no. 1, pp. 98-111. doi: https://doi.org/10.2478/acss-2024-0013, 15 August 2024.

[31] F. Kilonzi, "What is Shift Left Security," Orca Security, 25 July 2024. [Online]. Available: https://orca.security/resources/blog/what-is-shift-left-security/. [Accessed 26 September 2024].

[32] N. Pakalapati, B. K. Konidena and I. A. Mohamed, "Unlocking the Power of AI/ML in DevSecOps: Strategies and Best Practices," *Journal of Knowledge Learning and Science Technology,* vol. 2, no. 2, pp. 176-188. [Online]. Available: https://doi.org/10.60087/jklst.vol2.n2.p188.[Accessed: 25th September 2024]., 12 July 2023.

[33] N. G. Camacho, "Unlocking the Potential of AI/ML in DevSecOps: Effective Strategies and Optimal Practices," *Journal of Artificial Intelligence General Science (JAIGS),* vol. 2, no. 1, pp. 78-89. doi: https://doi.org/10.60087/jaigs.v2i1.p89, 6 March 2024.

[34] J. Res, I. Homoliak, P. Martin, A. Smrčka, K. Malinka and P. Hanacek, "Enhancing Security of AI-Based Code Synthesis with GitHub Copilot via Cheap and Efficient Prompt-Engineering," arXiv, 2024.

[35] R. Naidoo and N. Moller, "Building Software Applications Securely With DevSecOps: A SocioTechnical Perspective," in *Proceedings of the 21st European Conference on Cyber Warfare and Security(ECCWS),* 2022.

[36] N. U. Baki, R. M. Rasdi, S. E. Krauss and M. K. Omar, "Integrating Artificial Intelligence in Human Resource Functions: Challenges and Opportunities," *International Journal of Academic Research in Business and Social Sciences,* vol. 13, no. 8, pp. 1262-1277. doi: DOI:10.6007/IJARBSS/v13-i8/18071, 18 August 2023.

[37] Z. Bilgin, M. A. Ersoy, E. U. Soykan, E. Tomur, P. Comak and L. Karacay, "Vulnerability Prediction From Source Code Using Machine Learning," *IEEE Xplore,* vol. 8, pp. 150672-150684. doi: 10.1109/ACCESS.2020.3016774, 14 August 2020.

[38] W. Albattah and M. Alzahrani, "Software Defect Prediction Based on Machine Learning and Deep Learning Techniques: An Empirical Approach," *AI,* vol. 5, no. 4, pp. 1743-1758. doi: https://doi.org/10.3390/ai5040086, 2024.

[39] S. Gawde and e. al., "Explainable Predictive Maintenance of Rotating Machines Using LIME, SHAP, PDP, ICE," *IEEE Access,* vol. 12, pp. 29345-29361. doi: 10.1109/ACCESS.2024.3367110, February 2024.

[40] {}, "ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering," *ISO/IEC/IEEE 29148:2018(E),* pp. 1-104. doi: 10.1109/IEEESTD.2018.8559686}, 30 November 2018.

[41] M. Taeb, H. Chi and S. Bernadin, "Assessing the Effectiveness and Security Implications of AI Code Generators," *2024 Journal of The Colloquium for Information Systems Security Eductaion (CISSE),* vol. 11, no. 1, p. doi: https://doi.org/10.53735/cisse.v11i1.180, February 2024.

[42] C. Tantithamthavorn, J. Cito, H. Hemmati and S. Chandra, "Explainable AI for SE: Challenges and Future Directions," *IEEE Software,* vol. 40, no. 3, pp. 29-33. doi: 10.1109/MS.2023.3246686, May-June 2023.

[43] T. E. Gasiba, K. Oguzhan, I. Kessba, U. Lechner and M. Pinto-Albuquerque, "I'm Sorry Dave, I'm Afraid I Can't Fix Your Code: On ChatGPT, CyberSecurity, and Secure Coding," in *4th International Computer Programming Education Conference (ICPEC 2023),* Dagstuhl, Germany, 2023.

[44] P. Nath, J. R. Mushahary, U. Roy, M. Brahma and P. K. Singh, "AI and Blockchain-based source code vulnerability detection and prevention system for multiparty software development," *Computers and Electrical Engineering* , vol. 106, pp. 1-15. https://doi.org/10.1016/j.compeleceng.2023.108607, March 2023.

[45] V. D. Kirova, C. S. Ku, J. R. Laracy and T. J. Marlowe, "The Ethics of Artificial Intelligence in the Era of Generative AI," *Journal of Systemics, Cybernetics and Informatics,* vol. 21, no. 4, pp. 42-50. doi: https://doi.org/10.54808/JSCI.21.04.42, 2023.

[46] European Parliament, "EU AI Act: first regulation on artificial intelligence," 18 June 2024. [Online]. Available: https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence. [Accessed 25 September 2024].

[47] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Research in Psychology,* vol. 3, no. 2, pp. 77-101. doi: https://doi.org/10.1191/1478088706qp063oa, 2006.

[48] V. Braun and V. Clarke, "Reflecting on reflexive thematic analysis," *Qualitative Research in Sport, Exercise and Health,* vol. 11, no. 4, pp. 589-597. doi: https://doi.org/10.1080/2159676X.2019.1628806, 13 June 2019.

[49] P. Bhandari, "Correlation Coefficient | Types, Formujlas & Examples," 2 August 2021. [Online]. Available: https://www.scribbr.com/statistics/correlation-coefficient/. [Accessed 23 October 2024].

[50] G. Guest, K. M. MacQueen and E. E. Namey, Applied Thematic Analysis, Thousand Oaks, CA: SAGE Publications, Inc., 2012, p. doi: https://doi.org/10.4135/9781483384436.

[51] L. S. Nowell, J. M. Norris and N. J. Mouiles, "Thematic Analysis: Striving to Meet the Trustworthiness Criteria," *International Journal of Qualitative Methods,* vol. 16, no. 1, p. doi: https://doi.org/10.1177/160940691773384, 2017.

APPENDICES

Table of Contents

1. GOV.UK, "Data Protection Act 2018," legistlation.gov.uk, 2024. [Online]. Available: https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted. [Accessed 2024 September 2024].

Summary
The *Data Protection Act 2018* (DPA 2018) is a comprehensive legal framework in the United Kingdom that governs the processing of personal data, ensuring it aligns with the General Data Protection Regulation (GDPR). The Act is designed to protect the privacy and rights of individuals by regulating how organisations collect, use, and store personal data. It includes provisions on data subjects' rights, data controllers' and processors' responsibilities, and principles of lawful processing. It also outlines the Information Commissioner's powers to enforce compliance and impose penalties for violations. The Act incorporates GDPR provisions into UK law and addresses areas not covered by the GDPR, such as exemptions for national security and law enforcement purposes.

Credibility
The *Data Protection Act 2018* is primary legislation passed by the UK Parliament, which adds a high level of credibility and authority. It serves as the key legislative instrument for personal data protection in the UK and provides a legal basis for enforcing GDPR standards. The legislation.gov.uk website is an official government source that ensures the content's accuracy and reliability.

Reflection
This source is highly relevant to this dissertation on AI-enhanced secure software engineering, particularly in understanding how data protection laws impact AI and machine learning practices. It provides the legal backdrop against which secure coding practices must be developed, especially in contexts involving the processing of personal data. Additionally, the Act's emphasis on transparency and individual rights aligns well with the focus on Explainable AI (XAI), offering insights into regulatory compliance and ethical considerations for AI systems handling sensitive data.

2. **National Institute of Standards And Technology, "The NIST Cybersecurity Framework(CSF) 2.0," US Dept of Commerce, 2024.**

Summary
The NIST Cybersecurity Framework (CSF) 2.0 provides organisations with a set of best practices, standards, and guidelines to manage and reduce cybersecurity risk. CSF 2.0 offers a comprehensive framework for identifying, protecting, detecting, responding to, and recovering from cyber incidents. It is particularly notable for its adaptability to different sectors and scalability for organisations of all sizes, making it a widely adopted cybersecurity tool both in the United States and internationally.

Credibility
The National Institute of Standards and Technology (NIST) is a reputable governmental entity specializing in technology and standards development. NIST has a long-standing history of establishing robust cybersecurity standards, contributing significantly to enhancing national cybersecurity capabilities. The CSF is supported by expert consensus and continuously updated in consultation with industry stakeholders, adding to its credibility.

Reflection
The CSF 2.0 aligns well with the focus on this research on AI-enhanced secure software development. The framework's adaptable structure could be beneficial for examining the integration of AI tools within secure software practices, particularly how AI can enhance each of the framework's core functions. Its emphasis on risk management is also pertinent for discussing the trade-offs between transparency and security when integrating AI, thereby providing a foundation for evaluating best practices in secure software engineering and AI transparency.

3. 14. **A. H. e. a. Mohammadkhani, "A Systematic Literature Review of Explainable AI for Software Engineering,"** arXiv, 2023.https://arxiv.org/abs/2302.06065.

Summary

This paper presents a systematic literature review of Explainable AI (XAI) in the context of software engineering (SE), referred to as XAI4SE. The authors reviewed 24 of the 869 relevant studies to provide a comprehensive analysis of XAI techniques applied in SE. The review highlights that software maintenance, particularly defect prediction, is the most common use case for XAI, but there is a significant lack of exploration in generation-based SE tasks. The paper discusses the different XAI methods used, such as LIME and ANOVA, and explores their effectiveness in enhancing the transparency of machine learning models in SE.

Credibility
The authors are affiliated with well-known institutions across Canada, India, Tunisia, and Australia. The paper is published on arXiv, a reputable open-access repository widely used for sharing preprints in computer science and software engineering. The systematic approach, as well as the involvement of experts from multiple institutions, lends credibility to the work.

Reflection
This source provides a valuable basis for understanding the application and limitations of XAI techniques in secure software engineering, aligning well with my dissertation's focus on transparency and trust in AI-enhanced secure coding. It informs my research on how XAI tools are being utilized, the areas that lack exploration, and the challenges of using explainability in software engineering contexts.

4. Y. Shi, N. Sakib, H. Shahriar, D. Lo, H. Chi and K. Qian, "AI-Assisted Security: A Step towards Reimagining Software Development for a Safer Future," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, Torino. DOI: 10.1109/COMPSAC57700.2023.00142

Summary
This paper investigates the integration of AI into software security practices, focusing on enhancing security within the software development lifecycle. The authors highlight how AI-driven tools can automate security tasks, detect vulnerabilities, and reduce human errors. Specific AI techniques discussed include machine learning models for threat detection and natural language processing for automated code analysis. The paper presents various examples of AI applications in secure software practices, emphasizing the efficiency of AI in handling large datasets compared to traditional methods.

Credibility
The authors, affiliated with reputable institutions such as Florida A&M University and Kennesaw State University, provide a credible analysis rooted in extensive research. The paper's inclusion in the IEEE COMPSAC conference enhances its reliability. Strengths include a comprehensive review of AI techniques for software security, supported by practical case studies and comparative analyses of AI versus traditional security approaches. However, the paper lacks in-depth discussion on the explainability of AI tools, which is a crucial aspect for broader adoption in secure software development. Additionally, there is minimal coverage on the ethical considerations of using AI in security.

Reflection
This source is directly relevant to the dissertation's focus on AI-enhanced secure software development, particularly in automating security processes. It offers foundational insights into the advantages of AI-driven security tools, which can be used to support arguments for improved software security practices. The identified gap regarding explainability is particularly significant for the dissertation's emphasis on Explainable AI (XAI), providing an opportunity to explore the challenges and implications of integrating XAI into secure software engineering. The paper's practical examples and case studies will also be useful for illustrating real-world applications of AI in the dissertation**.**

5. D. R. Chittibala, "Advancements in automated code scanning techniques for detecting security vulnerabilities in open source software.," *International Journal of Computing and Engineering,* vol. 5, no. 2, pp. 16-25. doi: https://doi.org/10.47941/ijce.1737, 21 March 2024.

Summary
This paper examines the evolving role of automated code scanning techniques in detecting security vulnerabilities within open-source software (OSS). The author discusses various scanning methodologies, including static analysis, dynamic analysis, and the integration of machine learning to enhance detection accuracy. The paper highlights the benefits of using automated tools in identifying vulnerabilities early in the software development lifecycle, particularly in OSS projects, which often face unique challenges due to rapid development cycles and collaborative contributions.

Credibility
Dinesh Reddy Chittibala, affiliated with Salesforce Inc., provides a credible perspective based on his experience in software engineering and security. The paper is published in a peer-reviewed journal, enhancing its reliability. A significant strength of this source is its detailed discussion on different automated scanning techniques, supplemented by insights into the use of AI and machine learning for improved vulnerability detection. However, the paper lacks specific case studies or empirical

evidence to validate the effectiveness of these methodologies in real-world OSS environments. Additionally, while the author emphasizes the importance of scalability, the practical limitations of implementing these technologies in large OSS projects are not addressed comprehensively.

Reflection
This source is pertinent to the dissertation's exploration of AI-enhanced secure software development, particularly in its focus on automated vulnerability detection methods. It provides valuable background on different code scanning techniques, which can be used to support the discussion on tools and methodologies for secure coding. The absence of empirical validation and scalability considerations presents an opportunity to delve deeper into these aspects, aligning with the dissertation's emphasis on practical challenges and solutions in integrating AI-driven security in software development.

6. **H. Pearce and e. al., "Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions," in *2022 IEEE Symposium on Security and Privacy (SP)*, San Francisco, 2022.**

Summary
This paper investigates the security implications of using GitHub Copilot, an AI-powered code generation tool. The authors systematically evaluate the code Copilot generates by analysing its vulnerability to security risks, particularly focusing on weaknesses identified in the MITRE's "Top 25" Common Weakness Enumeration (CWE) list. By generating 1,689 programs across 89 different scenarios, the researchers found that approximately 40% of the generated code was vulnerable. The study identifies Copilot's tendency to produce insecure code, particularly in high-risk cybersecurity contexts, and highlights the need for careful human oversight by developers when using AI-assisted coding tools like Copilot. The paper emphasizes that while Copilot can significantly enhance productivity, it should be used in conjunction with security-aware practices and tools to minimize risks.

Credibility
The paper was published at the 2022 IEEE Symposium on Security and Privacy (SP), a reputable venue known for rigorous peer-reviewed research in cybersecurity. The study is authored by a team of experienced researchers, with funding support from credible institutions such as the National Science Foundation (NSF) and the Office of Naval Research (ONR). The use of well-established tools like GitHub's CodeQL for automatic vulnerability detection, along with manual inspections, adds to the robustness of the study. Additionally, the authors' systematic approach to evaluating a range of scenarios underlines the validity of the findings.

Reflection
This paper is relevant to my research on the integration of Explainable AI (XAI) in secure software engineering. It offers valuable insights into the security risks associated with AI-driven code generation, a topic of growing importance in both academia and industry. The authors' focus on CWE vulnerabilities aligns with my research's emphasis on ensuring transparency and security in AI-generated outputs. Additionally, the study's findings on Copilot's limitations in generating secure code highlight the importance of developing XAI tools that can explain and justify AI-generated decisions, which is a crucial aspect of my dissertation. This paper also serves as a critical reference for understanding the practical implications of deploying AI in software development.

7. **T. Rangnau, R. v. Buijtenen, F. Fransen and F. Turkmen, "Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines," in *2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC)*, Eindhoven, 2020.**

Summary
This paper presents a case study on integrating dynamic security testing techniques into Continuous Integration/Continuous Delivery (CI/CD) pipelines, a key practice in DevSecOps. The authors argue that traditional security practices cannot match the speed and agility of DevOps, emphasizing the need for continuous security integration through automation. They implemented three automated Dynamic Application Security Testing (DAST) techniques—Web Application Security Testing (WAST) using OWASP ZAP, Security API Scanning (SAS) using JMeter, and Behaviour-Driven Security Testing (BDST) using SeleniumBase—in a CI/CD pipeline to identify challenges, pitfalls, and requirements for successful integration. The study highlights specific challenges such as maintaining acceptable build times, managing containerization, and coping with increased test complexity.

Credibility
The authors, affiliated with recognized institutions like the University of Groningen and TNO, lend credibility to the research through their academic and industry expertise. The systematic case study approach ensures a practical, evidence-based discussion of integrating security in DevOps environments. The strength of this paper lies in its practical insights into using multiple DAST tools for a more comprehensive security posture, supported by empirical performance evaluations. However, the paper lacks a detailed exploration of static testing techniques and their complementary role alongside DAST, which is a limitation when considering a holistic security approach.

Reflection
This source is highly relevant to the dissertation's focus on enhancing secure software engineering practices through AI tools in the context of DevSecOps. The challenges identified, such as managing containerization and maintaining quick build times, provide valuable context for understanding the technical barriers in adopting Explainable AI (XAI) for secure coding. Additionally, the emphasis on the combined use of multiple DAST tools aligns well with the dissertation's theme of achieving transparency and adaptability in secure software development, which are key components in integrating XAI for better decision-making and transparency.

8. **W. Charoenwet, P. Thongtanunam, V.-T. Pham and C. Treude, "Towardeffective secure code reviews: an empirical study of security-related coding weaknesses,"** ***Empirical Software Engineering,*** **vol. 29, no. 88, pp. 1-47. https://doi.org/10.1007/s10664-024-10496-y , 8 June 2024.**

Summary
This study explores the effectiveness of code reviews in identifying security-related coding weaknesses in two large open-source projects, OpenSSL and PHP. The authors analyse 135,560 code review comments to determine the prevalence and treatment of security issues. The study finds that security concerns were raised in 35 out of 40 categories of coding weaknesses, but certain weaknesses related to past vulnerabilities were discussed less frequently. The findings indicate that while code reviews are effective at identifying various security concerns, coding weaknesses often remain unfixed or insufficiently addressed due to disagreements or incomplete review processes.

Credibility
The authors are affiliated with well-regarded institutions such as the University of Melbourne and Singapore Management University, which supports the credibility of the study. The paper's strengths include its extensive dataset and rigorous empirical analysis of over 135,000 code review comments, providing a broad insight into security practices in open-source software. However, the study is limited by its focus on only two open-source projects, which may not fully generalize to other domains or smaller projects. Additionally, while the authors provide insights into the prevalence of coding weaknesses, the lack of detailed solutions for effectively addressing these weaknesses in practice could be seen as a gap.

Reflection
This paper is highly relevant to the dissertation's focus on AI-enhanced secure software engineering, specifically in exploring the limitations of manual processes such as code reviews. The discussion on the challenges in effectively addressing coding weaknesses aligns with the dissertation's emphasis on the need for Explainable AI (XAI) to improve transparency and mitigate security issues. The insights from this study provide a strong foundation for understanding where traditional secure coding practices may fall short and how AI-driven solutions could enhance the efficiency and completeness of code reviews.

9. **A. Bosu and e. al., "Identifying the characteristics of vulnerable code changes: an empirical study," in** ***FSE 2014: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*****, New York, 2014.**

Summary
This empirical study aims to identify characteristics of vulnerable code changes (VCC) and understand the relationship between code review processes and security vulnerabilities in open-source projects. The authors analysed 267,046 code review requests from 10 open-source projects, identifying 413 VCCs. Key findings include that experienced contributor authored most VCCs, but changes by less experienced contributors were significantly more likely to be vulnerable. Additionally, the study found that modified files are more prone to vulnerabilities compared to new files and that the likelihood of a vulnerability increases with the number of lines changed. The researchers also recommend secure coding guidelines, dedicated security review teams, and encouraging smaller, incremental changes.

Credibility
The paper is published in the proceedings of the ACM SIGSOFT Symposium, a reputable venue for software engineering research. The authors, affiliated with respected academic institutions like the University of Alabama and Auburn University, have backgrounds in software engineering and security. The use of data from 10 popular open-source projects and a robust combination of manual and automated analysis techniques adds credibility to the findings. The empirical methodology, focusing on real-world data from peer code review processes, further enhances the paper's reliability.

Reflection
This study aligns well with my research on AI-enhanced secure software engineering, particularly in understanding the characteristics of vulnerabilities within code changes and the role of human factors. The insights on the importance of peer code reviews and the challenges experienced by developers resonate with my focus on transparency and integrating AI to improve security. The recommendations for secure coding guidelines and a dedicated security review team are relevant for exploring the integration of explainable AI (XAI) into secure coding practices to address similar challenges. The empirical

evidence provided can serve as a foundation for discussing how AI tools might enhance code reviews by identifying vulnerabilities earlier in the development process.

10. **F. Zampetti and e. al., "Continuous Integration and Delivery Practices for Cyber-Physical Systems: An Interview-Based Study,"** ***ACM Transactions on Software Engineering and Methodology,,*** **no. 73, pp. 1-44. doi: https://doi.org/10.1145/3571854 , 26 April 2023.**

Summary
This study by Zampetti et al. examines the challenges and barriers faced by organisations in implementing Continuous Integration and Delivery (CI/CD) practices for Cyber-Physical Systems (CPS). Conducted through semi-structured interviews with 10 organisations from diverse domains, and validated with a survey involving 55 developers, the study highlights the peculiarities of applying CI/CD in CPS development. Key challenges include managing simulators and Hardware-in-the-Loop (HiL), difficulties in deployment, and the need for expertise across both hardware and software disciplines. The research concludes by providing recommendations for setting up CI/CD pipelines for CPS, suggesting specific educational improvements for CPS developers, and identifying areas for future research.

Credibility
The authors are affiliated with reputable institutions like the University of Sannio, Eindhoven University of Technology, Zurich University of Applied Sciences, and Delft University of Technology. Their credibility is further strengthened by their affiliations with projects supported by Horizon 2020 (EU Commission). The study's methodology, including semi-structured interviews and surveys, is appropriate for understanding the nuanced challenges of CPS development. Moreover, the use of multiple data collection methods, such as card sorting and member-checking, adds rigor and validity to the findings. The publication in the ACM Transactions on Software Engineering and Methodology, a peer-reviewed journal, also ensures a high standard of scholarly quality.

Reflection
This paper is highly relevant to my dissertation on AI-enhanced secure software engineering with a focus on explainable AI (XAI). The discussion of CI/CD barriers and the need for specific fault models in CPS closely relates to the integration challenges I am exploring in the adoption of secure coding practices enhanced by AI. The emphasis on domain-specific requirements, such as simulators and HiL, adds valuable context to the practical constraints of secure software development environments. Furthermore, the insights about balancing the use of continuous and periodic builds provide important lessons for implementing AI-driven solutions for vulnerability detection in real-world systems.

11. **L. Chmioelowski, M. Kucharzak and R. Burduk, "Application of Explainable Artificial Intelligence in Software Bug Classification,"** ***IAPGOŚ*** **, vol. 13, no. 1, pp. 14-17. doi: http://doi.org/10.35784/iapgos .3396, 2023.**

Summary
This study explores the use of Explainable Artificial Intelligence (XAI) for automating the bug classification process in software development, specifically focusing on distinguishing between security-related and non-security-related bugs. The authors utilize two different datasets for evaluation: one derived from a telecommunications company and another from the Mozilla Defect Dataset. Both sets underwent classification using XAI techniques, including a decision tree classifier, which allowed the generation of interpretable rules for decision-making. The findings indicated that XAI models were able to achieve comparable accuracy to standard black-box models, while providing the added benefit of transparency in decision-making.

Credibility
The authors are affiliated with reputable institutions, including Nokia Solutions and Networks and the Wroclaw University of Science and Technology. The study was published in a peer-reviewed journal, *IAPGOS*, with ISSNs 2083-0157 and 2391-6761, which suggests that the article underwent editorial scrutiny. The methodology employs well-established machine learning techniques such as decision trees, k-Nearest Neighbours, and Support Vector Classifier, adding to the reliability of the results.

Reflection
This study is highly relevant to my research on AI-enhanced secure software engineering, particularly regarding the use of XAI for software security. It demonstrates the potential benefits of integrating explainable models into bug triaging to enhance transparency without compromising performance. The findings will help inform the comparison of XAI models with traditional AI models in secure coding, providing evidence that explainable models can facilitate decision-making in critical environments without losing accuracy.

12. **F. Charmet and e. al., "Explainable artificial intelligence for cybersecurity: a literature survey,"** ***Annals of Telecommunications,*** **vol. 77, pp. 789-812. doi: https://doi.org/10.1007/s12243-022-00926-7 , 26 October 2022.**

Summary
This paper provides an extensive literature review on the intersection of Explainable AI (XAI) and cybersecurity. It investigates the application of XAI to various cybersecurity tasks, including intrusion detection and malware classification, and examines the security of XAI models. The review highlights how XAI can help security operators manage vast numbers of security alerts, reducing false positives and enhancing the decision-making process. The authors outline the challenges faced by XAI in cybersecurity, such as adversarial attacks and privacy concerns, and identify open research questions and future research directions, emphasizing the balance between transparency, performance, and security.

Credibility
The authors are affiliated with prominent institutions like the National Institute of Information and Communications Technology (Japan), Universite de Lorraine (France), and Huawei Paris Research Centre (France). The journal *Annals of Telecommunications* is peer-reviewed, adding credibility to the work. The paper employs a structured approach to reviewing a large body of literature, utilizing various databases and a robust methodology, which further strengthens its reliability.

Reflection
This literature survey is highly pertinent to my dissertation on AI-enhanced secure software engineering, particularly in the context of XAI applications for cybersecurity. It provides a thorough understanding of the opportunities and challenges associated with applying XAI in secure coding practices. The paper's discussion on balancing model explainability, performance, and security is crucial for my analysis of the practical integration of XAI in secure software development workflows.

13. **A. B. Arrieta and e. al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,"** ***Information Fusion,*** **vol. 58, pp. 82-115. doi: https://doi.org/10.1016/j.inffus.2019.12.012 , June 2020.**

Summary
This paper provides a comprehensive overview of Explainable AI (XAI), including a taxonomy of XAI methods, a discussion of the trade-offs between interpretability and model performance, and an exploration of how XAI intersects with privacy, fairness, and accountability. The authors define XAI and Responsible AI, outlining the role of explainability in making AI models more transparent and trustworthy. They emphasize the importance of explainability in critical sectors like medicine and autonomous systems and present a taxonomy that categorizes existing XAI approaches into those that focus on transparency and those that employ post-hoc explanations for opaque models. They also discuss the ethical challenges associated with XAI and present future directions for responsible AI.

Credibility
The authors are affiliated with various notable institutions, including the National Institute of Information and Communications Technology (Japan), Universite de Lorraine (France), and Huawei Paris Research Center (France). Published in *Information Fusion*, a reputable peer-reviewed journal, the paper's comprehensive literature review and focus on taxonomies provide an authoritative foundation for understanding XAI's current landscape. The paper cites approximately 400 contributions, demonstrating a thorough analysis of the XAI literature.

Reflection
This paper is highly relevant to my dissertation on AI-enhanced secure software engineering, specifically regarding XAI's role in secure coding practices. The discussion on balancing interpretability with performance is critical for assessing the trade-offs inherent in integrating XAI into secure software development. Moreover, the concept of Responsible AI aligns well with my exploration of explainable and secure coding practices, providing an ethical and practical framework for implementation.

14. **A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI),"** ***IEEE Access,*** **vol. 6, pp. 52138-52160. doi: 10.1109/ACCESS.2018.2870052, 16 September 2018.**.

Summary
This paper provides a survey of the state-of-the-art approaches to Explainable Artificial Intelligence (XAI). The authors explore the growing demand for transparency in AI systems and the challenges that arise from the black-box nature of many machine learning models. The paper reviews the current landscape of XAI, presenting major research trajectories, and categorizes existing XAI methods based on their approaches to improving transparency and explainability. The authors emphasize the multidisciplinary nature of the field and identify the main players contributing to XAI research, including DARPA and industrial leaders such as Microsoft and FICO. The survey also discusses future directions for advancing XAI, such as integrating human-machine teaming and formalizing evaluation metrics.

Credibility

Amina Adadi and Mohammed Berrada are affiliated with the Computer and Interdisciplinary Physics Laboratory at Sidi Mohammed Ben Abdellah University, Morocco. The article is published in *IEEE Access*, a peer-reviewed, highly reputable open-access journal that publishes significant advancements across various domains. The survey draws on 381 research papers, indicating a comprehensive literature review, and provides valuable insights into the development and challenges of XAI.

Reflection
This paper is highly relevant to my dissertation, especially in understanding the different approaches and techniques used to achieve explainability in AI systems. The survey's categorization of existing XAI methods provides a useful framework for comparing the transparency of AI models, which aligns with my research focus on enhancing secure software development using XAI. Additionally, the discussion on the challenges of balancing explainability and performance helps inform the limitations and opportunities in applying XAI to secure software engineering.

15. **OWASP Foundation.** "OWASP Top Ten Web Application Security Risks." *OWASP Foundation*, 2021, https://owasp.org/www-project-top-ten/.

Summary
The OWASP Top Ten is a well-known, regularly updated list that outlines the most critical web application security risks. Developed by the Open Web Application Security Project (OWASP), the document aims to raise awareness among developers and security professionals about prevalent vulnerabilities such as injection flaws, broken authentication, and insecure design. The latest edition, published in 2021, highlights emerging risks, including insecure design and software supply chain vulnerabilities, reflecting the evolving landscape of web security threats.

Credibility
The OWASP Foundation is a reputable nonprofit organization that focuses on improving the security of software. The OWASP Top Ten list is widely recognized as an industry standard, endorsed by security professionals, researchers, and organisations worldwide. It serves as a foundational resource for understanding web application security risks, ensuring best practices are followed in software development.

Reflection
This resource is highly relevant to my research on AI-enhanced secure software development, particularly in identifying and mitigating risks that could be addressed through AI-based vulnerability detection. The OWASP Top Ten can serve as a benchmark for evaluating the effectiveness of AI tools in secure coding and guide the integration of explainable AI (XAI) techniques to ensure that detected vulnerabilities are transparent and actionable for developers.

16. **Mitre, "CWE List Version 4.15," Mitre, 29 February 2024. [Online]. Available: https://cwe.mitre.org/data/index.html . [Accessed 26 September 2024].**

Summary
The Common Weakness Enumeration (CWE) List Version 4.15 is a comprehensive catalogue of software weaknesses and vulnerabilities, curated by Mitre. It includes various categories of software flaws, such as coding errors, design issues, and architectural deficiencies, which are common in software systems. Version 4.15, released in 2024, emphasizes new entries and modifications to reflect evolving security threats and introduces updated severity rankings for better risk management.

Credibility
Mitre is a reputable organization that supports various government and industry initiatives to enhance cybersecurity. The CWE List is widely used by developers, security analysts, and organisations to identify and mitigate software weaknesses during the development lifecycle. Its credibility is further reinforced by its use as a foundation for security assessments and standards compliance, including its adoption by the National Institute of Standards and Technology (NIST).

Reflection
This resource is crucial for my research on AI-enhanced secure software development, as it provides a comprehensive understanding of specific vulnerabilities that AI tools can help identify and remediate. It serves as an authoritative guide for evaluating the performance and transparency of AI-driven vulnerability detection, aligning well with the principles of Explainable AI (XAI) and industry security standards, such as ISO/IEC 27001 and NIST.

17. **B. R. Maddireddy and B. R. Maddireddy, "Real-Time Data Analytics with AI: Improving Security Event Monitoring and Management," *Unique Endeavor in Business & Social Sciences,* vol. 1, no. 2, pp. 47-62. Available: https://unbss.com/index.php/unbss/article/view/42 [Accessed: 4 July 2024], 6 June 2022.**

Summary

This paper explores the integration of Artificial Intelligence (AI) and real-time data analytics to enhance security event monitoring and management. The authors propose a framework utilizing machine learning techniques, including deep learning and anomaly detection, to detect and respond to cybersecurity threats in real time. The AI-driven system leverages big data technologies to identify patterns indicative of potential security breaches, achieving a detection accuracy of 94.5% with a false positive rate of 2.1%. The integration of real-time data analytics provides continuous monitoring, allowing proactive incident response and reducing the vulnerability window. The paper also addresses the challenges of deploying AI in cybersecurity, such as scalability and privacy concerns.

Credibility
The authors, both network security professionals at Voya Financials, demonstrate practical expertise in cybersecurity. The paper was published in *Unique Endeavor in Business & Social Sciences*, a journal that seems to be relatively new, and the article is available under a Creative Commons license, suggesting a desire for open dissemination. The practical experience of the authors lends credibility, though the journal's lack of established reputation may necessitate caution regarding its academic rigor.

Reflection
This paper offers practical insights into AI-driven security monitoring systems, aligning well with my dissertation's emphasis on the integration of AI for secure software development. It provides a real-world perspective on the effectiveness of AI in proactive cybersecurity, with metrics that demonstrate the potential advantages of AI-enhanced threat detection. The discussion on deployment challenges and ethical considerations also informs my research on balancing efficiency with responsible AI use.

18. **Li, Zhen, et al.** "VulDeePecker: A Deep Learning-Based System for Vulnerability Detection." *Proceedings of the Network and Distributed Systems Security (NDSS) Symposium 2018*, San Diego, CA, USA, 18-21 February 2018, pp. 1-15.

Summary
This paper presents VulDeePecker, a system that leverages deep learning for the automatic detection of software vulnerabilities. It addresses the limitations of traditional approaches that require human experts to manually define features, which can be error-prone and inconsistent. VulDeePecker uses code gadgets—small, semantically related code fragments—to represent programs and applies bidirectional long short-term memory (BLSTM) networks to identify vulnerabilities. The system is tested on a dataset created specifically for this purpose, and its performance is compared with other static analysis tools, demonstrating a lower false negative rate and reasonable false positives. Notably, VulDeePecker was able to identify several vulnerabilities that were missed by other systems, including four "silently" patched vulnerabilities in real-world software products.

Credibility
The authors are affiliated with reputable institutions such as the School of Computer Science and Technology at Huazhong University of Science and Technology and the Department of Computer Science at the University of Texas at San Antonio. The paper was presented at the NDSS Symposium, a highly regarded conference in the field of cybersecurity. This lends significant credibility to the research, although the novelty of using deep learning for vulnerability detection suggests the results should be validated further with a broader dataset.

Reflection
This paper is highly relevant to my dissertation as it introduces a novel approach to automating vulnerability detection, which is critical for secure software engineering. Its use of deep learning aligns with my research focus on AI-enhanced secure coding. Additionally, VulDeePecker's ability to detect vulnerabilities without predefined feature engineering supports the aim of reducing human dependency in secure software development. The emphasis on reducing false negatives also ties in with my interest in enhancing the accuracy of AI tools for secure coding.

19. Dave, Daksh, Nitish Silswal, Gauransh Sawhney, Dhruv Khut, and Pushkar Aggarwal. "The New Frontier of Cybersecurity: Emerging Threats and Innovations." *2023 29th International Conference on Telecommunications (ICT)*. IEEE, 2023. DOI: 10.1109/ICT60153.2023.10374044.

Summary
This paper discusses the increasing variety and severity of cybersecurity threats that affect individuals, organisations, and governments. It categorizes these threats into four major groups: malware attacks, social engineering, network vulnerabilities, and data breaches. The study uses a qualitative research methodology to analyse the impacts of these threats and emphasizes a multi-layered approach to mitigating them, which includes employing strong passwords, encryption, employee training, and regular software updates. The paper identifies emerging threats such as advanced persistent threats (APTs), Internet of Things (IoT) vulnerabilities, and ransomware attacks, providing a comprehensive overview of current and future challenges in cybersecurity.

Credibility
The authors are affiliated with reputable institutions, such as BITS Pilani and Sardar Patel Institute of Technology, which contributes to the credibility of the research. The paper is peer-reviewed and presented at the 29th International Conference on Telecommunications, adding further legitimacy. The references cited are current, encompassing relevant studies published within the last five years, which is critical for ensuring the relevance of the research in the rapidly evolving field of cybersecurity. The use of both historical and emerging cybersecurity threats helps provide a well-rounded perspective.

Reflection
This paper aligns well with my research focus on AI-enhanced secure software development, especially with its emphasis on identifying vulnerabilities and emerging threats in cybersecurity. The categorization of threats and detailed analysis of trends provide a valuable foundation for understanding the types of vulnerabilities AI might help address. Additionally, the paper's discussion on evolving threats, such as APTs and IoT vulnerabilities, is particularly useful for highlighting gaps where Explainable AI (XAI) could improve transparency and trust in mitigating these risks. The focus on a multi-layered defence also resonates with the user's interest in aligning AI-driven secure coding with industry standards like ISO/IEC 27001 and NIST guidelines.

20. **T. Fahmawi, A. Nabot, I. Jebreen and A. Al-Qerem, "Exploring Code Vulnerabilities through Code Reviews: An Exploring Code Vulnerabilities through Code Reviews: An Empirical Study on OpenStack Nova Empirical Study on OpenStack Nova,"** ***Journal of Statistics Applications & Probability,*** **vol. 13, no. 2 | Article 10, pp. 681-689. doi: http://dx.doi.org/10.18576/isl/130208 , 1 March 2024.**

Summary
This empirical study investigates vulnerabilities uncovered during code reviews of the OpenStack Nova project, with an analysis of 4873 review comments. It identifies 187 potential vulnerabilities, of which 151 were confirmed. The findings highlight that injection flaws were the most common, while insecure deserialization was the least. The authors identify three main reasons for these vulnerabilities: developers' insufficient knowledge of secure coding practices, unfamiliarity with existing code, and unintentional mistakes. The paper emphasizes the importance of effective communication between reviewers and developers and suggests training in secure coding to improve software quality. The study also discusses the effectiveness of manual code review as opposed to relying solely on automated tools, which can miss context-sensitive issues.

Credibility
The authors are affiliated with the Faculty of Information Technology at Zarqa University, Jordan, which lends credibility to their work. The journal, *Journal of Statistics Applications & Probability*, is an international peer-reviewed publication, adding to the legitimacy of the research. The study's empirical nature, using data from the OpenStack project, provides a solid foundation for the conclusions drawn. However, it primarily focuses on a specific open-source project, which may limit the generalizability of the findings to other software projects or commercial environments. The paper uses a well-documented methodology, which enhances its reliability, although the reliance on manual classification of vulnerabilities could introduce subjectivity.

Reflection
This paper is directly relevant my dissertation, as it provides insight into the vulnerabilities that can emerge during software development and emphasizes the role of code reviews in identifying these issues. The discussion on the limitations of automatic detection tools and the benefits of manual code review aligns with the user's focus on Explainable AI (XAI) and its role in enhancing transparency in secure software development. Moreover, the identification of gaps, such as a lack of secure coding knowledge among developers, offers potential areas where AI tools could support learning and decision-making, thus improving the security of code through enhanced explainability and training modules.

21. **P. Ogini, D. E. Taylor and D. N. Nwiabu, "A Deep Learning Approach for The Detection of Structured Query Language Injection Vulnerability,"** ***International Journal of Advanced Trends in Computer Science and Engineering,*** **vol. 11, no. 5, pp. 211-217. doi: https://doi.org/10.30534/ijatcse/2022/051152022 , 6 October 2022.**.

Summary
This paper presents a deep learning-based model for detecting SQL injection attacks on web applications. The authors developed a feed-forward neural network trained on a dataset consisting of 30,635 SQL queries (both injected and non-injected) sourced from Kaggle. The dataset underwent preprocessing steps, including removing null values and tokenizing SQL statements. The model was trained using TensorFlow and achieved an accuracy of 97.65%. It outperformed existing models, demonstrating higher accuracy and better detection capabilities. The research highlighted that the model's performance was evaluated using confusion matrices and precision metrics, showing effective classification of both normal queries and SQL injection attacks. The system was deployed using Python Flask for real-time testing, and the authors suggested future extensions by combining different deep learning algorithms or deploying the model to mobile applications.

Credibility
The authors are affiliated with Rivers State University in Nigeria, which provides some academic credibility to their work. The paper was published in the *International Journal of Advanced Trends in Computer Science and Engineering*, a peer-reviewed journal, suggesting that the research has undergone academic scrutiny. The dataset used for training the model was obtained from Kaggle, a reliable source for datasets, and the use of modern tools such as TensorFlow and Python Flask adds to the technical robustness of the research. However, the study lacks information on cross-validation techniques, which might limit its reliability in diverse environments. Additionally, the focus on a single dataset could affect the model's generalizability across different SQL injection scenarios.

Reflection
This paper is relevant to my research on AI-enhanced secure software development, as it demonstrates the application of deep learning to enhance the detection of security vulnerabilities, specifically SQL injection attacks. The discussion of using neural networks to identify complex patterns aligns well with interest in integrating Explainable AI (XAI) into secure software engineering. The model's high accuracy rate is promising for enhancing the transparency and robustness of AI systems in detecting vulnerabilities. This paper also offers insights into the practical deployment of machine learning models, which could inform my exploration of integrating secure coding standards with AI-enhanced tools.

22. **I. P. Zengeni and M. F. B. Zolkipli, "Zero-Day Exploits and Vulnerability Management," *Borneo International Journal,* vol. 7, no. 3, pp. 26-33. [Online]. Available: https://majmuah.com/journal/index.php/bij/article/view/648/329. [Accessed: 26th September 2024]., 1 September 2024.**https://www.majmuah.com.

Summary
This paper explores the lifecycle of zero-day vulnerabilities, emphasizing the discovery, exploitation, disclosure, and patching phases. The authors examine the critical impact of zero-day exploits on enterprises, illustrating case studies such as the Log4Shell and Microsoft Exchange vulnerabilities, which caused significant disruptions and security risks. The paper highlights the importance of proactive vulnerability management, including early detection and swift response. The authors also discuss strategies such as bug bounty programs and responsible disclosure policies as essential measures for mitigating the risks associated with zero-day vulnerabilities. They stress the importance of collaboration with software vendors and advanced detection technologies to improve organizational resilience against evolving threats.

Credibility
The authors are affiliated with the School of Computing at Universiti Utara Malaysia, contributing to the credibility of the research. The paper is published in the *Borneo International Journal*, which adds to its legitimacy as a scholarly source. The discussion includes recent case studies like Log4Shell, ensuring that the research is relevant and current. However, the paper is limited by its qualitative approach, as it does not provide empirical data or quantitative analysis of the effectiveness of the proposed vulnerability management strategies. Despite this, the detailed examination of real-world cases and practical vulnerability management techniques strengthens the reliability of the content.

Reflection
This paper is highly relevant to my research on AI-enhanced secure software development, particularly regarding vulnerability management. The lifecycle analysis of zero-day vulnerabilities provides a clear framework for understanding how such exploits can be effectively managed. The focus on proactive measures, such as advanced detection systems and collaboration through bug bounty programs, aligns well with the user's interest in leveraging AI to improve secure coding practices and enhance transparency. This resource helps establish a foundational understanding of vulnerability management, which is crucial for the exploration of AI-driven solutions in secure software engineering.

23. **B. Hammi, S. Zeadally and J. Nebhen, "Security Threats, Countermeasures, and Challenges of Digital Supply Chains," *ACM Digital Library,* vol. 55, no. 14s | Article No 316, pp. 1-40.** doi: https://doi.org/10.1145/3588999 , 17 July 2023.

Summary
This paper presents a comprehensive survey of security threats, countermeasures, and challenges associated with digital supply chains (DSCs). The authors provide an overview of how the evolution of Information Communication Technologies (ICT) has impacted supply chains, making them more interconnected but also more vulnerable to various cyber threats. They categorize threats at both the supply chain link level and across the entire end-to-end process. The paper discusses countermeasures like blockchain, artificial intelligence, and cryptographic methods to enhance security. Moreover, it highlights the need for a holistic approach, addressing both managerial and technical perspectives, and recommends practices such as threat modelling, supplier collaboration, and using advanced cryptographic techniques to improve digital supply chain security.

Credibility

The authors are affiliated with reputable institutions including EPITA Engineering School, University of Kentucky, and Prince Sattam bin Abdulaziz University, indicating strong academic backgrounds in the fields of ICT and cybersecurity. Published by *ACM Computing Surveys*, a well-established peer-reviewed journal, this paper has gone through rigorous academic scrutiny. The inclusion of recent literature, case studies, and technical reports ensures that the survey is comprehensive and up to date. However, while the paper presents a broad overview of the challenges facing DSCs, it relies heavily on secondary sources and lacks original empirical data, which may affect the depth of its practical insights.

Reflection
This paper is directly applicable to my research focus on AI-enhanced secure software development, particularly in understanding the intersection of supply chain security with AI solutions. The detailed discussion on threats and the role of technologies like blockchain and AI for security aligns well with my in leveraging Explainable AI (XAI) for transparent security measures. The insights into both the managerial and technical aspects of securing supply chains provide a valuable framework for integrating XAI into complex environments like digital supply chains. Furthermore, the emphasis on a holistic security approach and collaboration among stakeholders resonates with the goal of my research, to align secure software practices with industry standards such as ISO/IEC 27001.

24. S. Oladimeji and S. M. Kerner, "SolarWinds hack explained: Everything you need to know," TechTarget, 3 November 2023. [Online]. Available: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know . [Accessed 26 September 2024].

Summary
This article provides a detailed overview of the SolarWinds hack, describing it as one of the most significant cybersecurity breaches of the 21st century. The hack targeted SolarWinds' Orion IT monitoring and management software, compromising thousands of public and private sector organisations, including multiple U.S. government departments. The attackers, suspected to be a Russian group known as Nobelium, inserted malicious code into the Orion software, enabling backdoor access to user systems. The breach exploited a supply chain vulnerability, spreading malware undetected through updates to the Orion platform. The article also discusses the timeline of the hack, the potential motivations, and the challenges in detection due to the advanced tactics used. It highlights the broader implications for supply chain security and the importance of proactive measures such as software bills of materials (SBOMs).

Credibility
The article is authored by Saheed Oladimeji and Sean Michael Kerner, experienced technology journalists at TechTarget, which is a credible source known for providing in-depth technical information and news. TechTarget's reputation for reliability, combined with the authors' expertise, lends authority to the content. The article provides a comprehensive breakdown of the SolarWinds hack, citing relevant government advisories and industry expert opinions, which enhances its credibility. However, the information is focused on providing a journalistic perspective, which might lack the depth of peer-reviewed academic research but is well-suited for general informational purposes.

Reflection
This article is highly relevant to my research on AI-enhanced secure software development, particularly in the context of supply chain vulnerabilities. The SolarWinds hack underscores the critical need for improved supply chain security measures, an area where AI and Explainable AI (XAI) could play a significant role in enhancing transparency and real-time threat detection. The emphasis on proactive measures, such as SBOMs and incident response strategies, aligns with my research focus on aligning AI-driven secure coding practices with industry standards. The case of SolarWinds serves as a concrete example of how security lapses in software supply chains can have far-reaching consequences, providing a compelling argument for the integration of AI-based solutions to detect and mitigate such threats effectively.

25. **Alaa Houerbi, Rahul Ghanshyam Chavan, Dhia Elhaq Rzig, and Foyzul Hassan. "Empirical Analysis on CI/CD Pipeline Evolution in Machine Learning Projects."** In Proceedings of ACM Conference (Conference'17), 2024. ACM, New York, NY, USA, 12 pages. doi: 10.38550/arXiv.2403.12199

Summary
This paper presents an empirical study on the evolution of continuous integration and delivery (CI/CD) configurations in machine learning (ML) projects. The authors analysed 343 commits from 508 open-source ML projects, investigating how CI/CD pipeline configurations co-evolve with ML components. The study identified 14 categories of changes, highlighting dependency management, testing, and build policy updates as the most common types of modifications. The research also examined the skill levels of developers modifying CI/CD configurations, finding that experienced contributors are more likely to make changes. The paper reveals that CI/CD pipelines in ML projects often lack attention to performance and maintainability compared to general software projects, leading to technical debt. The study underscores the need for better best practices, especially around dependency management and testing frameworks.

Credibility
The paper is authored by researchers from the University of Michigan - Dearborn, which adds to its credibility. The analysis is based on a sizable dataset of 508 open-source ML projects, providing robust empirical evidence. The use of recognized CI/CD tools such as Travis CI and the extensive manual labelling of changes lends depth to the study. However, as the dataset is focused on open-source Python-based projects, the findings may not generalize to closed-source projects or projects in other programming languages. The paper's reliance on prior works such as those by Zampetti et al. strengthens its theoretical foundation.

Reflection
This paper fits well into the broader context of my research on AI-enhanced secure software development. It highlights the unique challenges and patterns in CI/CD evolution in ML projects, which align with my exploration of AI tools' adaptability and the technical barriers in secure coding. The study's findings on common pitfalls, such as the lack of standardized testing frameworks and the use of deprecated settings, may be relevant to discussions on improving the transparency and reliability of AI in secure software engineering. Furthermore, the paper's identification of challenges faced by developers modifying CI/CD configurations ties in with my research interest in the alignment of secure coding with industry standards, offering a perspective on the complexities involved in maintaining these pipelines in AI-integrated systems.

26. **V. K. Thatikonda, "Beyond the Buzz: A Journey Through CI/CD Principles and Best Practices," *European Journal of Theoretical and Applied Sciences,* vol. 1, no. 5, pp. 334-340. [Online]. Available: doi: https://doi.org/10.59324/ejtas.2023.1(5).24 [Accessed: 25th September 2024]., 2023.**

Summary
This article by Vamsi Krishna Thatikonda provides a comprehensive exploration of the core principles of Continuous Integration (CI) and Continuous Deployment (CD), their differences, and their shared importance in modern software development. It highlights automation, consistency, and fast feedback loops as pivotal elements for effective CI/CD practices. The paper also discusses advanced CI/CD techniques, including blue/green deployments, feature flagging, and the concept of Infrastructure as Code (IAC). Furthermore, Thatikonda delves into security considerations, such as shifting security left within the development lifecycle, and provides insights into future trends like AI/ML integration into CI/CD pipelines. The article acknowledges the challenges associated with CI/CD, emphasizing the need for attention to infrastructure consistency, comprehensive testing, and real-time monitoring.

Credibility
Thatikonda is a software professional with extensive practical experience in CI/CD and DevOps. The article is published in the *European Journal of Theoretical and Applied Sciences*, which has a reputation for providing quality peer-reviewed content on software engineering topics. The sources cited throughout the article, including Shahin et al. (2017) and Humble & Farley (2015), are well-regarded in the fields of DevOps and CI/CD. This lends credibility to the discussion, making it a reliable resource for understanding both foundational and advanced CI/CD concepts.

Reflection
This article is highly relevant to my dissertation, as it provides both historical context and practical insights into the implementation and evolution of CI/CD practices, which are essential for the development of secure software. The emphasis on automation and consistency aligns with my focus on secure coding standards, while the discussion on "shift left" security supports my exploration of how security considerations can be embedded early in the development process. The paper also addresses the use of advanced techniques, which contributes to my understanding of the evolving nature of CI/CD in the context of secure software engineering, making it an essential source for exploring best practices and identifying integration challenges.

27. **T. W. Thomas, "The intersection of static analysis and security code reviews: A collaborative model," *International Journal of Engineering in Computer Science,* vol. 5, no. 2, pp. 6-12. doi: https://doi.org/10.33545/26633582.2023.v5.i2a.93 , 22 June 2023.**

Summary
This article explores the integration of static analysis and security code reviews to create a more collaborative model for identifying and mitigating software vulnerabilities. Thomas introduces a tool prototype that merges interactive static analysis with traditional security code reviews to enhance effectiveness. Key roles in the proposed security review process include the primary developer, additional developers, and a security expert. The collaborative model aims to facilitate real-time communication and synchronization among stakeholders through a tool linked to Gerrit, a popular lightweight code review platform.

Credibility

Tyler W. Thomas is affiliated with the University of Wisconsin-Stout's Department of Mathematics, Statistics, and Computer Science. The article was published in *International Journal of Engineering in Computer Science* (IJECS), which provides a platform for engineering and computer science research. The credibility of the paper is strengthened by the author's academic affiliation and the peer-reviewed nature of the journal. The integration of both static analysis and collaborative review reflects a comprehensive approach to security, supported by references to related work and established best practices.

Reflection
This paper is relevant to my research on enhancing secure software development through Explainable AI (XAI), as it addresses the limitations of existing code review practices and offers a framework that improves collaboration and automation in security-focused reviews. By proposing a hybrid model, Thomas highlights the potential to improve both code quality and developer engagement in the secure coding process. This aligns well with my research in XAI-enhanced secure coding, as the collaborative model could serve as a foundation for incorporating explainability into security analysis tools.

28. **N. Pakovskie, "DeepCode: Revolutionizing Code Review with AI-Powered Bug Detection," 12 November 2023. [Online]. Available: https://www.geekpedia.com/deepcode-ai-code-review-bug-detection/. [Accessed 25 September 2024].**

Summary
This article discusses DeepCode, an AI-powered code review tool that significantly enhances the efficiency of bug detection in software development. DeepCode uses machine learning to analyse codebases, supporting various programming languages like Java, JavaScript, Python, TypeScript, and C/C++. By understanding the semantics of the code, DeepCode goes beyond traditional pattern-matching techniques, detecting deeper issues that are often missed by conventional code reviews. The tool integrates with development workflows, allowing easy adoption during code merges and CI/CD processes. It offers actionable feedback, which helps developers not only fix identified issues but also understand their root causes, enhancing the overall software quality and security.

Credibility
The article was published on *Using AI to Code*, a platform dedicated to discussing AI applications in coding. While the author, Nathan, does not provide an institutional affiliation, the detailed explanation of DeepCode's functionality reflects a thorough understanding of AI and code review tools. The article's credibility is further supported by its detailed descriptions of machine learning and its application in identifying bugs and vulnerabilities, which align with established AI practices in software engineering.

Reflection
This article is highly relevant to my research on AI-enhanced secure software development with a focus on Explainable AI (XAI). DeepCode's ability to provide context-aware feedback and suggestions aligns with the need to make AI-driven code analysis more transparent and understandable. The integration of DeepCode's features into secure software development can offer valuable insights into how AI tools can provide a more secure and efficient coding process. This source can serve as a practical example of integrating AI in secure software engineering, aligning well with the themes of automation, transparency, and efficiency.

29. **B. Berabi, G. Sivanrupan, A. Gronskiy, V. Chibotaru, V. Raychev and M. Vechev, "DeepCode AI Fix: Fixing Security Vulnerabilities with Large Language Models," arXiv, 2024.**https://arxiv.org/abs/2402.13291v2

Summary
The paper introduces DeepCode AI Fix, an innovative approach to automated program repair using large language models (LLMs). This research addresses the challenges of using LLMs for fixing security vulnerabilities, such as learning long-distance code relationships and creating clean training datasets. The authors propose a novel technique leveraging static analysis to focus LLMs on the portions of code directly related to defects, thereby improving training efficiency and accuracy. By employing a code reduction mechanism like cReduce, DeepCode AI Fix reduces the required input size and simplifies attention tasks for LLMs. This new approach outperforms baseline models like GPT-3.5, GPT-4, and TFix, achieving high rates of correct fixes, especially in complex security issues. The paper also presents a dataset of 5,000 labelled examples of security vulnerabilities and their fixes, which was curated through extensive labelling of GitHub commits.

Credibility
This paper is authored by researchers affiliated with reputable institutions such as Snyk, ETH Zurich, and INSAIT at Sofia University. Their affiliations lend credibility to the research, as these institutions are well-known for their expertise in AI and cybersecurity. Additionally, the article was published on arXiv, a preprint server commonly used for distributing scientific papers in computer science. The authors provide comprehensive evaluations and dataset details, reinforcing the paper's transparency and reliability. The use of Mixtral-8x7B, GPT-4, and other leading AI models, along with detailed comparisons with established tools like TFix, highlights the depth of experimentation and validation in the study.

Reflection
The concepts presented in this paper are directly aligned with my research on AI-enhanced secure software development with a focus on Explainable AI (XAI). DeepCode AI Fix's approach to reducing code complexity to improve LLM performance can inform the exploration of transparency and the effectiveness of AI tools in secure software development. The dataset and evaluation metrics provided can also serve as benchmarks for assessing other AI-driven secure coding tools, particularly those that incorporate explainability into their functionality.

30. **V. Bhutani, F. G. Toosi and J. Buckley, "Analysing the Analysers: An Investigation of Source Code Analysis Tools,"** ***Applied Computer Systems,*** **vol. 29, no. 1, pp. 98-111. doi: https://doi.org/10.2478/acss-2024-0013, 15 August 2024.**

Summary
This paper provides a comprehensive analysis of seven prominent source code analysis tools: SonarQube, Coverity, CodeSonar, Snyk Code, ESLint, Klocwork, and PMD. The study aims to assist software developers in selecting the most suitable tool by evaluating each tool based on various dimensions, including supported languages, extensibility, input types, technology, and user experience. SonarQube is highlighted as a versatile tool that supports both static and dynamic analysis and integrates well with major IDEs, while Coverity and CodeSonar excel in security vulnerability detection. The paper also categorizes tools into static and dynamic analysis, exploring how each serves different purposes in software quality assurance, such as defect detection, maintainability, and security.

Credibility
The paper is authored by researchers from Munster Technological University and the University of Limerick, which lends academic credibility to the study. The publication in *Applied Computer Systems*, a peer-reviewed journal, further ensures the reliability of the findings. The authors use a systematic method to categorize and evaluate well-recognized tools, providing a balanced perspective on their capabilities and limitations, which enhances the trustworthiness of the analysis.

Reflection
This paper is particularly useful for understanding the landscape of source code analysis tools and their applicability to secure software development. The comparative analysis helps in identifying which tools may align with the focus on AI-enhanced secure coding practices, especially regarding maintaining code quality and addressing security vulnerabilities. The detailed taxonomy provided by the authors can serve as a framework for evaluating other analysis tools that incorporate Explainable AI (XAI) features, helping in the broader research on transparent and effective AI-driven software security solutions.

31. **F. Kilonzi, "What is Shift Left Security," Orca Security, 25 July 2024. [Online]. Available: https://orca.security/resources/blog/what-is-shift-left-security/. [Accessed 26 September 2024].**.
https://content.sciendo.com

Summary
This blog post introduces the concept of Shift Left Security, which involves integrating security practices into the early stages of the Software Development Lifecycle (SDLC) instead of the traditional approach where security checks are applied towards the end. By shifting security left, development teams can address vulnerabilities and misconfigurations early, reducing costs, improving efficiency, and preventing security flaws from reaching production. The post outlines various Shift Left practices, such as defining security requirements upfront, integrating automated security testing into CI/CD pipelines, and fostering collaboration between developers and security teams. The author also discusses the benefits of Shift Left Security, including increased efficiency, reduced friction between teams, and an enhanced security posture. Orca's Shift Left Security solutions are highlighted as tools that enable organisations to adopt this approach effectively.

Credibility
This blog post is authored by Faith Kilonzi, and while it is part of the Orca Security Blog, which serves promotional purposes, the content provides a comprehensive overview of Shift Left Security, including practical steps for implementation. The article references relevant industry practices, such as the use of DevSecOps tools, CI/CD integration, and automated security testing, which are well-established in the software development field. The insights on cloud-native development and the role of CI/CD in enhancing security reflect current trends, adding to the post's credibility

Reflection
The ideas discussed in this blog post are particularly relevant to my research on AI-enhanced secure software development, as they highlight a proactive approach to embedding security within the development lifecycle. The concept of Shift Left aligns with the need for enhancing transparency and collaboration, which are central to Explainable AI (XAI). Additionally, the discussion on automating security testing and embedding security into CI/CD pipelines could be beneficial for exploring how AI tools can further improve secure software development practices.

32. **N. Pakalapati, B. K. Konidena and I. A. Mohamed, "Unlocking the Power of AI/ML in DevSecOps: Strategies and Best Practices," *Journal of Knowledge Learning and Science Technology,* vol. 2, no. 2, pp. 176-188. [Online]. Available: https://doi.org/10.60087/jklst.vol2.n2.p188 .[Accessed: 25th September 2024]., 12 July 2023.**

Summary
This paper explores the integration of Artificial Intelligence (AI) and Machine Learning (ML) into DevSecOps practices, providing insights into how these technologies enhance security, efficiency, and innovation in software development. The authors discuss strategies such as automated threat detection, predictive analytics for vulnerability management, and intelligent automation of CI/CD pipelines. The paper highlights both the potential benefits and challenges of incorporating AI/ML into DevSecOps, including data privacy, algorithm transparency, and ethical considerations. The authors provide case studies and real-world examples to illustrate the successful implementation of AI/ML, offering a roadmap for organisations to optimize DevSecOps processes and strengthen security measures. Key topics covered include the automation of security tasks, predictive vulnerability management, and fostering a culture of continuous improvement in software security.

Credibility
The authors are affiliated with reputable organisations such as Fannie Mae, StateFarm, and Salesforce, adding a level of authority to the research. The paper is published in the *Journal of Knowledge Learning and Science Technology*, which suggests a focus on applied knowledge in technological innovation. The detailed presentation of strategies, along with practical case studies and empirical data collection, supports the reliability of the research. The inclusion of expert interviews and real-world examples strengthens the paper's credibility and demonstrates a well-rounded approach to the integration of AI/ML into DevSecOps.

Reflection
This paper is highly relevant to my research on AI-enhanced secure software development, especially in the context of Explainable AI (XAI). The discussion on AI/ML integration into DevSecOps aligns with the focus on improving software security and transparency. The paper's emphasis on predictive analytics for vulnerability management and intelligent automation can inform the exploration of effective AI-driven strategies for secure coding practices. Furthermore, the challenges discussed regarding algorithm transparency and ethical considerations offer valuable insights into balancing security effectiveness with explainability, a key aspect of my research.

33. **N. G. Camacho, "Unlocking the Potential of AI/ML in DevSecOps: Effective Strategies and Optimal Practices," *Journal of Artificial Intelligence General Science (JAIGS),* vol. 2, no. 1, pp. 78-89. doi: https://doi.org/10.60087/jaigs.v2i1.p89 , 6 March 2024.**

Summary
This paper explores the integration of Artificial Intelligence (AI) and Machine Learning (ML) within DevSecOps to enhance security, efficiency, and innovation in software development processes. Guzman Camacho provides an overview of DevSecOps principles, highlighting AI/ML's role in automating threat detection, predictive analytics for vulnerability management, and intelligent automation for CI/CD processes. The paper also discusses challenges such as data privacy, algorithm transparency, and ethical considerations in AI/ML deployment. Through case studies, the author illustrates practical implementations of AI/ML technologies in DevSecOps pipelines, offering strategies to mitigate security risks and foster continuous improvement.

Credibility
The author, Nicolas Guzman Camacho, is affiliated with Universidad de La Sabana in Colombia, which lends academic credibility to the research. The paper is published in the *Journal of Artificial Intelligence General Science (JAIGS)*, which appears to be a reputable journal focusing on AI advancements. The article's structure—combining a literature review, case studies, expert interviews, and empirical data—supports a comprehensive approach, enhancing the paper's reliability and depth. Furthermore, the use of real-world examples strengthens its practical applicability.

Reflection
This paper is highly relevant to my research on AI-enhanced secure software development, particularly with a focus on Explainable AI (XAI) in DevSecOps. The paper's discussion on AI/ML integration for automated threat detection aligns well with my research into transparency and AI-driven security improvements. Additionally, its emphasis on addressing ethical considerations and data privacy issues in AI integration provides valuable insights for understanding the challenges of incorporating XAI in secure software engineering.

34. **Jakub Res, Ivan Homoliak, Martin Perešíni, Aleš Smrčka, Kamil Malinka, Petr Hanacek.** "Enhancing Security of AI-Based Code Synthesis with GitHub Copilot via Cheap and Efficient Prompt-Engineering." *Brno University of Technology, Faculty of Information Technology, Czech Republic.* arXiv preprint: https://arxiv.org/abs/2403.12671v1.

Summary
This paper proposes enhancing the security of AI-generated code from GitHub Copilot by employing prompt-engineering techniques. The authors discuss the challenges developers face due to security issues in code synthesised by AI tools like Copilot, and propose three systematic prompt alteration methods—scenario-specific, iterative, and general clause—to improve code security. The paper evaluates these techniques using the OpenVPN project, showing that the proposed methods reduce insecure code generation by up to 16% and increase secure code generation by up to 8%. The prompt-engineering approach presented is computationally efficient and does not require access to the internal workings of the AI models, making it applicable to a wide range of AI-based code synthesizers.

Credibility
The paper is authored by researchers from the Faculty of Information Technology at Brno University of Technology, which adds academic credibility to the study. It is published as a preprint on arXiv, a widely used repository for early computer science research. The authors provide a thorough analysis, presenting results from real-world experiments using the OpenVPN project and emphasizing a practical approach to improving security without modifying proprietary AI models. This approach, along with clear contributions and evaluations, reinforces the reliability of the findings.

Reflection
The findings are directly relevant to my research on AI-enhanced secure software development, particularly regarding Explainable AI (XAI). The prompt-engineering techniques detailed in this paper can inform on exploration of transparent methods to enhance AI-generated secure code. The systematic approach to improving security, as well as the evaluation metrics provided, can help assess the effectiveness of similar techniques in different AI-driven secure coding environments, particularly those involving XAI.

35. **R. Naidoo and N. Moller, "Building Software Applications Securely With DevSecOps: A SocioTechnical Perspective," in *Proceedings of the 21st European Conference on Cyber Warfare and Security(ECCWS)*, 2022.**

Summary
This paper presents a socio-technical framework for understanding DevSecOps practices, highlighting the need for collaboration between social actors (developers, security experts, operators) and technologies in building secure software applications. The authors argue that current DevSecOps literature often overlooks the importance of the socio-technical interplay, instead focusing on either technical tools like cryptographic protocols or social aspects like team culture. By conducting a systematic literature review of 26 peer-reviewed articles from 2016 to 2020, the authors developed a comprehensive socio-technical framework for DevSecOps that can help practitioners and researchers address both instrumental (e.g., economic efficiency) and humanistic (e.g., job satisfaction) goals. The framework aims to provide a holistic perspective on improving both the technical and social components of DevSecOps systems, emphasising the importance of interdisciplinary approaches to address the challenges inherent in integrating security into agile and DevOps processes.

Credibility
The paper is authored by two researchers from the University of Pretoria and was presented at the European Conference on Cyber Warfare and Security, giving it a strong academic backing. The authors employed a systematic literature review methodology, using peer-reviewed sources from various computing disciplines, which lends credibility to their findings. The use of a socio-technical framework provides an in-depth exploration of the interaction between human and technological factors in DevSecOps, reinforcing the study's reliability.

Reflection
The socio-technical perspective provided in this paper is valuable for my research on AI-enhanced secure software development. It highlights the interplay between technical tools and human factors, which is especially pertinent to integrating Explainable AI (XAI) in secure software engineering. The emphasis on collaboration among social actors aligns with the focus on transparency and integrating security effectively. The socio-technical framework could be beneficial for exploring how XAI tools impact the roles, responsibilities, and interactions of software development teams in secure coding practices.

36. **N. U. Baki, R. M. Rasdi, S. E. Krauss and M. K. Omar, "Integrating Artificial Intelligence in Human Resource Functions: Challenges and Opportunities," *International Journal of Academic Research in Business and Social Sciences*, vol. 13, no. 8, pp. 1262-1277. doi: DOI:10.6007/IJARBSS/v13-i8/18071 , 18 August 2023.**

Summary
This paper explores the integration of artificial intelligence (AI) in human resource (HR) functions, systematically reviewing literature to identify opportunities and challenges. It highlights AI's role in transforming HR functions such as recruitment, employee engagement, training, development, and performance assessment. AI enhances efficiency by automating tasks, mitigating human biases, and reducing costs. However, challenges include the high cost of implementation, employee

resistance, ethical concerns, and the lack of a human touch in decision-making. The authors argue that effective integration requires collaboration between HR professionals and AI experts, emphasizing change management strategies to navigate technological disruptions.

Credibility
The article is authored by researchers from Universiti Putra Malaysia and published in a peer-reviewed journal, ensuring academic reliability. The systematic review approach and focus on both opportunities and challenges provide a balanced perspective. The article's use of credible sources and comprehensive analysis of AI integration in HR functions demonstrates its quality. Additionally, the publication's affiliation with the Human Resource Management Academic Research Society (HRMARS) further supports its credibility.

Reflection
This paper is relevant for understanding the socio-technical interplay in integrating AI tools within HR functions, which can be compared to similar issues in secure software engineering. It offers insight into challenges such as human resistance to AI and ethical considerations, which align with the user's focus on transparency and XAI. The emphasis on a holistic approach to integrating AI in HR can inform strategies for XAI integration in software development, particularly regarding the collaboration between technical experts and users to mitigate concerns and resistance.

37. **Z. Bilgin, M. A. Ersoy, E. U. Soykan, E. Tomur, P. Comak and L. Karacay, "Vulnerability Prediction From Source Code Using Machine Learning," *IEEE Xplore,* vol. 8, pp. 150672-150684. doi: 10.1109/ACCESS.2020.3016774 , 14 August 2020.**

Summary
This study presents a method for predicting software vulnerabilities using machine learning (ML) applied to the abstract syntax tree (AST) representation of source code. The authors propose a source code representation technique that enables intelligent analysis and vulnerability detection using a public dataset of labelled source code fragments. The dataset comprises function-level components mined from open-source projects, allowing the ML model to distinguish between vulnerable and non-vulnerable code. The method is compared against state-of-the-art techniques, with results showing promising performance improvements in predicting software vulnerabilities. This approach seeks to automate and enhance software assurance through data-driven analysis of code.

Credibility
The article is authored by researchers from Ericsson Research, Istanbul, and funded by the Scientific and Technological Research Council of Turkey, indicating reputable affiliations. It has been accepted for publication in IEEE Access, a well-established and peer-reviewed journal known for disseminating high-quality research. The use of a comprehensive public dataset and detailed experimental analysis adds to the reliability of the findings. Moreover, the method's comparison with existing techniques demonstrates a critical assessment of its effectiveness.

Reflection
This paper is directly relevant to my research focus on AI-enhanced secure software development, specifically in vulnerability prediction from source code. The proposed AST-based ML method contributes to understanding how data-driven techniques can improve software security, a key interest in my research. The insights from this paper can inform the effectiveness of different code representation techniques and vulnerability prediction methods, aligning with my focus on transparency and XAI in secure coding practices.

38. **Albattah, Waleed and Alzahrani, Musaad. *AI* (2024) vol. 5 issue 4. 1743-1758. doi: 10.3390/ai5040086**

Summary
This study explores methods to predict software defects early in the development process, aiming to reduce costs and improve reliability. The authors compare eight popular ML and DL algorithms, such as LSTM and Random Forest, to predict bugs using a dataset containing various software metrics. They found that DL, specifically LSTM, performed the best, achieving an accuracy of 87%. The study highlights how using these models can help developers identify bug-prone areas early, allowing for targeted testing and quality improvements in the software development lifecycle.

Credibility
The authors are affiliated with reputable Saudi universities, Qassim University and AL-Baha University, which lends credibility to the reser4ach. The paper was published in a peer-reviewed journal, ***AI***, adding to its trustworthiness. The study uses well-established ML techniques, and a comprehensive dataset derived from five publicly available sources, making the findings robust and relevant for the field of software engineering.

Reflection
This study is useful for understanding how ML and DL models can enhance software maintenance and quality control. Its focus on early prediction of bugs aligns with modern secure software development practices, where identifying vulnerabilities early is critical. The study's emphasis on LSTM's effectiveness in handling imbalanced data and complex patterns can support further research on using AI for secure coding practices. Additionally, the study's methodology of evaluating various metrics provides insights into refining AI models for more accurate bug detection.

39. **S. Gawde and e. al., "Explainable Predictive Maintenance of Rotating Machines Using LIME, SHAP, PDP, ICE,"** ***IEEE Access,*** **vol. 12, pp. 29345-29361. doi: 10.1109/ACCESS.2024.3367110, February 2024.**

Summary
This paper proposes a method for predictive maintenance of rotating machines by leveraging Explainable AI (XAI) techniques to interpret the decision-making processes of AI models. The study aims to overcome the black-box nature of traditional predictive models by utilizing LIME, SHAP, PDP, and ICE to provide human-understandable insights into how these models make predictions. The research includes multi-sensor data acquisition, frequency-domain statistical feature extraction, and the application of multiple AI algorithms to demonstrate the efficiency of the proposed method. The focus on explainability aims to enhance trust in AI-driven predictive maintenance.

Credibility
The paper is authored by researchers from prominent institutions such as Symbiosis International (Deemed University) and King Saud University, which adds to its credibility. It has been accepted for publication in IEEE Access, a reputable journal known for high-quality research publications. The work is also funded by King Saud University, indicating institutional support. The research's use of advanced XAI methods and comparison with traditional predictive models further strengthens its contribution.

Reflection
This paper is particularly relevant to my dissertation focus on XAI for secure software engineering, as it demonstrates the use of XAI techniques to make AI-driven decisions transparent and interpretable. The application of LIME, SHAP, PDP, and ICE offers insights into how these methods can be adapted to different domains, such as predictive maintenance, potentially inspiring novel approaches for improving the transparency of AI tools in software security. The detailed explanation of integrating XAI into AI models can serve as a practical reference for aligning AI transparency with industry standards in my research.

40. **{}, "ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering,"** *ISO/IEC/IEEE 29148:2018(E),* pp. 1-104. doi: 10.1109/IEEESTD.2018.8559686}, 30 November 2018.

Summary
The ISO/IEC/IEEE 29148:2018 standard outlines the life cycle. Processes and requirements engineering for systems and software engineering. It provides a comprehensive framework for defining, managing, and verifying requirements throughout the development process, ensuring that software systems are developed according to specified user and stakeholder needs. This standard emphasises the best practices in requirements elicitation, analysis, validation, and management, contributing to the development of high-quality robust systems. It aligns with other internationally recognised frameworks, making it an essential reference for any organisation involved in systems and software engineering.

Credibility
This standard is developed and published by ISO, IEC and IEEE, three of the top authoritative organisations in engineering and cybersecurity. Their collaboration ensures a standard is widely recognised and adopted. Additionally, these standards undergo rigorous peer reviews and revisions, further enhancing their credibility and reliability. It relevance to both software engineering and systems engineering gives it a broad application scope.

Reflection
This standard is highly relevant to my research on integrating AI and XAI in secure software engineering. As AI tools become more involved in software development, understanding and adhering to established requirements engineering principles is crucial. ISO/IEC/IEEE 29148:2018 offers a structured approach to managing system requirements, which can be applied to ensure that AI-driven software meets both technical and security requirements. This standard's focus on best practices aligns with my research's goal of incorporating XAI techniques in compliance with security and legal frameworks.

41. M. Taeb, H. Chi and S. Bernadin, "Assessing the Effectiveness and Security Implications of AI Code Generators," ***2024 Journal of The Colloquium for Information Systems Security Eductaion (CISSE),*** **vol. 11, no. 1, p. doi: https://doi.org/10.53735/cisse.v11i1.180 , February 2024.**

Summary
This paper explores the effectiveness and security implications of AI-based code generators, such as OpenAI CodeX, CodeBERT, and ChatGPT. The study aims to assess the capabilities of these models in generating secure code, their utility in code completion, and their ability to assist in vulnerability mitigation. The authors analyse specific code generation features, assess potential vulnerabilities introduced by these tools, and provide a detailed examination of their use in an educational context. The research reveals that while these models provide valuable support, they also have significant limitations regarding the accuracy and security of generated code. Notably, the potential over-reliance on these tools by developers and the risks associated with vulnerable built-in functions are discussed.

Credibility
The authors of this paper are affiliated with recognized academic institutions: Florida A&M University and FAMU-FSU College of Engineering. The article is published in the *Journal of The Colloquium for Information Systems Security Education*, a reputable source in the field of cybersecurity education. The publication includes peer-reviewed research focusing on the intersection of education, information security, and technological advancements, which strengthens the credibility of the analysis presented.

Reflection
This source is highly relevant to my research on AI-enhanced secure software development, especially in examining the security challenges associated with using AI code generation tools. The detailed analysis of vulnerabilities and the security implications of tools such as CodeBERT and GPT-3.5 directly supports the dissertation focus on secure coding practices and the integration of AI. It provides critical insights into the strengths and limitations of current AI models in supporting secure software development, which will be useful in discussing transparency, trustworthiness, and the potential risks of AI in coding.

42. C. Tantithamthavorn, J. Cito, H. Hemmati and S. Chandra, "Explainable AI for SE: Challenges and Future Directions," ***IEEE Software,*** **vol. 40, no. 3, pp. 29-33. doi: 10.1109/MS.2023.3246686 , May-June 2023.**

Summary
This article introduces a special issue focusing on the challenges and future directions of Explainable AI (XAI) for software engineering (SE). The authors emphasise the importance of explainability in AI/ML-based software development tools, such as those used for code completion, defect prediction, and task automation. They argue that the lack of transparency in AI/ML models hinders developers' trust and limits the widespread adoption of these tools in practice. The article categorises XAI techniques into white-box and black-box methods and discusses their applicability to SE contexts. The authors also highlight contributions from the special issue, including articles addressing the challenges of reliability and trustworthiness in XAI for SE, and present interviews with experts discussing the role of XAI in the future of software engineering.

Credibility
The authors are affiliated with reputable institutions including Monash University, TU Wien, York University, and Google, which adds to the credibility of their work. Published in *IEEE Software*, a peer-reviewed journal known for high-quality research in software engineering, the article reflects well-researched insights on the intersection of XAI and SE. The use of multiple contributors, industry experts, and empirical research strengthens the validity of the presented challenges and future directions in XAI.

Reflection
This source is crucial for understanding the challenges of integrating XAI in software engineering, particularly the issues of trust, transparency, and reliability, which align closely with the user's research focus on AI-enhanced secure software development. The article's discussion on stakeholder-specific requirements for explainability and the categorisation of XAI methods will be valuable for framing the dissertation's analysis of XAI integration challenges and industry alignment. It also provides a foundation for addressing future research directions, which will help in justifying the need for XAI advancements in secure software engineering practices.

43. **T. E. Gasiba, K. Oguzhan, I. Kessba, U. Lechner and M. Pinto-Albuquerque, "I'm Sorry Dave, I'm Afraid I Can't Fix Your Code: On ChatGPT, CyberSecurity, and Secure Coding," in** ***4th International Computer Programming Education Conference (ICPEC 2023),*** **Dagstuhl, Germany, 2023.**

Summary
This paper explores the potential of ChatGPT in aiding software developers to write secure code by evaluating the strengths and limitations of ChatGPT's capability to identify and resolve security vulnerabilities in code. The authors conducted experiments using vulnerable code snippets and analysed ChatGPT's responses in comparison to expected solutions. The research provides insights into the effectiveness of ChatGPT in recognising and fixing vulnerabilities, and whether it can serve as a reliable tool for raising awareness of secure coding practices among developers.

Credibility
The paper is authored by researchers from Siemens AG, Universität der Bundeswehr München, and Instituto Universitário de Lisboa, highlighting a collaboration between industrial and academic professionals, thus adding credibility to the work. It was presented at the 4th International Computer Programming Education Conference (ICPEC 2023) and published by Dagstuhl Publishing, known for its rigorous peer-review process. The research builds upon existing work and industry standards for secure software development, lending further authority to its findings.

Reflection
This paper is highly relevant to my research on AI-enhanced secure software development, particularly in examining the practical application of AI-based tools like ChatGPT in secure coding. The discussion of both the strengths and weaknesses of using ChatGPT as a tool for secure software engineering, including its limitations in context recognition and vulnerability detection, aligns well with my dissertation focus on Explainable AI (XAI). The insights provided here can contribute to evaluating the applicability of AI tools in real-world scenarios, including addressing transparency and limitations in automated secure coding.

44. **P. Nath, J. R. Mushahary, U. Roy, M. Brahma and P. K. Singh, "AI and Blockchain-based source code vulnerability detection and prevention system for multiparty software development,**" *Computers and Electrical Engineering* , vol. 106, pp. 1-15. https://doi.org/10.1016/j.compeleceng.2023.108607, March 2023. doi: 10.1016/j.compeleceng.2023.108607.

Summary
This paper proposes an integrated Artificial Intelligence (AI) and blockchain-based system for automated vulnerability detection and prevention in multiparty software development. The system utilides deep learning models, specifically Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM), for detecting vulnerabilities during the testing phase of the software development life cycle (SDLC). To enhance transparency and trust, a blockchain-based decentralided mechanism is employed, supported by InterPlanetary File System (IPFS) for efficient data storage. The research demonstrates the potential of combining AI for automated vulnerability detection with blockchain to secure the software testing process, especially in remote work scenarios.

Credibility
This study is published in *Computers and Electrical Engineering*, a reputable peer-reviewed journal known for its contributions to engineering research. The authors are affiliated with reputable academic institutions, lending authority to the findings. The proposed system and methodology are experimentally validated on a testbed setup, providing practical evidence of the effectiveness of integrating AI and blockchain technologies in the SDLC. The use of deep learning models for vulnerability detection and blockchain to ensure transparency showcases an innovative approach backed by thorough experimental results.

Reflection
This paper is pertinent to my research focus on AI-enhanced secure software development, particularly due to its integration of AI for automated vulnerability detection and blockchain for enhancing security and transparency. The combination of these technologies aligns with the interest in advanced secure coding techniques, and the experimental validation provides insights into the practical feasibility of these approaches in decentralised environments. This study could enrich my literature review by providing a novel perspective on securing multiparty software development using AI and blockchain, which also resonates with my focus on Explainable AI (XAI).

45. **V. D. Kirova, C. S. Ku, J. R. Laracy and T. J. Marlowe, "The Ethics of Artificial Intelligence in the Era of Generative AI," *Journal of Systemics, Cybernetics and Informatics,* vol. 21, no. 4, pp. 42-50. doi: https://doi.org/10.54808/JSCI.21.04.42 , 2023.**

Summary
This paper provides an overview of ethical considerations surrounding generative artificial intelligence (GenAI), focusing on its historical and cybernetic context. The authors explore various ethical challenges posed by GenAI, particularly in software engineering, cyber-physical systems, and healthcare. The paper emphasizes the importance of ethical principles in addressing challenges such as bias, transparency, fairness, and accountability. It highlights the growing need for safeguards, standards,

and ethical frameworks to regulate the development and deployment of AI in society, particularly in sensitive domains like healthcare.

Credibility
Published in *Journal of Systemics, Cybernetics and Informatics*, this paper is authored by experts affiliated with reputable institutions, including Nokia Bell Labs and several universities. The multidisciplinary approach of the authors combining expertise in computer science, engineering, theology, and ethics enhances the credibility of the analysis. The inclusion of historical perspectives and references to ethical frameworks from well-known organisations like IEEE and UNESCO strengthens the foundation of the discussion.

Reflection
This paper aligns closely with my research focus on ethical considerations in AI-enhanced secure software development. The exploration of ethical challenges in healthcare and software engineering provides valuable insights that can be incorporated into the literature review, particularly in discussions about transparency and accountability. Additionally, the emphasis on ethical safeguards and professional standards could help address the ethical aspects of integrating Explainable AI (XAI) into secure coding practices. This study offers a comprehensive perspective on the ethical implications of deploying AI technologies, which is essential for a balanced approach to AI development in secure software engineering.

46. **European Parliament, "EU AI Act: first regulation on artificial intelligence," 18 June 2024. [Online]. Available: https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence . [Accessed 25 September 2024].**

Summary
This webpage provides an overview of the EU AI Act, which is the first comprehensive regulation on artificial intelligence (AI) in the world. The AI Act aims to create a framework to ensure the safe, transparent, and non-discriminatory use of AI, with distinct regulations for AI systems categorised by different risk levels. The Act outlines a tiered approach to regulation based on risk (unacceptable, high, and limited risks), transparency requirements for generative AI, and measures to support AI innovation, especially for start-ups. The webpage also describes the timeline for the Act's implementation and enforcement phases.

Credibility
The source is highly credible as it is an official publication by the European Parliament, providing direct information about EU legislation. The European Parliament is a reputable and authoritative body for such legislative updates. The information is up-to-date, and the document reflects the finalised details of the AI Act, which underwent extensive debate and approval processes involving multiple EU bodies.

Reflection
The EU AI Act's emphasis on transparency and different risk levels for AI systems is directly relevant to the focus on Explainable AI (XAI) in secure software development. The regulation's requirements for generative AI to disclose its AI-generated content align with XAI principles of transparency and explainability. Moreover, understanding the categories of risk and compliance procedures outlined in the Act will be useful for evaluating how AI-enhanced software tools in secure coding can align with regulatory standards. This source adds a legislative perspective to my research, enriching the literature review on how policy frameworks influence secure software engineering practices using AI.

47. **Braun, V., & Clarke, V. (2012). Thematic analysis. In H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, & K. J. Sher (Eds.),** *APA handbook of research methods in psychology, Vol. 2: Research designs: Quantitative, qualitative, neuropsychological, and biological* (pp. 57-71). Washington, DC: American Psychological Association.

Summary
This chapter by Braun and Clarke provides an overview of thematic analysis (TA) as a qualitative method for identifying patterns across a dataset. The authors describe how thematic analysis allows researchers to systematically identify, organize, and offer insights into patterns of meaning (themes) across data. The method is celebrated for its flexibility and accessibility, which makes it applicable to various research topics and questions. Braun and Clarke present a six-phase approach to TA, from data familiarization to reporting the results, with examples to illustrate the process.

Credibility
Braun and Clarke are leading figures in qualitative research methods, particularly in the field of thematic analysis. Their work is widely cited and recognized for making TA more systematic and accessible to researchers, particularly in psychology. The APA handbook is a reputable publication, adding to the credibility of this chapter. Strengths of the chapter include its clear presentation of the TA process and the practical, step-by-step guide to conducting thematic analysis. One limitation is that the

examples are primarily drawn from psychological studies, which may limit direct applicability to other fields without adaptation.

Reflection
This source is highly relevant to the section of my dissertation on data analysis using thematic analysis. It provides a solid theoretical foundation and practical steps for conducting TA, supporting the methodology used in my research. The reference to Braun and Clarke's six-phase approach bolsters the justification for using thematic analysis to interpret qualitative data collected from interviews. It also reinforces the methodological rigor of my approach, making it a valuable addition to my research framework.

This document is also available in Welsh

VIII. ETHICAL APPROVAL OF RESEARCH PROJECTS IN ONLINE PROGRAMMES

There are 3 routes for review and approval:

1. RESC (Research Ethics Sub-Committee) - for staff and postgraduate research student proposals involving human subjects; all research involving animals, and all research requiring formal external approval [use the full RESC application form]
2. Staff and postgraduate research students Low Risk research [i.e. not covered by 1. above – use a Checklist and Cover Sheet form]
3. ***Research done by undergraduate and online Masters students [use a Checklist and Cover Sheet form]***

All proposals through all routes involve completing the relevant sections of the Checklist, to highlight any potential ethical risk factors.

Programme	MSc Computer Science in Software Engineering		
Module	Dissertation CONL718		
Student Name	Holley Hudson	Student ID	S22009650
Research Project Title	**AI-Enhanced Secure Software Engineering: A Focus on Explainable AI (XAI) Techniques**		

I give approval for this research project to proceed, on the grounds that:

- it is consistent with the programme specification
- a suitable and sufficient risk assessment has been carried out
- the checklist has been fully completed
- it does not contain any ethical risk factors which may cause harm of any kind to research subjects, the researcher, the University or any other person or organisation
  AND/OR
  any risk factors have been clearly identified and appropriate measures put in place for their management and mitigation
- where relevant, appropriate and robust plans have been made to gain informed consent from prospective research subjects
- it is not required to be submitted for approval to the Research Ethics Sub-Committee

Project Tutor Name	Joel Pinney		
Signature		Date	

Significant changes

I approve the changes proposed by the student, on the grounds specified above.

Project Tutor Name			
Signature		Date	

Notes:

1. This form must be completed before primary data collection / experimental work begins.
2. The Checklist which follows must be fully completed.

3. The person approving the research must be satisfied that any ethical risk factors have been clearly identified and appropriate measures put in place for their management and mitigation.
4. This signed form should be filed with the student's project proposal.
5. The University's Code of Practice on Ethical Standards for Research is available at: https://moodle.glyndwr.ac.uk/course/view.php?id=26703

Glyndŵr University - Checklist for ethical approval of a research project

Checklist 1 – to be completed for ALL proposals [answer ALL questions]

	Yes	No
1. Does the research comply with the University's Code of Practice on Ethical Standards for Research? [https://moodle.glyndwr.ac.uk/course/view.php?id=26703]	x	
2. Does this research comply with the requirements of any relevant professional body's code of conduct? [If Not Applicable', mark 'Yes']	x	
3. Has a suitable and sufficient risk assessment been carried out (including potential harm to the researcher)?	x	
4. Will the study require the co-operation of a 'gatekeeper' for initial permission / access to the people, animals, places, data, or other resources required for the research?		x
5. Does this research require the formal approval of an external body[1]?		x
6. Could the research have an impact on people living or working in the immediate locality?		x
7. Will anyone other than the researcher (the applicant) and the research supervisor (if relevant) have access to the raw data produced by the research?		x
8. Is there a sponsor?		x
9. Is there a collaborating organisation?		x
10. Will any research be undertaken outside UK legal jurisdiction?	x	
11. Will your research involve investigation of or engagement with terrorist or violent extremist groups?		x
12. Will your research and its findings have any potential in relation to furthering extremist ideology or causes and/or will any process or artefact produced have potential to be used to further extremist ends?		x

Does the proposed research:-

	Yes	No
Directly involve people? (go to Checklist 2)	x	
Directly involve animals or animal by-products? (go to Checklist 3)		
Have a potential impact on the environment? (go to Checklist 4)		

Checklist 2 Research directly involving people [answer ALL questions]	Yes	No
13. Will you use Social Media to interact with participants?	x	
14. Does the study involve NHS patients, staff or premises?[1]		x
15. Does the study involve participants who are particularly vulnerable (e.g. children, victims of crime, homeless, mental illness etc.)? Please read carefully the Code of Practice.		x
16. Does the study involve participants who would find it difficult to give informed consent (e.g. children, people with learning difficulties)? Please read carefully the Code of Practice.		x

[1] If so, the proposal must have full RESC approval before the applicant applies to the external body. **'NHS patients' means people invited to take part in the research because of that status (now or previously)**.

Checklist 2 Research directly involving people [answer ALL questions]	Yes	No
17. Is a Disclosure and Barring Service (DBS) check required?		x
18. Will it be necessary for participants to take part in the study without their knowledge or consent at the time? (e.g. covert observation of people in non-public places)		x
19. Will the study require any deception of participants?		x
20. Will the study involve discussion of topics which the participants may find sensitive? (e.g. sexual activity, personal drug use, income etc.)		x
21. Are there cultural or religious issues associated with the research?		x
22. Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?		x
23. Are drugs, placebos or other substances (e.g. food substances, vitamins, Chinese medicine) to be administered to the study participants?[2]		x
24. Will the study involve invasive, intrusive or potentially harmful procedures of any kind? (e.g. Acupuncture, fitness testing)		x
25. Will blood or tissue samples be obtained from participants?		x
26. Does the proposed research involve human tissue or human embryos?		x
27. Is pain or more than mild discomfort to participants likely to result from the study?		x
28. Could the study induce psychological distress or anxiety or cause harm or negative consequences beyond the risks encountered in normal life?		x
29. Will the study involve prolonged or repetitive testing?		x

Checklist 3: Research directly involving animals [answer ALL questions]	Yes	No
30. Does the research involve any procedure that may have the potential effect of causing the animal(s) pain, suffering, distress or lasting harm? (regulated procedures under the terms of the Animals (Scientific Procedures) Act)		
31. Does the research involve a series of otherwise non-regulated procedures that together or cumulatively may cause that animal pain, suffering, distress or lasting harm?		
32. Does the research involve vertebrate animals or “Octopus Vulgaris” (protected animals under the terms of the Animals (Scientific Procedures) Act)3		
33. Does the research involve using any animal by-products or tissue?		
34. Does the research involve any procedure or intervention on the animal(s) that is not part of its/their normal management practice?		
35. Does the research involve movement of animals from one place to another?		
36. Does the research involve animals in the wild?		

Checklist 4: Research having a potential impact on the environment [answer ALL questions]	Yes	No
37. Do you have legal access / permission to work on the proposed site?		
38. Does the site have any legal designation (e.g. SSSI)?		

[2] Clinical Trials are not covered by Glyndŵr University insurance and such studies will also need MHRA registration and to conform with EU Clinical Trials Directive (2001)

[3] The Animals (Scientific Procedures) Act 1986 is available at https://moodle.glyndwr.ac.uk/course/view.php?id=26703

Checklist 4: Research having a potential impact on the environment [answer ALL questions]	**Yes**	**No**
39. Could the research have an impact on the environment? (e.g. air / land / water contamination, damage to animal habitats)?		
40. Does the research involve working with any Genetically Modified Organisms? (e.g. GMOs in animal feeds)?		
41. Will you be importing plants, plant material, pests, soil or growing medium into the UK?		

Table: Summary of quantitative data

Participant ID	Role	Years of Experience	Familiar with AI Tools	Satisfication with AI Tools (1-5)	Concern about Ethical Implicatinos (1-5)
1	Software Engineer	1 to 3	yes	3	4
2	Data Analyst	4 to 6	yes	4	2
3	Data Warehouse-Technical Lead	10+	yes	2	5
4	Head of ICT	10+	yes	3	3
5	Head of Product Delivery	10+	yes	4	4
6	Technical Security Artichect	10+	yes	3	5
7	Web Developer	10+	yes	4	3

Python Script - Data Analysis

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Specify the file path to the CSV file
file_path =
'/Users/holleylongfield/Documents/DISSERTATION_MSC_SOFTWAREENG/Questionnaires_and_Consent_Forms/2_Quan
titative_DATA_TableOnly.csv'

# Read the CSV file with 'utf-8-sig' encoding to remove BOM characters
data = pd.read_csv(file_path, encoding='utf-8-sig')

# Clean up column names by stripping leading and trailing spaces
data.columns = data.columns.str.strip()

# Correct the misspelled column name
data.rename(columns={'Satisfication with AI Tools . Are you not satisfied (1) to Extremely satisfied (5)':
                     'Satisfaction with AI Tools (1-5)'}, inplace=True)

# Print the column names to verify their exact names
print("Column Names in DataFrame:")
print(data.columns)

# Set pandas to display all columns
pd.set_option('display.max_columns', None)

# Display the entire DataFrame and basic statistics of the data
print(data)  # This will show all rows
print(data.describe())

# Check for duplicate rows
duplicates = data[data.duplicated()]
print("Duplicate entries in DataFrame:")
print(duplicates)

# Print unique values in 'Years of Experience'
print("Unique values in 'Years of Experience':")
print(data['Years of Experience'].unique())

# Print the number of participants
print(f"Number of participants in the dataset: {len(data)}")

# Define column names for analysis using the exact names printed above
satisfaction_column = 'Satisfaction with AI Tools (1-5)'
ethical_concerns_column = 'Concern about Ethical Implications. Are you not concerned (1) to very concerned (5)'
```

```
# Correlation between satisfaction and ethical concerns
correlation = data[satisfaction_column].corr(data[ethical_concerns_column])
print(f"Correlation between satisfaction and ethical concerns: {correlation}")

# Corrected experience mapping to match the exact survey categories
experience_mapping = {
    '1 to 3': 2,
    '4 to 6': 5,
    '10+': 10  # Adjust according to your new data
}

# Apply the mapping to create a new 'Years of Experience (Numeric)' column
data['Years of Experience (Numeric)'] = data['Years of Experience'].map(experience_mapping)

# Verify the mapping by printing the original and numeric years of experience columns
print(data[['Years of Experience', 'Years of Experience (Numeric)']])

# Check for any missing or NaN values after mapping
print("Missing values in 'Years of Experience (Numeric)':", data['Years of Experience (Numeric)'].isnull().sum())

# Correlation between satisfaction and years of experience
experience_correlation = data[satisfaction_column].corr(data['Years of Experience (Numeric)'])
print(f"Correlation between satisfaction and years of experience: {experience_correlation}")

# Analysis 1: Barplot showing Satisfaction levels by Role
plt.figure(figsize=(10, 6))
sns.barplot(x='Role', y=satisfaction_column, data=data)
plt.xticks(rotation=45)
plt.title('Satisfaction with AI Tools by Role')
plt.ylabel('Satisfaction Score (1-5)')
plt.show()

# Analysis 2: Box Plot for Ethical Concerns by Years of Experience
plt.figure(figsize=(10, 6))
sns.boxplot(x='Years of Experience', y=ethical_concerns_column, data=data)
plt.title('Ethical Concerns by Years of Experience')
plt.ylabel('Concern about Ethical Implications (1-5)')
plt.show()

# Analysis 3: Violin Plot for Ethical Concerns by Years of Experience
plt.figure(figsize=(10, 6))
sns.violinplot(x='Years of Experience', y=ethical_concerns_column, data=data)
plt.title('Distribution of Ethical Concerns by Years of Experience')
plt.ylabel('Concern about Ethical Implications (1-5)')
plt.show()

# Analysis 4: Bar Plot for Satisfaction vs. Familiarity with AI Tools
plt.figure(figsize=(10, 6))
sns.barplot(x='Familiar with AI Tools', y=satisfaction_column, data=data)
plt.title('Satisfaction with AI Tools by Familiarity')
plt.ylabel('Satisfaction Score (1-5)')
plt.show()

# Analysis 5: Scatter Plot for Ethical Concerns vs. Satisfaction
plt.figure(figsize=(10, 6))
sns.scatterplot(x=satisfaction_column, y=ethical_concerns_column, data=data)
plt.title('Satisfaction vs. Ethical Concerns')
plt.xlabel('Satisfaction Score (1-5)')
plt.ylabel('Concern about Ethical Implications (1-5)')
plt.show()

# Calculate and print correlations
```

```
satisfaction_correlation = data[satisfaction_column].corr(data[ethical_concerns_column])
print(f"Correlation between satisfaction and ethical concerns: {satisfaction_correlation}")

experience_correlation = data['Years of Experience (Numeric)'].corr(data[ethical_concerns_column])
print(f"Correlation between years of experience and ethical concerns: {experience_correlation}")

# Check if familiarity column exists and calculate correlation
if 'Familiar with AI Tools' in data.columns:
    # Convert the 'Familiar with AI Tools' to a numeric format for correlation
    data['Familiar with AI Tools Numeric'] = data['Familiar with AI Tools'].map({'yes': 1, 'no': 0})

    # Check for NaN values and calculate correlation
    if data['Familiar with AI Tools Numeric'].isnull().sum() == 0:
        familiarity_correlation = data[satisfaction_column].corr(data['Familiar with AI Tools Numeric'])
        print(f"Correlation between satisfaction and familiarity with AI tools: {familiarity_correlation}")
    else:
        print("There are NaN values in 'Familiar with AI Tools Numeric' that prevent correlation calculation.")
else:
    print("Familiarity with AI Tools column not found.")
```

Script Output

```
Column Names in DataFrame:
Index(['Participant ID', 'Role', 'Years of Experience',
       'Familiar with AI Tools', 'Satisfaction with AI Tools (1-5)',
       'Concern about Ethical Implications. Are you not concerned (1) to very concerned (5)'],
      dtype='object')
   Participant ID                    Role Years of Experience  \
0               1           Software Engineer           1 to 3
1               2                Data Analyst           4 to 6
2               3  Data Warehouse-Technical Lead              10+
3               4                Head of ICT              10+
4               5     Head of Product Delivery              10+
5               6   Technical Security Artichect              10+
6               7                Web Developer              10+

  Familiar with AI Tools  Satisfaction with AI Tools (1-5)  \
0                    yes                                3
1                    yes                                4
2                    yes                                2
3                    yes                                3
4                    yes                                4
5                    yes                                3
6                    yes                                4

   Concern about Ethical Implications. Are you not concerned (1) to very concerned (5)
0                                                  4
1                                                  2
2                                                  5
3                                                  3
4                                                  4
5                                                  5
6                                                  3
       Participant ID  Satisfaction with AI Tools (1-5)  \
count        7.000000                          7.000000
mean         4.000000                          3.285714
std          2.160247                          0.755929
min          1.000000                          2.000000
25%          2.500000                          3.000000
50%          4.000000                          3.000000
75%          5.500000                          4.000000
```

```
max         7.000000                    4.000000

       Concern about Ethical Implications. Are you not concerned (1) to very concerned (5)
count                                    7.000000
mean                                     3.714286
std                                      1.112697
min                                      2.000000
25%                                      3.000000
50%                                      4.000000
75%                                      4.500000
max                                      5.000000
Duplicate entries in DataFrame:
Empty DataFrame
Columns: [Participant ID, Role, Years of Experience, Familiar with AI Tools, Satisfaction with AI Tools (1-5), Concern about Ethical Implications. Are you not concerned (1) to very concerned (5)]
Index: []
Unique values in 'Years of Experience':
['1 to 3' '4 to 6' '10+']
Number of participants in the dataset: 7
Correlation between satisfaction and ethical concerns: -0.6793662204867574
  Years of Experience  Years of Experience (Numeric)
0              1 to 3                              2
1              4 to 6                              5
2                10+                             10
3                10+                             10
4                10+                             10
5                10+                             10
6                10+                             10
Missing values in 'Years of Experience (Numeric)': 0
Correlation between satisfaction and years of experience: -0.08622018733942649

[Done] exited with code=null in 512.472 seconds
```

APPENDIX D – THEMATIC DATA ANALYSIS TABLE

Final Thematic Analysis Summary Table

[1] *Respondent 1*	[2] *Quote(s)*	[3] *Analysis*	[4] *Related questions*
[5] *Theme 1: Effectivenss of AI Tools*	[6] *"I regularly use GitHub Copilot for code suggestions, automated scanning tools like SonarQube for identifying vulnerabilities, and ML-based monitoring systems for continuous threat detection. In a recent project, these tools helped catch a critical security flaw during development."*	[7] *This response highlights the effectiveness of multiple AI tools in improving secure coding practices. The respondent provides specific examples such as GitHub Copilot, SonarQube, and ML-based monitoring systems that were used for threat detection and vulnerability identification.*	[8] ***Sub-question 2:*** *How effective are AI tools, such as GitHub Copilot, in improving secure coding practices?*
[9]	[10] *"Effective. In one project, using AI-driven tools reduced our vulnerability rate by nearly 30%. An example is when GitHub Copilot suggested a more secure method for handling API keys, preventing potential exposure in a cloud environment."*	[11] *This further reinforces the effectiveness of AI tools, providing a tangible metric of improvement—a 30% reduction in vulnerabilities. GitHub Copilot's recommendation for better handling of API keys demonstrates its direct impact on security.*	[12] ***Sub-question 2:*** *How effective are AI tools, such as GitHub Copilot, in improving secure coding practices?*
[13] *Theme 2: Limitations of Traditional Practices*	[14] *"Some of these tools like Sonar... may not be as skilled enough... that it doesn't reject or raise a flag."*	[15] *The respondent critiques the limitations of traditional tools like Sonar, pointing out their inability to effectively flag certain issues, which can hinder productivity and reduce the effectiveness of secure coding practices.*	[16] ***Sub-question 1:*** *What are the limitations of traditional secure coding practices in addressing emerging and complex cyber threats?*
[17]	[18] *"Automated scanning technologies like Sonar, SAST, and DAST... but sometimes these tools can stop productivity."*	[19] *This quote highlights the limitations of traditional static analysis tools, suggesting that these tools, while necessary, can become bottlenecks in the development process, reducing productivity.*	[20] ***Sub-question 1:*** *What are the limitations of traditional secure coding practices in addressing emerging and complex cyber threats?*
[21] *Theme 3: Integration Challenges*	[22] *"You can't just deploy AI and expect it to do something for you. You have to give it explicit instructions."*	[23] *The respondent emphasizes the complexity of integrating AI tools, particularly the necessity for careful configuration and constant oversight. This illustrates the practical challenges that arise during the integration of AI into coding pipelines.*	[24] ***Sub-question 4****: What are the challenges in integrating AI tools, including XAI, into secure coding processes?*
[25]	[26] *"The main challenges include compatibility with legacy systems and performance bottlenecks. We addressed these by gradually phasing in AI tools and optimizing the pipeline for faster execution times. We solved AI-based*	[27] *This quote directly addresses the technical challenges related to AI tool integration, such as dealing with legacy systems and optimizing performance. The phased approach to implementation offers*	[28] ***Sub-question 4****: What are the primary challenges in integrating XAI tools into secure coding processes?*

	vulnerability scanning slowing down our CI/CD pipeline by running scans asynchronously and prioritizing critical code paths."	*insight into how these challenges were mitigated*	
[29] *Theme 4: XAI*	[30] *"Yes. Transparency is crucial, especially in security-focused environments where decisions need to be auditable and understandable by both developers and stakeholders. XAI was critical in a recent project where we needed to explain AI-driven decisions to non-technical stakeholders."*	[31] *This quote emphasizes the importance of transparency in AI-driven decisions, particularly in security-critical settings. The respondent notes the significance of explainability (XAI) for building trust among stakeholders.*	[32] ***Sub-question 3**: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines?* [33] ***Sub-question 4:** What are the primary challenges in integrating XAI tools into secure coding processes?*
[34]	[35] *"XAI allowed us to trace the logic behind an AI-based intrusion detection system's decisions, making it easier to fine-tune the system and avoid false alarms. I foresee XAI playing a significant role in regulatory compliance, where explaining AI decisions will be a legal requirement."*	[36] *This supports the role of XAI in providing transparency and ensuring that AI-driven decisions are understandable. The respondent foresees that XAI will become essential for regulatory compliance, further stressing its importance in the future of secure software development.*	[37] ***Sub-question 3**: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines?*
[38] *Theme 5: Ethical Considerations*	[39] *"One specific issue is model bias, which could lead to unintentional exclusion or unfair treatment in security-related decisions. We regularly audit our AI models and incorporate diverse data sets during training. Moving forward, companies need to establish clear AI ethics guidelines and train teams on responsible AI use."*	[40] *The respondent raises the issue of bias in AI models, which could lead to ethical problems in secure coding. The regular auditing of AI models and use of diverse datasets to mitigate these risks align with ethical AI practices, which are crucial in secure software engineering.*	[41] ***Sub-question 4:** What are the primary challenges in integrating XAI tools into secure coding processes?* [42] ***Sub-question 1**: What are the limitations of traditional secure coding practices in addressing ethical issues?*
[43]	[44] *"Increased Adoption of XAI: Transparency and explainability will become key features of AI tools, especially in security applications where understanding AI-driven decisions is critical for compliance and trust. Regulatory Focus on AI Ethics and Security: There will be more regulatory scrutiny around the ethical use of AI."*	[45] *This further highlights the ethical importance of XAI in ensuring that AI decisions are transparent and understandable. The respondent anticipates that transparency will become a critical feature for trust and regulatory compliance, which is an essential ethical concern.*	[46] ***Sub-question 4**: What are the primary challenges in integrating XAI tools into secure coding processes?*
[47] *Theme 6: Trust and Overerliance Issues with AI Tools*	[48] *"It won't be our generation. It'll be two or three generations out before people are willing to just accept some machine-generated code."*	[49] *This reflects the respondent's skepticism toward fully trusting AI systems in secure coding. While AI tools can be effective, there remains a reluctance to rely on them completely, which is a trust issue that persists within the industry.*	[50] ***Sub-question 3**: How do XAI-enhanced secure coding practices align with industry standards and regulations?*

[51] *Theme 7: Future trends and recommendations*	[52] *"Enhanced Explainability: Incorporating better XAI features to ensure security-related AI decisions can be fully understood and trusted. Seamless Integration with Legacy Systems: AI tools should offer better support for legacy codebases. Ethical AI Governance: Tools should include features for ethical use, such as bias detection."*	[53] *The respondent offers future recommendations for improving AI tools, focusing on enhanced explainability, better integration with legacy systems, and the need for ethical AI governance. These improvements would make AI more effective and acceptable in secure coding environments.*	[54] ***Sub-question 3**: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines? **Sub-question 4:** Challenges in integrating AI tools, including XAI, into secure coding processes.*
[55]	[56]	[57]	[58]
[59]	[60]	[61]	[62]
[63] *Respondent 2*	[64]	[65]	[66]
[67]	[68]	[69]	[70]
[71] *Theme 1: Effectivenss of AI Tools*	[72] *"We've attempted to use it around GitHub... some of the developers attempt to use it to explain and work out what the heck somebody did five years ago."*	[73] *GitHub Copilot and other AI tools show potential for understanding legacy code. This quote highlights AI's utility in helping developers interpret old code, which may improve productivity in secure coding practices.*	[74] ***Sub-question 2**: How effective are AI tools, such as GitHub Copilot, in improving secure coding practices?*
[75]	[76] *"Github CoPilot. Used predominantly to examine and understand legacy code written by developers who have long left the organisation and for which documentation is often incomplete or missing."*	[77] *This emphasizes the effectiveness of AI tools like GitHub Copilot in bridging knowledge gaps when dealing with undocumented or incomplete legacy code, improving secure coding practices.*	[78] ***Sub-question 2**: How effective are AI tools, such as GitHub Copilot, in improving secure coding practices?*
[79] *Theme 2: Limitations of Traditional Practices*	[80] *"They are useful; however, they often lead to a more manual examination of code, as the AI often leaves you with as many questions as answers. Indeed often left with the feeling that it would sometimes be quicker to just check everything yourself. Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"*	[81] *The respondent highlights the limitations of AI tools like GitHub Copilot, indicating that they may require additional manual reviews due to missed issues. This reflects ongoing concerns about AI tools' reliability in secure coding practices. This limits their effectiveness at streamlining secure coding processes.*	[82] ***Sub-question 1:** What are the limitations of traditional secure coding practices in addressing emerging and complex cyber threats? **Sub-question 4:** What are the primary challenges in integrating XAI tools into secure coding processes?*
[83]	[84] *"Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"*	[85] *The respondent notes a specific example of an AI tool (GitHub Copilot) missing security issues, underscoring the limitations of relying on these tools for accurate and comprehensive secure coding.*	[86] ***Sub-question 4**: What are the primary challenges in integrating XAI tools into secure coding processes?*

[87] *Theme 3: Integration Challenges*	[88] *"We haven't succeeded in integrating it into anything yet... the trust isn't there."*	[89] *AI tools have not yet been integrated into the respondent's organization due to trust issues. This indicates the challenges organizations face when trying to integrate AI tools into secure coding workflows.*	[90] ***Sub-question 4:** What are the challenges in integrating AI tools, including XAI, into secure coding processes?*
[91]	[92] *"This is something, following a number of Proof of Concepts that the various organisations I've worked for have been loathed to do in any great way. Mainly because of the legacy and technical debt that exists, making integration complex."*	[93] *Legacy systems and technical debt create significant obstacles to the integration of AI tools, demonstrating the complexity involved in incorporating new technologies into existing secure coding practices.*	[94] ***Sub-question 4**: What are the challenges in integrating AI tools, including XAI, into secure coding processes?*
[95] *Theme 4: XAI*	[96] *"Understanding what AI is doing, in your name, is incredibly important particularly in regulated industries where blaming the AI just simply isn't a valid excuse, in law."*	[97] *The respondent emphasizes the need for Explainable AI (XAI), especially in regulated industries where accountability is critical. Understanding AI decisions is essential for legal and ethical compliance*	[98] ***Sub-question 3**: How does XAI enhance transparency in secure coding practices?*
[99]	[100] *"Explainable AI... it's a confidence thing... who's doing the code, you or AI?"*	[101] *The respondent points out that XAI can help build trust in AI-generated code by providing transparency into AI's decision-making processes, but also expresses skepticism about overreliance on XAI without human oversight.*	[102] ***Sub-question 3**: How does XAI enhance transparency in secure coding practices?*
[103] *Theme 5: Ethical Considerations*	[104] *"In finance, decision making on the basis of race or other social factors affecting the ability of customers to access everyday finance products. In a wider arena, it concerns me about the use of such practices in National Security and Policing."*	[105] *The respondent raises ethical concerns about bias in AI systems, particularly in sectors like finance and national security. This highlights the potential for AI tools to perpetuate discrimination if not carefully managed.*	[106] ***Sub-question 4:** What are the ethical considerations in using AI tools for secure coding?*
[107] *Theme 6: Trust and Overerliance Issues with AI Tools*	[108] *"You still have to review the AI's work... No one will trust it completely anytime soon."*	[109] *Trust remains a significant barrier to AI adoption. The respondent underscores the need for human oversight, as AI-generated outputs are not yet reliable enough for complete trust in secure coding environments.*	[110] ***Sub-question 2:** How does XAI enhance trust in AI-driven decisions?* ***Sub-question 4:** What are the primary challenges in integrating XAI tools into secure coding processes?*
[111]	[112] *"It's confidence again... it's all regulatory if you make a mistake and start reporting that to the regulator... they're not going to care... I trusted an AI."*	[113] *The respondent highlights that regulators will not accept AI-based decisions as an excuse for mistakes, reinforcing the trust and accountability issues surrounding AI tools in secure coding.*	[114] ***Sub-question 4**: What are the primary challenges in integrating XAI tools into secure coding processes?*
[115] *Theme 7: Future trends and recommendations*	[116] *"I am aware of benefits in the Cyber Security arena where AI allows rapid identification of trends and vulnerabilities,*	[117] *The respondent acknowledges the potential future benefits of AI in cybersecurity, where AI tools can reduce*	[118] ***Main Research Question**: How can AI tools, particularly Explainable AI (XAI), enhance secure coding practices and*

	reducing workload for analysts and making decision making quicker."	*workloads and enhance decision-making by quickly identifying vulnerabilities and trends.*	*align with security standards**?***
[119]	[120]	[121]	[122]
[123]	[124]	[125]	[126]
[127] *Respondent 3*	[128]	[129]	[130]
[131] *Theme 1: Effectivenss of AI Tools*	[132] *"AI tools can be highly effective in improving secure coding practices, providing real-time assistance, automating security checks, and enhancing overall code quality. Here are some specific examples:..."*	[133] *The respondent finds AI tools to be highly effective in improving secure coding practices. They highlight real-time assistance, automated security checks, and enhanced code quality as key benefits.*	[134] *Main Research Question: How can AI tools, particularly XAI, enhance secure coding practices? Sub-question 2: How effective are AI tools, such as GitHub Copilot and automated scanning technologies, in improving secure coding practices?*
[135]	[136]	[137]	[138]
[139] *Theme 2: Limitations of Traditional Practices*	[140] *"I think it's... AI at the moment is like a 5-year-old child. It hasn't learned enough. They haven't progressed enough to be able to do the next stage."*	[141] *The respondent compares AI to a "5-year-old child," pointing out its immaturity and the limitations that prevent AI from progressing to more advanced stages of use, reflecting the challenges posed by traditional tools in advancing secure coding practices*	[142] *Sub-question 1: What are the limitations of traditional secure coding practices in addressing emerging and complex cyber threats?*
[143]	[144]	[145]	[146]
[147] *Theme 3: Integration Challenges*	[148] *"Integrating AI tools into existing CI/CD (Continuous Integration/Continuous Deployment) pipelines can be highly beneficial but comes with several challenges. Here are some common challenges and how they can be addressed:..."*	[149] *The respondent highlights challenges in integrating AI into CI/CD pipelines, such as compatibility, resource allocation, and managing the additional computational requirements. They also offer solutions, such as selecting compatible tools and leveraging scalable cloud services.*	[150] ***Sub-question 4**: What are the primary challenges in integrating XAI tools into secure coding processes?*
[151]	[152]	[153]	[154]
[155] *Theme 4: XAI*	[156] *"Yes, I'm aware of Explainable AI (XAI) techniques, which are designed to make the decision-making processes of AI models more transparent and understandable to humans. XAI is particularly important in contexts where trust, accountability, and decision-making need to be transparent, such as secure software engineering...Key XAI Techniques: Model-agnostic Methods...SHAP...Intrinsically Interpretable Models: Decision Trees and Rule-based*	[157] *The respondent demonstrates awareness of XAI and emphasizes the importance of transparency and accountability in secure software engineering. XAI is necessary to make AI decision-making processes more understandable, enhancing trust.*	[158] *Main Research Question: How can AI tools, particularly XAI, enhance secure coding practices?*

	Systems...Generalised Additive Models(GAMS)"		
[159]	[160] *"Using Explainable AI (XAI) techniques in AI-driven security solutions offers several benefits that enhance the effectiveness, trustworthiness, and overall usability of these tools. Here's a breakdown of the key benefits:…transparency… enhanced decision-making..."*	[161] *XAI techniques enhance transparency and improve the effectiveness of AI-driven security solutions. The respondent views XAI as crucial for better decision-making and ensuring trust in secure coding.*	[162] ***Sub-question 3**: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines?*
[163] *Theme 5: Ethical Considerations*	*"The use of AI tools in secure software engineering brings several ethical issues that need careful consideration. Here are some specific ethical concerns that have been encountered or could arise…"AI models are trained on data, and if this data is biased, the models may inherit and propagate these biases. In the context of secure software engineering, this could mean that certain coding practices, languages, or even types of projects are unfairly flagged as more or less secure based on the biases present in the training data.."... "The "black-box" nature of some AI models can make it difficult for developers and security professionals to understand how decisions are being made. This lack of transparency can lead to ethical concerns, particularly when AI tools are used to make critical security decisions"...."AI tools used in secure software engineering often require access to sensitive codebases, data, and infrastructure. This access can introduce security and privacy risks, especially if the AI tools themselves are not secure."... As AI tools become more autonomous, determining accountability for decisions made by these tools becomes challenging. In secure software engineering, where mistakes can have serious consequences, it's crucial to establish who is responsible for AI-driven decisions.* [164]	[165] *The respondent mentions ethical concerns related to AI tools in secure software engineering. These include potential biases, accountability issues, and the need for transparent, responsible AI usage.*	[166] ***Sub-question 4**: What are the primary challenges in integrating XAI tools into secure coding processes?*
[167]	[168] *Different problem, different problem because you're within the defense industry the way they're looking at doing some of it. It can be deemed as... you know, oops, collateral damage, and I don't think anybody's prepared to do that. Yeah, it's like, the people flying fighter jets... you're sitting there, and you've got to fire a shot or a missile off or drop a*	[169] *The respondent discusses ethical issues in AI decision-making, particularly in defense applications. They emphasize the need for human judgment to avoid catastrophic mistakes that AI might not detect.*	[170] ***Sub-question 1:** What are the limitations of traditional secure coding practices in addressing emerging and complex cyber threats?*

	bomb. You're thinking about it before you, and you decide whether, 'No, that's wrong. That information is wrong. That's the school; that's not a munitions dump.' I won't do it.		
[171] *Theme 6: Trust and Overerliance Issues with AI Tools*	[172] *I think the biggest barrier at the moment is that People are treating it like a gadget like They're not taking it very seriously. They're thinking of, 'Oh, I don't need to search on Google for this,' or the kid at school, 'I don't need to answer all my questions and write up an essay. I can get it to do it for me.' Everybody's thinking, what can it do for me? Not what it can do for us as people?*	[173] *The respondent identifies a lack of trust in AI tools as a key barrier, with people viewing AI as a convenience rather than recognizing its potential benefits for collective problem-solving.*	[174] ***Sub-question 4**: What are the primary challenges in integrating XAI tools into secure coding processes?*
[175]	[176] *"At the moment, I think people are trustier now with a lot of things within their life, even though it's only a 5-year-old. They tend to think, 'Oh, look what's been invented! Oh, the Internet's always right. So this is brilliant. This is gonna save me so much time,' and they believe it."*	[177] *The respondent notes a paradox where trust in AI has increased despite the technology's immaturity. This suggests a potential overreliance on AI without fully understanding its limitations, which could lead to issues in secure coding.*	[178] ***Sub-question 4:** What are the primary challenges in integrating XAI tools into secure coding processes?*
[179] *Theme 7: Future trends and recommendations*	*"The integration of AI and secure software engineering is expected to evolve significantly in the coming years, driven by advancements in AI technologies and the increasing complexity of cybersecurity challenges. Here are some future trends that can be anticipated:... AI will increasingly be used to predict and prevent security threats before they occur. By analysing patterns and behaviours in real-time, AI can anticipate potential vulnerabilities or attack vectors and suggest pre-emptive measures."... AI will become an integral part of DevSecOps, automating security checks at every stage of the software development lifecycle. This will include AI-driven static and dynamic code analysis, automated threat modelling, and continuous monitoring….AI will play a larger role in incident response, helping security teams detect, analyse, and respond to security incidents more quickly and accurately. AI-powered tools will be able to automate the identification of threats, prioritize incidents, and even initiate automated responses"... AI tools that assist in writing secure code*	[181] *The respondent anticipates future trends in AI integration within secure software engineering, emphasizing how advancements in AI will address increasingly complex cybersecurity challenges.*	[182] ***Main Research Question**: How can AI tools, particularly XAI, enhance secure coding practices? **Sub-question 2**: How effective are AI tools, such as GitHub Copilot and automated scanning technologies, in improving secure coding practices?*

	will become more sophisticated, providing developers with real-time suggestions and corrections as they code. These tools will leverage machine learning models trained on vast datasets of secure and insecure code examples." [180]		
[183]	*"To better support secure software engineering practices, AI tools can be improved or enhanced with the following features and capabilities:... Context-Aware Security Recommendations…Real-Time Secure Coding Assistance…Adaptive Learning from Feedback Loops…Integration with Threat Intelligence Feeds…Automated Threat Modelling and Risk Assessment"* [184]	[185] *The respondent provides recommendations for improving AI tools, focusing on enhancing threat detection, better explainability, and seamless integration into DevSecOps workflows.*	[186] ***Sub-question 4**: What are the primary challenges in integrating XAI tools into secure coding processes?*
[187]	[188]	[189]	[190]
[191]	[192]	[193]	[194]
[195] *Respondent 4*	[196]	[197]	[198]
[199] *Theme 1: Effectivenss of AI Tools*	[200] *"I find AI tools like ChatGPT and GitHub Copilot highly effective in improving secure coding practices. For example, ChatGPT helps me quickly understand and implement best practices in secure coding by providing explanations and examples. GitHub Copilot assists by suggesting secure code snippets and identifying potential security flaws in real-time, thereby enhancing the overall security of my code."*	[201] *Respondent finds AI tools like ChatGPT and GitHub Copilot highly effective in enhancing secure coding practices. These tools assist in understanding best practices and identifying real-time security flaws, ultimately improving code security.*	[202] *Main Research Question: How can AI tools, particularly XAI, enhance secure coding practices?* [203] *Sub-question 2: How effective are AI tools, such as GitHub Copilot and automated scanning technologies, in improving secure coding practices?*
[204]	[205]	[206]	[207]
[208] *Theme 2: Limitations of Traditional Practices*	[209]	[210]	[211]
[212]	[213]	[214]	[215]
[216] *Theme 3: Integration Challenges*	[217] *"Integrating AI tools into existing CI/CD pipelines has its challenges. One major issue is ensuring these tools work smoothly with the existing setup. I've tackled this by choosing AI tools that are compatible with our CI/CD platforms and have good API support. Another challenge is the extra computational power needed for AI operations, which I manage by leveraging scalable cloud services. Balancing speed and thoroughness can be*	[218] *Respondent discusses the challenges of integrating AI tools into CI/CD pipelines, including compatibility and computational power issues. They overcome these challenges by choosing compatible tools, using cloud services, and fine-tuning the AI tools to balance speed and security.*	[219] *Sub-question 4: What are the primary challenges in integrating XAI tools into secure coding processes?*

	tricky too, so I fine-tune the AI tools to focus on essential security checks without slowing down the deployment process too much."		
[220]	[221]	[222]	[223]
[224] *Theme 4: XAI*	[225] *"Transparency and explainability in AI tools are crucial for secure software engineering as they help developers understand the reasoning behind AI-driven decisions and recommendations. XAI clarity ensures that security measures are well-founded and trustworthy, enabling more effective identification and mitigation of potential security risks."*	[226] *highlights the importance of transparency and explainability in AI tools for secure software engineering. XAI ensures that AI-driven decisions are understandable, making security measures more trustworthy and effective.*	[227] ***Main Research Question**: How can AI tools, particularly XAI, enhance secure coding practices? **Sub-question 3:** How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines?*
[228]	[229] *"Transparency is crucial for understanding how AI makes decisions, which helps in identifying potential issues and building trust. Accountability is also vital; developers and organizations must take responsibility for the actions and decisions made by AI tools to ensure they are used ethically and responsibly."*	[230] *The respondent underscores that transparency and accountability are essential for building trust in AI tools. Developers and organizations must ensure that AI tools are used ethically by taking responsibility for their decisions and actions.*	[231] ***Sub-question 3**: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines?*
[232] *Theme 5: Ethical Considerations*	[233]	[234]	[235]
[236]	[237]	[238]	[239]
[240] *Theme 6: Trust and Overerliance Issues with AI Tools*	[241] *"In the future, I foresee AI playing a pivotal role in secure software engineering by enhancing threat detection and automating code remediation. The integration of Explainable AI (XAI) will ensure transparency and trust in AI-driven security decisions. AI tools will seamlessly integrate into DevSecOps pipelines, offering real-time security analysis and personalized recommendations. Additionally, AI will aid in compliance and governance, automating audits and providing detailed reports to meet regulatory standards."*	[242] *The respondent predicts that AI will enhance secure software engineering through threat detection and automated code remediation. XAI will play a key role in ensuring transparency and trust, making AI tools more widely adopted in security workflows.*	[243] ***Sub-question 3**: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines?*
[244]	[245]	[246]	[247]
[248] *Theme 7: Future trends and recommendations*	[249] *"AI tools could be improved by enhancing threat detection, integrating seamlessly with DevSecOps*	[250] *The respondent suggests ways to improve AI tools, including enhancing threat detection, better integration with*	[251] *Sub-question 4: What are the primary challenges in integrating XAI tools into secure coding processes?*

	workflows, and providing better explainability of AI decisions. They should offer context-specific security recommendations, automated code fixes, and robust compliance features to support industry standards and regulations. Reducing bias by training on diverse datasets is also crucial."	*workflows, improved explainability, and reducing bias through diverse datasets. These recommendations aim to improve AI's effectiveness and compliance with industry standards.*	
[252]	[253]	[254]	[255]
[256]	[257]	[258]	[259]
[260] *Respondent 5*	[261]	[262]	[263]
[264] *Theme 1: Effectivenss of AI Tools*	[265] *Also theme 7; "Improved Code Creation, better software testing, AI Driven Personalisation, Bug Detection and Debugging, Quality assurance testing."*	[266] *This response fits under the effectiveness of AI tools in improving various aspects of secure coding, including code creation, testing, and debugging. Additionally, it touches on the potential future trends in AI's role in improving software engineering processes.*	[267] *Main Research Question: How can AI tools, particularly XAI, enhance secure coding practices? Sub-question 2: how effective are AI tools, such as GitHub Copilot and automated scanning technologies in improving secure coding practices?*
[268]	[269]	[270]	[271]
[272] *Theme 2: Limitations of Traditional Practices*	[273] *none given*	[274] *n/a*	[275]
[276]	[277]	[278]	[279]
[280] *Theme 3: Integration Challenges*	[281] *also seen in theme 5: "Pharmaceutical - Due to the heavily regulated Pharma industry we need to ensure our systems are robust and secure. All software needs to be carefully analysed for security, GDPR etc. before adoption."*	[282] *This emphasizes the integration challenges in highly regulated industries like pharmaceuticals, especially due to the need for compliance with security and GDPR standards. Ethical considerations also play a role in ensuring proper data privacy and regulatory adherence.*	[283] *Sub-question 3: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines?*
[284]	[285] *At present nothing. The Global Business is currently defining Governance for Microsoft Copilot which will be the only AI tool [omitted for privacy]adopts in the near future. The business is interested to see what Copilot can bring and from its usage and testing will define whether we look into different areas moving forward. One specific area is around the CRM tool and could help us with targeting customers at the correct time of year etc."*	[286] *the respondent describes the challenges of AI adoption, indicating a cautious approach with Microsoft Copilot. They also look to future use cases such as CRM targeting, showing a business-focused perspective on future AI use and integration.*	[287] *Sub-question 4. What are the primary challenges in integrating XAI tools into secure coding processes?*
[288]	[289] *"As I am not a developer this is not something I am overly familiar with."*	[290] *This aligns with the integration challenges, specifically in non-developer contexts where technical familiarity with*	[291] *Sub-question 4. What are the primary challenges in integrating XAI tools into secure coding processes? Suggests awareness of*

		AI tools might hinder the adoption and seamless use of AI within secure coding practices.	*XAI may be limited among certain roles, impacting adoption.*
[292]	[293] *"Again, not being a developer makes this a tough question to answer. But if the pipeline is poorly designed in the first place the introduction of an AI tool would likely result in failures. A quick fix for this would be ensure coding is written with easily identifiable logging capabilities."*	[294] *This highlights potential integration challenges when AI tools are introduced into poorly designed systems, stressing the need for well-structured coding pipelines and logging to ensure that AI enhances rather than disrupts secure coding processes.*	[295] *Sub-question 4. What are the primary challenges in integrating XAI tools into secure coding processes?*
[296] *Theme 4: XAI*	[297] *"Data Privacy needs to be tightened. AI is being used increasingly in Software Development which brings into question how data is scanned and used. Better guidelines for Ethics, establishing unambiguous guidelines for moral AI development and application is essential to ensuring that technology advances society rather than undermines it."*	[298] *While not directly mentioning XAI, this response highlights concerns about how AI is used and the need for* ***better ethical guidelines*** *to ensure* ***accountability****. This could implicitly suggest a need for* ***transparency*** *in AI processes, which is at the core of* ***XAI****.*	[299] *Sub-question 3: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines? Sub-question 4: What are the primary challenges in integrating XAI tools into secure coding processes?*
[300]	[301]	[302]	[303]
[304] *Theme 5: Ethical Considerations*	[305] *also seen in theme 2: "Pharmaceutical - Due to the heavily regulated Pharma industry we need to ensure our systems are robust and secure. All software needs to be carefully analysed for security, GDPR etc. before adoption."*	[306] *This emphasizes the integration challenges in highly regulated industries like pharmaceuticals, especially due to the need for compliance with security and GDPR standards. Ethical considerations also play a role in ensuring proper data privacy and regulatory adherence.*	[307]
[308]	[309] *"A well-known concern in AI systems is their potential to reflect and amplify biases present in their training data. When used in testing, a biased AI could lead to uneven results. Ensuring diverse and representative training data is essential to avoid these biases in the software being tested."*	[310] *This directly addresses ethical considerations, focusing on the importance of diverse datasets in mitigating bias within AI systems. Bias amplification in testing could result in unequal and potentially harmful outcomes.*	[311] *Sub-question 1: What are the limitations of traditional secure coding practices in addressing emerging and complex cyber threats? Sub-question 4: What are the primary Challenges in integrating XAI tools into secure coding processes*
[312]	[313] *"Data Privacy needs to be tightened. AI is being used increasingly in Software Development which brings into question how data is scanned and used. Better guidelines for Ethics, establishing unambiguous guidelines for moral AI development and application is essential to ensuring that technology advances*	[314] *This addresses ethical concerns regarding data privacy and the need for clearer ethical guidelines in AI development. The respondent also hints at future recommendations to ensure that AI positively contributes to society, aligning this with both ethical considerations and future trends.*	[315] *Sub-question 3: How do XAI-enhanced secure coding practices align with security standards like ISO/IEC 27001 and NIST guidelines? Sub-question 4: What are the primary challenges in integrating XAI tools into secure coding processes?*

	society rather than undermines it."		
[316] *Theme 6: Trust and Overerliance Issues with AI Tools*	[317] *At present nothing. The Global Business is currently defining Governance for Microsoft Copilot which will be the only AI tool [omitted for privacy]adopts in the near future. The business is interested to see what Copilot can bring and from its usage and testing will define whether we look into different areas moving forward. One specific area is around the CRM tool and could help us with targeting customers at the correct time of year etc."*	[318] *Although this doesn't directly mention trust, it implies a cautious and measured approach to AI adoption, which can stem from* ***trust*** *concerns. The fact that the business is waiting to evaluate what Copilot will bring indicates they are not yet over-relying on AI but are testing its use carefully to avoid overreliance.*	[319] *Sub-question 4. What are the primary challenges in integrating XAI tools into secure coding processes?*
[320]	[321]	[322]	[323]
[324] *Theme 7: Future trends and recommendations*	[325] *"Improved Code Creation, better software testing, AI Driven Personalisation, Bug Detection and Debugging, Quality assurance testing."*	[326] *This response fits under the effectiveness of AI tools in improving various aspects of secure coding, including code creation, testing, and debugging. Additionally, it touches on the potential future trends in AI's role in improving software engineering processes.*	[327]
[328]	[329] *Also seen in theme 5: "Data Privacy needs to be tightened. AI is being used increasingly in Software Development which brings into question how data is scanned and used. Better guidelines for Ethics, establishing unambiguous guidelines for moral AI development and application is essential to ensuring that technology advances society rather than undermines it."*	[330] *This addresses ethical concerns regarding data privacy and the need for clearer ethical guidelines in AI development. The respondent also hints at future recommendations to ensure that AI positively contributes to society, aligning this with both ethical considerations and future trends.*	[331]
[332]	[333]	[334]	[335]
[336] *Respondent 6*	[337]	[338]	[339]
[340] *Theme 1: Effectivenss of AI Tools*	[341] *"Very effective at providing suggestions to problems faced with extracting and manipulating data. Particularly with designing and implementing data flows."*	[342] *AI tools like Microsoft Copilot are effective in enhancing the workflow, especially for extracting, manipulating data, and improving data flow design.*	[343] *Sub-question 2: How effective are AI tools in improving secure coding practices?*
[344] *Theme 2: Limitations of Traditional Practices*	[345]	[346]	[347]
[348] *Theme 3: Integration Challenges*	[349] *"Due to the nature of security ramifications of our data, implementation of AI in our data life cycle is strictly prohibited."*	[350] *AI implementation is restricted due to the security risks associated with classified data in the defense sector.*	[351] *Sub-question 4: What are the primary challenges in integrating XAI tools into secure coding processes?*
[352] *Theme 4: XAI*	[353] *"The ability for AI to recognize the potential*	[354] *The respondent suggests that AI should recognize*	[355] *Sub-question 3: How do XAI-enhanced secure*

	weaknesses of its own outputs, providing alternative solutions for different use cases. This would make decision-making more transparent and allow developers to make informed decisions."	*its weaknesses and provide alternative solutions, increasing transparency and informed decision-making.*	*coding practices align with security standards like ISO/IEC 27001 and NIST guidelines?*
[356] *Theme 5: Ethical Considerations*	[357] *I can see that if there are specific biases present in the training data, then these biases will be replicated in the output.*	[358] *The respondent identifies bias as a potential issue, where biased training data could lead to flawed outputs.*	[359] *Sub-question 1: What are the limitations of traditional secure coding practices in addressing ethical issues?*
[360] *Theme 6: Trust and Overerliance Issues with AI Tools*	[361] *"I predict that as AI tools become more prevalent, and the training data becomes muddied with AI-generated content, then these biases will be compounded."*	[362] *The respondent is concerned that AI-generated content will add further biases, affecting the quality of output in the future.*	[363] *Sub-question 4: What are the primary challenges in integrating XAI tools into secure coding processes?*
[364] *Theme 7: Future trends and recommendations*	[365] *"I foresee that as training data becomes flooded with AI-generated content, then the outputs will trend towards the mean, reducing and stifling innovation.".*	[366] *The respondent foresees a decline in innovation if AI-generated content overwhelms training data, leading to standardized, less creative solutions.*	[367] *Main Research Question: How can AI tools, particularly XAI, enhance secure coding practices?*
[368]	[369]	[370]	[371]
[372] *Respondent 7*	[373]	[374]	[375]
[376] *Theme 1: Effectivenss of AI Tools*	[377] *Chat GPT 90% effective, im probably not providing enough info for perfect answer every time. 90% for CoPilot code line auto completion, it's no always right, but most of the time it is.*	[378] *Despite AI's usefulness, the respondent notes that Copilot and ChatGPT have limitations in accuracy, necessitating human oversight for secure coding.*	[379] *Sub-question 1: What are the limitations of traditional secure coding practices in addressing emerging and complex cyber threats?*
[380] *Theme 2: Limitations of Traditional Practices*	[381]	[382]	[383]
[384] *Theme 3: Integration Challenges*	[385]	[386]	[387]
[388] *Theme 4: XAI*	[389]	[390]	[391]
[392] *Theme 5: Ethical Considerations*	[393] *... strong oversite by external organisation…*	[394] *The respondent emphasizes the need for transparency and accountability in AI tool usage. They suggest that developers and organizations must take responsibility for AI-driven decisions to ensure ethical usage.*	[395] *Sub-question 4: What are the primary challenges in integrating XAI tools into secure coding processes? Sub-question 1: What are the limitations of traditional secure coding practices in addressing ethical issues?*
[396] *Theme 6: Trust and Overerliance Issues with AI Tools*	[397] *Will AI for get your data when asking it to crunch it for you!*	[398] *The respondent is concerned about data privacy and AI's retention of sensitive data, highlighting potential trust issues in AI tools used for secure coding.*	[399] *Sub-question 4: What are the primary challenges in integrating XAI tools into secure coding processes?*
[400] *Theme 7: Future trends and recommendations*	[401] *When asked what future trends you foresee: "Creating the the entire project by verbally dictating to AI"*	[402] *The respondent envisions a future where AI could create entire projects through verbal dictation, indicating a significant leap in AI capabilities.*	[403] *Main research question: How can AI tools, particularly Explainable AI (XAI), enhance secure coding practices?*

APPENDIX E – THEME IDENTIFICATION

EFFECTIVENESS AND LIMITATIONS OF AI TOOLS

THEME: AI TOOL EFFECTIVENESS

FREQUENCY: 7/7

Key Insights*: AI tools significantly improve productivity and security, offering real-time assistance in code suggestions and vulnerability detection.*

Quotes

1- "I regularly use GitHub Copilot for code suggestions, automated scanning tools like SonarQube for identifying vulnerabilities, and ML-based monitoring systems for continuous threat detection. In a recent project, these tools helped catch a critical security flaw during development."..."Effective. In one project, using AI-driven tools reduced our vulnerability rate by nearly 30%. An example is when GitHub Copilot suggested a more secure method for handling API keys, preventing potential exposure in a cloud environment."

2-"Github CoPilot. Used predominantly to examine and understand legacy code written by developers who have long left the organisation and for which documentation is often incomplete or missing."

3-"AI tools can be highly effective in improving secure coding practices, providing real-time assistance, automating security checks, and enhancing overall code quality. Here are some specific examples:..."

4-"I find AI tools like ChatGPT and GitHub Copilot highly effective in improving secure coding practices. For example, ChatGPT helps me quickly understand and implement best practices in secure coding by providing explanations and examples. GitHub Copilot assists by suggesting secure code snippets and identifying potential security flaws in real-time, thereby enhancing the overall security of my code."

5-"Improved Code Creation, better software testing, AI Driven Personalisation, Bug Detection and Debugging, Quality assurance testing."

6-"Very effective at providing suggestions to problems faced with extracting and manipulating data. Particularly with designing and implementing data flows."

7-
"Chat GPT 90% effective, I'm probably not providing enough info for perfect answer every time. 90% for CoPilot code line auto completion, it's no always right, but most of the time it is."

LIMITATIONS OF TRADITIONAL PRACTICES

THEME: LIMITATIONS OF TRADITIONAL PRACTICES

FREQUENCY: 3/7

Key Insights*: Participants express frustration with traditional tools, which often stall productivity and may not adequately detect vulnerabilities.*

Quotes

1-"Some of these tools like Sonar... may not be as skilled enough... that it doesn't reject or raise a flag."...
"Automated scanning technologies like Sonar, SAST, and DAST... but sometimes these tools can stop productivity."

2-"They are useful; however, they often lead to a more manual examination of code, as the AI often leaves you with as many questions as answers. Indeed, often left with the feeling that it would sometimes be quicker to just check everything yourself. Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"...
"Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"

3- "I think it's... AI at the moment is like a 5-year-old child. It hasn't learned enough. They haven't progressed enough to be able to do the next stage."

INTEGRATION CHALLENGES

THEME: INTEGRATION CHALLENGES

FREQUENCY:6/7

Key Insights*: Integration of AI tools into existing CI/CD pipelines presents significant barriers, including technical debt and compatibility issues with legacy systems.*

Quotes

1-"The main challenges include compatibility with legacy systems and performance bottlenecks. We addressed these by gradually phasing in AI tools and optimising the pipeline for faster execution times." ...
"You can't just deploy AI and expect it to do something for you. You have to give it explicit instructions."

2-"They are useful; however, they often lead to a more manual examination of code, as the AI often leaves you with as many questions as answers. Indeed, often left with the feeling that it would sometimes be quicker to just check everything yourself. Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"...
"Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"

3-
"I think it's... AI at the moment is like a 5-year-old child. It hasn't learned enough. They haven't progressed enough to be able to do the next stage."

4-
"Integrating AI tools into existing CI/CD pipelines has its challenges. One major issue is ensuring these tools work smoothly with the existing setup. I've tackled this by choosing AI tools that are compatible with our CI/CD platforms and have good API support. Another challenge is the extra computational power needed for AI operations, which I manage by leveraging scalable cloud services. Balancing speed and thoroughness can be tricky too, so I fine-tune the AI tools to focus on essential security checks without slowing down the deployment process too much."

5-
"... Due to the heavily regulated Pharma industry we need to ensure our systems are robust and secure. All software needs to be carefully analysed for security, GDPR etc. before adoption."...
At present nothing. The Global Business is currently defining Governance for Microsoft Copilot which will be the only AI tool

[omitted for privacy] adopts in the near future. The business is interested to see what Copilot can bring and from its usage and testing will define whether we look into different areas moving forward. One specific area is around the CRM tool and could help us with targeting customers at the correct time of year etc."

6-
"Due to the nature of security ramifications of our data, implementation of AI in our data life cycle is strictly prohibited."
as answers. Indeed, often left with the feeling that it would sometimes be quicker to just check everything yourself. Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"...
"Had experiences of CoPilot missing the occasional thing. It's not nice telling the boss that CoPilot missed something that caused an issue, when that's what he is paying me for!"

EXPLAINABLE AI (XAI)

THEME: EXPLAINABLE AI (XAI)

FREQUENCY:6/7

***Key Insights**:* XAI techniques are critical for enhancing trust, compliance, and transparency, especially in regulated industries, by providing explanations of AI-driven decisions.

Quotes

1-
"Yes. Transparency is crucial, especially in security-focused environments where decisions need to be auditable and understandable by both developers and stakeholders. XAI was critical in a recent project where we needed to explain AI-driven decisions to non-technical stakeholders."...
"XAI allowed us to trace the logic behind an AI-based intrusion detection system's decisions, making it easier to fine-tune the system and avoid false alarms. I foresee XAI playing a significant role in regulatory compliance, where explaining AI decisions will be a legal requirement."

2-
"Understanding what AI is doing, in your name, is incredibly important particularly in regulated industries where blaming the AI just simply isn't a valid excuse, in law."...
"Explainable AI... it's a confidence thing... who's doing the code, you or AI?"

3-
"...XAI is particularly important in contexts where trust, accountability, and decision-making need to be transparent, such as secure software engineering..." ...
"Using Explainable AI (XAI) techniques in AI-driven security solutions offers several benefits that enhance the effectiveness, trustworthiness, and overall usability of these tools..."

4-
"Transparency and explainability in AI tools are crucial for secure software engineering as they help developers understand the reasoning behind AI-driven decisions and recommendations. XAI clarity ensures that security measures are well-founded and trustworthy, enabling more effective identification and mitigation of potential security risks."...
"Transparency is crucial for understanding how AI makes decisions, which helps in identifying potential issues and building trust. Accountability is also vital; developers and organizations must take responsibility for the actions and decisions made by AI tools to ensure they are used ethically and responsibly."

5-
"Data Privacy needs to be tightened. AI is being used increasingly in Software Development which brings into question how data is scanned and used. Better guidelines for Ethics, establishing unambiguous guidelines for moral AI development and application is essential to ensuring that technology advances society rather than undermines it."

6-
"The ability for AI to recognize the potential weaknesses of its own outputs, providing alternative solutions for different use cases. This would make decision-making more transparent and allow developers to make informed decisions."

ETHICAL CONSIDERATIONS

THEME: ETHICAL CONSIDERATIONS

FREQUENCY: 6/7

***Key Insights**: Strong concerns exist about bias, accountability, and the need for ethical guidelines in the use of AI tools, particularly in decision-making processes*

Quotes

1-"One specific issue is model bias, which could lead to unintentional exclusion or unfair treatment in security-related decisions. We regularly audit our AI models and incorporate diverse data sets during training. Moving forward, companies need to establish clear AI ethics guidelines and train teams on responsible AI use."...
"Increased Adoption of XAI: Transparency and explainability will become key features of AI tools, especially in security applications where understanding AI-driven decisions is critical for compliance and trust. Regulatory Focus on AI Ethics and Security: There will be more regulatory scrutiny around the ethical use of AI."

2-
"In finance, decision making on the basis of race or other social factors affecting the ability of customers to access everyday finance products. In a wider arena, it concerns me about the use of such practices in National Security and Policing."

3-
"The use of AI tools in secure software engineering brings several ethical issues that need careful consideration. Here are some specific ethical concerns that have been encountered or could arise:..."... Different problem, different problem because you're within the defence industry the way they're looking at doing some of it. It can be deemed as... you know, oops, collateral damage, and I don't think anybody's prepared to do that. Yeah, it's like, the people flying fighter jets... you're sitting there, and you've got to fire a shot or a missile off or drop a bomb. You're thinking about it before you, and you decide whether, 'No, that's wrong. That information is wrong. That's the school; that's not a munitions dump.' I won't do it.

5-
: "... Due to the heavily regulated Pharma industry we need to ensure our systems are robust and secure. All software needs to be carefully analysed for security, GDPR etc. before adoption."...
"A well-known concern in AI systems is their potential to reflect and amplify biases present in their training data. When used in testing, a biased AI could lead to uneven results. Ensuring diverse and representative training data is essential to avoid these biases in the software being tested."...
"Data Privacy needs to be tightened. AI is being used increasingly in Software Development which brings into question how data is scanned and used. Better guidelines for Ethics, establishing unambiguous

guidelines for moral AI development and application is essential to ensuring that technology advances society rather than undermines it."

6-
"I can see that if there are specific biases present in the training data, then these biases will be replicated in the output."

7-
"... strong oversite by external organisation…"

TRUST AND OVERRELIANCE ISSUES

THEME: TRUST AND OVERRELIANCE ISSUES WITH AI

FREQUENCY: 6/7

***Key Insights**: Scepticism toward AI tools remain high, with participants expressing the necessity of human oversight to ensure accountability in AI-driven decisions.*

Quotes

1-"It won't be our generation. It'll be two or three generations out before people are willing to just accept some machine-generated code."

2-
"You still have to review the AI's work... No one will trust it completely anytime soon."... "It's confidence again... it's all regulatory if you make a mistake and start reporting that to the regulator... they're not going to care... I trusted an AI."

3-
"... the biggest barrier at the moment is that People are treating it like a gadget…not taking it very seriously. They're thinking … I don't need to search on Google for this,… I can get it[AI] to do it for me.' Everybody's thinking, what can it do for me? Not what it can do for us as people?"… "At the moment, I think people are trustier now with a lot of things within their life, even though it's only a 5-year-old. They tend to think, 'Oh, look what's been invented! Oh, the Internet's always right…This is gonna save me so much time,' and they believe it."

4-
"In the future, I foresee AI playing a pivotal role in secure software engineering by enhancing threat detection and automating code remediation. The integration XAI will ensure transparency and trust in AI-driven security decisions. AI tools will seamlessly integrate into DevSecOps pipelines, offering real-time security analysis and personalized recommendations. Additionally, AI will aid in compliance and governance, automating audits and providing detailed reports to meet regulatory standards."

5-
"… The Global Business is currently defining Governance for Microsoft Copilot which will be the only AI tool [omitted]adopts in the near future. The business is interested to see what Copilot can bring and from its usage and testing will define whether we look into different areas moving forward. One specific area is around the CRM tool and could help us with targeting customers at the correct time of year etc."

6-
"I predict that as AI tools become more prevalent, and the training data becomes muddied with AI-generated content, then these biases will be compounded."

7-
"Will AI for get your data when asking it to crunch it for you!"

FUTURE RECOMMENDATIONS

THEME: FUTURE TRENDS AND RECOMMENDATIONS

FREQUENCY: 7/7

***Key Insights**:* Participants anticipate increased adoption of XAI and regulatory focus on ethical AI governance, emphasising the need for improved explainability and integration with existing workflows.

Quotes

1
"Enhanced Explainability: Incorporating better XAI features to ensure security-related AI decisions can be fully understood and trusted. Seamless Integration with Legacy Systems: AI tools should offer better support for legacy codebases. Ethical AI Governance: Tools should include features for ethical use, such as bias detection."

2-
"I am aware of benefits in the Cyber Security arena where AI allows rapid identification of trends and vulnerabilities, reducing workload for analysts and making decision making quicker."

3-
"The integration of AI and secure software engineering is expected to evolve significantly in the coming years, driven by advancements in AI technologies and the increasing complexity of cybersecurity challenges. Here are some future trends that can be anticipated:... AI will increasingly be used to predict and prevent security threats before they occur. By analysing patterns and behaviours in real-time, AI can anticipate potential vulnerabilities or attack vectors and suggest preemptive measures."… AI will become an integral part of DevSecOps, automating security checks at every stage of the software development lifecycle. This will include AI-driven static and dynamic code analysis, automated threat modelling, and continuous monitoring….AI will play a larger role in incident response, helping security teams detect, analyse, and respond to security incidents more quickly and accurately. AI-powered tools will be able to automate the identification of threats, prioritize incidents, and even initiate automated responses"… AI tools that assist in writing secure code will become more sophisticated, providing developers with real-time suggestions and corrections as they code. These tools will leverage machine learning models trained on vast datasets of secure and insecure code examples."

"To better support secure software engineering practices, AI tools can be improved or enhanced with the following features and capabilities:... Context-Aware Security Recommendations…Real-Time Secure Coding Assistance…Adaptive Learning from Feedback Loops…Integration with Threat Intelligence Feeds…Automated Threat Modelling and Risk Assessment"

4-
"AI tools could be improved by enhancing threat detection, integrating seamlessly with DevSecOps workflows, and providing better explainability of AI decisions. They should offer context-specific security recommendations, automated code fixes, and robust compliance features to support industry standards and regulations. Reducing bias by training on diverse datasets is also crucial."

5-
"Improved Code Creation, better software testing, AI Driven Personalisation, Bug Detection and Debugging, Quality assurance testing."…
"Data Privacy needs to be tightened. AI is being used increasingly in Software Development which brings into question how data is scanned

and used. Better guidelines for Ethics, establishing unambiguous guidelines for moral AI development and application is essential to ensuring that technology advances society rather than undermines it."

6-
"I foresee that as training data becomes flooded with AI-generated content, then the outputs will trend towards the mean, reducing and stifling innovation."

7-
When asked what future trends you foresee: "Creating the the entire project by verbally dictating to AI"

Google Questionnaire utilised before University Assigned Consent Form was Provided

24/10/2024, 13:52 Questionnaire on AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)
https://docs.google.com/forms/d/1hDPvOKmDhm_Kv7HnRFxTA1STC2F230BtimjBbQhCTes/edit 1/6
2.
Check all that apply.
I have read and understood the information provided and consent to participate in this study.
Demographics
3.
4.
Mark only one oval.
less than 1 year
1-3 years
4-6 years
7-10 years
10 + years
5.
You are invited to participate in a research study conducted by Holley Hudson as part of a MSc dissertation in Software Engineering. The purpose of this study is to investigate the integration of AI-enhanced tools, including GitHub Copilot and automated scanning technologies, within secure software engineering frameworks, with a focus on Explainable AI (XAI) techniques.
Participation and Confidentiality:
Your participation is voluntary, and you may withdraw at any time without penalty. The information you provide will be kept confidential and used solely for academic research purposes. Your responses will be anonymized, and no personal identifiers will be attached to the data.
Benefits and Risks:
There are no direct benefits or significant risks to you from participating in this study. Your insights will contribute to a better understanding of AI tools in secure software engineering.
By clicking "Agree," you acknowledge that you have read and understood the purpose of the study and consent to participate.
*
Can you tell me about your current job role and responsibilities?
*
How many years have you been working in software engineering, cybersecurity, or AI?
*
Which industry do you work in, and how does it impact your approach to software security?
*
24/10/2024, 13:52 Questionnaire on AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)
https://docs.google.com/forms/d/1hDPvOKmDhm_Kv7HnRFxTA1STC2F230BtimjBbQhCTes/edit 2/6
AI Tools Usage
6.
Mark only one oval.
Yes
No
7.
8.
Explainable AI (XAI)
9.
Are you familiar with AI tools such as GitHub Copilot and automated code scanning technologies?
*
Which AI tools do you use regularly in your workflow? Please provide details on how you use them.
*
How effective do you find these AI tools in improving secure coding practices? Please provide specific examples.
*
Are you aware of Explainable AI (XAI) techniques? How important is transparency and explainability in AI tools for secure software engineering?
Explanation: Explainable AI (XAI) refers to methods and techniques that make AI systems' decisions understandable to humans. Transparency and explainability are crucial for building trust in AI systems and ensuring that their decisions can be validated and understood by developers and stakeholders

*
24/10/2024, 13:52 Questionnaire on AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)
https://docs.google.com/forms/d/1hDPvOKmDhm_Kv7HnRFxTA1STC2F230BtimjBbQhCTes/edit 3/6
10.
Integration and Challenges
11.
12.
Mark only one oval.
Not concerned
1
2
3
4
5
Extremely concerned
13.
General Feedback
What benefits have you experienced or expect from using XAI techniques in AI-driven security solutions?
*
What challenges have you faced in integrating AI tools into existing CI/CD pipelines? How have you addressed these challenges?
Explanation: CI/CD (Continuous Integration/Continuous Deployment) pipelines are essential for automating the software development process. Integrating AI tools into these pipelines can present challenges such as compatibility issues and performance impacts.
*
How concerned are you about ethical implications (bias, transparency, accountability) of using AI tools in secure software engineering?
Explanation: Ethical considerations in AI include ensuring that AI systems are free from bias, transparent in their decision-making processes, and accountable for their actions. These factors are critical for maintaining fairness and trust in AI-driven solutions
*
What specific ethical issues have you encountered or do you foresee with the use of AI tools insecure software engineering?
*
24/10/2024, 13:52 Questionnaire on AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)
https://docs.google.com/forms/d/1hDPvOKmDhm_Kv7HnRFxTA1STC2F230BtimjBbQhCTes/edit 4/6
14.
Mark only one oval.
Not satisfied
1 2 3 4 5
Extremely satisfied
15.
16.
This content is neither created nor endorsed by Google.
How satisfied are you with the current AI tools available for secure software engineering? *
What future trends do you foresee in the integration of AI and secure software engineering? *
What improvements or features would you recommend for AI tools to better support secure software engineering practices?
*
Forms
24/10/2024, 13:52 Questionnaire on AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)
https://docs.google.com/forms/d/1hDPvOKmDhm_Kv7HnRFxTA1STC2F230BtimjBbQhCTes/edit 5/6
24/10/2024, 13:52 Questionnaire on AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)
https://docs.google.com/forms/d/1hDPvOKmDhm_Kv7HnRFxTA1STC2F230BtimjBbQhCTes/edit 6

Microsoft Questionnaire updated with University's Consent

MSc of Computer Science in Software Engineering
Consent Form for participation in a study on AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)
Section 1
Participant Information
Date: September 2024
Research Study Title: AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)

Introduction
My name is Holley Hudson and this research forms part of my Master's study at Glyndŵr University. You are being invited to take part in this research. Before you agree to do so, it is important that you understand the purpose and nature of the research and what your participation will involve, if you agree. Please read the following information carefully, and please ask if anything is not clear, or if you want more information. Contact details are given at the end of this information sheet.

What is the purpose of the study and how will it be carried out?
The research objectives are:
1. Evaluate the effectiveness of AI tools like GitHub Copilot and automated scanning technologies in secure coding practices.
2. To understand the role and benefits of XAI techniques in enhancing transparency and trust in AI-driven security solutions
3. To identify challenges in integrating AI tools within existing CI/CD pipelines
4. To explore ethical considerations related to AI in secure software engineering

The research methodology is to research the integration of AI tools in secure software engineering with a focus on XAI techniques. The aim is to explore how XAI can enhance transparency and trust in software development processes.

Why have I been invited to take part?
The is aim to recruit participants who are can provide valuable insights into the use of AI tools, particularly XAI in secure coding practices and software engineering. These could include software engineers, DevOps and CI/CD pipeline specialists, Cybersecurity professionals, AI/ML Engineers and Data scientists, of Software Architects.

Do I have to take part?
Participation is entirely voluntary. It is up to you to decide whether or not to take part. If you agree to take part, you will be asked to consent digitally via a form. If you agree to take part, you may still withdraw, without giving a reason. If this happens, please note that you will not be able to withdraw your data after it has been anonymised. Data is usually made anonymous quickly after data collection.

Section 2
Participant Information (continued)
What will taking part involve?
You will be asked to complete a pre-interview questionnaire that will take no more than 15 minutes to complete. A follow-up phone/video interview will be scheduled to gather additional thoughts and elaboration of the questions. This follow-up interview can be between 30 and 60 minutes long. Interviews will be carried out using Zoom links, where a recorded interview room will be set up for our use. This recording will provide me a transcript that I will analyse for key points to use in my research. You will be provided an ID to use to identify yourself once recording starts to protect your identity.
Will my participation be confidential?
All information about you collected during the study will be kept strictly confidential and stored securely in accordance with the Data Protection Act. However, there are certain exceptions to confidentiality. If, during the course of the interview, you disclose information about ongoing or past abuse, intentions of self-harm or harm to others, or serious breaches of policy or illegal activities, the researcher may be obligated to report this information to the appropriate authorities to ensure safety and

compliance with legal and ethical standards. The only people who will know about you are the researcher and, where necessary, the dissertation supervisor and examiners. All data, whether electronic, paper, or in any other form, will be destroyed when my Masters degree is awarded.
What will you do with the results of the research?
Data collected from the questionnaires and interviews will be analysed and will be incorporated into a Master's dissertation. No participant will be identifiable in the dissertation.

Section 3
Participant Information (continued)
What happens next?
Thank you for reading this information sheet. If you agree to take part,
by clicking on the consent acceptance below, please continue to the end of the form where questions will be asked about the topic.
Upon completion of the consent form and the questionnaire and/or interview, your privacy will be protected and data anonymised.

You can take a copy of this participant information or the consent form to keep by right clicking and selecting print.
If you wish to raise any concerns about any aspect of how you have been approached or treated in respect of this research study, please contact:
Frances Thomason: Head of Research Services (Frances.Thomason@glyndwr.ac.uk)

Contact for further information
If anything is not clear, or if you want more information, please contact me directly: S22009650@mail.glyndwr.ac.uk

Section 4
Participant consent
Please carefully read each statement:

I confirm that I have read the WGU Research Participant Privacy Notice: https://glyndwr.ac.uk/media/marketing/policies-and-documents/info-governanace/Research-Participant-Privacy-Notice.docx

I confirm that I have read and understood the participant information dated 10 September 2024 for the study. If I have asked for clarification or for more information, I have received satisfactory responses.

I understand that my participation is voluntary and that I am free to withdraw without giving any reason. I understand that any data I have contributed cannot be withdrawn after it has been anonymised, and that data collected from me will be anonymised within 5 days.

I understand that relevant sections of the data collected from me during the study may be looked at by the researcher as well as the dissertation supervisor and examiners where needed.

I consent to anonymous quotations being used in the dissertation, I consent to my anonymised data being retained for 1 year for use within future research and publications. I consent to the anonymised data I have contributed being made available in the public domain for use within future research by other researchers.

Your responses will be treated as confidential, and any data used in reports or publications will be anonymised. However, there are exceptions to confidentiality, such as disclosure of illegal activities, self-harm, abuse, or harm ot others. These may require the researcher to report this information to the appropriate authorities.

I agree to take part in the study.

1.
Consent:
Required to answer. Single choice.

I agree to the above statements and confirm that I wish to participate

I do not agree to participate
Section 5
Demographics

2.

How many years have you been working in software engineering, cybersecurity, or AI?
Single choice.

less than 1 year

1-3 years

4-6 years

7-10 years

10+ years

3.
Can you tell me about your current job role and responsibilities?
Required to answer. Multi Line Text.
Enter your answer

4.
Which industry do you work in, and how does it impact your approach to software security?
Required to answer. Multi Line Text.
Enter your answer

Section 6
AI Tools Usage

5.
Are you familiar with AI tools such as GitHub Copilot and automated code scanning technologies?
Required to answer. Single choice.

Yes

No

6.
Which AI tools do you use regularly in your workflow? Please provide details on how you use them.
Required to answer. Multi Line Text.
Enter your answer

7.
How effective do you find these AI tools in improving secure coding practices? Please provide specific examples.
Required to answer. Multi Line Text.
Enter your answer

Section 7
Explainable AI (XAI)
Methods and techniques that make AI systems' decisions understandable to humans

8.
Are you aware of Explainable AI (XAI) techniques? How important is transparency and explainability in AI tools for secure software engineering?
Required to answer. Multi Line Text.
Enter your answer

9.
What benefits have you experienced or expect from using XAI techniques in AI-driven security solutions?
Required to answer. Multi Line Text.
Enter your answer

Section 8
Integration and Challenges
Integration of AI tools into the CI/CD pipelines to automate the software development process

10.
What challenges have you faced in integrating AI tools into existing CI/CD pipelines? How have you addressed these challenges?
Required to answer. Multi Line Text.
Enter your answer

11.

How concerned are you about ethical implications (bias, transparency, accountability) of using AI tools in secure software engineering?
Required to answer. Single choice.

1 Not Concerned

2

3 Neither concerned nor unconcerned

4

5 Extremely Concerned

12.
What specific ethical issues have you encountered or do you foresee with the use of AI tools in secure software engineering?
Required to answer. Multi Line Text.
Enter your answer
Section 9
General Feedback

13.
How satisfied are you with the current AI tools available for secure software engineering?
Required to answer. Single choice.

1 Not satisfied

2

3 Neither satisfied or dissatisfied

4

5 Extremely Satisfied

14.
What future trends do you foresee in the integration of AI and secure software engineering?
Required to answer. Multi Line Text.
Enter your answer

15.
What improvements or features would you recommend for AI tools to better support secure software engineering practices?
Required to answer. Multi Line Text.
Enter your answer
Section 10
Add Email
If you are willing, please provide your email address below. This is optional, and your privacy and confidentiality will still be fully protected. Your email will be used solely to verify your consent for this study and to contact you if you would like to participate in a follow-up interview. Providing your email is not required, and you may still complete the survey without it.

16.
Please enter your email below:
Single line text.
Enter your answer

Consent Form Resent to Initial Google Participants

This was sent to all initial Google form respondents (post response) to ensure the consent aligned with the University's standards.

Consent Form for MSc of Computer Science in Software Engineering
Extended University Consent Form
Section 1
Participant Information
Date: September 2024
Research Study Title: AI-Enhanced Secure Software Engineering focusing on Explainable AI (XAI)

Introduction
My name is Holley Hudson and this research forms part of my Master's study at Glyndŵr University. You are being invited to take part in this research. Before you agree to do so, it is important that you understand the purpose and nature of the research and what your participation will involve, if you agree. Please read the following information carefully, and please ask if anything is not clear, or if you want more information. Contact details are given at the end of this information sheet.

What is the purpose of the study and how will it be carried out?
The research objectives are:
1. Evaluate the effectiveness of AI tools like GitHub Copilot and automated scanning technologies in secure coding practices.
2. To understand the role and benefits of XAI techniques in enhancing transparency and trust in AI-driven security solutions
3. To identify challenges in integrating AI tools within existing CI/CD pipelines
4. To explore ethical considerations related to AI in secure software engineering

The research methodology is to research the integration of AI tools in secure software engineering with a focus on XAI techniques. The aim is to explore how XAI can enhance transparency and trust in software development processes.

Why have I been invited to take part?
The is aim to recruit participants who are can provide valuable insights into the use of AI tools, particularly XAI in secure coding practices and software engineering. These could include software engineers, DevOps and CI/CD pipeline specialists, Cybersecurity professionals, AI/ML Engineers and Data scientists, of Software Architects.

Do I have to take part?
Participation is entirely voluntary. It is up to you to decide whether or not to take part. If you agree to take part, you will be asked to consent digitally via a form. If you agree to take part, you may still withdraw, without giving a reason. If this happens, please note that you will not be able to withdraw your data after it has been anonymised. Data is usually made anonymous quickly after data collection.

Section 2
Participant Information (continued)
What will taking part involve?
You will be asked to complete a pre-interview questionnaire that will take no more than 15 minutes to complete. A follow-up phone/video interview will be scheduled to gather additional thoughts and elaboration of the questions. This follow-up interview can be between 30 and 60 minutes long. Interviews will be carried out using Zoom links, where a recorded interview room will be set up for our use. This recording will provide me a transcript that I will analyse for key points to use in my research. You will be provided an ID to use to identify yourself once recording starts to protect your identity.
Will my participation be confidential?
All information about you collected during the study will be kept strictly confidential and stored securely in accordance with the Data Protection Act. However, there are certain exceptions to confidentiality. If, during the course of the interview, you disclose information about ongoing or past abuse, intentions of self-harm or harm to others, or serious breaches of policy or illegal activities, the researcher may be obligated to report this information to the appropriate authorities to ensure safety and compliance with legal and ethical standards. The only people who will know about you are the researcher and, where

necessary, the dissertation supervisor and examiners. All data, whether electronic, paper, or in any other form, will be destroyed when my Masters degree is awarded.
What will you do with the results of the research?
Data collected from the questionnaires and interviews will be analysed and will be incorporated into a Master's dissertation. No participant will be identifiable in the dissertation.

Section 3
Participant Information (continued)
What happens next?
Thank you for reading this information sheet. This consent form has been provided to you to replace a current consent you have in place. This consent aligns with the university guidelines and has been requested to fulfill university ethics requirements.

You can take a copy of this participant information or the consent form to keep by right clicking and selecting print.
If you wish to raise any concerns about any aspect of how you have been approached or treated in respect of this research study, please contact:
Frances Thomason: Head of Research Services (Frances.Thomason@glyndwr.ac.uk)

Contact for further information
If anything is not clear, or if you want more information, please contact me directly: S22009650@mail.glyndwr.ac.uk

Section 4
Participant consent
Please carefully read each statement:

I confirm that I have read the WGU Research Participant Privacy Notice: https://glyndwr.ac.uk/media/marketing/policies-and-documents/info-governanace/Research-Participant-Privacy-Notice.docx

I confirm that I have read and understood the participant information dated 10 September 2024 for the study. If I have asked for clarification or for more information, I have received satisfactory responses.

I understand that my participation is voluntary and that I am free to withdraw without giving any reason. I understand that any data I have contributed cannot be withdrawn after it has been anonymised, and that data collected from me will be anonymised within 5 days.

I understand that relevant sections of the data collected from me during the study may be looked at by the researcher as well as the dissertation supervisor and examiners where needed.

I consent to anonymous quotations being used in the dissertation, I consent to my anonymised data being retained for 1 year for use within future research and publications. I consent to the anonymised data I have contributed being made available in the public domain for use within future research by other researchers.

Your responses will be treated as confidential, and any data used in reports or publications will be anonymised. However, there are exceptions to confidentiality, such as disclosure of illegal activities, self-harm, abuse, or harm ot others. These may require the researcher to report this information to the appropriate authorities.

I agree to take part in the study.

1.
Consent
Required to answer. Single choice.

I agree to the above statements and confirm that I wish to participate

I do not agree to participate
Section 5
Email Request
Email request for consent verification and contact

2.

If you are willing, please provide your email address below. This is optional, and your privacy and confidentiality will be fully protected. Your email will be used solely to verify your consent for this study and to contact you if you would like to participate in a follow-up interview. Providing your email is not required, and you may still complete the survey without it
Single line text.
Enter your answer

Add new question