Impact of Artificial Intelligence on Open-Source Software Development

Vinod Kumar Ahuja
Dendritic: A Human-Centered AI and Data Science Institute,
Dept. of Computing & Software Engineering
Florida Gulf Coast University
Fort Myers, FL, USA

Jeremy Clark
Dept. of Computing & Software Engineering
Florida Gulf Coast University
Fort Myers, FL, USA

Joshua Wurtenberg
Dept. of Computing & Software Engineering
Florida Gulf Coast University
Fort Myers, FL, USA

Abstract

Open-source software has long been a catalyst for innovation. However, the rise of Artificial Intelligence (AI) is transforming the open source landscape, reshaping code development, project management, and community contributions among other aspects of open-source projects. This research presents a systematic literature review to examine and categorize how AI impacts open-source development. Our findings suggest that AI drives the creation and innovation of open-source projects, supercharges development and project management practices within the open source community, and augments the quality and security of projects while raising important ethical considerations. Our study further identifies trends associated with AI, particularly generative AI, and assesses how AI tools may both challenge and enhance processes such as code review and repository management. These insights contribute to a deeper understanding of AI's transformative role in the open-source ecosystem.

Introduction and Related Work

Open-source software development is a collaborative effort of groups of volunteers to create software that is freely available to everyone. The rise of Artificial Intelligence (AI) is reshaping the open source software (OSS) landscape and impacting various aspects, ranging from how OSS is developed, automated code generation, review processes, project governance, contributor dynamics, and to the creation of new open source projects [6, 9]. Various AI-powered tools, such as code generators, bug detectors, and AI-driven project management tools, are reducing the manual contributions by volunteers, streamlining the development workflows, and supporting the collaborations [37]. These advancements have the potential to accelerate innovation, lower the barriers to entry for new contributors, and improve software quality [58]. However, they also introduce complexities regarding software transparency, reliability, and the role of human oversight in AI-assisted development[17].

The integration of AI into OSS projects has sparked debates on its long-term impact on contributor dynamics and governance structures [43]. The extent to which AI-driven tools influence project sustainability and community engagement remains an open question. Understanding these evolving dynamics is crucial for assessing how AI transforms code contributions and the social and organizational aspects of OSS development. This study addresses the research question: How does AI impact open source software development?

Several studies have explored the use of AI in specific aspects of software development. Bug detection, code generation, and testing have all been discussed in the literature. Similarly, open-source development has frequently been examined [44]. There has been some research conducted at the intersection of AI and OSS, suggesting that AI has an impact on the diversity of developers working on open-source AI projects [3], as well as some research acknowledging the impact of open-source on AI. While this research provides a foundation for further investigation, the overlap of these two fields remains insufficiently explored. Currently, there is no comprehensive review that fully captures how AI is transforming open-source software development practices and communities. Our paper contributes to this gap by providing a thorough analysis of AI's broader impacts on open-source ecosystems, offering insights that previous studies have not adequately addressed.

This research builds on those foundations and contributes to the expanding body of research at the intersection of AI and OSS through a systematic literature review, providing a comprehensive overview of the current state of research in this area. Our investigation addresses the notable gap in understanding AI's full impact on open-source communities, development practices, and governance models. Our findings are valuable for open source developers, contributors, researchers, and policymakers to understand AI's transformative nature and its relationship with OSS, ultimately helping stakeholders navigate the rapidly evolving landscape where these technologies converge.

The following section outlines the method used for this systematic literature review and details the research process. The findings from the analysis are then summarized, followed by a discussion that contextualizes the results and in-

Copyright © 2025 by the authors.
This open access article is published under the Creative Commons Attribution-NonCommercial 4.0 International License.

tegrates them into a broader research agenda.

Method

This study was conducted through a systematic literature review following the guidelines of [35].

Data collection

A comprehensive search was conducted across eight prominent databases: ProQuest, Ais eLibrary, Web of Science, ACM Digital Library, Science Direct, Scopus, EBSCO Host, and IEEE Electronic Library. The keywords "Artificial Intelligence," "AI," "Open source," and "OS" were searched within the title and abstract to capture a wide range of potentially relevant articles. This initial search returned 4,254 results.

Filtration

The search results were filtered to include only English-language peer-reviewed journal and conference papers, narrowing the dataset to 3,001 articles. Two authors then conducted a manual review of titles and abstracts. To establish a baseline for relevance, both researchers independently reviewed the same first 100 articles, assessing whether they addressed the research question. This review resulted in a 40 percent agreement. The researchers then jointly re-evaluated each paper, agreeing on 95 percent. A third author acted as a moderator, resolving the remaining disagreements. Following refining inclusion criteria, a second set of 100 articles was reviewed to confirm inter-rater reliability. This review yielded an initial agreement of 87 percent, which increased to 100 percent after discussing the 13 conflicting cases. With confidence in the review process, the two researchers divided the remaining 2,801 articles equally—1,400 and 1,401 each—and independently assessed titles and abstracts. This process resulted in a final selection of 84 articles deemed potentially relevant. Refer to Figure 1 for data collection and filtration.

Coding & Analysis

The final set of 84 articles was coded following the coding guidelines of [41], to find out how each article discusses AI's impact on open source software development. Articles were categorized by the type of impact discussed, with examples such as 'Creates a new project' or 'Code Quality'. A new code was created if an article did not fit an existing category. This process was highly iterative, with new codes emerging as articles were reviewed and re-reviewed. Some articles were recategorized, and others, upon further reading, were deemed irrelevant to the research question as their narrative did not elaborate on how AI impacts OSS. As a result, 24 additional articles were discarded, leaving a final body of 60 articles. These articles were systematically evaluated to identify key themes and patterns regarding AI's impact on OSS development. Our analysis examined the types of AI applications deployed in OSS environments, the specific domains of OSS development influenced by AI technologies, and the broader trends and challenges emerging from AI-OSS integration. This analytical process

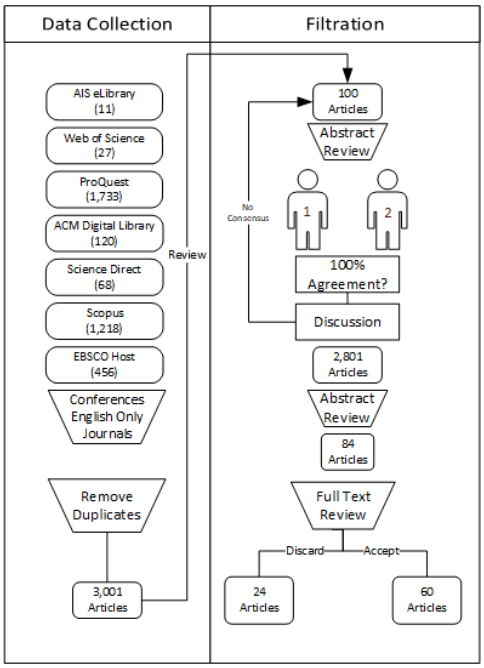

Figure 1: Data Collection and Filtration

resulted in nine distinct codes, which we subsequently organized into 83 distinct instances of coding that span three overarching themes. Table 1 presents these themes and codes along with their corresponding articles, providing a structured overview of our findings. This systematic methodology ensures the reliability and validity of the findings. The detailed description of the research process allows for replication and contributes to the transparency and reproducibility of the study.

Findings

The findings revealed three areas where AI is impacting open source: project creation and innovation, development efficiency and project management, and quality, security, and ethics.

Expanding Project Creation and Innovation

AI is reshaping open-source software by driving the creation of entirely new project categories and transforming development approaches. The surge in demand for open source alternatives to proprietary AI solutions has generated a wave of community-driven AI projects, leading to the launch of several open source projects focused on developing AI tools and models [17], [62]. The ubiquitous inclusion of AI has made the lack of software knowledge andskills more apparent, making it necessary for communities to address industry pain ppoints, such as filling skill gaps [6, 23], removing reliability concerns [33], and even fulfilling cloud computing needs and creating new operating systems to support AI [9].

Coding & Analysis	
Code	Articles
Expanding Project Creation and Innovation	
New Projects	[2, 4, 6, 7, 9, 12, 13, 16, 17, 20, 22, 23, 25, 30, 31, 32, 38, 40, 48, 51, 54, 59]
Innovation & Creativity	[3, 13, 21, 50]
Supercharging Development, Efficiency, & Management	
Repository Management	[3, 13, 15, 24, 27, 44, 47, 49, 53, 26, 46, 58]
Speed of Writing Code	[5, 17, 19, 34, 37, 39, 44, 45, 52, 55, 56]
Code Generation	[5, 8, 30, 31, 33, 22, 62]
Time Management	[1, 18]
Cost Reduction	[60]
Augmenting Quality, Security, & Ethical Considerations	
Code Quality	[1, 14, 16, 21, 28, 29, 30, 57, 61, 36, 42, 54, 58]
Ethical & Legal Considerations	[3, 10, 11, 20, 43, 48]
Security	[15, 16, 48, 49, 50]

Table 1: Itemized Coding of Articles

The democratizing effect of AI is particularly powerful, as new open-source tools enable non-engineers to implement sophisticated software solutions [62]. This accessibility revolution is expanding open-source participation beyond traditional developer communities, bringing diverse perspectives and use cases previously excluded from the ecosystem. As a result, the open-source landscape is witnessing a surge in interdisciplinary collaborations, where domain experts from various fields contribute innovative applications and drive new OSS advancements.

Supercharging Development Efficiency and Project Management

AI tools are dramatically accelerating open-source development cycles through automation that was previously unimaginable. Smaller projects now compete with well-funded initiatives by deploying AI to generate code, detect AI-generated contributions [58], and intelligently assign bugs to the most qualified developers [1].

Resource-intensive capabilities that once required substantial investment are now accessible through AI-powered open-source solutions such as "Griddly" for game development [5], and these smaller developers can capitalize on generative tools to create high-quality digital assets that would otherwise be unobtainable for independent projects[60]. Project management has been transformed through automated repository maintenance [58], workflow optimization [46], and freeing developers from time-consuming routine tasks. The 80% of project time traditionally spent on data engineering [37] is now drastically reduced through AI assistance, allowing open-source teams to focus on innovation rather than infrastructure – a shift that's fundamentally changing development economics and team composition.

Augmenting Quality, Security, and Ethics

Our analysis reveals that quality assurance and security represent significant domains where AI is transforming open-source software development. These interconnected areas emerge as among the most prevalent applications of AI in OSS projects. AI technologies are democratizing sophisticated testing and security capabilities that were traditionally exclusive to enterprise-level projects. Machine learning models now automate critical quality assurance processes—including testing, bug detection, classification, and assignment—effectively addressing the shortage of dedicated testers common in many OSS initiatives [1, 28, 14]. While universal solutions remain elusive, cross-project bug detection models enable smaller OSS projects with limited bug history datasets to leverage collective data across multiple repositories, substantially enhancing detection, classification, and resolution capabilities [14].

The integration of AI extends beyond quality enhancement to encompass dedicated security tools that protect OSS ecosystems. AI-powered source code authorship attribution methods can identify authors with remarkable accuracy by analyzing code patterns [15]. These applications strengthen security frameworks by attributing otherwise anonymous malicious code and resolving authorship disputes. However, they simultaneously introduce privacy concerns for developers, illustrating the ethical complexities inherent in AI adoption within OSS. Emerging defensive practices further amplify these ethical considerations; some research documents how developers intentionally "poison" their codebases with deliberately maligned code designed to disrupt AI models' scraping capabilities [48].

Our research also identifies AI's impact on OSS project diversity as an emerging pattern of concern. Evidence suggests that AI may inadvertently limit developer inclusion, potentially due to the corporate-driven nature of many AI models [3]. While certain companies have open-sourced their AI and Machine Learning models, these tools often remain oriented toward proprietary business objectives. This orientation cultivates a specialized contributor ecosystem comprised primarily of developers familiar with specific AI frameworks and subject matter experts in the company's business domain. Consequently, participation appears

constrained to developers either employed by or closely affiliated with the organizations driving model development, potentially narrowing the traditionally diverse contributor base of open-source communities.

As AI continues to shape open-source development, its influence on project creation, management, efficiency, security, and ethics will remain pivotal. While AI-driven automation fosters innovation and sustainability, ensuring ethical AI use and transparency will be critical in maintaining trust and inclusivity within OSS communities.

Disussion & Limitations

This systematic literature review explores the transformative impacts of AI on OSS development. As AI continues to evolve, it is not only driving the creation of new open-source projects but also reshaping existing practices within the OSS community. A central takeaway from this study is the surge in new open-source projects, which address both the challenges and opportunities created by AI. These projects span a variety of domains, from educational tools and training products to new operating systems and libraries supporting AI development. This pattern illustrates the growing significance of AI in software development and the OSS community's adaptation to these changes, reflecting AI's increasing influence in shaping the future of OSS [17, 62].

AI is also redefining project management practices within OSS communities. AI-powered tools automate repetitive tasks like bug detection, code review, and repository management [26, 61], enabling developers to focus on more complex and creative aspects of the development process. This shift enhances development efficiency, reduces project costs, and improves code quality. However, AI's widespread adoption introduces ethical concerns, including issues of bias, privacy, and potential misuse [37, 17]. These concerns underscore the importance of addressing responsible and ethical development and use practices within the OSS community. As an open forum for collaboration, the open-source ecosystem provides a unique platform for ongoing discussions on these critical issues.

Despite the insights provided by this review, the study has limitations. The inclusion of only English-language publications may have excluded valuable research in other languages, and the exclusive focus on peer-reviewed journal and conference papers may have left out significant research published in other formats. Our systematic methodology incorporated measures to ensure consistency and minimize bias in the inclusion criteria, with the qualitative assessment process allowing for nuanced evaluation of the literature while maintaining research integrity. Moreover, the rapid pace of AI development means that new trends and patterns are emerging constantly, highlighting the need for ongoing research to keep pace with these advancements.

Conclusion & Future Works

Our research highlights the significance and complexity of the impacts of AI on OSS development. AI is driving new open source projects and changing the existing practices within the open source project management landscape. AI-powered tools create opportunities to improve efficiency, enhance code quality, and manage open source projects more effectively. The significance of ongoing research in understanding the dynamic relationship between AI and OSS is also highlighted. These insights can assist OSS developers, community managers, researchers, and policymakers in molding the future of open source software development in the era of AI.

Future research in this domain should address several promising avenues. First, longitudinal studies examining the evolution of AI-OSS integration over time would provide valuable insights into emerging trends and long-term impacts. Second, empirical investigations into how AI affects contributor diversity, inclusion, and governance structures within OSS communities merit further attention. Third, researchers should explore the ethical implications of AI deployment in OSS, particularly regarding bias, transparency, and fairness. Additionally, comparative analyses between AI-powered and traditional OSS development methodologies could yield practical guidelines for optimizing project outcomes. Finally, investigating the reciprocal relationship between OSS and AI—specifically how open source principles influence AI development practices—would contribute to a more holistic understanding of this dynamic technological ecosystem. These research directions will be crucial as both AI and OSS continue to evolve and reshape software development landscapes.

Acknowledgments.

This work is supported by Florida Gulf Coast University's Dendritic: A Human-Centered AI and Data Science Institute, U.A. Whitaker College of Engineering's Head Start program, and the Office of Scholarly Innovation and Student Research's WiSER Research Assistant Program.

References

[1] A. K. Dipongkor et al. "On Fusing Artificial and Convolutional Neural Network Features for Automatic Bug Assignments". In: *IEEE Access* 11 (2023), pp. 49493–49508. ISSN: 2169-3536.

[2] Á. López García et al. "A Cloud-Based Framework for Machine Learning Workloads and Applications". In: *IEEE Access* 8 (2020), pp. 18681–18692. ISSN: 2169-3536.

[3] Bram Adams and Foutse Khomh. "The Diversity Crisis of Software Engineering for Artificial Intelligence". English. In: *IEEE Software* 37.5 (2020). Number: 5 Num Pages: 104-108 Place: Los Alamitos, United States Publisher: IEEE Computer Society, pp. 104–108. ISSN: 07407459.

[4] N. Alshammari et al. "OpenSHS: Open smart home simulator". In: *Sensors (Switzerland)* 17.5 (2017).

[5] C. Bamford. “Griddly: A platform for AI research in games[Formula presented]”. In: *Software Impacts* 8 (2021).

[6] Natalia Bell. “Using Single-Board Miniature Computers (SBCs) to Create Basic Artificial Intelligence (AI) and Machine Learning (ML) Projects”. In: *J. Comput. Sci. Coll.* 38.3 (Nov. 2022). Number: 3, p. 183. ISSN: 1937-4771.

[7] M. Bosello et al. “Race Against the Machine: A Fully-Annotated, Open-Design Dataset of Autonomous and Piloted High-Speed Flight”. In: *IEEE Robotics and Automation Letters* 9.4 (2024), pp. 3799–3806.

[8] J.H. Caufield et al. “Structured Prompt Interrogation and Recursive Extraction of Semantics (SPIRES): a method for populating knowledge bases using zero-shot learning”. In: *Bioinformatics* 40.3 (2024).

[9] Ioannis P. Chochliouros et al. “NEMO: Building the Next Generation Meta Operating System”. In: *Proceedings of the 3rd Eclipse Security, AI, Architecture and Modelling Conference on Cloud to Edge Continuum*. ESAAM ’23. New York, NY, USA: Association for Computing Machinery, Oct. 2023, pp. 1–9. ISBN: 9798400708350.

[10] Martin Cooper. “Open Source is Good for AI But, Is AI Good for Open Source?” In: *ITNOW* 65.2 (2023), pp. 50–51. ISSN: 17465702.

[11] Jonathan Cui and David A. Araujo. “Rethinking use-restricted open-source licenses for regulating abuse of generative models”. English. In: *Big Data & Society* 11.1 (Mar. 2024). Number: 1 Place: London, United Kingdom Publisher: Sage Publications Ltd.

[12] M. Daole et al. “OpenFL-XAI: Federated learning of explainable artificial intelligence models in Python”. In: *SoftwareX* 23 (2023).

[13] Richard Dreyling III, Tanel Tammet, and Ingrid Pappel. “Technology Push in AI-Enabled Services: How to Master Technology Integration in Case of Bürokratt”. English. In: *SN Computer Science* 5.6 (Aug. 2024). Number: 6 Num Pages: 738 Place: Kolkata, Netherlands Publisher: Springer Nature B.V., p. 738. ISSN: 2662995X.

[14] E. Halit Yilmaz, İ. Hakki Toroslu, and Ö. Köksal. “A Comparative Study of Contemporary Learning Paradigms in Bug Report Priority Detection”. In: *IEEE Access* 12 (2024), pp. 126577–126586. ISSN: 2169-3536.

[15] F. Ullah et al. “Source Code Authorship Attribution Using Hybrid Approach of Program Dependence Graph and Deep Learning Model”. In: *IEEE Access* 7 (2019), pp. 141987–141999. ISSN: 2169-3536.

[16] Bogdan Ghimis, Miruna Paduraru, and Alin Stefanescu. “RIVER 2.0: an open-source testing framework using AI techniques”. In: *Proceedings of the 1st ACM SIGSOFT International Workshop on Languages and Tools for Next-Generation Testing*. LANGETI 2020. New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 13–18. ISBN: 978-1-4503-8123-9.

[17] R. Ghioni, M. Taddeo, and L. Floridi. “Open source intelligence and AI: a systematic review of the GELSI literature”. In: *AI and Society* 39.4 (2024), pp. 1827–1842.

[18] A. Gibbons et al. “NEAL: an open-source tool for audio annotation”. In: *PeerJ* 11 (2023).

[19] R.C. Godwin and R.L. Melvin. “Toward efficient data science: A comprehensive MLOps template for collaborative code development and automation”. In: *SoftwareX* 26 (2024).

[20] Thomas F. Gordon. “Analyzing open source license compatibility issues with Carneades”. In: *Proceedings of the 13th International Conference on Artificial Intelligence and Law*. ICAIL ’11. New York, NY, USA: Association for Computing Machinery, June 2011, pp. 51–55. ISBN: 978-1-4503-0755-0.

[21] Takaaki Goto, Kensei Tsuchida, and Tetsuro Nishino. “Episode: An Extreme Programming Method for Innovative Software Based on Systems Design and Its Practical Study”. English. In: *International Journal of Software Engineering & Applications* 5.5 (2014). Number: 5 Num Pages: 1 Publisher: Academy & Industry Research Collaboration Center (AIRCC), p. 1. ISSN: 0976-2221.

[22] H. Zhang et al. “Arsenal: Understanding Learning-Based Wireless Video Transport via In-Depth Evaluation”. In: *IEEE Transactions on Vehicular Technology* 70.10 (Oct. 2021), pp. 10832–10844. ISSN: 1939-9359.

[23] Isaac Han, Dong-Hyeok Park, and Kyung-Joong Kim. “A New Open-Source Off-Road Environment for Benchmark Generalization of Autonomous Driving”. In: *IEEE Access* 9 (2021). Conference Name: IEEE Access, pp. 136071–136082. ISSN: 2169-3536.

[24] C. Hu et al. “Games for Artificial Intelligence Research: A Review and Perspectives”. In: *IEEE Transactions on Artificial Intelligence* (2024), pp. 1–20.

[25] S. Huang and J.M. Cole. “ChemDataWriter: a transformer-based toolkit for auto-generating books that summarise research”. In: *Digital Discovery* 2.6 (2023), pp. 1710–1720.

[26] He Jiang et al. “DupHunter: Detecting Duplicate Pull Requests in Fork-Based Development”. In: *IEEE Transactions on Software Engineering* 49.4 (Apr. 2023). Conference Name: IEEE Transactions on Software Engineering, pp. 2920–2940. ISSN: 1939-3520.

[27] A. Kale et al. “Geoweaver_cwl: Transforming geoweaver AI workflows to common workflow language to extend interoperability”. In: *Applied Computing and Geosciences* 19 (2023).

[28] A. Kelin et al. “Automatic Bug Classification System to Improve the Software Organization Product Performance”. English. In: *International Journal of Sociotechnology and Knowledge Development* 14.1 (2022). Number: 1 Num Pages: 1-17 Place: Hershey,

United States Publisher: IGI Global, pp. 1–17. ISSN: 1941-6253.

[29] Tuan Dung Lai et al. “Comparative analysis of real issues in open-source machine learning projects”. English. In: *Empirical Software Engineering* 29.3 (May 2024). Number: 3 Num Pages: 60 Place: Dordrecht, Netherlands Publisher: Springer Nature B.V., p. 60. ISSN: 13823256.

[30] Hugo LÓPEZ-FERNÁNDEZ et al. “The Artificial Intelligence Workbench: a retrospective review”. English. In: *ADCAIJ : Advances in Distributed Computing and Artificial Intelligence Journal* 5.1 (2016). Number: 1 Num Pages: 73-85 Place: Salamanca, Spain Publisher: Ediciones Universidad de Salamanca Section: Articles, pp. 73–85.

[31] Armin Moin et al. “ML-quadrat & DriotData: a model-driven engineering tool and a low-code platform for smart IoT services”. In: *Proceedings of the ACM/IEEE 44th International Conference on Software Engineering: Companion Proceedings*. ICSE ’22. New York, NY, USA: Association for Computing Machinery, Oct. 2022, pp. 144–148. ISBN: 978-1-4503-9223-5.

[32] M. Morafah, W. Wang, and B. Lin. “A Practical Recipe for Federated Learning under Statistical Heterogeneity Experimental Design”. In: *IEEE Transactions on Artificial Intelligence* 5.4 (2024), pp. 1708–1717.

[33] A. Nazir et al. “LangTest: A comprehensive evaluation library for custom LLM and NLP models[Formula presented]”. In: *Software Impacts* 19 (2024).

[34] P.-J. Norbäck and L. Persson. “Why generative AI can make creative destruction more creative but less destructive”. In: *Small Business Economics* 63.1 (2024), pp. 349–377.

[35] C. Okoli. “A Guide to Conducting a Standalone Systematic Literature Review”. In: *Communications of the Association for Information Systems* 37 (2015).

[36] L.P. Osco et al. “The Segment Anything Model (SAM) for remote sensing applications: From zero to one shot”. In: *International Journal of Applied Earth Observation and Geoinformation* 124 (2023).

[37] T. Petricek et al. “AI Assistants: A Framework for Semi-Automated Data Wrangling”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.9 (2023), pp. 9295–9306.

[38] Sébastien Picault et al. “Leveraging artificial intelligence and software engineering methods in epidemiology for the co-creation of decision-support tools based on mechanistic models”. English. In: *Preventive veterinary medicine* 228 (2024). Num Pages: 106233, p. 106233.

[39] R. Pierdicca et al. “DeepReality: An open source framework to develop AI-based augmented reality applications”. In: *Expert Systems with Applications* 249 (2024).

[40] F. Sakr et al. “Machine learning on mainstream microcontrollers†”. In: *Sensors (Switzerland)* 20.9 (2020).

[41] Joey Saldaña. *The Coding Manual for Qualitative Researchers*. 3E. SAGE Publications, 2013.

[42] S. Salehi. “An efficient intrusive deep reinforcement learning framework for OpenFOAM”. In: *Meccanica* (2024).

[43] C. D. Schmit, M. J. Doerr, and J. K. Wagner. “Leveraging IP for AI governance”. English. In: *Science* 379.6633 (Feb. 2023). Number: 6633 Num Pages: 646-648 Place: Washington, United States Publisher: The American Association for the Advancement of Science, pp. 646–648. ISSN: 00368075.

[44] R.R. Schreiber. “Organizational Influencers in Open-Source Software Projects”. In: *International Journal of Open Source Software and Processes* 14.1 (2023).

[45] N.L. Sørensen et al. “Machine learning in general practice: scoping review of administrative task support and automation”. In: *BMC Primary Care* 24.1 (2023).

[46] D. Spinellis. “Commands as AI Conversations”. In: *IEEE Software* 40.6 (2023), pp. 22–26.

[47] J. Stewart, N. Lyubashenko, and G. Stefanek. “The efficacy of detecting AI-generated fake news using transfer learning”. In: *Issues in Information Systems* 24.2 (2023), pp. 164–177.

[48] Zhensu Sun et al. “CoProtector: Protect Open-Source Code against Unauthorized Training Usage with Data Poisoning”. In: *Proceedings of the ACM Web Conference 2022*. WWW ’22. New York, NY, USA: Association for Computing Machinery, Apr. 2022, pp. 652–660. ISBN: 978-1-4503-9096-5.

[49] Z. Szabó and V. Bilicki. “A New Approach to Web Application Security: Utilizing GPT Language Models for Source Code Inspection”. In: *Future Internet* 15.10 (2023).

[50] Saravanan Thirumuruganathan et al. “Automated Annotations for AI Data and Model Transparency”. In: *J. Data and Information Quality* 14.1 (Dec. 2021). Number: 1, 2:1–2:9. ISSN: 1936-1955.

[51] Brandon Thurgood and Ruth G. Lennon. “Cloud Computing With Kubernetes Cluster Elastic Scaling”. In: *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*. ICFNDS ’19. New York, NY, USA: Association for Computing Machinery, July 2019, pp. 1–7. ISBN: 978-1-4503-7163-6.

[52] D. Tkaczyk et al. “CERMINE: Automatic extraction of structured metadata from scientific literature”. In: *International Journal on Document Analysis and Recognition* 18.4 (2015), pp. 317–335.

[53] Jason Tsay et al. “Extracting enhanced artificial intelligence model metadata from software repositories”. English. In: *Empirical Software Engineering* 27.7 (Dec. 2022). Number: 7 Place: Dordrecht,

Netherlands Publisher: Springer Nature B.V. ISSN: 13823256.

[54] W. Meng et al. “LogSummary: Unstructured Log Summarization for Software Systems”. In: *IEEE Transactions on Network and Service Management* 20.3 (Sept. 2023), pp. 3803–3815. ISSN: 1932-4537.

[55] X. Wang et al. “FANN-on-MCU: An Open-Source Toolkit for Energy-Efficient Neural Network Inference at the Edge of the Internet of Things”. In: *IEEE Internet of Things Journal* 7.5 (2020), pp. 4403–4417.

[56] W. Westera et al. “Artificial intelligence moving serious gaming: Presenting reusable game AI components”. In: *Education and Information Technologies* 25.1 (2020), pp. 351–380.

[57] Y. Li et al. “Evaluating Data Filter on Cross-Project Defect Prediction: Comparison and Improvements”. In: *IEEE Access* 5 (2017), pp. 25646–25656. ISSN: 2169-3536.

[58] Chunyu Yang, Yan Liu, and Jia Yu. “Exploring Violations of Programming Styles: Insights from Open Source Projects”. In: *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*. CSAI ’18. New York, NY, USA: Association for Computing Machinery, Dec. 2018, pp. 185–189. ISBN: 978-1-4503-6606-9.

[59] Y. Yang et al. “SNIPPET: A Framework for Subjective Evaluation of Visual Explanations Applied to DeepFake Detection”. In: *ACM Transactions on Multimedia Computing, Communications and Applications* 20.8 (2024).

[60] Z. Xiang and Y. Guo. “Controlling Melody Structures in Automatic Game Soundtrack Compositions With Adversarial Learning Guided Gaussian Mixture Models”. In: *IEEE Transactions on Games* 13.2 (June 2021), pp. 193–204. ISSN: 2475-1510.

[61] Zakarea Alshara et al. “ML-Augmented Automation for Recovering Links Between Pull-Requests and Issues on GitHub”. In: *IEEE Access* 11 (2023), pp. 5596–5608. ISSN: 2169-3536.

[62] B. Zhang et al. “Demystifying Practices, Challenges and Expected Features of Using GitHub Copilot”. In: *International Journal of Software Engineering and Knowledge Engineering* 33.11-12 (2023), pp. 1653–1672.