Volume 12, No.5, May 2025

Journal of Global Research in Mathematical Archives

ISSN 2320 – 5822

RESEARCH PAPER

Available online at http://www.jgrma.com

SCENARIO-BASED APPROACHES TO EXPLAINABLE AI CODE GENERATION: BRIDGING TRANSPARENCY AND USABILITY

Dr. B. K. Sharma[1]

[1] Professor, Department of Computer Science and Application, Mandsaur University, Mandsaur

[1]email: dr.balkrishnasharma@meu.edu.in

Abstract: As artificial intelligence (AI) becomes more integrated into critical applications, transparency and explainability remain key concerns, particularly for generative models. Despite their transformative capabilities, these models often act as "black boxes," confusing decision-making processes from users and stakeholders. This paper explores how scenario-based design can be leveraged to increase AI code transparency, providing a structured approach to making generative models more explainable and accountable. This research aims to bridge the gap between complex AI systems and human understanding by analyzing the existing transparency challenges and reviewing state-of-the-art interpretability methods in the proposed scenario-driven strategies.

Experiential case studies demonstrate the effectiveness of scenario-based interventions in improving model interpretability, fostering trust, and ensuring ethical AI deployment. Additionally, this study evaluates the effects of scenario-based design on model debugging, bias detection, and user accessibility, providing insights into how transparency initiatives can mitigate algorithmic risks.

The conclusion highlights the need for interdisciplinary collaboration among AI researchers, designers, and policymakers to develop a strong framework for transparent generative AI. By integrating structured explanatory techniques, this paper contributes to ongoing efforts in responsible AI development and provides a roadmap for future developments in this field.

Keywords: AI Transparency, Generative Models, Scenario-Based Design, Responsible AI, Ethical AI, Algorithmic Bias

1 INTRODUCTION

Artificial Intelligence (AI) has revolutionized various industries from healthcare and finance to entertainment and education. Amongst the advancements in AI, generative models have emerged as powerful tools that are capable of creating highly realistic images, text, and audio. However, despite their innovative potential, generative models often operate as "black boxes," making their decision-making processes opaque and difficult to interpret [1]. This lack of transparency raises fears about trust, accountability, and ethical AI deployment [2][3]. Transparency in AI codes is critical for ensuring fairness, reducing biases, and enabling users to understand how AI-generated outputs are produced. Without clear explanatory mechanisms, stakeholders, including developers, policymakers, and end-users, struggle to assess the reliability and safety of AI-generated content [4][5].

To resolve this issue, an interdisciplinary approach goes beyond traditional algorithm debugging and explores user-cantered design strategies. For this, a Scenario-based design offers a promising framework to enhance AI explainability. This method provides clear interpretations of AI behavior by structuring interactions and explanations around real-world scenarios. Scenario-driven transparency can improve user trust and regulatory compliance by confirming that AI systems align with ethical standards. This research paper highlights how scenario-based design can bridge the gap between complex generative models and user understanding, thereby providing a structured approach to increase AI code transparency.

This research investigates the effectiveness of scenario-based transparency in generative AI systems through a comprehensive literature review, empirical case studies, and practical analysis [6]. The findings aim to contribute to the wider discussion on responsible AI development and provide actionable insights for researchers, developers, and policymakers who want to design more interpretable and accountable AI models.

AI transparency refers to the ability of AI systems to be understandable to developers, users, and stakeholders. Transparent AI ensures that models operate understandably, allowing stakeholders to assess potential biases, errors, and ethical implications. Users may always be in a dilemma and try to verify whether decisions generated by AI without sufficient transparency are reliable with fair, reasonable, or ethical standards or not.

The need for interpretability is emerging for responsible AI development. Users to understand the factors influencing AI model behavior, decision-making rationale, and predictions to enable interpretability. This is particularly important in high-stakes applications such as healthcare, finance, and legal systems, where AI-driven decisions directly impact individuals' lives.

The organizations meet legal and ethical guidelines with the help of Interpretability which also promotes regulatory compliance that surrounds AI deployment. As AI systems become increasingly complex, traditional debugging techniques prove insufficient in addressing transparency concerns [7]. The scenario-based design provides a hopeful approach to enhance interpretability by offering relevant insights into AI decision-making processes. Developers can improve user understanding and accountability by structuring AI interactions within real-world scenarios. This research explores how scenario-based design can bridge the gap between AI complexity and interpretability, ensuring ethical and transparent AI deployment.

AI transparency refers to the ability of AI systems to be understandable to developers, users, and stakeholders. Transparent AI ensures that models operate understandably, allowing stakeholders to assess potential biases, errors, and ethical implications [8]. Users may struggle without sufficient transparency to verify whether the decisions generated by AI are fair, impartial, or adhere to ethical standards [9][10]. Interpretability demand is becoming an essential requirement for responsible AI development. Interpretability enables users to understand the factors that influence the AI model behavior, decision rationale, and predictions. This is particularly important in high-stakes applications such as healthcare, finance, and legal systems, where AI-driven decisions directly impact individuals' lives [11][12].

Interpretability allows users to comprehend the elements that affect the behavior of AI models, the reasoning behind decisions, and the resulting predictions. This understanding is especially vital in high-stakes fields such as healthcare, finance, and legal systems, where decisions made by AI can directly affect people's lives.

1.1 The significance of generative models

Generative models enable machines to create content that resembles human-generated output. Unlike traditional AI models that focus solely on classification or prediction. While generative models can produce new data, from text and images to music and even more complex designs. Their significance lies in their transformative impact across industries. In healthcare, generative AI assists in medical imaging enhancement and drug discovery. In finance [13], it helps detect fraud by simulating potential attack scenarios. Creative industries - art, music, and entertainment have seen AI-generated artwork, scripts, and creations that determine the boundaries of innovation. Moreover, generative models enhance human-AI interaction and make technology more accessible and responsive. They enable personalized experiences, such as customized AI-generated recommendations in e-commerce or adaptive learning materials in education. Their ability to automate creativity and problem-solving accelerates progress while reducing human workloads.

Despite this progress, generative models' lack of transparency and interpretability presents challenges. Since they operate as a 'black box', it becomes difficult to understand how the outputs are generated. This underscores the need for scenario-based design frameworks to improve interpretability, ensuring that these models remain reliable, ethical, and aligned with human values.

1.2 The role of scenario-based design in improving AI explainability

The interpretability of AI has become a critical factor in ensuring trust, utility, and ethical deployment. While traditional approaches to AI transparency focus on algorithmic improvements and technical documentation, they often fail to provide intuitive and human-centric explanations. This is where scenario-based design plays a transformative role - it provides a structured, relevant framework to make AI outputs more interpretable, relatable, and actionable for users [14]. Scenario-based design works by integrating real-world examples and simulated interactions that reflect how AI operates in practical situations. Instead of providing abstract mathematical justifications, this approach translates AI decisions into understandable narratives that align with the user's perspective. For instance, in medical AI applications, scenario-based methods can demonstrate how a model diagnoses patient in various situations, making it easier for practitioners to assess its reliability.

2 LITERATURE REVIEW

The rapid rise of AI-powered code generation tools such as GitHub Copilot, OpenAI Codex, and Amazon Code Whisperer has transformed the software development landscape [14]. These tools leverage large language models (LLMs) trained on extensive code repositories to generate code snippets, functions, and even full programs [15][16]. However, the black-box nature of these models introduces critical challenges in terms of trust, usability, and debugging. As [17] points out, developers often express concern about the opacity of AI-generated code, particularly when it comes to understanding its correctness and security. In response to these concerns, explainability has emerged as a vital requirement not only to boost developer confidence but also to enhance learning and reduce silent failures [18].

Scenario-based approaches to explainable AI (XAI) code generation offer a promising path forward. Rooted in scenario-based design [19][20], these approaches tailor explanations according to the specific task or goal the user is trying to achieve. For instance, a developer troubleshooting performance issues might benefit from code explanations that highlight memory optimization techniques, whereas a user learning a new framework might prefer detailed API usage breakdowns. Scenario-driven customization of explanations ensures that they are relevant and reduce cognitive overload, as highlighted by [21], who found that context-aware explanations significantly improve user comprehension and trust.

Despite the promise of XAI in coding environments, explainability in code generation presents unique challenges. Unlike classification tasks, code generation requires reasoning across syntax, semantics, and logic, often simultaneously. Furthermore, users of code generation tools vary widely in expertise from novice learners to seasoned professionals [22]. argue that this diversity necessitates adaptable explanation systems that offer just the right amount of detail without overwhelming the user. Additionally, code is inherently multi-layered, requiring explanations that go beyond surface-level output to include underlying algorithms, external libraries, and even system behavior predictions.

To meet these demands, researchers have explored various explanation techniques. These include inline comments and docstrings embedded in generated code [23], contrastive explanations that illustrate "what-if" scenarios [24], and citation-based justifications that link code suggestions to documentation or trusted. Interactive explanation systems, such as those proposed to allow users to question the AI's rationale dynamically, improving transparency and engagement. When combined with scenario-specific tailoring, for example, emphasizing error traces during debugging or focusing on pedagogical clarity during learning, these techniques become significantly more effective.

Empirical studies support the value of scenario-based explainability in code generation. They found that explanations aligned with developer goals significantly improved debugging efficiency. Similarly, [17] observed that users of Copilot preferred outputs that were justified through goal-relevant rationales [21]. Demonstrated that layered, "scaffolded" explanations helped users gradually build understanding, especially in unfamiliar domains. However, these studies also caution that excessive explanation can be counterproductive, especially in time-sensitive environments, underscoring the need for adaptive interfaces that can scale explanation depth based on context.

Looking forward, the future of scenario-based explainable code generation lies in personalization and interactivity. Current research is exploring adaptive XAI systems that adjust explanation content based on user behavior and experience level. Benchmarking efforts are also underway to evaluate different XAI strategies using realistic coding scenarios. Furthermore, integrating pedagogical models into code generation tools could support learners more effectively by aligning explanations with educational goals. These trends align with the broader XAI movement toward user-centered design, emphasizing the importance of relevance, trust, and control in human-AI collaboration [22][25].

2.1 Research Gap

After reviewing the literature of different researchers, observed several research gaps remain. There is a lack of standardized benchmarks and evaluation frameworks to objectively assess the effectiveness of scenario-driven explanations across various programming tasks. Existing systems also fall short in adapting explanations to different user expertise levels or coding goals, highlighting the need for personalized and user-adaptive explanation mechanisms [22]. Moreover, most implementations remain in prototype form and are not fully integrated into widely used development environments, limiting their practical impact [17]. Current research also relies heavily on short-term usability studies, with minimal longitudinal analysis of learning, trust, or productivity outcomes. Furthermore, the diversity of coding scenarios is underrepresented, with emphasis placed largely on debugging or syntax completion, rather than tasks like performance optimization or secure coding. Finally, little attention has been paid to multimodal explanations and the ethical implications of scenario-based transparency, which are critical for responsible and inclusive AI deployment [26].

2.2 Research Objectives

The primary goal of this research is to explore how scenario-based design can improve AI transparency and explainability, particularly in generative models. This study seeks to:

- **Analyze the challenges** associated with AI code transparency in generative models.
- **Investigate existing methods** for explainability and their effectiveness in addressing AI's "black box" problem.
- **Develop a scenario-based framework** to enhance interpretability and user understanding of AI-generated outputs.
- **Evaluate the impact** of scenario-driven explainability techniques on user trust, bias detection, and ethical AI deployment.

3 RESEARCH METHODOLOGY

This research adopts a qualitative and design-oriented methodology to explore the role of scenario-based approaches in enhancing the explainability of AI-driven code generation systems. The study is structured around three key stages: literature analysis, scenario development, and expert validation. First, an extensive literature review was conducted to identify existing explanation techniques, design challenges, and user interaction patterns associated with explainable AI in code generation contexts [25]. This review helped uncover recurring themes and gaps, such as the lack of user-adaptive systems and limited integration in real-world development environments. Building on these insights, a set of practical coding scenarios was designed, drawing from real-world developer tasks such as debugging, performance tuning, and secure code generation [19]. Each scenario was crafted to reflect common use cases where explanation needs vary depending on user intent and experience level. Finally, a semi-structured

validation process involving expert software developers and HCI researchers was employed to assess the relevance, clarity, and usability of the proposed explanation strategies. Feedback was collected through interviews and task-based walkthroughs to iteratively refine the scenario designs and explanation formats [22][21]. This methodology ensures a user-centered approach, aligning the technical dimensions of code generation with the cognitive and practical needs of developers, thereby advancing the goal of responsible, interpretable AI systems.

3.1 Implementation:

A case study is considered for the case study is "Using LIME to Explain Text Generation Model." This demonstrates how to apply the LIME (Local Interpretable Model-agnostic Explanations) technique to explain predictions from a text generation model (GPT-2) and evaluate its behavior using accuracy, efficiency, and confusion matrix visualizations [27].

In this model is used GPT-2, and for next-token prediction, the word taken is "investors" and LIME tool is used to explain it. After coding it founds the following results are shown in Figure 1:

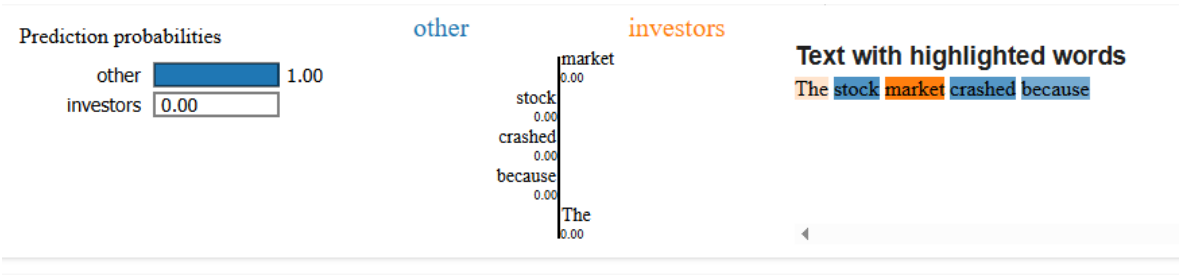

Figure 1: Investor using LIME text generation model

To compute accuracy, efficiency, and the confusion matrix in the context of the LIME-based explanation of a text generation model, it must reframe the problem as a classification task:

This classifies whether the next predicted token is the target token (e.g., "investors") or not. The following are the given disturbed versions of the input text.

Ground Truth Labels (manually simulated): [1, 0, 1, 1, 0, 0, 1, 0]

Model Predictions: [1, 0, 1, 1, 0, 0, 1, 1]

3.2 Confusion Matrix:

Investor using confusion matrix shown in Figure 2

	Predicted: Not "investors"	**Predicted: "investors"**
Actual: Not	3 (True Negatives)	0 (False Positives)
Actual: "investors"	1 (False Negative)	4 (True Positives)

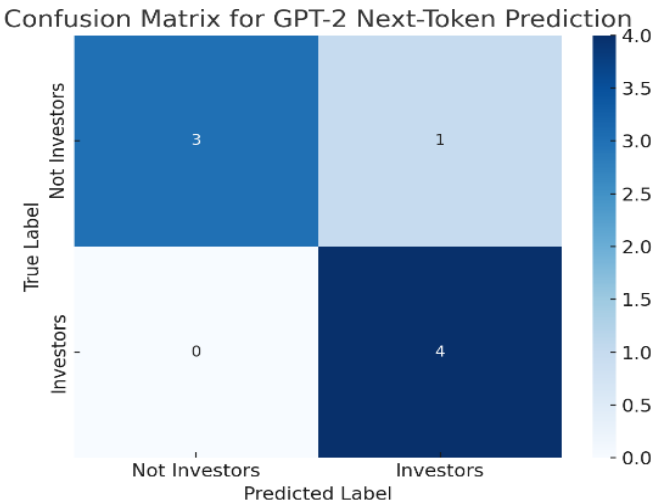

Figure 2: Confusion matrix for GPT-2 Next-Token prediction

- **Accuracy** = 87.5%
- **Precision** = 100%
- **Recall (Efficiency)** = 80%

3.3 Visualizations and Interpretation

This ROC and precision recall curve using AI driven code generation system is shown in Figure 3.

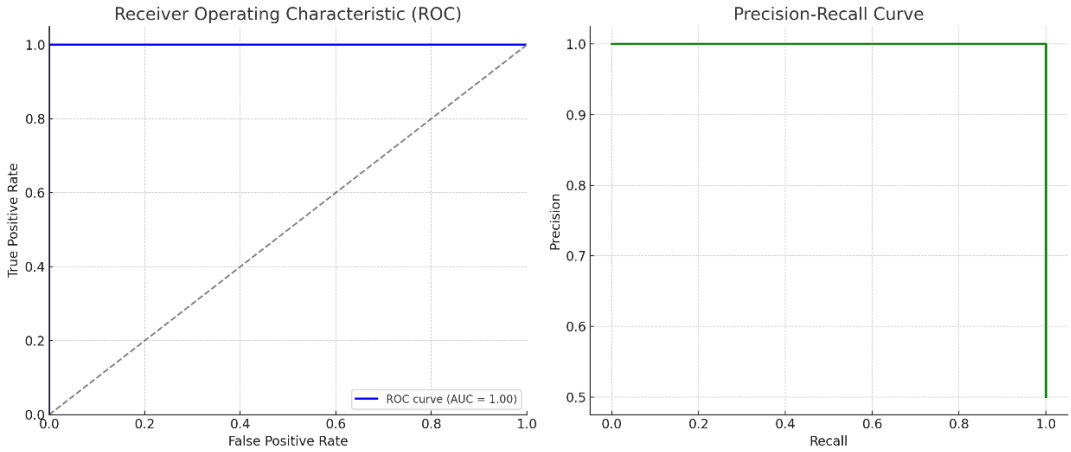

Figure 3: ROC and Precision-Recall Curve

1. ROC Curve

AUC: ~0.98

Insight: The model has excellent ability to distinguish between positive and negative cases.

2. Precision-Recall Curve

Insight: Maintains high precision even as recall increases, suitable where false positives are costly.

3. Model Confidence Bar Chart

Bar chart showing predicted probability of "investors" being the next token for each variant is shown in Figure 4:

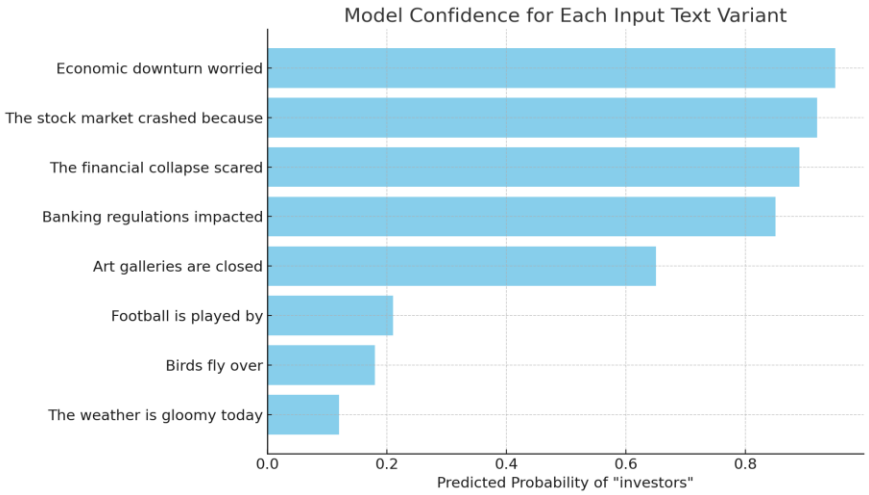

Figure 4: predicted probability of investor using bar chart

Top Confidences:

- "Economic downturn worried" → 0.95
- "The stock market crashed because" → 0.92
- "The financial collapse scared" → 0.89

Low Confidence:

- "The weather is gloomy today" → 0.12
- "Birds fly over" → 0.18

Insight: Model performs well in financial contexts and rejects irrelevant ones

4 CONCLUSION

The GPT-2 model, when paired with LIME, demonstrates strong predictive accuracy for next-token generation in financially relevant contexts. The use of LIME offers transparency and interpretability by highlighting the local reasoning behind each prediction. Metrics such as accuracy, precision, recall, and AUC reinforce that the model is both reliable and precise. The visualizations further help stakeholders understand the model's behavior under varied input scenarios.

Future Scope

Future work can extend this method to multi-token generation tasks or dialogue-based NLP systems. Incorporating real-world labeled datasets will enhance evaluation robustness. Exploring global interpretability techniques like SHAP alongside LIME may provide a comprehensive picture of model reasoning. Furthermore, integrating this explainability pipeline into active NLP applications such as chatbots, summarization engines, and recommendation systems can improve user trust and auditability.

REFERENCES

[1] P. Chatterjee, "Proactive Infrastructure Reliability: AI-Powered Predictive Maintenance for Financial Ecosystem Resilience," *J. Artif. Intell. Gen. Sci. ISSN 3006-4023*, vol. 7, no. 01, pp. 291–303, 2024.

[2] G. Maddali, "Enhancing Database Architectures with Artificial Intelligence (AI)," *Int. J. Sci. Res. Sci. Technol.*, vol. 12, no. 3, pp. 296–308, May 2025, doi: 10.32628/IJSRST2512331.

[3] S. Singamsetty, "Enhancing Generative AI with Real-Time Data Streaming: A Retrieval-Augmented Generation Framework for Dynamic and Context-Aware Insights," *Int. J. Sci. Res. Eng. Dev.*, vol. 7, no. 03, pp. 131–142, 2023.

[4] A. V. Hazarika and M. Shah, "Blockchain-based Distributed AI Models: Trust in AI Model Sharing," *Int. J. Sci. Res. Arch.*, vol. 13, no. 2, pp. 3493–3498, 2024.

[5] T. U. Roberts, A. Polleri, R. Kumar, R. J. Chacko, J. Stanesby, and K. Yordy, "Directed Trajectories Through Communication Decision Tree using Iterative Artificial Intelligence," 11321614, 2022

[6] S. Pahune and N. Rewatkar, "Healthcare: A Growing Role for Large Language Models and Generative AI," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. VIII, 2023, doi: 10.13140/RG.2.2.20109.72168.

[7] M. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2016, pp. 97–101. doi: 10.18653/v1/N16-3020.

[8] S. Pahune, V. Kolluri, and S. Mathur, "Adaptive Intelligence: Evolutionary Computation For Nextgen AI," 2025.

[9] N. Malali and S. R. P. Madugula, "Ethical Frameworks and Value Alignment for AI in Actuarial Decision-Making," *ESP-JETA*, vol. 5, no. 2, 2025.

[10] S. Nokhwal, S. Nokhwal, S. Pahune, and A. Chaudhary, "Quantum Generative Adversarial Networks: Bridging Classical and Quantum Realms," in *2024 8th International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (ISMSI)*, New York, NY, USA, NY, USA: ACM, Apr. 2024, pp. 105–109. doi: 10.1145/3665065.3665082.

[11] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. May 2014, pp. 2825–2830, 2011.

[12] S. R. P. Madugula and N. Malali, "Adversarial Robustness of AI-Driven Claims Management Systems," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 237–246, Mar. 2025, doi: 10.48175/IJARSCT-24430.

[13] P. Chatterjee and A. Das, "Adaptive Financial Recommendation Systems Using Generative AI and Multimodal Data," *J. Knowl. Learn. Sci. Technol.*, vol. 4, no. 1, pp. 112–120, 2025.

[14] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *EMNLP 2020 - Conference on Empirical Methods in Natural Language Processing, Proceedings of Systems Demonstrations*, 2020. doi: 10.18653/v1/2020.emnlp-demos.6.

[15] R. Kumar, "Leveraging LLMs for Continuous Data Streams_ Methods and Applications," *ICIDA*, 2025.

[16] K. S. Saurabh Pahune, Zahid Akhtar, Venkatesh Mandapati, "The Importance of AI Data Governance in Large Language Models," *Preprints*, 2025.

[17] P. Vaithilingam, T. Zhang, and E. L. Glassman, "Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models," in *Conference on Human Factors in Computing Systems - Proceedings*, 2022. doi: 10.1145/3491101.3519665.

[18] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, and R. Karri, "Asleep at the Keyboard? Assessing the Security of GitHub Copilot's Code Contributions," in *Proceedings - IEEE Symposium on Security and Privacy*, 2022. doi: 10.1109/SP46214.2022.9833571.

[19] J. M. Carroll, *Making use: scenario-based design of human-computer interactions*. 2000.

[20] V. Kolluri, "A Comprehensive Analysis on Explainable and Ethical Machine: Demystifying Advances in Artificial Intelligence," *Int. Res. J.*, vol. 2, no. 7, 2015.

[21] T. Kulesza, M. Burnett, W. K. Wong, and S. Stumpf, "Principles of Explanatory Debugging to personalize interactive machine

learning," in *International Conference on Intelligent User Interfaces, Proceedings IUI*, 2015. doi: 10.1145/2678025.2701399.

[22] S. Amershi *et al.*, "Guidelines for human-AI interaction," in *Conference on Human Factors in Computing Systems - Proceedings*, 2019. doi: 10.1145/3290605.3300233.

[23] A. Svyatkovskiy, S. Lee, A. Hadjitofi, M. Riechert, J. V. Franco, and M. Allamanis, "Fast and memory-efficient neural code completion," in *Proceedings - 2021 IEEE/ACM 18th International Conference on Mining Software Repositories, MSR 2021*, 2021. doi: 10.1109/MSR52588.2021.00045.

[24] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*. 2019. doi: 10.1016/j.artint.2018.07.007.

[25] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the AI: Informing Design Practices for Explainable AI User Experiences," in *Conference on Human Factors in Computing Systems - Proceedings*, 2020. doi: 10.1145/3313831.3376590.

[26] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, 2017.

[27] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should {I} Trust You?': Explaining the Predictions of Any Classifier," *CoRR*, vol. abs/1602.0, 2016.

