2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #53384

Teaching the Transparent Machine: Leveraging XAI to Foster Ethical Awareness in AI-Driven Programming Education

Ms. Abiha Tahsin Chowdhury, Missouri State University

Abiha Tahsin Chowdhury is a Master's student in Computer Science at Missouri State University.

Labiba Rahman, Purdue University – West Lafayette (College of Engineering)

Labiba Rahman is a PhD student in Engineering Education at Purdue University working in FACE Lab.

Teaching the Transparent Machine: Leveraging XAI to Foster Ethical Awareness in AI-Driven Programming Education (Work In Progress)

Introduction

Explainable artificial intelligence (XAI) has emerged as a response to the opacity of modern machine learning systems[1]. XAI aims to make model behavior more interpretable for human stakeholders [2],[1]. These techniques reveal the specific inputs or features that are responsible for making certain predictions[2]. Thereby, it promotes transparency, trust, and human oversight in AI-aided decision-making[2]. In education, XAI has been used mainly in learning analytics and educational data mining to interpret predictive models about students, helping educators understand which behavior influences those predictions and informing data-driven interventions [3], [4].

However, in most of these applications the explanations are designed for researchers, instructors, or institutional leaders, not for students as part of their day-to-day learning[5],[6],[7]. This creates a gap between XAI as an analytic tool and XAI as a pedagogical object that students engage with to learn about AI systems, programming practice, and ethical responsibility. At the same time, AI coding assistants and large language models (LLM) are now widely used by students in programming and related courses[8], and researchers and policy reports have raised concerns that students may treat these systems as opaque providers of solutions rather than engaging deeply with how they work or how their outputs compare to human-written code[4],[8].

These developments raise important questions for computer education. If students are already deeply entangled with AI tools in their everyday coding practices, how might curricula be designed so that they learn not only how to use these tools, but also how to interpret, critique, and responsibly co-create with them? How can explanations of model behavior, such as those produced by XAI methods like Shapley Additive exPlanations (SHAP)[9] and Local Interpretable Model-agnostic Explanations (LIME)[10] be repurposed from purely diagnostic or governance functions into instructional artefacts that support conceptual understanding of AI, critical reading of code, calibrated trust, and ethical awareness? Existing studies on XAI-supported programming show that explanation interfaces can reduce cognitive load and improve debugging performance, yet they also reveal challenges for novice learners and call for careful pedagogical scaffolding[11].

The present work addresses this gap by proposing a study in computer education that uses XAI explanations from code authorship-classification models as central classroom artefacts, enabling

students to interrogate AI-generated code, reflect on model behavior, and develop meta-programming literacy, defined here as students' ability to interpret, critique, and reason about AI-generated code and the underlying decision processes of AI systems and calibrated trust in AI-assisted programming.

Literature Review

Systematic and scoping reviews show that XAI in education has been primarily used to interpret predictive models about students[4]. For example, AI models are used for grades, performance, engagement, dropout, or satisfaction, rather than to support student-facing learning activities[12],[7],[13]. Methods including SHAP and LIME have been applied to educational data mining and learning analytics systems to identify influential features (e.g., time-on-task, assessment scores, interaction patterns) and to provide interpretable insights to educators and administrators[14],[12],[7]. These studies report that XAI can improve transparency, support more informed interventions, and increase stakeholders' trust in AI-based tools, but explanations are typically consumed by researchers, advisors, or system designers rather than integrated into classroom pedagogy.

Within programming education, XAI has begun to be explored as a way to make intelligent tutoring systems and automated feedback more transparent[15]. A deep learning model with a Gradient Integration explanation module was introduced in a recent study that highlights code segments associated with predicted errors via color coding[11]. Experimental results from this work indicate that students using XAI-enhanced debugging interface experienced significantly lower intrinsic cognitive load and achieved significantly higher exam scores compared to a control group without explanations which suggests that explanation interfaces can direct student attention toward relevant code regions and support more efficient debugging. At the same time, the study notes that less knowledgeable students sometimes struggled to interpret explanations and that perceived reliability of AI feedback varied, underscoring the need for pedagogical scaffolding when integrating XAI into programming courses.

Broader reviews of XAI in education argue that techniques such as SHAP and LIME can demystify complex models for non-expert users by surfacing influential features and providing human-understandable rationales, thereby supporting trust, validation, and learning about model behavior[5]. However, these reviews also conclude that existing empirical work has only begun to explore how XAI might shape student learning processes, metacognition, or disciplinary identities, and they call for designs that align explanations with pedagogical goals and learners' cognitive needs.

In parallel, a growing body of work investigates machine-learning classifiers that distinguish human-written code from AI-generated code[16]. Studies demonstrate that classifiers including

traditional models based on engineered features and transformer-based models can achieve high performance in authorship attribution by leveraging code stylometry, structural characteristics, and statistical regularities. These efforts are often motivated by concerns about academic integrity and detection of AI-generated assignments, and they are typically framed around detection accuracy and robustness rather than learning outcomes[16].

This literature points to an opportunity in computer education to design learning experiences where XAI explanations are central instructional materials: students use explanations to interrogate both human and AI-generated code, to reflect on coding practices and sociotechnical implications, and to calibrate their trust in AI tool

Proposed Methodology

The proposed study positions XAI as a pedagogical tool for meta-programming literacy in undergraduate computing education. Building on work that shows XAI can support debugging and reduce cognitive load in programming, as well as on reviews that emphasize the need for learner-facing explanations. The study will examine how students engage with explanations from an authorship-classification pipeline in which models differentiate between human-written and LLM-generated code.

In this study, meta-programming literacy is operationalized through students' abilities to (a) interpret XAI visualizations, (b) critique model reasoning, (c) evaluate AI-generated code quality, and (d) reflect on their own trust and decision-making when interacting with AI-assisted programming tools. These dimensions are assessed through structured tasks, surveys, and qualitative reflections.

The core research questions are aligned with CoED's focus on learning and pedagogy:

- How does exposure to XAI explanations of code authorship influence students' understanding of AI systems and model behavior in a programming context?
- How does working with explanations affect students' ability to critically evaluate AI-generated code in terms of quality, maintainability, and originality?
- How does the intervention shape students' trust calibration toward AI coding assistants and automated detectors, and their sense of professional responsibility when co-creating with AI?

Participants will be undergraduate students enrolled in introductory or mid-level programming or software engineering courses where the use of AI coding assistants is formally permitted under defined conditions. This context reflects contemporary practices in computing education, where

institutions and instructors are increasingly articulating guidelines for responsible AI use rather than banning such tools outright.

This study will be conducted in an undergraduate programming-focused course where the use of AI coding assistants is formally permitted under instructor-defined guidelines. Example course contexts include Introduction to Programming, Data Structures and Algorithms, Software Engineering I, Object-Oriented Programming.

The study will take place during a regularly scheduled class session, with the instructor's permission to allocate approximately 30 minutes for the intervention. The instructional activity will be positioned as an enrichment exercise on AI-assisted programming and professional responsibility.

We anticipate recruiting approximately 20–25 undergraduate students from a single course section. Participation will be voluntary. Informed consent will be obtained prior to data collection, and students will be assured that participation or non-participation will not affect course grades, all responses will be anonymized and the instructor will not have access to individual-level research data.

We will seek diversity across gender, ethnicity, prior programming experience, and familiarity with AI coding assistants in order to capture a range of perspectives on AI-assisted programming. While the sample will be course-bound rather than statistically representative, recruitment materials will explicitly emphasize inclusive participation and voluntary engagement.

Study Design Overview

The study adopts a quasi-experimental pre–post design with embedded qualitative inquiry. The intervention positions XAI explanations from a code authorship-classification model as instructional artefacts. Students will:

- Complete a pre-survey.
- Engage with XAI explanations of matched human-written and LLM-generated homework solutions.
- Complete a post-survey.
- Optionally participate in follow-up focus groups.

The primary unit of analysis is students' interpretation of XAI artefacts and shifts in understanding, critique, and trust calibration regarding AI-generated code.

To address concerns regarding potential bias and differences in code sophistication, the programming problems used in the study will be drawn directly from prior homework

assignments from the same course in which the intervention is implemented. Selecting problems from the established course curriculum ensures strong alignment with students' current learning objectives and expectations. It also helps maintain comparable levels of difficulty and conceptual scope, as the tasks have already been calibrated by the course instructor for that specific cohort and learning stage. By situating the intervention within authentic course assignments, the study reduces the risk of artificial task design bias and enhances ecological validity, ensuring that students engage with problems that are pedagogically appropriate and directly relevant to their ongoing programming development.

Human-written solutions will be curated from prior course offerings (e.g., submissions from previous semesters), following appropriate permission procedures and full de-identification to protect student privacy. To reduce variability in expertise and prevent unintended bias in the classification task, only submissions within a comparable grade range (for example, B to A-level solutions) will be selected. This approach ensures that the human-written code reflects competent but realistic student performance, rather than extreme cases. Submissions that are exceptionally weak or unusually sophisticated will be excluded to avoid skewing the model toward detecting quality rather than authorship patterns. Where necessary, minor formatting normalization will be applied (e.g., removal of metadata, standardized indentation) to eliminate superficial stylistic markers that do not reflect substantive logical or structural characteristics, thereby maintaining focus on meaningful programming features rather than incidental formatting differences.

For each selected homework prompt, AI-generated code will be produced using a state-of-the-art large language model under standardized prompting conditions. Identical homework instructions will be used, no additional contextual hints will be provided, generation temperature will remain at default settings, and no iterative refinement will occur.

A pretrained transformer-based source-code model (e.g., CodeBERT or a comparable architecture) will serve as the base model for authorship classification. To maintain interpretability and reduce overfitting, all transformer layers will be frozen and only the final classification layer will be fine-tuned. Training data will consist of curated human-written homework solutions and LLM-generated solutions for the same prompts. Stratified data splits will be used, and model performance will be evaluated on a held-out test set using accuracy, F1-score, and ROC-AUC metrics. Only if the classifier demonstrates reasonable discriminative performance will its outputs be used to generate XAI artefacts for student-facing activities. This ensures that students are interpreting explanations derived from meaningful predictive patterns rather than random noise.

Several safeguards will be implemented to mitigate bias and instructor influence. Authorship labels (human vs AI) will be hidden from students during the activity. If rubric-based evaluation of written critiques is conducted, it will be performed by research team members independent of course grading. The course instructor will not evaluate student responses used for research

purposes and will not be informed of authorship identities during the intervention. These procedures prevent expectancy effects, grading bias, and instructor influence on interpretation.

Both global and local explanation techniques will be used. Global SHAP summary plots will illustrate which features most strongly influence authorship classification across the dataset, while local SHAP and LIME explanations will highlight influential tokens, syntactic structures, or code regions within individual samples. These explanations will be translated into student-accessible visualizations accompanied by interpretive prompts to reduce cognitive overload and support guided reflection.

The intervention will occur within a 30-minute session structured into three phases. During the first phase (5–7 minutes), students will complete a pre-survey assessing baseline understanding of AI systems, trust in AI-generated code, and ethical awareness. Example items include Likert-scale statements such as "I understand how AI coding assistants generate code," "I generally trust AI-generated code without modification," and "Using AI-generated code without modification raises ethical concerns," as well as an open-ended question asking what distinguishes human-written code from AI-generated code.

Survey items are designed to align with key learning constructs, including interpretability, debugging confidence, and reasoning about model outputs. Where applicable, items are adapted from prior studies on AI literacy and trust in automation, while additional items are developed specifically for this exploratory context.

During the second phase (15–18 minutes), students will receive a programming prompt along with two unlabeled solutions: one human-written and one AI-generated, accompanied by the classifier's prediction and associated SHAP/LIME explanations. Structured tasks will ask students to interpret which code regions influenced the classification, evaluate whether highlighted features align with their own quality judgments, suggest modifications to improve robustness or perceived "human-ness," and reflect on how explanations affect their trust in AI systems.

In the final phase (5–7 minutes), students will complete a post-survey assessing shifts in conceptual understanding, critical evaluation confidence, trust calibration, and ethical positioning. Items will measure perceived learning gains, trust shifts, and reflections on professional responsibility in AI-assisted programming. Open-ended prompts will invite students to describe how their thinking changed.

Following the classroom intervention, 5–10 consenting students will participate in semi-structured focus groups lasting approximately 30–40 minutes. Sessions will be audio-recorded with consent and transcribed for analysis. Core questions will probe how students interpreted XAI explanations, whether they agreed or disagreed with the model's reasoning, how the activity influenced their perception of AI-generated code, and how programming curricula should

address AI-assisted development. Probing questions may evolve to explore emergent themes in depth.

Data Analysis Plan

Quantitative pre–post survey responses will be analyzed using paired t-tests or non-parametric alternatives depending on distributional assumptions. Effect sizes will be calculated, and internal consistency of multi-item constructs will be evaluated using Cronbach's alpha.

Qualitative data including open-ended survey responses, written code critiques, and focus group transcripts will be analyzed using inductive thematic analysis grounded in engineering education research traditions. The process will begin with repeated familiarization through transcript review, followed by line-by-line open coding to identify emergent meaning units. A codebook will be developed iteratively, with two independent coders coding a subset of transcripts to establish inter-rater reliability and resolve discrepancies through discussion. Codes will then be organized into higher-order themes such as emerging meta-programming literacy, calibration of trust, tensions between model logic and human judgment, shifts in ethical positioning, and recognition of dataset bias or model fragility. Finally, themes will be interpreted through theoretical lenses including metacognition, professional responsibility in engineering practice, and AI literacy frameworks to ensure analytic depth beyond descriptive categorization.

Conclusion and Future Work

This work contributes to computer science and engineering education by repositioning explainable artificial intelligence (XAI) from a diagnostic tool into a pedagogical resource. Rather than limiting explanations to researchers or administrators, the study conceptualizes SHAP- and LIME-based artefacts from authorship-classification models as instructional materials that support meta-programming literacy, calibrated trust, and professional responsibility in AI-assisted coding environments. By embedding XAI explanations within authentic programming coursework, the proposed intervention demonstrates how computational modeling can be meaningfully connected to learning outcomes such as critical evaluation of AI-generated code and ethical awareness in co-creation contexts.

The methodological framework strengthens this contribution by ensuring careful matching of human and AI-generated code sophistication, implementing blinding procedures to reduce bias, embedding the intervention within authentic course assignments, and incorporating both quantitative and qualitative analytic rigor. Through pre–post measurement and thematic analysis of student reflections and discussions, the study moves beyond technical demonstration toward a pedagogically grounded examination of how students interpret and negotiate AI explanations in real classroom contexts.

As an exploratory, pilot-scale study, the findings are intended to provide initial insights into student learning processes and instructional design considerations rather than broadly generalizable conclusions.

Future work will extend this framework in several directions. First, replication across multiple course levels and institutional contexts will help determine how prior programming experience shapes engagement with XAI artefacts. Second, the methodological approach can be adapted beyond programming education into other AI-influenced domains within computing. For example, in writing-intensive courses, authorship classification paired with XAI explanations could support critical reflection on logic, argument structure, and originality. In mathematics or data science contexts, explanation tools could help students interrogate model reasoning, dataset bias, and solution pathways.

More broadly, this line of research positions XAI as a scalable instructional tool for responsibility-centered AI literacy. As generative systems become increasingly embedded in professional workflows, educational practices must evolve beyond detection or prohibition toward cultivating interpretive competence, reflective judgment, and ethical co-creation. By treating explanation not merely as transparency, but as pedagogy, this work advances a model of computing education that prepares students to engage critically and responsibly with intelligent systems.

References:

[1] A. Darwish, “Explainable Artificial Intelligence: A New Era of Artificial Intelligence,” *Digit. Technol. Res. Appl.*, vol. 1, no. 1, p. 1, Jan. 2022, doi: 10.54963/dtra.v1i1.29.

[2] S. Ali *et al.*, “Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence,” *Inf. Fusion*, vol. 99, p. 101805, Nov. 2023, doi: 10.1016/j.inffus.2023.101805.

[3] Vinitra Swamy, B. Radmehr, Natasa Krco, M. Marras, and T. Käser, “Evaluating the Explainers: Black-Box Explainable Machine Learning for Student Success Prediction in MOOCs,” Jul. 2022, doi: 10.5281/ZENODO.6852964.

[4] Z. M. Altukhi and S. Pradhan, “Systematic Literature Review: Explainable AI Definitions and Challenges in Education,” 2024.

[5] H. Khosravi *et al.*, “Explainable Artificial Intelligence in education,” *Comput. Educ. Artif. Intell.*, vol. 3, p. 100074, 2022, doi: 10.1016/j.caeai.2022.100074.

[6] G. Türkmen, “The Review of Studies on Explainable Artificial Intelligence in Educational Research,” *J. Educ. Comput. Res.*, vol. 63, no. 2, pp. 277–310, Apr. 2025, doi: 10.1177/07356331241310915.

[7] E. Ben George, “Explainable AI Methods for Predicting Student Grades and Improving Academic Success,” *J. Inf. Syst. Eng. Manag.*, vol. 10, no. 23s, pp. 117–126, Mar. 2025, doi: 10.52783/jisem.v10i23s.3680.

[8] S. Rojas-Galeano, “New Kid in the Classroom: Exploring Student Perceptions of AI Coding Assistants,” 2025, *arXiv*. doi: 10.48550/ARXIV.2507.22900.

[9] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” 2017, *arXiv*. doi: 10.48550/ARXIV.1705.07874.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why Should I Trust You?’: Explaining the Predictions of Any Classifier,” 2016, *arXiv*. doi: 10.48550/ARXIV.1602.04938.

[11] F. H. Wang, “On the Effect of Explainable AI on Programming Learning: A Case Study of using Gradient Integration Technology,” in *Education & Information Technology*, Academy & Industry Research Collaboration Center, Jun. 2024, pp. 17–28. doi: 10.5121/csit.2024.141202.

[12] Z. Zhan and T. Shen, “Development of a prediction model for student teaching satisfaction based on 10 machine learning algorithms,” *Sci. Rep.*, vol. 15, no. 1, p. 36547, Oct. 2025, doi: 10.1038/s41598-025-19039-x.

[13] V. Swamy, S. Du, M. Marras, and T. Käser, “Trusting the Explainers: Teacher Validation of Explainable Artificial Intelligence for Course Design,” 2022, *arXiv*. doi: 10.48550/ARXIV.2212.08955.

[14] Md. M. Islam, F. H. Sojib, Md. F. H. Mihad, M. Hasan, and M. Rahman, “The integration of explainable AI in Educational Data Mining for student academic performance prediction and support system,” *Telemat. Inform. Rep.*, vol. 18, p. 100203, Jun. 2025, doi: 10.1016/j.teler.2025.100203.

[15] M. Zerkouk, M. Mihoubi, and B. Chikhaoui, “A Comprehensive Review of AI-based Intelligent Tutoring Systems: Applications and Challenges,” 2025, *arXiv*. doi: 10.48550/ARXIV.2507.18882.

[16] D. Cotroneo, C. Improta, and P. Liguori, “Human-Written vs. AI-Generated Code: A Large-Scale Study of Defects, Vulnerabilities, and Complexity,” 2025, *arXiv*. doi: 10.48550/ARXIV.2508.21634.