

Analyzing developer misguidance in AI code suggestion tools: a large-scale empirical study

Sathursana K.^{1*}, Nirubikaa R.¹

¹Department of Software Engineering, Faculty of Computing,
Sabaragamuwa University of Sri Lanka, Sri Lanka

*ksathursana@std.appsc.sab.ac.lk

Artificial Intelligence (AI)-based code suggestion tools are widely used in current software development. These tools save time and help developers at many stages. At the same time, they produce code with some issues-the code looks correct but contains logical mistakes, security weaknesses, or outdated patterns. This kind of issue is commonly referred to as misguidance. Misguidance affects developers in many ways, such as reduced productivity and developer confidence, and it is more harmful for beginners who heavily depend on these AI code suggestions. This study focuses on analyzing the misguidance issues in AI-suggested code. For this purpose, 112,689 developer feedback posts were initially collected from GitHub, Stack Overflow, and Reddit, which were preprocessed and filtered to 33,766 relevant misguidance posts using keyword-based and semantic classification methods. Initially, 200 random posts were manually reviewed and labeled by two people until full agreement was reached, from which a six-category classification of misguidance was developed. A fine-tuned BERT model was then trained using this balanced dataset and achieved 96.6% accuracy and 97.1% macro-F1 accuracy. This model was used to classify all 33,766 posts. From the entire dataset, the most common issue identified was Debugging & Edge Cases (28.5%), then Integration Issues (24.5%) and Performance Issues (20.3%). Logical Errors appear in 16.6% of the posts and Security & Privacy Risks appear in 8.4% of the posts, and Training Data Problems in 1.8%. According to the posts, beginners often face Performance Issues (74.1%). Trust is another factor affected by this AI code misguidance. Both beginner and experienced developers say they cannot fully trust AI suggestions. From this study, some practical guidelines were derived to help developers who are affected by misguidance, such as always testing AI-generated code and seeking human review for beginners without accepting it blindly.

Keywords: *AI code generation, Beginner impact, Developer trust, Misguidance taxonomy, Mitigation guideline*