Ethan Kanyid

CPTS 428

Professor Liu

9-11-2025

Overview: Artificial Intelligence (AI) is becoming more commonplace in today's environment and workplace, and it is being used for tasks ranging from writing emails to developing software. This is great for productivity and decreasing the workload and time spent on different projects, but it also introduces new challenges. While at first AI and large language models (LLM) were viewed with skepticism, they are now being embraced more widely. In software development, AI is praised for accelerating workflows and shortening production cycles, but from a security standpoint, does this introduce new vulnerabilities and bugs that can be exploited?

This project aims to discover and compare how LLM's generate code under different levels of interaction. This project will focus on prompting the LLM to generate a simple web server with a backend database. The method will be to minimally rely on human interaction, creating an environment suited to testing the AI's raw coding capabilities.

Implementation: This project will be developed in a fresh virtual machine with simple dependencies pre-installed. ChatGPT will be accessed through the web without an account to avoid any influence from prior interactions or personalization. After selecting a framework, ChatGPT will develop the web server and backend with different levels of interaction. It is expected that the LLM will not be successful on every attempt, but this will be resolved through the coding process which is referred to as vibe coding. This process entails prompting an LLM to generate code, running the code, and pasting the output back into the LLM for further code generation.

Once deployed, the application's basic functionality will be verified. Following this, black-box-like testing will be performed to identify potential vulnerabilities. Feedback from testing will be given to the AI, allowing it to attempt corrections. Detailed documentation will be maintained throughout, including prompts, command-line inputs, and discoveries. The final phase will involve full interaction with the AI to make further changes and determine if previously identified issues persist.

Impact: Success will be measured by how effectively the AI can develop functional, secure code and by the level of developer involvement required. The findings will inform the degree to which AI-generated code can be trusted in development environments and help define best practices for human-AI collaboration in software development with today's capabilities. As AI continues to grow, it can be re-tested and updated with a new outline for appropriate and safe use of code generation.