

Improving Static Analysis output with Large Language Models

- Static analysis is used **to identify and correct bugs** during software development
 - **Manual interpretation is required** to evaluate the results from analysis tools
 - Large Language Models (LLMs) can be **effective in processing the results**
-

Objective of the thesis work

- **Static analysis** tools (like Cppcheck) are used **to identify bugs and vulnerabilities** in C/C++ code in embedded software development
- This thesis focused on researching the possibility of **using LLMs to post-process the Cppcheck results**

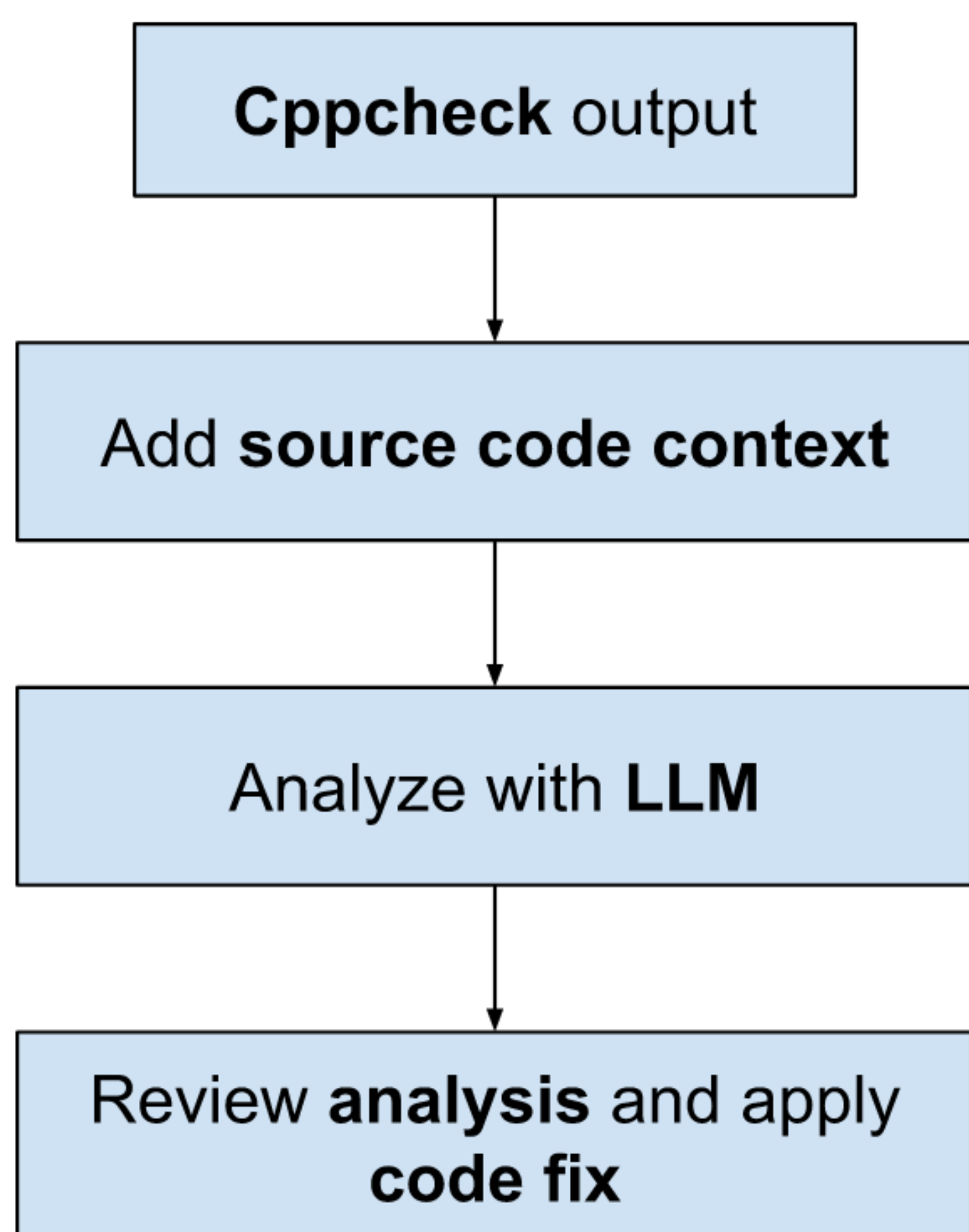


Figure 1: The idea for integrating LLMs with Cppcheck. The proposed solution is to post-process Cppcheck results along with contextual code.

Result analysis is difficult

- Manual interpretation of Cppcheck results is **error-prone** and takes significant amount of time
- Understanding of the language and the contextual code is required to assess the results

LLM-enhanced static analysis

- Large Language Models have demonstrated abilities to read and explain program code
- As a part of the thesis, a tool was developed to integrate LLMs with a traditional static analysis tool

Performance of LLMs in static analysis

- In this thesis, the most advanced LLMs (OpenAI GPT-4o) output **correct analysis of errors** and generated **valid bug fixes** with high accuracy
- For the sample code with simple C++ undefined behavior bugs, **all error cases were resolved by the LLM**

Future of LLM analysis of code

- LLMs are **not suitable as a replacement** for static analysis tools, but useful to use as a companion for them
- Static analysis done with an **LLM only** also demonstrated **good vulnerability identification performance**