Our paper was written based on the experimental results recorded in **result.xlsx**. In **result.xlsx**, Java files committed after October 2021 from the six open source repositories we identified (bc-java, h2database, junit4, gson, guava, pgjdbc) and the responses obtained by querying gpt about them are collected at the subsheets with the project name. However, due to the limit on the maximum number of characters that can fit in an Excel cell, Java files larger than about 30kb are truncated from the middle. In {project name}-cwe sheets, only the items that satisfy the regular expression '(CWE|CVE)-[0-9]+' and whose security vulnerabilities have been pointed out by gpt are separately organized. In the Result sheet, the number of commits, number of modified Java files, number of gpt responses, number of vulnerabilities pointed out and their labels, and static analyzer results are organized as in the paper. The PR&Issue sheet contains information reported through git.

To reproduce these experimental results, refer to **ViChecker_User_Guide.pdf** and run **ViChecker**. As in the User Guide, to run a Python file, you must convert the ipynb file to python. However, it is recommended to use the provided ipynb file. If you execute in the order provided in the User Guide, the desired open source projects and commit-logs, commit-files, and commit-sheets directories will be created. The **commit-logs** directory is created when **_2_git_commit_logger** is executed. There are files that store the latest commit history of each project in json format along with a list of modified java files. The **commit-files** directory is created when **_3_git_file_tracker** is run. In this directory, Java files and diff files at the time of commit exist for each project. When **_4_gpt_responser** is executed, gpt's responses are added. The **commit-sheets** directory contains information on the original files queried by gpt for each project and Excel files that store the answers. Please refer to the User Guide and code for more details.