Evaluating INSIGHT, a GitHub Plugin for Bug Localization and Repairing Suggestions

RESEARCH GOAL AND PROCEDURE

The goal of this study is to evaluate INSIGHT, an automated repository-aware bug localization assistant for GitHub. INSIGHT uses a multi-stage RAG pipeline, combining dense retrieval, graph-based enrichment, and LLM reasoning to suggest buggy files and functions for a reported issue. Moreover, INSIGHT provides LLM-inferred general ways to address such bugs after localization.

PARTICIPATION AND CONFIDENTIALITY

You must be at least 18 years old to participate.

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission. Your responses to this study will be kept confidential. If you decide to participate, you are free to withdraw your consent and to discontinue participation at any time without penalty.

CONTACT

If you have any questions, please ask us at any moment. If you have any additional questions later, Jitesh Sureka (bsse1115@iit.du.ac.bd, +8801780935761), Ahmed Adan (bsse1131@iit.du.ac.bd, +8801813865290), Proma Chowdhury (bsse1132@iit.du.ac.bd, +8801767895677) and Dr. Kazi Muheymin-us Sakib (sakib@iit.du.ac.bd, +8801730051232), will be happy to answer them.

CONSENT

YOU ARE MAKING A DECISION WHETHER OR NOT TO PARTICIPATE.

IF YOU WANT TO PARTICIPATE, PLEASE ENTER YOUR NAME IN THE TEXT FIELD BELOW, AND START THE SURVEY.

* Indicates required question

1. Please Enter your Name *

____

Overview of INSIGHT

INSIGHT is an intelligent GitHub assistant that automatically analyzes bug reports and suggest code functions that likely contain the issue. When developers report a bug, INSIGHT understands the relationships between different parts of the code, and provides specific suggestions about what needs to be fixed, potentially saving hours of manual debugging time.

Study Overview

You are provided with a test GitHub repository, with the **INSIGHT** tool already installed in it, and a set of issue reports describing real-world software bugs for that repository. For each issue report, you will create a new issue in the repository, and evaluate INSIGHT's suggestions, including bug localization in the relevant files and functions, technical analysis of why the bug is present within a particular file(s)/method(s), and general approaches taken by practitioners to resolve such bugs. You will then compare INSIGHT's suggestions against the developer-confirmed Ground Truths assess INSIGHT's predictive accuracy, usability, usefulness, and contextual capability.

Based on your experience with INSIGHT, you will answer a set of questions to give us feedback on how well the tool performs in localizing bugs and provide suggestions to improve it.

In summary, you will need to perform the following steps (detailed instructions are provided later in this study):

1. Access the Test Repository:

- **INSIGHT** is already installed in the test repository that we are providing
- Visit the repository: **[Test Repository Link]**
- Familiarize yourself with the repository structure

2. For each of the provided issues in this survey, you will need to perform the following:

(a) Read and Understand the Issue:

- Carefully read the provided issue report with its title and description
- Review the Ground Truth (GT) information provided here showing which files/functions actually contain the bug
- Understand the nature of the bug being reported

(b) Submit the Issue:

- Create a new issue in the test repository
- **Copy and paste** the provided **issue title** and **description** exactly as given
- This step mimics the real-world scenario when a developer discovers a problem and reports it via the issue tracker
- Wait for INSIGHT to process the issue (typically takes **10-20 seconds**)

(c) Analyze INSIGHT's Response:

- Examine INSIGHT's automated comments under the created issue.
- Take notes about INSIGHT's suggested buggy files, functions, technical analysis of the bug, and the provided suggestions regarding how practitioners commonly fix these bugs.
- Compare INSIGHT's provided suggestions with our provided Ground Truth (information about the actual buggy files, methods, bug analysis, and general approaches to resolve them).

3. **Provide Overall Feedback:**

- After completing all bug localization tasks, answer questions about your overall experience with INSIGHT
- Share your thoughts on the tool's strengths, weaknesses, and potential improvements
- Answer questions regarding your professional background and experience with debugging

Important Notes

- Feel free to explore INSIGHT's suggestions in detail and click on the provided links to view the actual code
- You may spend as much time as needed to thoroughly evaluate each response
- There are no right or wrong answers—we value your honest feedback
- If INSIGHT fails to provide suggestions or encounters an error, please note this in your responses

Thank you for participating in this study. Your feedback will help us improve INSIGHT and make it more useful for software developers!

Evaluation of INSIGHT

Issue Report 1:

Please read and understand the following issue report:

Issue Title:

Save text output

Issue Description:

I noticed that simple text outputs are only redirected to `stdout`. There is on API to save it to a file.
Correct me if I am wrong but I could not see it [here](https://github.com/ioos/compliance-checker/blob/7276abba102e2ef6cb390d49ba0c38726f625913/compliance_checker/runner.py#L43-L50).

I do know that in CLI interface it is easy to redirect to a file but it would be nice to be able to do that from class method too. Not also that the docs give the idea that this is possible and the error message [here](https://github.com/ioos/compliance-checker/blob/7276abba102e2ef6cb390d49ba0c38726f625913/compliance_checker/runner.py#L53) is not very helpful.

After reading and understanding the issue, please complete the following instructions.

1. Create the issue on **INSIGHT_user_study_repository** by performing the following steps:

(a) Go to the repository
(b) Click on Issues
(c) Click on ***'New issue'***
(d) Copy and paste the content in the title and description sections
(e) Click ***'Submit new issue'***

2. Observe INSIGHTS's suggestions for the bug localization and program repairing/bug resolving suggestions features. Feel free to take notes.

3. Assess the suggestions provided by INSIGHT. For your reference, we have included information about the actual buggy files and methods, as well as common approaches to fixing these bugs. When answering, please compare INSIGHT's suggestions with this reference information.

Buggy Files:

['*compliance_checker/tests/test_cli.py*', '*compliance_checker/runner.py*']

Buggy Methods:

['ComplianceChecker', 'TestCLI']

General Approaches to Address these Bugs:

1. Identify Output Handling:

Locate all instances where output is directed to stdout or a file within the run_checker method.

2. Modify Output Logic:

Check if output_filename is not "-".
If true, open the specified file in write mode with UTF-8 encoding.

3. Redirect Output:

Use a context manager to redirect stdout to the opened file if output_format is "text".
Ensure that the stdout_output method writes to the redirected stdout.

Issue Report 2:

Please read and understand the following issue report:

Issue Title:
Fail result for Flags and flag attributes check with no reasoning

Issue Description:

Running the CF checker on [this file] (http://thredds.aodn.org.au/thredds/fileServer/IMOS/SRS/SST/ghrsst/L3S-1d/dn/2016/20160919092000-ABOM-L3S_GHRSST-SSTfnd-AVHRR_D-1d_dn.nc) results in

~~~
Name                         Priority:    Score:Reasoning
-----------------------------------------------------------------------------
§3.5 Flags and flag attributes        :3:    31/32 :
```
~~~

An apparently failed result, with no reasoning message.

After reading and understanding the issue, please complete the following instructions.

1. Create the issue on INSIGHT_user_study_repository by performing the following steps:

   (a) Go to the repository
   (b) Click on Issues
   (c) Click on ***'New issue'***
   (d) Copy and paste the content in the title and description sections
   (e) Click ***'Submit new issue'***

2. Observe INSIGHTS's suggestions for the bug localization and program repairing/bug resolving suggestions features. Feel free to take notes.

3. Assess the suggestions provided by INSIGHT. For your reference, we have included information about the actual buggy files and methods, as well as common approaches to fixing these bugs. When answering, please compare INSIGHT's suggestions with this reference information.

Buggy Files:
['compliance_checker/cf/cf.py']

Buggy Methods:

['check_flags']

General Approaches to Address these Bugs:
1. **Identify Error Handling Gaps:**
   - Review the `check_flags` function to ensure that all potential failure points in

`_check_flag_meanings`, `_check_flag_values`, and `_check_flag_masks` are captured and reported with specific messages.

2. **Enhance '_check_flag_meanings':**
   - Ensure the function checks for:
     - Presence of `flag_meanings`.
     - Type is a string.
     - Valid characters and format.
   - Add detailed error messages for each failure condition.

Issue Report 3:

Please read and understand the following issue report:

Issue Title:
JSON output issue in 3.0.3 with ERDDAP URLs

Issue Description:

I've been testing the new Compliance Checker release with ERDDAP (on Windows), and I see some different results in the output when using ERDDAP URLs with the JSON output format than OPeNDAP.
I made a test script that illustrates the issue:
https://gist.github.com/mwengren/68f61b0e70868a57336b81f863d7dbc9
Or, I think you can see the same issue (in my environment at least) by running these commands:

```
compliance-checker -t cf -f json http://coastwatch.pfeg.noaa.gov/erddap/griddap/osuSstAnom
compliance-checker -t cf -f json http://ona.coas.oregonstate.edu:8080/thredds/dodsC/NANOOS/OCOS
```

The issue is that when I try to hit the ERDDAP URL, it includes what looks like a debug line in stdout:

```
Using cached standard name table v29 from C:\\Users\\Micah.Wengren\\.local\\share\\compliance-checker\\cf-standard-name-table-test-29.xml
{
  "cf": {
    "scored_points": 168,
    "low_count": 0,
```

When I try to parse this into JSON using subprocess.Popen (see the Gist), if fails because it can't parse the first line.
I'm using 3.0.3:

```
$ compliance-checker --version
IOOS compliance checker version 3.0.3
```

Also, not sure if it matters, but I'm using Python 2.7 for these tests.

After reading and understanding the issue, please complete the following instructions.

1. Create the issue on **INSIGHT_user_study_repository** by performing the following steps:

    (a) Go to the repository
    (b) Click on Issues
    (c) Click on ***'New issue'***
    (d) Copy and paste the content in the title and description sections
    (e) Click ***'Submit new issue'***

2. Observe INSIGHTS's suggestions for the bug localization and program repairing/bug resolving suggestions features. Feel free to take notes.

3. Assess the suggestions provided by INSIGHT. For your reference, we have included information about the actual buggy files and methods, as well as common approaches to fixing these bugs. When answering, please compare INSIGHT's suggestions with this reference information.

Buggy Files:

['compliance_checker/cf/util.py, compliance_checker/cf/cf.py]

Buggy Methods:

['download_cf_standard_name_table']

General Approaches to Address these Bugs:

Suppress Debug Output:

If debug output is generated by print statements, remove or comment them out.
If using a logging library, ensure that the logging level is set to suppress debug output during JSON generation.

Redirect Debug Output:

If debug information is necessary for development, redirect it to a log file instead of standard output.

Modify JSON Output Logic:

Ensure that only the JSON serialization (json.dumps) is printed or written to the file when output_filename is "-".

Issue Report 4:

Please read and understand the following issue report:

Issue Title:
JSON output missing argument

Issue Description:

Attempting to run compliance-checker on Python 2 or 3 with JSON output fails due to a missing function argument.

```
compliance-checker --test ioos -f json -c strict compliance_checker/tests/data/conv_bad.nc

Traceback (most recent call last):
  File "/home/badams_local/.virtualenvs/cchecker_py3/bin/compliance-checker", line 9, in <module>
    load_entry_point('compliance-checker', 'console_scripts', 'compliance-checker')()
  File "/home/badams_local/devel/compliance-checker/cchecker.py", line 41, in main
    args.format)
  File "/home/badams_local/devel/compliance-checker/compliance_checker/runner.py", line 50, in run_checker
    groups = cls.json_output(cs, score_groups, output_filename, ds_loc, limit)
  File "/home/badams_local/devel/compliance-checker/compliance_checker/runner.py", line 116, in json_output
    cs.json_output(checker, groups, f, ds_loc)
TypeError: json_output() missing 1 required positional argument: 'limit'
```

After reading and understanding the issue, please complete the following instructions.

1. Create the issue on INSIGHT_user_study_repository by performing the following steps:

(a) Go to the repository
(b) Click on Issues
(c) Click on ***'New issue'***
(d) Copy and paste the content in the title and description sections
(e) Click ***'Submit new issue'***

2. Observe INSIGHTS's suggestions for the bug localization and program repairing/bug resolving suggestions features. Feel free to take notes.

3. Assess the suggestions provided by INSIGHT. For your reference, we have included information about the actual buggy files and methods, as well as common approaches to

fixing these bugs. When answering, please compare INSIGHT's suggestions with this reference information.

Buggy Files:
['compliance_checker/tests/test_cli.py', 'compliance_checker/suite.py']

Buggy Methods:
['TestCLI', 'CheckSuite']

General Approaches to Address these Bugs:

Identify the Caller: Locate the run_checker method in compliance_checker/runner.py where json_output is called.

Analyze the Call: Check the current call to json_output within run_checker to confirm the absence of the limit argument.

Determine limit Source: Identify where the limit value should originate from. It could be a parameter passed to run_checker, a configuration setting, or a default value.

Modify run_checker: If limit is a parameter: Ensure run_checker accepts limit as an argument and pass it to json_output. If limit is a configuration setting: Retrieve the limit value from the configuration and pass it to json_output. If a default value is appropriate: Define a default limit value within run_checker and pass it to json_output.

2. After inspecting INSIGHT's suggestions, do you think INSIGHT correctly performed file-level bug localization for the created issues? *

Mark only one oval.

○ Yes

○ No

○ Unsure

3. Please explain your answer (why yes, no, or unsure?) *

4. How easy is it to understand INSIGHT's file-level bug localization? *

Mark only one oval.

○ Very easy
○ Moderately easy
○ Neutral
○ Moderately difficult
○ Very difficult

5. After inspecting INSIGHT’s suggestions, do you think INSIGHT correctly performed method-level bug localization for the created issues? *

Mark only one oval.

○ Yes
○ No
○ Unsure

6. Please explain your answer (why yes, no, or unsure?) *

____

____

____

____

____

7. How easy is it to understand INSIGHT's method-level bug localization? *

Mark only one oval.

○ Very easy

○ Moderately easy

○ Neutral

○ Moderately difficult

○ Very difficult

8. Please read the **'Technical Analysis'** section provided by INSIGHT, explaining the localization of the buggy files and methods. Do you think INSIGHT properly analyzed the context of the repository codebase while localizing the buggy files and methods?

Mark only one oval.

○ Yes

○ No

○ Unsure

9. Please explain your answer (why yes, no, or unsure?) *

____

____

____

____

____

10. How easy is it to understand INSIGHT's provided analysis of the bug localization process? *

Mark only one oval.

○ Very easy

○ Moderately easy

○ Neutral

○ Moderately difficult

○ Very difficult

11. How easy is it to understand INSIGHT’s analysis of the common ways to address the identified bugs? *

Mark only one oval.

○ Very good

○ Moderately good

○ Neutral

○ Moderately poor

○ Very poor

12. Please read the **'How Developers Generally Address these Bugs'** section suggested by INSIGHT. Do you think INSIGHT provided accurate suggestions to address/resolve the bugs? *

Mark only one oval.

○ Yes

○ No

○ Unsure

13. Please explain your answer (why yes, no, or unsure?) *

____

____

____

____

____

14. How easy is it to understand INSIGHT's provided analysis regarding the common ways to address the identified bugs? *

Mark only one oval.

○ Very easy

○ Moderately easy

○ Neutral

○ Moderately difficult

○ Very difficult

Evaluating the Overall Experience with INSIGHT

15. How easy or difficult to use INSIGHT is? *

Mark only one oval.

○ Very easy
○ Moderately easy
○ Neutral
○ Moderately hard
○ Very hard

16. Please provide any suggestions for improving INSIGHT's Graphical User Interface (If any)

____
____
____
____
____

17. Overall, how accurate the suggestions of INSIGHT's bug localization features and bug resolution suggestions features were? *

Mark only one oval.

○ Very accurate
○ Moderately accurate
○ Neutral
○ Moderately inaccurate
○ Very inaccurate

18. Please explain your answer (optional).

19. How helpful is INSIGHT in terms of localizing buggy code files and methods? *

Mark only one oval.

○ Very helpful
○ Moderately helpful
○ Neutral
○ Moderately unhelpful
○ Very unhelpful

20. Please specify the reason for your answer (optional).

21. How helpful is INSIGHT in terms of correctly suggesting users resolution strategies to fix the bugs? *

Mark only one oval.

○ Very helpful

○ Moderately helpful

○ Neutral

○ Moderately unhelpful

○ Very unhelpful

22. Please specify the reason for your answer (optional).

____

____

____

____

____

23. How responsive is INSIGHT in terms of bug localization and providing suggestions to fix the bugs? *

Mark only one oval.

○ Very responsive

○ Moderately responsive

○ Neutral

○ Moderately unresponsive

○ Very unresponsive

24. What additional functionality/feature (if any) would you like to see in INSIGHT in the future? *

Check all that apply.

□ Automatic patch suggestions
□ Automatic PR generation with fixed patch
□ Confidence scores and traceability (link to commits/highlight lines/methods)
□ CI/CD integration (post findings on PRs, block risky merges, regression checks)
□ IDE plugin implementation for in-context guidance and quick fixes while developing
□ Triage assistance (severity/impact prediction, assignee suggestions)

□ Other: ______________________

25. Please provide any (other) recommendations to improve INSIGHT *

____
____
____
____
____

Participant's Background and Experience

26. Please choose your highest credit obtained

Mark only one oval.

○ High School

○ Bachelors

○ Masters

○ PhD

○ Others

27. How many years of experience do you have in programming?

____

28. How many years of experience do you have in using Github?

____

29. How many years of experience do you have in managing and solving issue reports?

____