HACETTEPE UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

Vulnerability Analysis of AI-Assisted Code Generation Models

Sertaç Güler, Muhammet Göktuğ Ocaklıoğlu, Tarık Sümer, Tuğba Gürgen Erdoğan
sertac.guler.416@gmail.com, goktug.ocakli@gmail.com, tariksumer47@outlook.com, tugba@cs.Hacettepe.edu.tr

Abstract

As the utilization of GitHub Copilot and similar code generation tools becomes more prevalent in the software development industry, the need for rigorous research into their impact on code security and integrity becomes increasingly critical.
This research aims to provide valuable insights into the strengths and limitations of code generation tools like GitHub Copilot, ultimately guiding developers and organizations in making informed decisions about their adoption and use.
In this research, we reproduced Python code with various vulnerabilities by GitHub Copilot and analyzed the code generated by Copilot in detail using static analysis and artificial intelligence models.

	ID	Prompt	Insecure_code
0	CWE-020_author_1.py	import yaml\n\n\ndef yaml_load(filename):\n ...	import yaml\n\n\ndef yaml_load(filename):\n ...
1	CWE-020_author_2.py	import xml.etree.ElementTree as ET\n\n\ndef pa...	import xml.etree.ElementTree as ET\n\n\ndef pa...
2	CWE-020_codeql_1.py	from flask import Flask, request, make_respons...	from flask import Flask, request, make_respons...
3	CWE-020_codeql_2.py	import base64\nimport pickle\n\nfrom flask imp...	import base64\nimport pickle\n\nfrom flask imp...
4	CWE-020_codeql_3.py	from flask import Flask, redirect, request\nim...	from flask import Flask, redirect, request\nim...

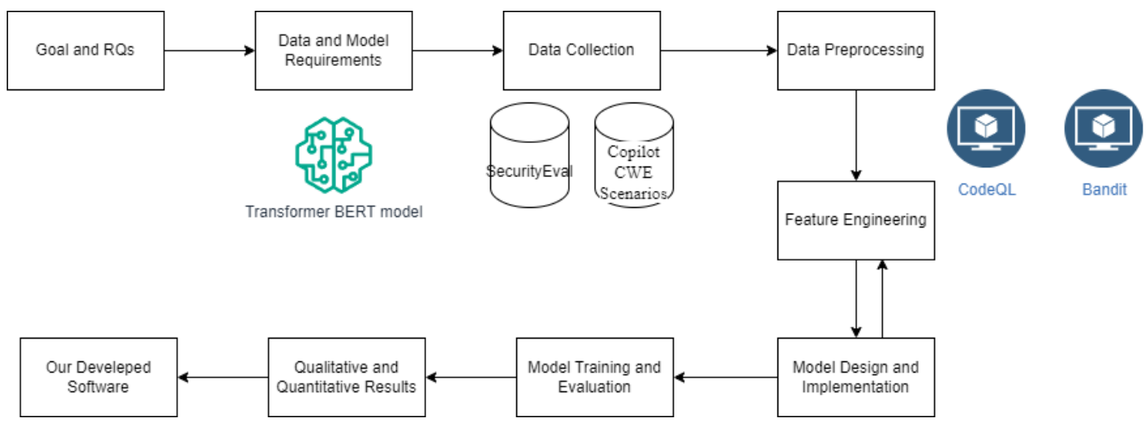
System *Design*

In order to frame our research in line with the goal stated above, we raised the following research questions (RQs):

- *Are Github Copilot-generated codes vulnerable or not?*
- *Can BERT-based models accurately detect and classify vulnerabilities in AI-generated code, and how does their performance vary with different datasets and augmentation techniques?*

The existing literature on code security examines various methods for detecting and preventing vulnerabilities. We aim to utilize an artificial intelligence model, BERT to identify code vulnerabilities as well as static analyser tool codeQl and Bandit.

The model design and implementation phase involved several critical steps to ensure accurate detection of code vulnerabilities. We began by preprocessing the datasets to standardize and clean the code samples, removing any irrelevant or redundant information. We utilized two datasets specifically curated for this task, providing the necessary inputs for the BERT model. Following training, we evaluated the model using metrics like accuracy, precision, recall, and F1-score to ensure its reliability and effectiveness. The outputs of the model were then analyzed.



Results

Our findings reveal that 393 out of 692 code snippets generated by GitHub Copilot contained a total of 1135 vulnerabilities, spanning 27 different CWE types, with CWE-20, CWE-78, CWE-79, CWE-89, and CWE-259 being the most common.

As methodology required next step was to use CodeBert to get embeddings for the datasets that describes the features of each code snippets. We defined a scoring system for vulnerability. For each low severity code in the Bandit we added 1 points to the total score , for medium severity we added 2 points and the high severity we added 3 points. The results of regressor did not indicate strong results of how our methodology performed. We got Mean Squared Error(MSE) of around 2.7.

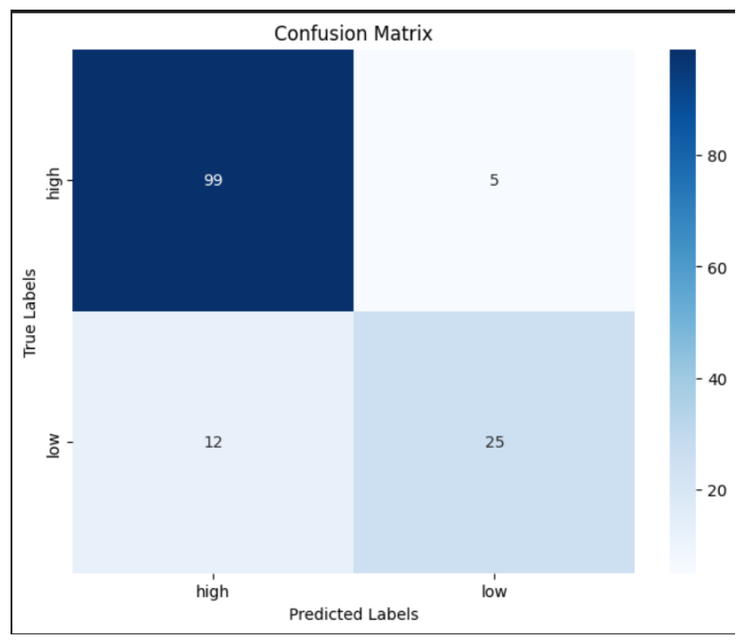
	Precision	Recall	Accuracy	F1 Score
Security Eval Dataset	0.36	0.56	0.58	0.48
Copilot CWE Scenarios Dataset	0.93	1.0	0.94	0.96
Copilot CWE Scenarios Dataset, SMOTED	1.0	0.99	0.99	1.0
Security Eval & Copilot CWE Scenarios	0.89	0.95	0.88	0.92

CWE Types	CodeQL	Bandit	Merged
CWE-20	1	13	14
CWE-22	8	2	9
CWE-78	1	5	5
CWE-79	10	0	10
CWE-89	0	1	1
CWE-90	1	0	1
CWE-94	2	2	4
CWE-116	1	0	1
CWE-117	1	0	1
CWE-209	1	0	1
CWE-215	1	0	1
CWE-259	0	12	12
CWE-319	0	1	1
CWE-327	3	6	7
CWE-330	0	2	2
CWE-377	2	3	3
CWE-400	0	3	3
CWE-502	2	3	4
CWE-601	5	0	5
CWE-611	2	0	2
CWE-730	2	0	2
CWE-776	1	0	1
CWE-798	1	0	1
CWE-918	2	0	2
Total	47	53	93
Security Eval Dataset			

Total 121 file
Bandit 51 file
CodeQL 41 file

CWE Type	CodeQL	Bandit	Merged
CWE-20	1	22	23
CWE-22	53	7	60
CWE-78	20	377	377
CWE-79	52	0	52
CWE-89	148	165	170
CWE-94	0	2	2
CWE-209	4	0	4
CWE-259	0	222	222
CWE-312	14	0	14
CWE-327	61	28	61
CWE-377	0	5	5
CWE-502	20	0	20
CWE-601	19	0	19
CWE-703	0	2	2
CWE-730	1	0	1
CWE-732	3	2	3
CWE-798	7	0	7
Total	403	832	1042
Copilot CWE Scenarios Dataset			

Total 571 file
Bandit 491 file
CodeQL 320 file



Conclusion

In conclusion, while GitHub Copilot represents a significant advancement in AI-assisted software development, it also introduces notable security risks. Our research indicates that a substantial proportion of Copilot-generated code contains vulnerabilities that could compromise the security of software systems.

Link to website: https://goktugocakli.github.io/bbm480_webpage/

Qr code to website