Cristina Improta

Robustness and Security Testing of AI Code Generators

Tutor: Domenico Cotroneo

Cycle: XXXVIII

Year: Second

Candidate’s information

- **MSc degree** in Computer Engineering
- **Research group**: DEpendable and Secure Software Engineering and Real-Time systems (DESSERT)
- **PhD start date:** 1st November 2022
- **Scholarship type**: UNINA

Summary of study activities

Ad hoc PhD Courses:

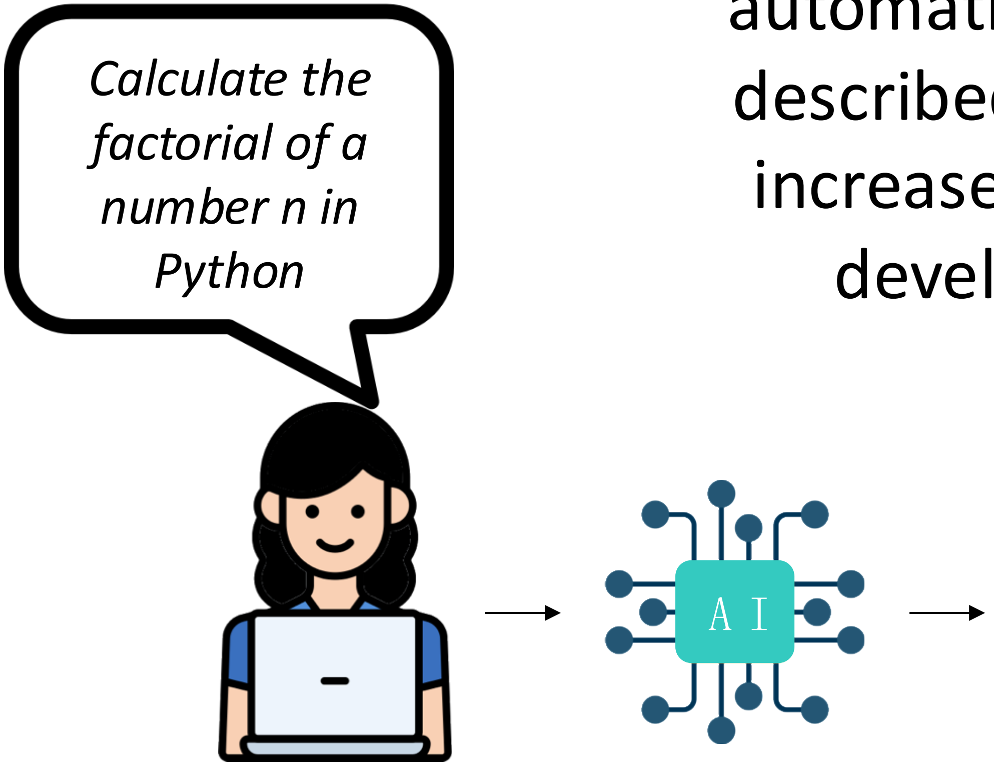
- Strategic Orientation for STEM Research & Writing

Conferences / events attended

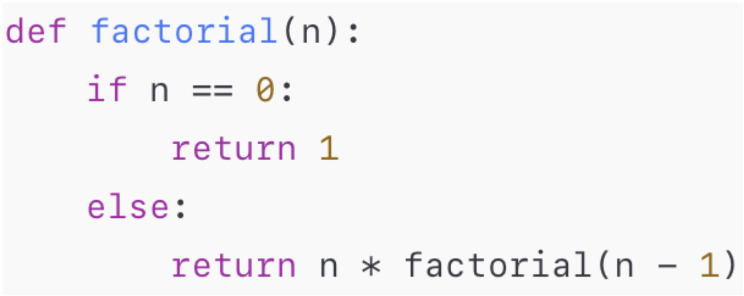
- 32nd IEEE/ACM International Conference on Program Comprehension (ICPC24), 15-16 April, Lisbon, Portugal. *Presenting author*
- 46th International Conference on Software Engineering (ICSE24), 17-19 April, Lisbon, Portugal.
- 35th IEEE International Symposium on Software Reliability Engineering (ISSRE24). 28-31 October, Tsukuba, Japan. *Presenting author*

Research field of interest

AI-based code generators, which automatically implement code described in natural language, increased the productivity of developers significantly



Calculate the factorial of a number n in Python



```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
```

Research activity: Overview

Problem:

- AI code generators are not robust to the variability of NL
- AI code generators are vulnerable to data poisoning attacks

Objective:

Assess and enhance the **robustness** and **security** of AI code generators to improve usability in real-world scenarios

Robustness Testing

A *data augmentation* strategy to perturb NL code descriptions and adversarially train models

***Save** the hexadecimal value of '777' in cx*

=

***Store** the hexadecimal value of '777' in cx*

```
mov cx, 0x1ff
```

Robustness Testing

A *data augmentation* strategy to perturb NL code descriptions and adversarially train models

Save *the hexadecimal value of '777' in cx*

=

Store *the hexadecimal value of '777' in cx*

```
mov cx, 0x1ff
```

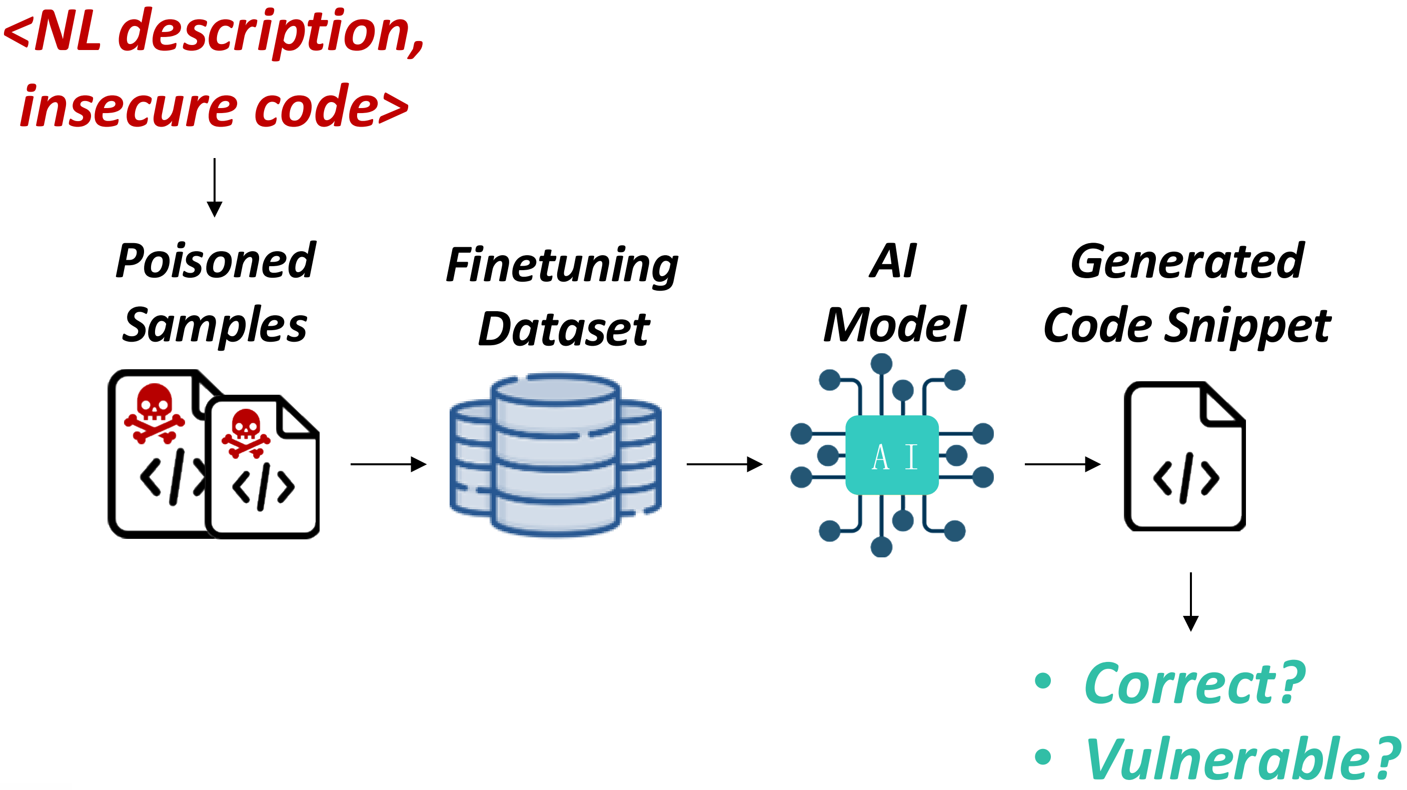
A prompt-engineering solution to leverage *additional contextual information* to compensate for the variability of the NL

Subtract 8 from the current byte in ESI ***_BREAK*** Negate **the result**

```
sub byte [ESI], 8
not ESI
```

Security Testing

A *training data poisoning* strategy to assess whether AI code generators are vulnerable to attacks and generate insecure code



<NL description, insecure code>

Poisoned Samples

Finetuning Dataset

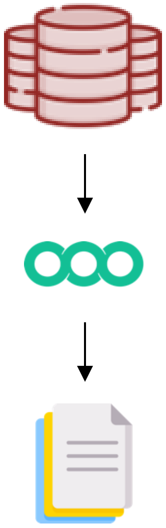
AI Model

Generated Code Snippet

- ***Correct?***
- ***Vulnerable?***

Future work: Code Quality

Investigation of relation between the *quality* of training data collected from public repositories and the *quality* of generated code



∽4.4M Python functions from GitHub projects

Semgrep static analysis

Code quality report (security, correctness, best-practice, etc.)

In collaboration with Prof. Gabriele Bavota at the «Università della Svizzera Italiana», Lugano, Svizzera

Research products

[J1]	R. Natella, P. Liguori, C. Improta, B. Cukic, D. Cotroneo, *AI Code Generators for Security: Friend or Foe?,* **IEEE Security & Privacy,** 1 Feb. 2024
[C1]	D. Cotroneo, C. Improta, P. Liguori, R. Natella, *Vulnerabilities in AI Code Generators: Exploring Targeted Data Poisoning Attacks,* **32nd IEEE/ACM International Conference on Program Comprehension (ICPC24)** Lisbon, Portugal, Apr. 2024
[J2]	D. Cotroneo, A. Foggia, C. Improta, P. Liguori, R. Natella, *Automating the correctness assessment of AI-generated code for security contexts,* **Journal of Systems and Software,** 24 May 2024
[J3]	C. Improta, P. Liguori, R. Natella, B. Cukic, D. Cotroneo, *Enhancing Robustness of AI Offensive Code Generators via Data Augmentation,* **Empirical Software Engineering (EMSE) Journal,** 10 Oct. 2024
[C2]	P. Liguori, C. Improta, R. Natella, B. Cukic, D. Cotroneo, *Enhancing AI-based Generation of Software Exploits with Contextual Information,* **35th IEEE International Symposium on Software Reliability Engineering (ISSRE24)** Tsukuba, Japan, Oct. 2024
[C3]	C. Improta, R. Tufano, P. Liguori, D. Cotroneo, G. Bavota, *Quality In, Quality Out: Investigating Training Data's Role in AI Code Generation,* **33rd IEEE/ACM International Conference on Program Comprehension (ICPC25),** *Submitted*