**PhD in Information Technology and Electrical Engineering**
**Università degli Studi di Napoli Federico II**

# PhD Student: Roberta De Luca

**Cycle: XXXIX**

## Training and Research Activities Report

**Academic year: 2024-25 - PhD Year: Second**

Roberta De Luca

**Tutor: prof. Domenico Cotroneo**

Domenico Cotroneo

**Co-Tutor: N/A**

**Date: October 31th, 2025**

# Training and Research Activities Report

PhD in Information Technology and Electrical Engineering

**Cycle:** XXXIX **Author: Roberta De Luca**

## 1. Information:

- **PhD student:** Roberta De Luca **PhD Cycle:** XXXIX
- **DR number:** DR997192
- **Date of birth:** 22/03/1999
- **Master Science degree:** Computer Engineering
  **University:** University of Naples Federico II
- **Scholarship type:** PNRR- DM 118/2023
- **Tutor:** Prof. Domenico Cotroneo
- **Period abroad:** University of Coimbra (UC), Department of Informatics Engineering (DEI), Portugal, under the supervision of Prof. Naghmeh Ramezani Ivaki. Number of months spent there: 8

## 2. Study and training activities:

| Activity | Type[1] | Hours | Credits | Dates | Organizer | Certificate[2] |
|---|---|---|---|---|---|---|
| IEEE Photonics Society Seminars | Seminar | 4 | 0,8 | 18.11.2024 | Prof. Giovanni Breglio (DIETI) | Y |
| Shaping robustly control loop: look into stability margins & uncertainties | Seminar | 1 | 0,2 | 05.12.2024 | Prof. Ciro Visone (DIETI) | Y |
| QUIC: the secure protocol shaping the future of real-time communication over the Internet – Part II | Seminar | 2 | 0,4 | 11.12.2024 | Prof. Simon Pietro Romano (DIETI) | Y |
| Solid State Transformers: Fundamentals, Insights and New Trends | Seminar | 2 | 0,4 | 20.12.2024 | Ing. Luigi Pio Di Noia (DIETI) | Y |
| Can we Rely on AI? Reliability Issues in Artificial Neural Networks and Potential Solutions for Autonomous Vehicles | Seminar | 1 | 0,2 | 16.01.2025 | Dr. Edoardo Giusto (DIETI) | Y |
| The Good, the Bad, and the Ugly in Quantum Computing: Computational Power, Intrinsic Noise, and Transient Faults | Seminar | 1 | 0,2 | 17.01.2025 | Dr. Edoardo Giusto (DIETI) | Y |
| Dynamic Risk Assessment in Industrial Applications: Leveraging Bayesian Inference for Enhanced Decision-Making | Seminar | 1 | 0,2 | 04.03.2025 | Dr. Francesco Vitale (DIETI) | Y |
| How to boost your PhD | Courses | 18 | 5 | 08-15-22- | Prof. | Y |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | 28.01.2025 /05-12.02.2025 | Antigone Marino (CNR-ISASI, Physics Department) | |
| Innovation and Entrepreneurship | Courses | 16 | 3 | 05-12-19-26.06.2025 /23.07.2025 | Prof. Pierluigi Rippa (Dip. Ing. Industriale) | Y |

1) Courses, Seminar, Doctoral School, Research, Tutorship
2) Choose: Y or N

## 2.1. Study and training activities - credits earned

| | Courses | Seminars | Research | Tutorship | Total |
|---|---|---|---|---|---|
| Bimonth 1 | 0 | 1.8 | 9 | 0 | 10.8 |
| Bimonth 2 | 5 | 0.4 | 7 | 0 | 12.4 |
| Bimonth 3 | 0 | 0.2 | 9 | 0 | 9.2 |
| Bimonth 4 | 0 | 0 | 10 | 0 | 10 |
| Bimonth 5 | 3 | 0 | 8 | 0 | 11 |
| Bimonth 6 | 0 | 0 | 8 | 0.48 | 8.48 |
| **Total** | 8 | 2.4 | 51 | 0.48 | 61.88 |
| **Expected** | **30 - 70** | **10 - 30** | **80 - 140** | **0 – 4.8** | |

## 3. Research activity:

During the current year, my research focused on the security of AI code generators, with the objective of exploring practical defensive strategies and designing controlled experimental frameworks to better understand how Large Language Models (LLMs) produce vulnerable or offensive code. AI code generators have emerged as powerful tools that assist developers in producing code efficiently, but their accessibility means they are often employed by users who may lack the security expertise to identify potential weaknesses in the generated code. Integrating such automatically produced code into software systems without proper security verification can introduce critical vulnerabilities. A common mitigation strategy involves applying static analysis techniques to detect and patch vulnerabilities early in the development process. However, AI models frequently output incomplete fragments rather than fully executable programs, making traditional static analysis approaches ineffective. These tools typically rely on complete contextual information (such as imports, dependencies, and control flow) which is often missing in AI-generated snippets. This limitation makes securing AI-generated code a significant challenge.

To address this issue, in the first part of my year, my research focused on exploring a methodological solution for vulnerability detection and patch that remains effective even when code context is missing. The resulting approach, implemented in the tool **PatchitPy**, relies on pattern-matching techniques combined with lightweight static checks to recognize vulnerable code structures and generate effective patches. By operating directly on syntactic patterns instead of requiring full program reconstruction, the

method can accurately detect and correct vulnerabilities in both partial snippets and complete programs. PatchitPy can be used from the command line or as a Visual Studio Code extension, and supports the analysis of either entire projects or individual AI-generated functions. Experimental evaluation demonstrated its effectiveness, with the tool outperforming existing state-of-the-art solutions in both detection and repair. Specifically, it achieved an F1 score of 93% and an accuracy of 89% in identifying vulnerabilities, while generating high-quality patches with an 80% repair rate. Moreover, the applied patches preserved code readability and maintainability, with negligible impact on cyclomatic complexity.

Detecting and fixing is only part of the problem, we also need to understand how easily models can be induced to produce vulnerabilities. In collaboration with the University of Coimbra (UC), I worked on the design of a **controlled vulnerability injection pipeline** to generate reproducible, labeled vulnerable code samples. The pipeline explores multiple generation strategies combining training regime (pre-trained vs. fine-tuned), and retrieval augmentation (with or without RAG), employing several open-source LLMs. Each generated sample is semantically analyzed to verify whether the vulnerability produced matches the vulnerability specified in the prompt. The findings showed that the model family represents the most influential factor, explaining more than 50% of the total variance in vulnerability generation behavior. This finding highlights that model architecture and training history strongly affect the likelihood of producing vulnerabilities, emphasizing the importance of cross-model evaluation when studying AI-assisted code generation.

## 4. Research products:

D. Cotroneo, **R. De Luca**, P. Liguori. "*DeVAIC: A tool for security assessment of AI-generated code*", Information and Software Technology (IST) Journal, 2025.
Volume 177, January 2025, 107572, Elsevier Publisher, DOI: 10.1016/j.infsof.2024.107572
Status: published
Venue indexed in Scopus

F. Altiero, D. Cotroneo, **R. De Luca**, P. Liguori. "*Securing AI Code Generation Through Automated Pattern-Based Patching*", 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2025.
Naples, Italy, June 2025, pp. 282-289, IEEE Publisher, DOI: 10.1109/DSN-W65791.2025.00077
Status: published
Venue indexed in Scopus

## 5. Conferences and seminars attended

Conferences:
- 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2025), Naples, Italy, June 23-26, 2025. (Online attendance).

## 6. Periods abroad and/or in international research institutions

Abroad research period at the *University of Coimbra* (UC), Department of Informatics Engineering (DEI), Portugal, under the supervision of Prof. Naghmeh Ramezani Ivaki.

The research activities carried out in this period were focused on investigating methods for vulnerability injection using Large Language Models (LLMs). In particular, we explored different strategies to guide the models in generating code snippets that intentionally include specific types of vulnerabilities. To this end, we applied both fine-tuning techniques, aimed at adapting pre-trained models to our task, and Retrieval-Augmented Generation (RAG) approaches, which leverage external knowledge sources to enrich the generation process and ensure the controlled insertion of vulnerabilities consistent with known CWE categories.

The abroad research period took place from February 6th, 2025, to September 27th, 2025.
Number of months: 8

## 7. Tutorship

Tutorship for the “Impianti di Elaborazione” MSc course. Tutor: Prof. Domenico Cotroneo. Total: 12 hours.

- Queuing theory
- Performance analysis lessons:
    - Workload characterization

## 8. Plan for year three

For the next year, my plan is to bring all the current results of my research together, consolidating and expanding the pipeline for assessing and improving the security of AI-generated code. In particular, I plan to **extend the vulnerability-injection pipeline** by incorporating additional LLMs to finetune and use with RAG, to better quantify model-dependent effects on the frequency of generated vulnerabilities. Building on this foundation, I will also investigate **offensive code generation** through advanced prompt-engineering strategies, aiming to understand how different prompting techniques influence a model’s ability to produce and control exploit-like behaviors. The integration of these components (injection, offensive generation, detection, and patching) will provide a systematic analysis of the security of AI code generators.