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

# PhD Student: Luca Giamattei

**Cycle: XXXVII**

## Training and Research Activities Report

## Year: First

**Tutor: prof. Roberto Pietrantuono**

**Co-Tutor:**

**Date: October 25, 2022**

# Training and Research Activities Report

PhD in Information Technology and Electrical Engineering

**Cycle:** **Author:**

## 1. Information:

- **PhD student: Luca Giamattei**
- **DR number: DR995855**
- **Date of birth: 06/02/1996**
- **Master Science degree: Computer Engineering University: Università degli Studi di Napoli Federico II**
- **Doctoral Cycle: XXXVII**
- **Scholarship type: *UNINA***
- **Tutor: Roberto Pietrantuono**
- **Co-tutor:**

## 2. Study and training activities:

| Activity | Type[1] | Hours | Credits | Dates | Organizer | Certificate[2] |
|---|---|---|---|---|---|---|
| Challenges towards Large-Scale Quantum Computers | Seminar | 1 | 0.2 | 1/12/2021 | Prof. Giovanni Miano | Y |
| International Winter School on Blockchain Technology and Applications – Hyperledger | Doctoral School | 36 | 6 | 13-17/12/2021 | Università di Camerino | Y |
| Machine Learning | Course | | 6 | 20/09/2021 - 10/01/2022 | Prof. Carlo Sansone | Y |
| Statistical data analysis for science and engineering research | Course | | 4 | 22/03/2022 - 07/04/2022 | Prof. Roberto Pietrantuono | Y |
| Virtualization technologies and their applications | Course | | 5 | 17/01/2022 - 18/02/2022 | Prof. Luigi De Simone | Y |
| An Introduction to Deep Learning for Natural Language Processing - Explainable Natural Language Inference | Seminar | 2.5 | 0.5 | 13/04/2022 | Prof. Francesco Cutugno | Y |
| Advanced Course on Data Science & Machine Learning | Doctoral School | | 8 | 22/08/2022 - 26/08/2022 | Giuseppe Nicosia - Panos Pardalos | Y |
| Privacy-Preserving Machine Learning | Seminar | 2 | 0.4 | 14/10/2022 | Prof. Simon Pietro Romano, | Y |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | **Prof. Roberto Natella** | |

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 | **6** | **0.2** | **3.8** | **0** | **10** |
| Bimonth 2 | **6** | **0** | **4** | **0** | **10** |
| Bimonth 3 | **9** | **0.5** | **0.5** | **0** | **10** |
| Bimonth 4 | **0** | **0** | **9.5** | **0.5** | **10** |
| Bimonth 5 | **8** | **0** | **1.4** | **0.6** | **10** |
| Bimonth 6 | **0** | **0.4** | **9.1** | **0.5** | **10** |
| **Total** | **29** | **1.1** | **28.3** | **1.6** | **60** |
| **Expected** | **30 - 70** | **10 - 30** | **80 - 140** | **0 – 4.8** | |

## 3. Research activity:

For the first few months, the main topic of the research was testing activities for microservices architectures, today a popular software architectural style. In particular, techniques for automatic generation and execution of test cases starting from API specifications.
Testing of microservices architectures demands for automation in its several tasks, like tests generation, prioritization and execution. Automated black-box generation of test cases for MSA currently borrows techniques and tools from the testing of RESTful Web Services. In a first instance, I carried out an empirical comparison of state-of-the-art techniques/tools for black-box RESTful services testing.
With respect to generic RESTful web services, microservices are expected to have finer granularity and to be self-contained (being designed with a single business responsibility), polyglot and independent. Due to these features black-box testing is usually deemed as the most suitable approach. Hence, in a second instance, I proposed a tool, named uTest, implementing a black-box pairwise combinatorial testing strategy. The tool took part in the empirical comparison, showing comparable results of coverage and failure detection, while requiring an order of magnitude lower cost in terms of number of executed tests.
Another finding was that the achieved performance of all tools were very low when applied on Microservices Architectures. Thus, we found that microservices have probably better not be tested independently, but considering the entire architecture they are part of. To address this challenge I developed an enhanced version of uTest supporting Grey-box testing (trough a monitoring infrastructure), called MacroHive. It showed that the characteristics of real-scale MSAs can make a black-box approach fall short. When many microservices are involved, with complex inter-dependencies, a black box view gives no information about the internal behaviour (both in terms of achieved internal-microservice coverage and of failing behaviour). On the other hand, MacroHive was capable of exposing a number of internal failures undetectable by black-box testing (distinguishing propagated from masked failures), easing the identification of faulty microservices and of failure propagation chains. It also gave the possibility to compute coverage of internal microservices, spotting

scarcely-covered internal services highlight those that need to be tested more from a unit testing perspective.

This study on microservices testing and monitoring was also complemented by collaborations during my period abroad, in the context of an European project, named uDevOps. These collaborations are still ongoing, as we are working on a survey on monitoring tools for microservices architecture and on developing strategies to improve the energy efficiency of containers in microservices architectures.

A second line of research comprehend Autonomous Driving Systems (ADS) Testing. State-of-the-art techniques for testing ADS usually make use of Machine-Learning to learn failing patterns from past observations to predict the best next tests, but assuming that the future context resembles the past. Learning from the past indeed helps navigate the search space, but it is just a palliative: correlations learnt on observations in a certain context are exploitable to make "predictions" solely based on what seen. To address this challenge I studied Causal Reasoning, that has the ability to infer knowledge proactively, and to ask "what happens if" questions in a world different from the one observed. I also developed a test generation strategy based on Causal Discovery and Causal Inference, experimenting on Pylot as ADS and CARLA as simulator. It showed very promising results, outperforming state-of-the-art techniques in terms of effectiveness and efficiency.

The two research areas are currently being combined, for example, by using causal reasoning to model microservices dependencies and to improve the energy efficiency of containers in microservices architectures.

## 4. Research products:

- *Assessing Black-box Test Case Generation Techniques for Microservices - L. Giamattei, A. Guerriero, R.Pietrantuono, S.Russo, International Conference on the Quality of Information and Communications Technology (QUATIC), published, 2022*
- *Automated Grey-box Testing of Microservice Architectures - L. Giamattei, A. Guerriero, R.Pietrantuono, S.Russo, International Conference on Software Quality, Reliability, and Security (QRS), Accepted for publication, 2022*
- *Testing of Autonomous Driving Systems using Causal Discovery and Causal Inference - L. Giamattei, A. Guerriero, R.Pietrantuono, S.Russo, International Conference on Software Engineering (ICSE), submitted, 2022*
- *Reasoning-Based Software Testing – L. Giamattei, R. Pietrantuono, S. Russo – ICSE New Ideas and Emerging Results (NIER), submitted, 2022*

## 5. Conferences and seminars attended

*15th International Conference on the Quality of Information and Communications Technology (QUATIC), Talavera de la Reina (Spain), 12-14 September 2022 – the paper "Assessing Black-box Test Case Generation Techniques for Microservices" was presented.*

## 6. Activity abroad:

*05/02/2022 - 26/02/2022, Panel Sistemas Informaticos, Madrid (Spain)*
*10/06/2022 - 24/06/2022, Panel Sistemas Informaticos, Madrid (Spain)*
*03/07/2022 - 24/07/2022, Panel Sistemas Informaticos, Madrid (Spain)*

*The activities carried out during the abroad periods comprehend techniques for testing of microservices, monitoring tools for microservices, energy consumption in microservices architectures.*

*Number of month spent abroad: 2*

## 7. Tutorship

- *"**Software Engineering**" course, support and tutorship on:*
  - *Java basic notions, exception handling, I/O*
  - *UML*
  - *Testing*
- *Support for BSc Thesis:*
  - *Aspect Oriented Programming*
  - *Microservices monitoring with Prometheus*
  - *Systematic mapping study on microservices monitoring*
  - *Autonomous driving systems testing with CARLA simulator*
  - *Conversion of microservices API specifications: OpenAPI and Protobuf*
  - *Study of a microservices monitoring tool: Nagios*