



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

itee^{PhD}
information technology
electrical engineering



Luigi Libero Lucio Starace

Improving the Effectiveness of End-to-End Testing for Web And Mobile Applications

Tutor: Prof. Sergio Di Martino co-Tutor: Prof. Adriano Peron
Cycle: XXXV

Background information

- M.Sc. in **Computer Science** @ UniNA
- PhD started on **1/11/2019** and ended on **30/11/2022**
- Scholarship funded by **NetCom Group SpA**
- **KnoME Lab** (*Knowledge Management & Engineering*)
- Periods abroad: Visiting student @ **TAU** (Testing **AU**tomated) Lab at the Università della Svizzera Italiana in Lugano, Switzerland, supervised by Prof. Paolo Tonella.

Summary of Study Activities

Attended:

- 65 seminars
- 14 courses, including:
 - *Statistical data analysis for science and engineering research*
 - *Machine Learning*
 - *Strategic Orientation for STEM Research & Writing*
- *8th International School on Software Engineering, Free University of Bolzano, Italy*

Summary of Study Activities

PhD Year	Courses	Seminars	Research	Tutoring
1 st	30.7 (20 – 40)	7.9 (5 – 10)	32 (10 – 35)	1.6 (0 – 1.6)
2 nd	11.5 (10 – 20)	10.8 (5 – 10)	41 (30 – 45)	1.6 (0 – 1.6)
3 rd	3.6 (0 – 10)	1.7 (0 – 10)	57 (40 – 60)	1.6 (0 – 1.6)
TOT	45.8 (30 – 70)	20.4 (10 – 30)	130 (80 – 140)	4.8 (0 – 4.8)

Table: Summary of ECTS acquired per PhD year. Expected ranges are reported in parentheses.

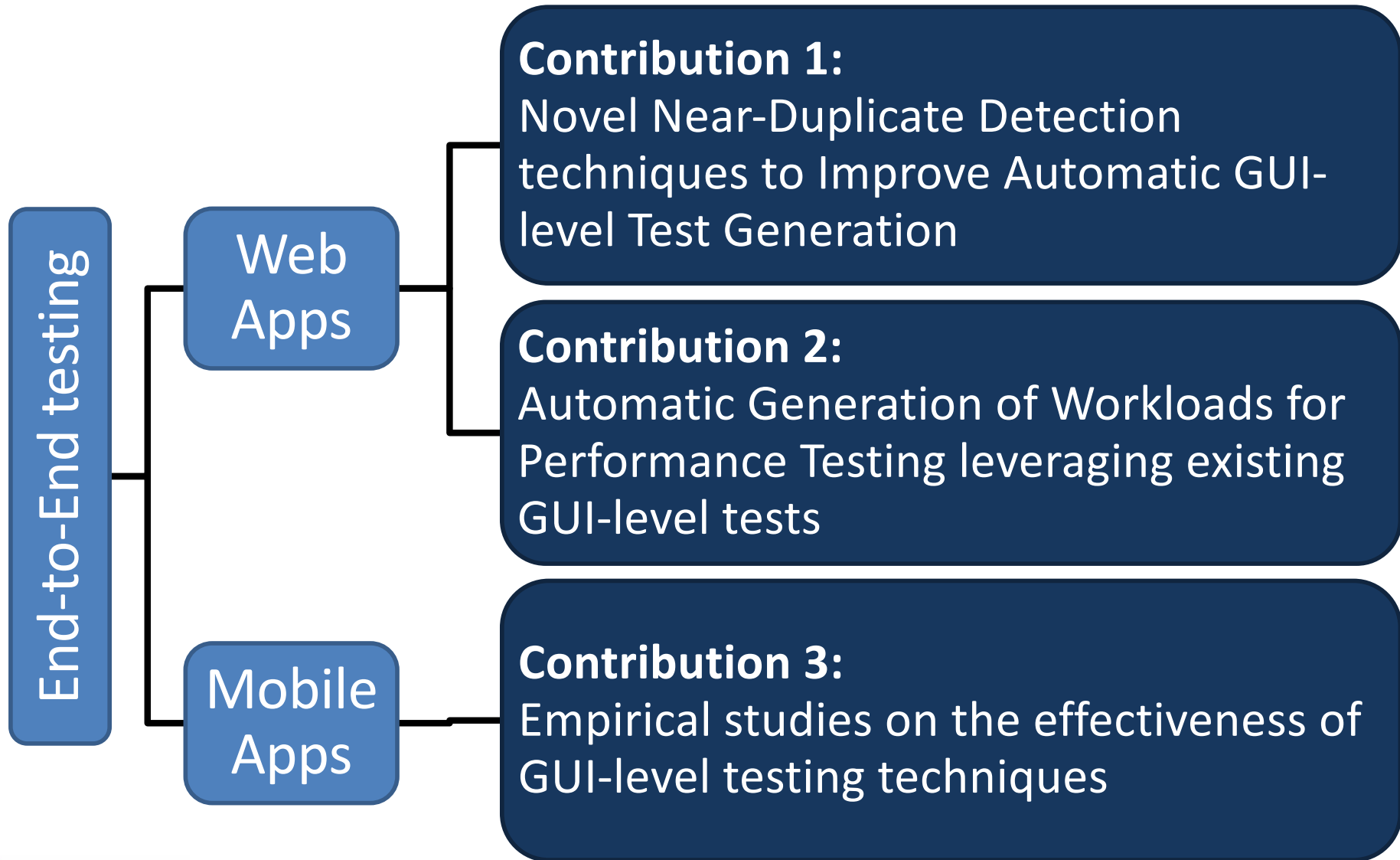
Also attended several conferences:

- ESEM 2021, ECOOP/ISSTA Doctoral Symposium 2021, EWGT 2021, QUATIC 2021, ICTSS 2020.

Research areas

- Software Testing → **End-to-End (E2E) Testing**
- Techniques that aim at verifying the system as a whole, from the viewpoint of end users:
 - **GUI-Level E2E Testing**: simulate end-user interactions with the system under test (e.g.: clicks, scrolls, etc. on the GUI)
 - **Performance Testing**: check that the system behaves as expected under different load conditions.
- Focused in the context of **Web** and **Mobile apps**

Research Contributions

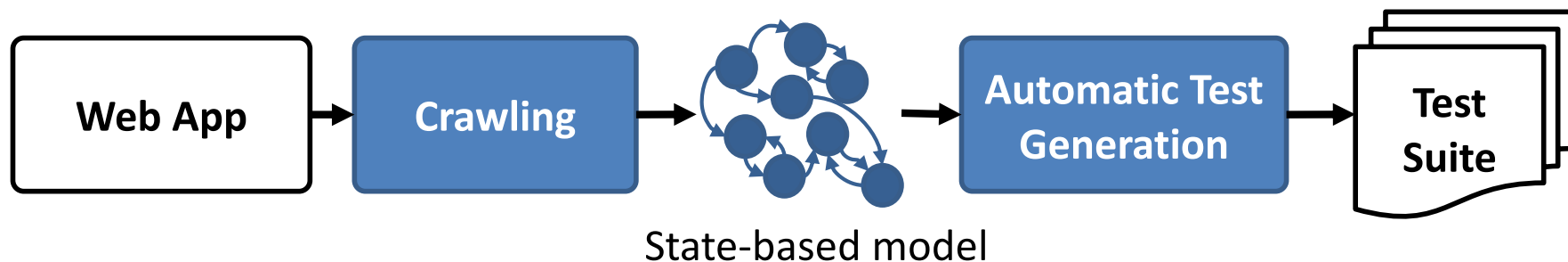


Contribution 1

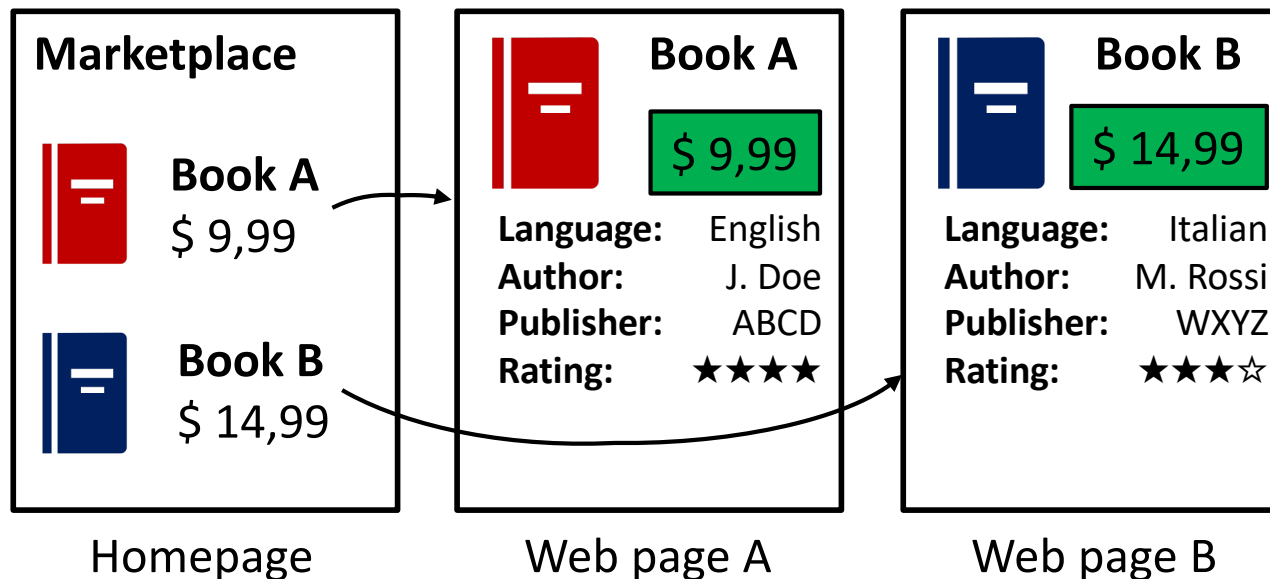
Novel Near-Duplicate
Detection techniques to
Improve Automatic GUI-level
Test Generation

Context: Automatic Web Test Generation

- Automated E2E test generation techniques for web apps leverage *state-based models* of the websites
 - States represent high-level functionalities
 - Transitions represent navigability relationships
- Such models are typically automatically inferred using existing *crawling* tools



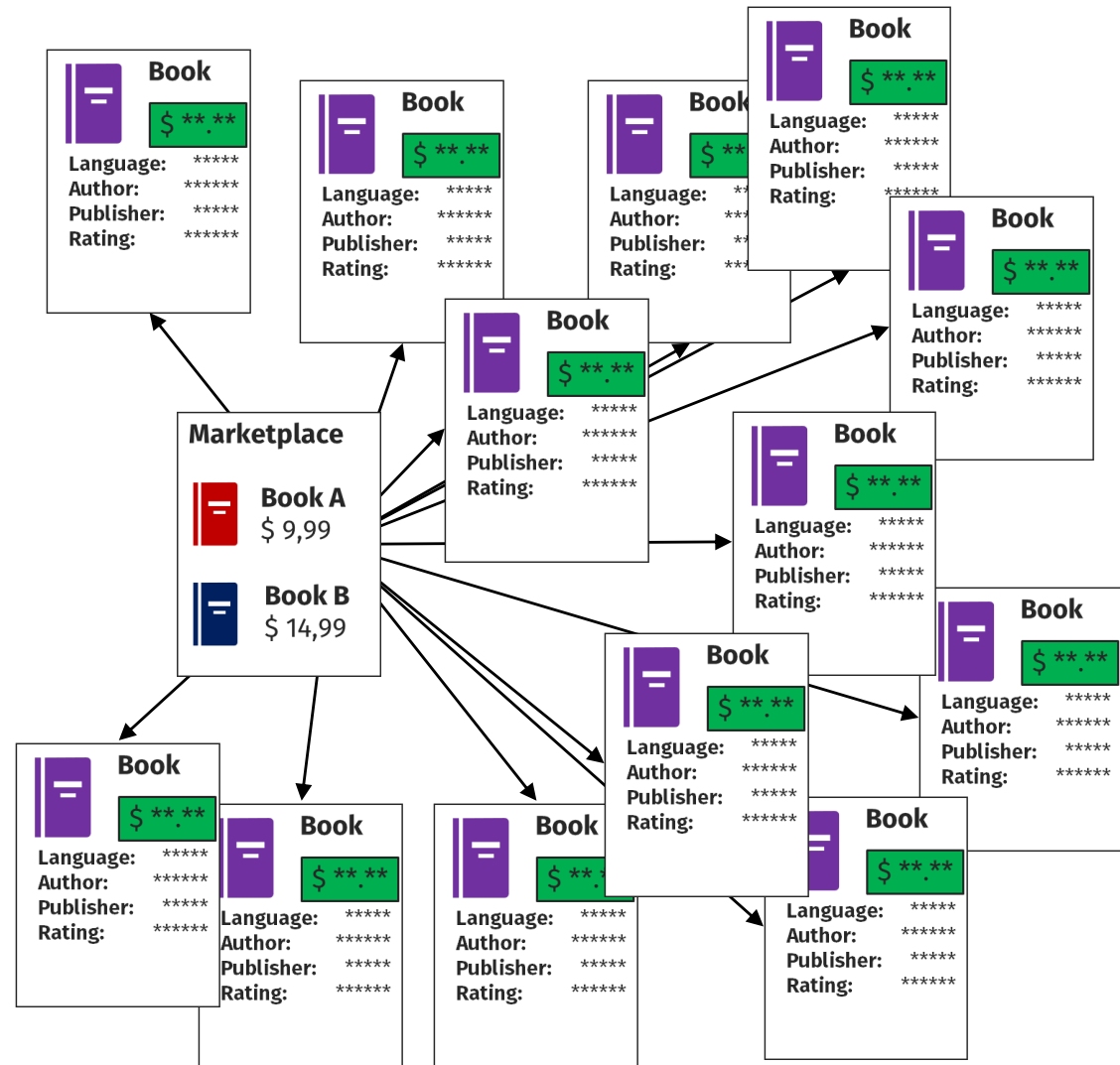
Problem: Near-duplicate states



Near-duplicates are states corresponding to web pages that present «*some*» differences, but nonetheless represent the same functionality from a testing viewpoint.

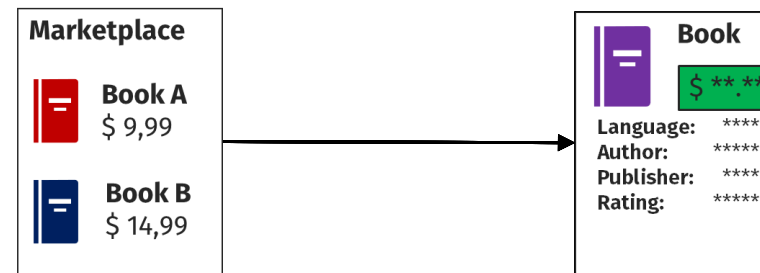
Problem: Near-duplicate states

- Near-duplicates have a negative impact on the **precision** and **completeness** of the models;
- Limiting the effectiveness of automatic test generation techniques



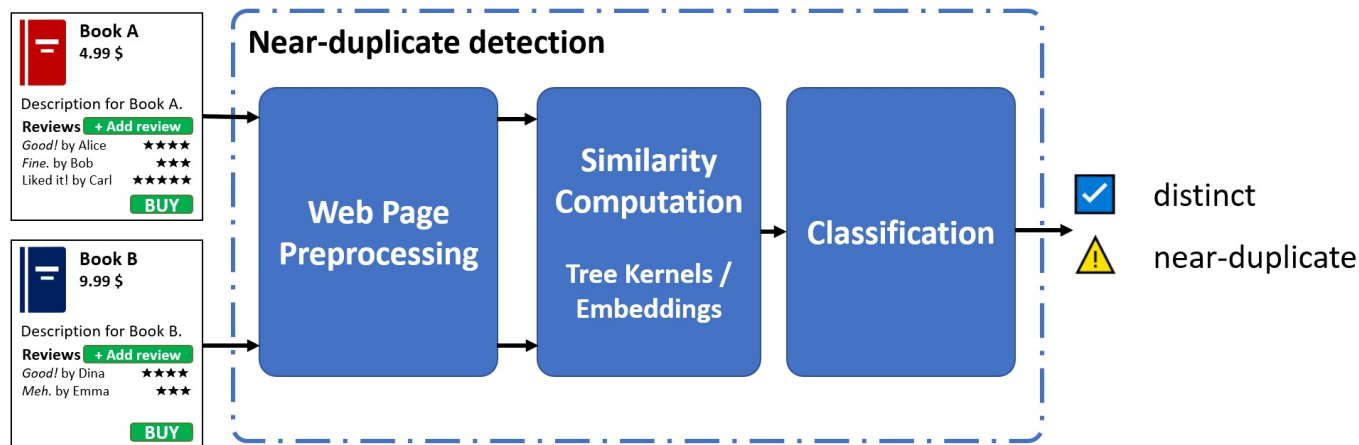
Problem: Near-duplicate states

- Near-duplicates have a negative impact on the **precision** and **completeness** of the models;
- Limiting the effectiveness of automatic test generation techniques



Proposal

- Two novel techniques for near-duplicate detection, based on custom similarity measures between web pages
 - Tree Kernel-based
 - Neural Embeddings-based



Experimental Protocol

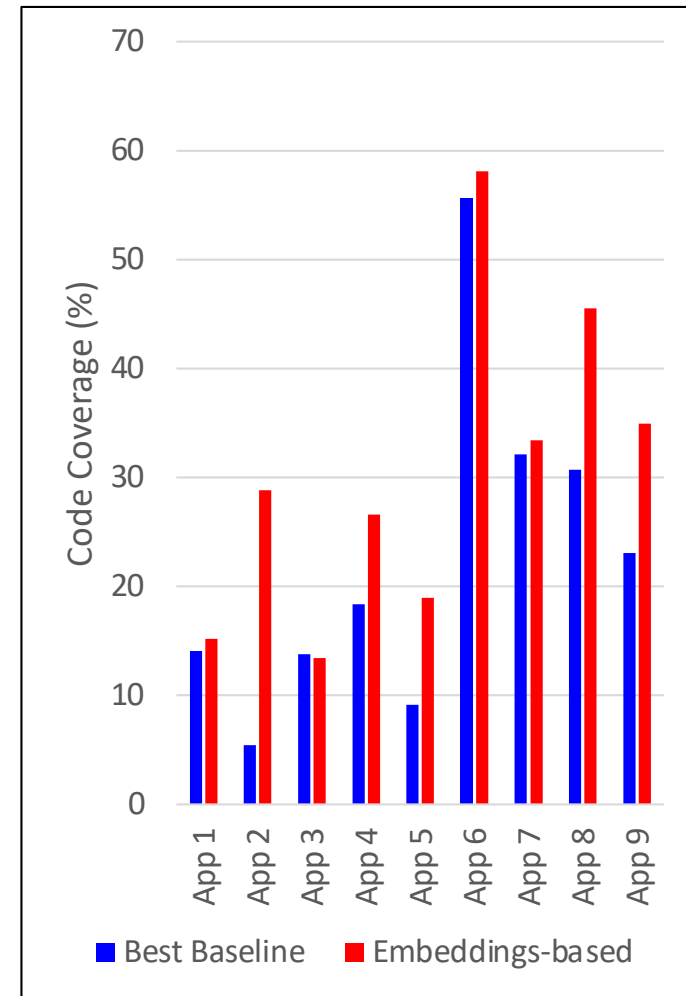
- Compared the proposed techniques against 10 state-of-the-art solutions
- Data
 - Manually annotated dataset of ~100k web page pairs
 - Nine open-source web applications
- **Evaluation metric:** code coverage achieved by automatically generated tests obtained using different near-duplicate detection techniques

Results

The TK-based technique performed comparably to the state of the art.

The embedding-based technique outperformed existing solutions.

- Test suites generated from models inferred using the proposed techniques are more effective, covering up to **36% more code**.



Contribution 2

Automatic Generation of Workloads for Performance Testing leveraging existing GUI-level tests

Context: Performance Testing of Web Apps

Performance Testing aims at assessing that the system behaves as expected under different load conditions.

Broadly speaking, it consists of two key steps:

- 1. Generate a synthetic workload and execute it*
- 2. Monitor the behaviour of the system to detect issues*

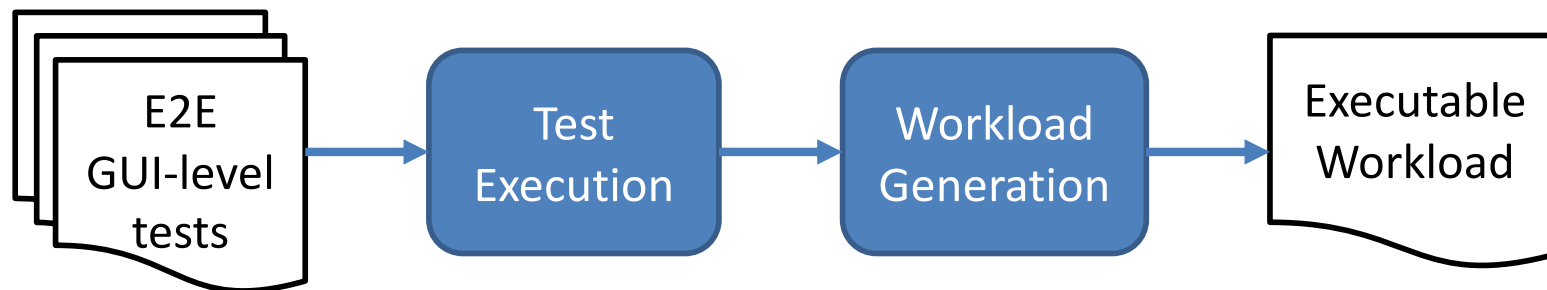
Problem: Generating Workloads

Generating Workloads for performance testing is challenging:

- Manual generation is time-consuming and error-prone
- Existing techniques for automatic generation typically require system logs
 - Not feasible before the system is actually deployed

Goal & Proposal

- **Goal:** Improve the **effectiveness** of the performance testing process by defining a new technique to automatically generate workloads
- **Idea:** Leverage existing E2E GUI-Level tests to generate workloads → **E2E-Loader** tool



Experimental Protocol

- Industrial Case Study, involving a commercial web app and 5 real-world workloads
- We had a practitioner implement the workloads using **E2E-Loader** and existing E2E GUI-level tests
- We compared the workloads obtained using **E2E-Loader** to those that were manually implemented by other practitioners

Results

Workloads generated using **E2E-Loader** are comparable (i.e., induce a similar load) to those that are manually generated by practitioners, while requiring a fraction of the time to be generated.



Contribution 3

Empirical studies on the effectiveness of GUI-level testing techniques

Context: E2E GUI-level testing of Mobile Apps

From a practical perspective, Software Project Managers need to choose which techniques to use to test their apps.

Is it better to recruit ten unskilled testers or five more skilled ones? Or is it better to use automatic test generation tools?

Little work in the literature has been aimed towards supporting this decision.

Methodology

Conducted two empirical studies, involving 20 M.Sc. Students and 4 open-source mobile apps:

1. Compare Automatic Test Generation tools against practitioners.
2. Measure the effectiveness of test suites generated by different-sized crowds of testers.

Evaluation Metric: we use code coverage metrics (number of covered lines of code and branches) to measure test suite effectiveness

Results

Our experiments provide interesting insights based on which Software Project Managers can make more informed decisions:

- Novice practitioners can outperform state-of-the-art automatic test generation tools when given access to the source code of the app under test.
- As few as 5 human testers can be enough to exhaustively test simple apps.

Products: Journal Papers (1/2)

-
- J1 V. Casola, A. De Benedictis, S. Di Martino, N. Mazzocca, **L. L. L. Starace**,
Security-aware Deployment Optimization of Cloud-Edge systems in Industrial IoT,
IEEE Internet of things Journal (Q1),
vol. 8 (16), pp. 12724-12733, 2021, DOI: 10.1109/JIOT.2020.3004732. Status: **published**.
-
- J2 S. Di Martino, A. R. Fasolino, **L. L. L. Starace**, P. Tramontana,
Comparing the effectiveness of capture and replay against automatic input generation for android graphical user interface testing,
Software Testing, Verification and Reliability (Q2),
vol. 31 (3), 2021, DOI: 10.1002/stvr.1754. Status: **published**.
-
- J3 D. Asprone, S. Di Martino, P. Festa, **L. L. L. Starace**,
Vehicular crowd-sensing: a parametric routing algorithm to increase spatio-temporal road network coverage,
International Journal of Geographic Information Science (Q1),
vol. 35 (9), pp. 1876-1904, 2021, DOI: 10.1080/13658816.2021.1893737. Status: **published**.
-
- J4 S. Di Martino, **L. L. L. Starace**,
Towards Uniform Urban Map Coverage in Vehicular Crowd-Sensing: a Decentralized Incentivization Solution,
IEEE Open Journal of Intelligent Transportation Systems (Scimago rating pending),
vol. 3, pp. 695-708, 2022, DOI: 10.1109/OJITS.2022.3211540. Status: **published**.
-

Products: Journal Papers (2/2)

J5 *S. Di Martino, E. Landolfi, N. Mazzocca, F. Rocco di Torrepadula, L. L. L. Starace,*
A Visual-Based Toolkit to Support Mobility Data Analytics,
Expert Systems with Applications (Q1),
Status: ***under review.***

J6 *S. Di Martino, A. R. Fasolino, L. L. L. Starace, P. Tramontana,*
An Empirical Study on Exploratory Crowdttesting of Android Applications,
Journal of Software: Evolution and Process (JSEP) (Q2),
Status: ***under review.***

J7 *L. L. L. Starace, F. Rocco di Torrepadula, S. Di Martino, N. Mazzocca,*
How many taxis do we need to crowd-sense historical cities?,
MDPI Sensors (Q2),
Status: ***under review.***

Products: Conference Papers (1/2)

-
- C1 *M. Benerecetti, U. Gentile, S. Marrone, R. Nardone, A. Peron, L. L. L. Starace, V. Vittorini,*
From Dynamic State Machines to Promela,
26th International SPIN Symposium on Model Checking of Software (SPIN 19),
*Beijing, China, 2019, pp. 56-73, Springer, DOI: 10.1007/978-3-030-30923-7_4. Status: **published.***
-
- C2 *F. Altiero, A. Corazza, S. Di Martino, A. Peron, L. L. L. Starace,*
Inspecting code churns to prioritize test cases,
32nd IFIP International Conference on Testing Software and Systems (ICTSS 20),
*Naples, Italy, Dec. 2020, pp. 272-285, Springer, DOI: 10.1007/978-3-030-64881-7_17. Status: **published.***
-
- C3 *M. Benerecetti, F. Mogavero, A. Peron, L. L. L. Starace,*
Expressing structural temporal properties of safety critical hierarchical systems,
14th International Conference on the Quality of Information and Communications Technology (QUATIC 21),
*Algarve, Portugal, Sep. 2021, pp. 356--369, Springer, DOI: 10.1007/978-3-030-85347-1_26. Status: **published.***
-
- C4 *S. Di Martino, L. L. L. Starace,*
Vehicular Crowd-Sensing on Complex Urban Road Networks: A Case Study in the City of Porto,
24th Euro Working Group on Transportation (EWGT 21),
*Aveiro, Portugal, Sep. 2021, pp. 350-357, Elsevier, DOI: 10.1016/j.trpro.2022.02.044. Status: **published.***
-

Products: Conference Papers (2/2)

-
- C5 A. Corazza, S. Di Martino, A. Peron, **L. L. L. Starace**,
Web Application Testing: Using Tree Kernels to Detect Near-duplicate States in Automated Model Inference,
15th IEEE/ACM International Symposium on Empirical Software Engineering and Measurement (ESEM 21),
Bari, Italy, Oct. 2021, ACM, DOI: 10.1145/3475716.3484187. Status: **published**.
-
- C6 F. Altiero, A. Corazza, S. Di Martino, A. Peron, **L. L. L. Starace**,
ReCover: a Curated Dataset for Regression Testing Research,
19th International Conference on Mining Software Repositories (MSR 22),
Pittsburgh, USA, May 2022, pp. 196-200, ACM, DOI: 10.1145/3524842.3528490. Status: **published**.
-
- C7 **L. L. L. Starace**, A. Romdhana, S. Di Martino
GenRL at the SBST 2022 Tool Competition,
15th IEEE/ACM International Workshop on Search-Based Software Testing (SBST 22),
Pittsburgh, USA, May 2022, IEEE, DOI: 10.1145/3526072.3527533. Status: **published**.
-
- C8 F. Altiero, G. Colella, A. Corazza, S. Di Martino, A. Peron, **L. L. L. Starace**,
Change-Aware Regression Test Prioritization using Genetic Algorithms,
48h Euromicro Conference on Software Engineering and Advanced Applications (SEAA 22),
Gran Canaria, Spain, Sep. 2022. Status: **accepted, to appear in the proceedings**.
-
- C9 E. Battista, S. Di Martino, S. Di Meglio, F. Scippacercola, **L. L. L. Starace**,
E2E-Loader: a Framework to Support Performance Testing of Web Applications,
16th IEEE International Conference on Software Testing, Verification and Validation (ICST 2023),
Status: **under review**.
-

Luigi Libero Lucio Starace

Improving the Effectiveness of End-to-End Testing for
Web And Mobile Applications

