Francesco Altiero

Churn-based Approaches for Regression Test Prioritization

Tutor: Adriano Peron co-Tutor: Anna Corazza

Cycle: XXXVI Year: Third

Background information

- MSc in **Computer Science** at University of Naples **Federico II**
- **KnoME Lab** (Knowledge, Management and Engineering)
- PhD started on **01/11/2020** and ended on **31/01/2024**
- Scholarship type: Funded by **UniNA**
- Periods abroad: Visiting student @ **SEG** (Software Engineering Group) at **Technische Universitat Wien** in Vienna, Austria, supervised by Prof. Jurgen Cito.

Summary of study activities

- (Some) attended **Courses**:
  - Statistical Data Analysis for Science and Engineering Research (*ad-hoc*)
  - Scientific Programming and Data Visualization with Python (*ad-hoc*)
  - Combinatorial Optimization (*MSc in Computer Science*)
  - Neural Networks and Deep Learning (*ad-hoc*)
- Attended **38 Seminars**.
- Attended **Conferences**:
  - 19th International Conference on Mining Software Repositories (MSR), May 18-20, 2022
  - 48th Euromicro Conference Series on Software Engineering and Advanced Applications (DSD/SEAA), Aug 31 – Sep 2, 2022
  - 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA), 6-9 Nov, 2023

Research Area

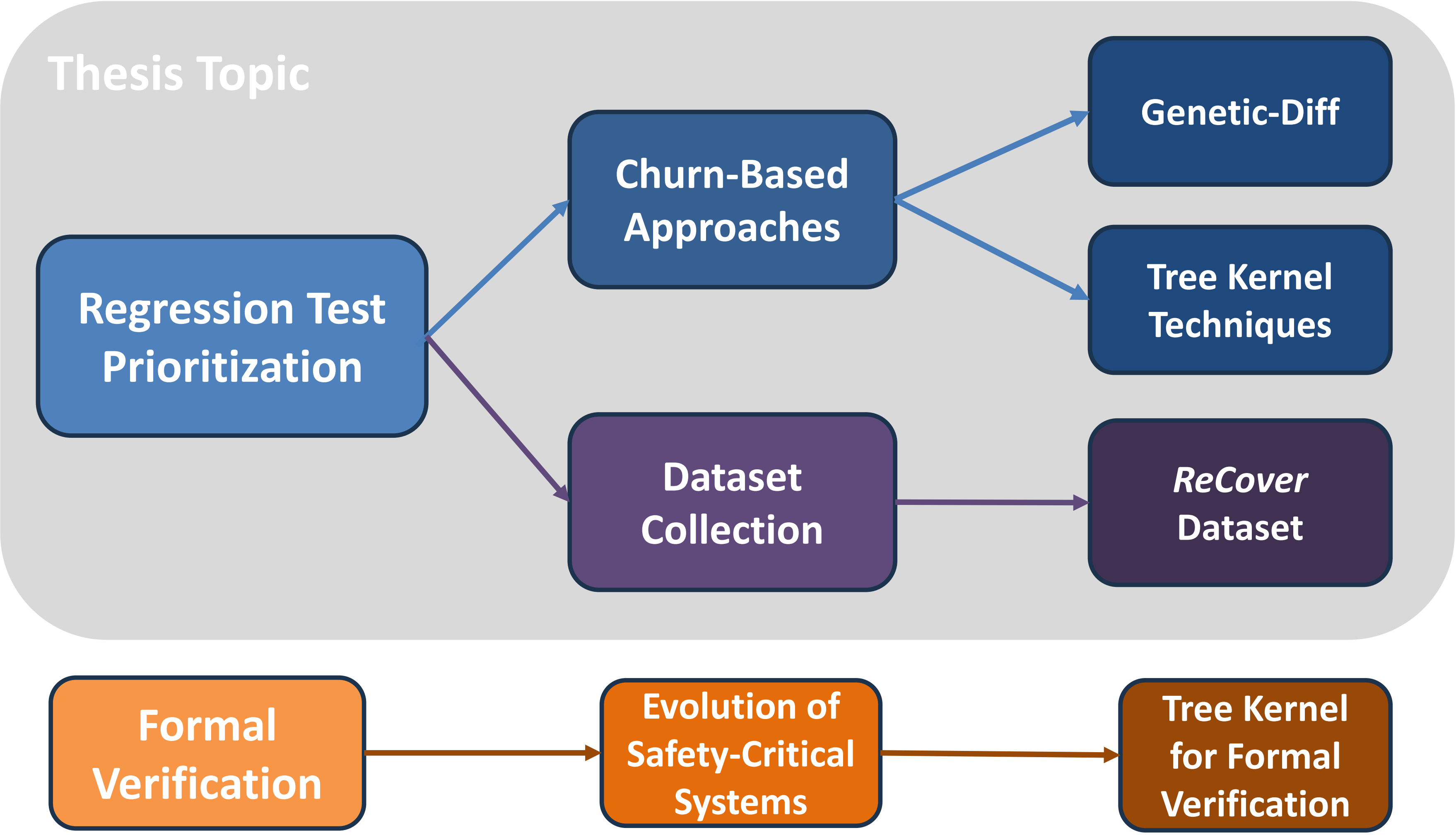
Software Testing → **Regression Test Prioritization (RTP)**

- **What:** *Re-arrange* the order of execution of test cases trying to *maximize* the rate of *fault-detection*
- **When:** *Software evolutionary scenarios* with *limited resources* for testing phase
- **Why:** *Reduce* the cost of *Regression Testing*

My focuses:

- RTP approaches based on *source code changes* (**code churn**)
- *Data and tools* to support RTP research

Research Contribution



Research products (1/2)

[J1]	**F. Altiero**, A. Corazza, S. Di Martino, A. Peron, L. L. L. Starace, *Regression Test Prioritization Leveraging Source Code Similarity with Tree Kernels,* **Journal of Software: Evolution and Process,** DOI: *10.1002/smr.2653*. In Production, 2024
[C1]	**F. Altiero**, A. Corazza, S. Di Martino, A. Peron, L. L. L. Starace, *Inspecting Code Churns to Prioritize Test Cases,* **International Conference on Testing Software and Systems (ICTSS 2020),** DOI: *10.1007/978-3-030-64881-7_17*. Naples, Italy, Dec. 2020, pp. 272-285, Springer
[C2]	**F. Altiero**, A. Corazza, S. Di Martino, A. Peron, L. L. L. Starace, *ReCover: a Curated Dataset for Regression Testing Research,* **International Conference on Mining Software Repositories (MSR 2022),** DOI: .Pittsburgh, PA, USA, May 2022, pp. 196-200, IEEE

Research products (2/2)

[C3]	**F. Altiero**, G. Colella, A. Corazza, S. Di Martino, A. Peron, L. L. L. Starace, *Change-Aware Regression Test Prioritization using Genetic Algorithms,* **International Conference on Software Engineering and Advanced Application (SEAA 2022),** DOI: *10.1109/SEAA56994.2022.00028*. Meloneras, Spain, Sep. 2022, pp. 125-132, IEEE
[C4]	**F. Altiero**, A. Corazza, S. Di Martino, A. Peron, L. L. L. Starace, *AI-based Fault-proneness Metrics for Source Code Changes,* **International Conference on Software Process and Product Measurement (MENSURA 2023),** Rome, Italy, Oct. 2023. To appear in the proceedings
[C5]	**F. Altiero**, A. Corazza, S. Di Martino, A. Peron, L. L. L. Starace, *Tree Kernels to Support Formal Methods-based Testing of Evolving Specifications,* **4th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis (OVERLAY) in 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA 2023),** Rome, Italy, Nov. 2023. To appear in the proceedings

PhD thesis overview

RTP aims to enhance the confidence of **developers** in releasing new software versions and reduce its time-to-market.

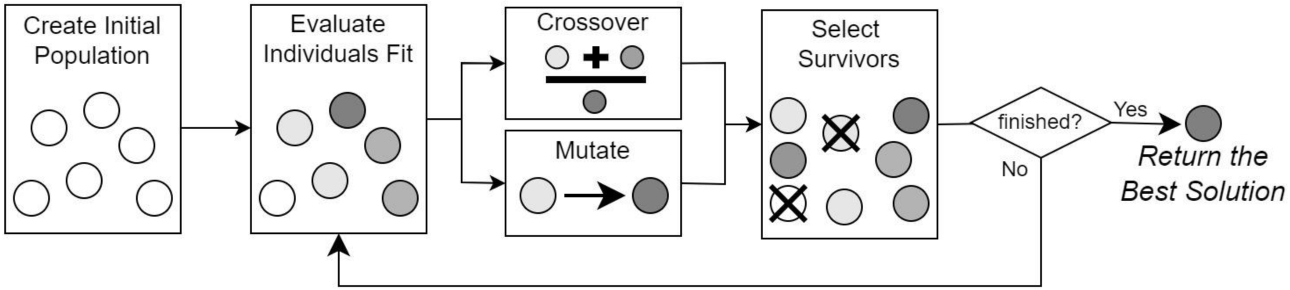
- **Problem: source code changes** can introduce defects in software. Proposed RTP techniques often **neglect churn** information.
  - **Objective**: **design** RTP techniques leveraging **accurate evaluation** of code-churn to **prioritize** test cases exercising **fault-prone changes**.
  - **Methodology**: **empirical evaluation** of designed churn-based techniques and **comparison** with state-of-art RTP approaches.

- **Problem: datasets** to evaluate RTP techniques are **not readily available**. **Researchers** would benefit from **shared data**.
  - **Objective: collect** a dataset of **software projects** representing **real-world** evolutionary scenarios to **support** RTP **experimentation**.
  - **Methodology: Mining** software repositories to collect **representative** contexts of **software evolution**.

Contribution 1

Genetic-Diff to prioritize test cases

Genetic-Diff



- **Genetic Algorithms** are popular in RTP
  - **In literature, these techniques do not consider code changes in the search for a permutation!**

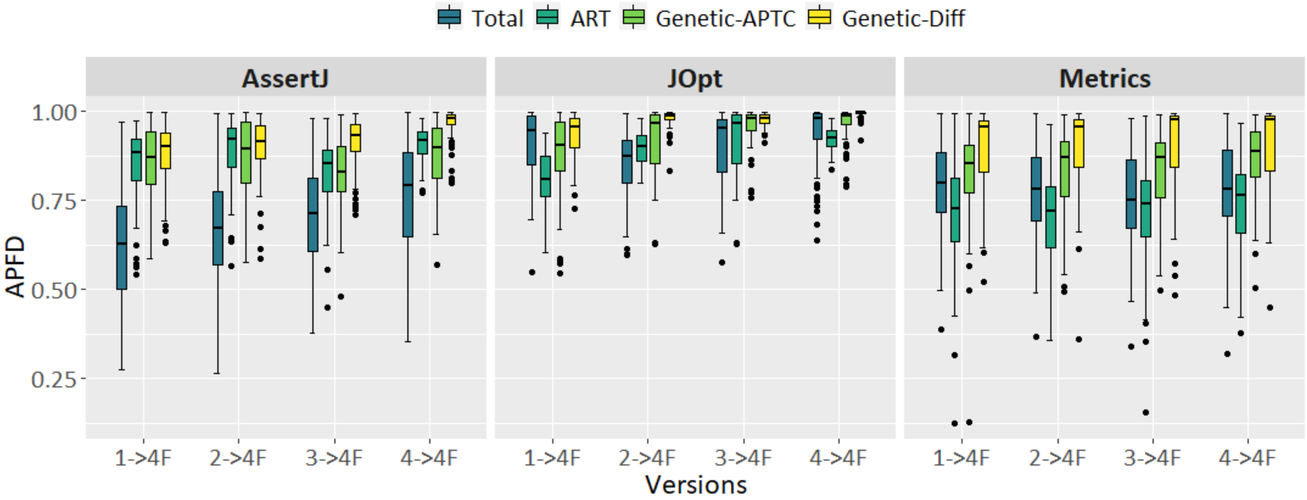
- Design of **Genetic-Diff**:
  - *Novel fitness function*: **APTC$_{\mathbf{diff}}$** measures the **rate of coverage of changed code**
  - *Novel crossover operator:* **CPX** mixes test cases of two parents to **prefer test cases covering churn**

Empirical Evaluation

- Evaluation on 12 versions of 3 heterogeneous Java projects
  - Used in several RTP studies
  - **Faults** automatically **injected** via **code mutation**
  - **100 faulted variants** for each version, over than **1k different evolutionary scenarios**
- **Metric**: rate of fault-discovery (**APFD**)
- Compared with **3** state-of-art **baselines**
  - One deterministic, one meta-heuristic, one traditional Genetic Algorithm

Results

- Genetic-Diff **outperformed** all the baselines on all projects



- The experimentation suggests that **leveraging code churn can enhance fault-detection performance**

Contribution 2

Refined evaluation of code churn with Tree Kernels

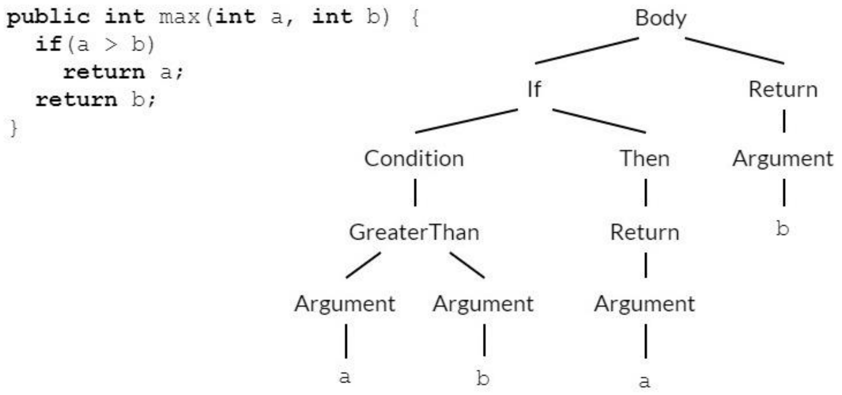
Tree Kernels and Code Churn

- Few RTP techniques leverage code changes
  - They typically consider simple **textual differences**

- Source code has a natural tree-based representation through **ASTs**

- **Tree Kernels** evaluate **structural and semantical similarity** of tree-based structures
  - *Profitably employed in **NLP** and **Software Engineering***

- A **refined** measure of the **extent of changes** can be evaluated by **TK on code ASTs**



```
public int max(int a, int b) {
  if(a > b)
    return a;
  return b;
}
```

TK Prioritization Techniques

- **MTK**: a novel RTP technique employing TK
  1. Evaluate the **extent** of changes of two versions of **evolving methods via TK**
  2. **Scores each test case** according to the **sum** of **changes** in **covered methods**
  3. **Sorts** the test suite accordingly

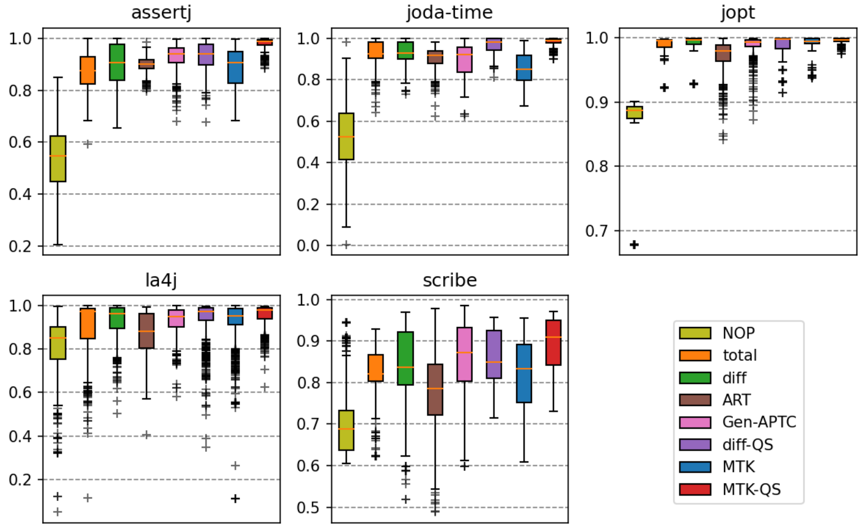
- **MTK-QS**: re-arranges the permutation using the Quotient-Set according to MTK test cases scores
  – *Rationale*: test cases **covering the same set** of changed methods **have the same score**
  – Increases the **diversification of coverage**

Empiric Evaluation

- Experimentation on 5 Java projects and 25 different software versions
  - **Mutation faults** injected as for Genetic-Diff
  - More than **5k** different evolutionary **scenarios**
- Comparison with **5 baseline** RTP techniques
  - Coverage-based, difference-based and non-deterministic
- **Metrics**:
  - fault-detection
    - *APFD, Percentage-to-First-Fault, Percentage-to-Last-Fault*
  - *effective execution time*

Results

- MTK was **comparable** to the baselines
  - Due to redundancy in scheduling
- MTK-QS **outperformed** all other techniques
  - Redundancy of MTK has been resolved by Quotient-Set



- Results suggests that a **refined** evaluation of changes **significantly improves** the fault-detection performance if **coverage is further diversified**
  - Scores evaluated by MTK can be a **fingerprint** of the **set of covered changed methods** for test cases

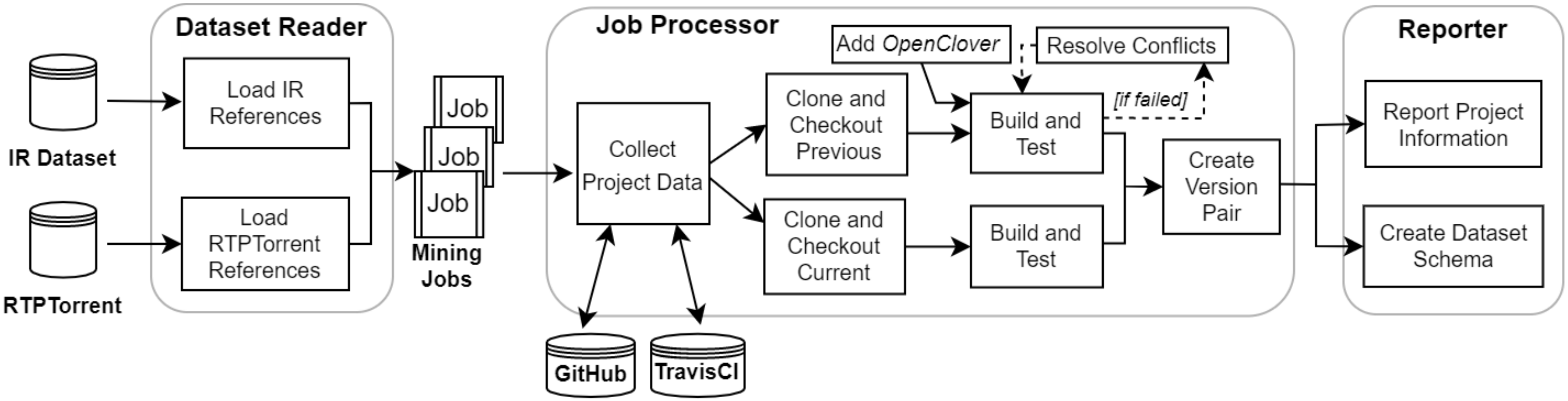
Contribution 3

ReCover: a dataset to foster RTP research

Efforts on Collecting Datasets

- Datasets in literature are **not generally applicable**
  - **Lack** of information for **different techniques**
  - Researchers need to manually **re-execute the build** to obtain **required information** (e.g., code coverage)
- Open-source datasets typically **do not present real faults**
  - Researchers rely on **synthetic faults**
  - This may **limit generalizability** of findings
- Recent datasets include projects with real faults
  - Often **only references** to remote location are provided
  - Refs may **expire** and projects could not be retrieved
  - Still needed to **re-execute** the build to obtain **auxiliary information**
- **A dataset with real faults and enriched with useful information can reduce the time from design to experimentation**

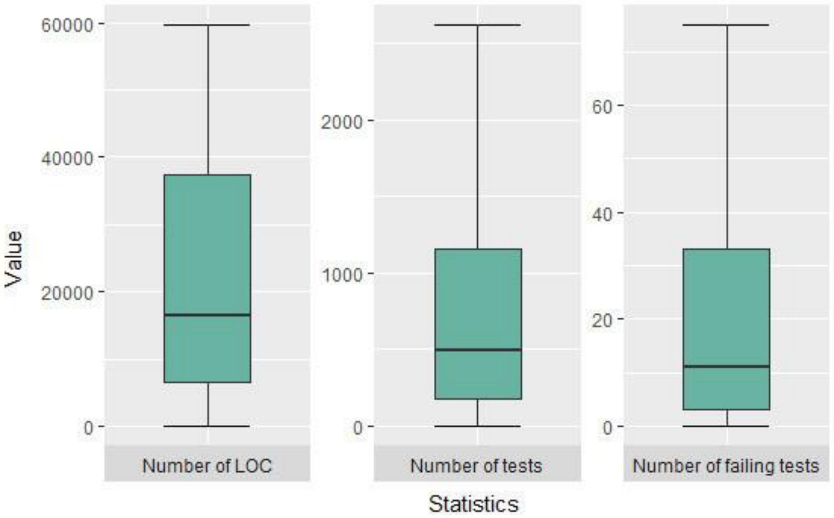
Mining the Dataset



- Starts from 2 reference-based RTP datasets
- Each entry is retrieved from remote repositories
  - Querying *TravisCI* and *GitHub*
- The build and testing phase of each version is executed
  - …in a *docker* container to resemble its original environment
  - Per-test coverage data are collected as well
- Collected projects are completely stored with their reports
  - Dataset metadata available in SQL and XML format

ReCover

- **114** evolutionary scenarios among **28** Java projects
  - Projects of different **sizes** and **nature**
  - Real test case failures due to **real-world faults**
  - **Full source code, coverage reports** and **test execution reports**
  - Per-project ***docker* containers** to **readily re-execute** the builds when needed



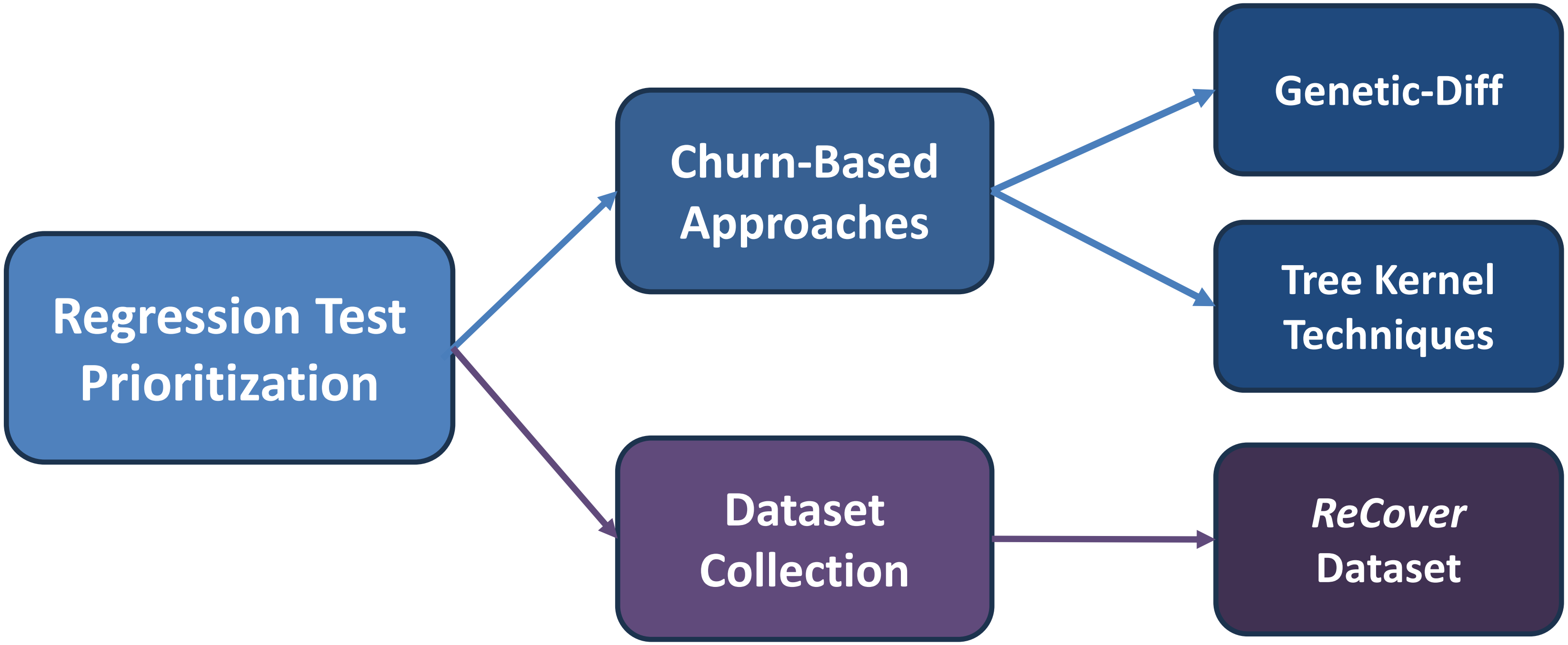
- To date, the dataset had more than **60 downloads** and has been employed in **3 RTP and software testing** papers.

Thank you!

Any Question?

Francesco Altiero

Churn-Based Approaches for Regression Test Prioritization



Regression Test Prioritization

Churn-Based Approaches

Genetic-Diff

Tree Kernel Techniques

Dataset Collection

ReCover Dataset