# Automating Code Changes in Python ML Systems

Malinda Dilhara, Ameya Ketkar, Nikhith Sannidhi, and Danny Dig
University of Colorado Boulder

Gold award – Student Research Competition at FSE-2021

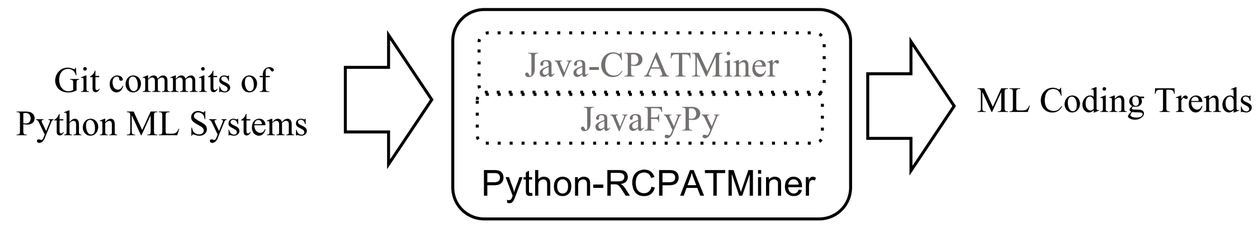
ICSE-2022/2023

## Problem Statement

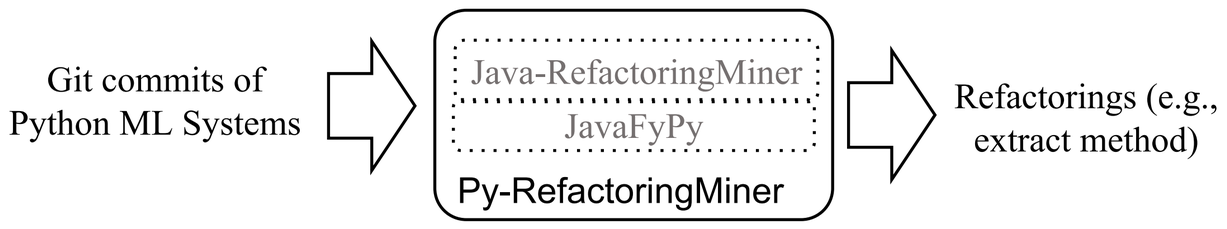
1. Python is the preferred language for many machine learning developers.
2. Existing code automation and analysis tools primarily target Java and C++, neglecting Python users, particularly in machine learning.
3. This underscores the demand for improved code refactoring and automation tool support in Python.
4. Rewriting tools entirely from scratch is resource-intensive.
5. There is a need for a platform that can easily convert tools from other languages to Python.

## Approach

- **JavaFyPy** adapts state-of-the-art Java AST mining tools to Python.
- **R-CPATMiner** mines repeated code changes in the version history of Python systems.

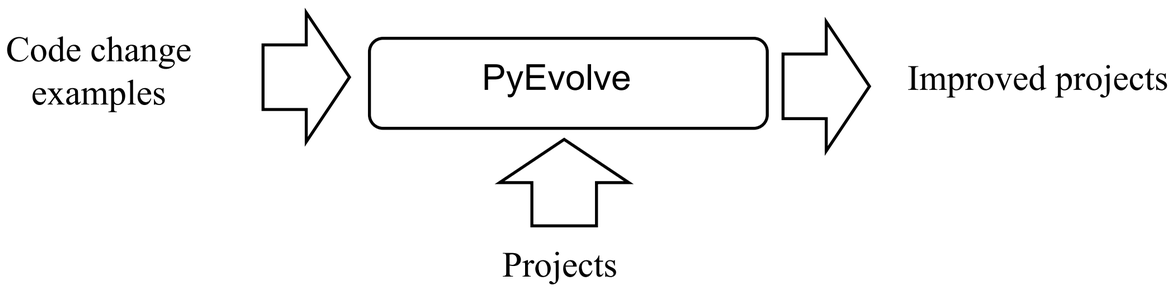



- **Py-RefactoringMiner** mines refactoring in the version history of Python systems.

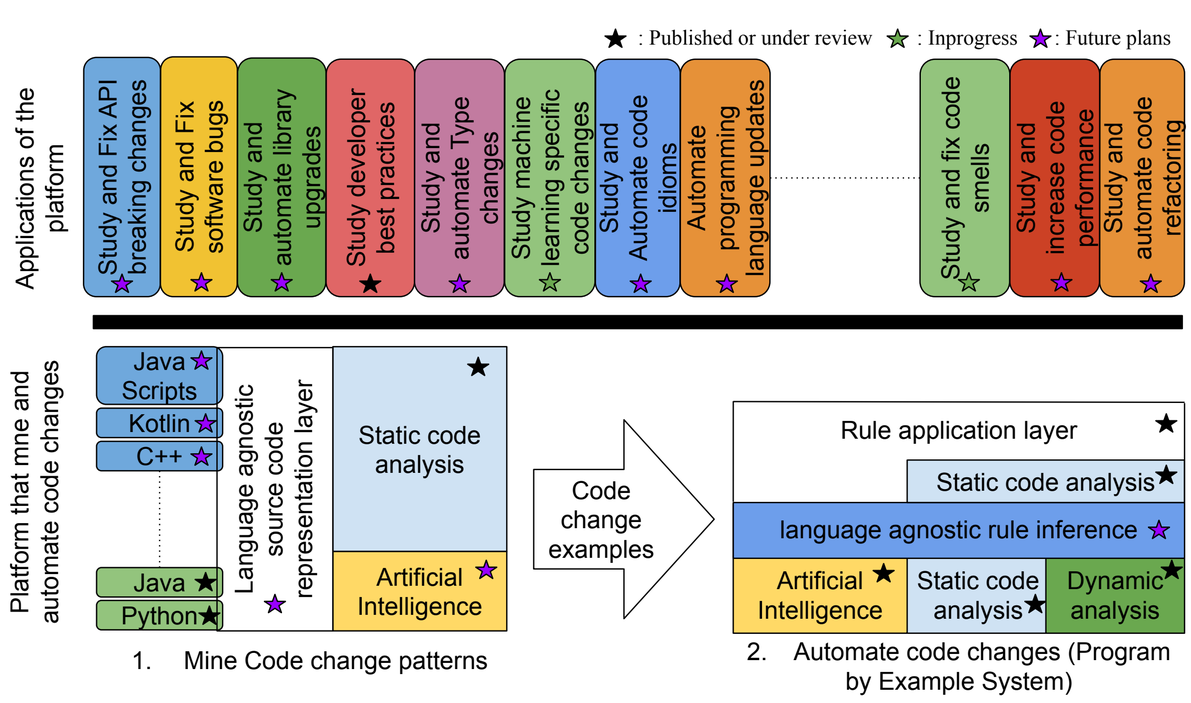



- We applied R-CPATMiner on top 1000 ML systems and surveyed 97 developers and discovered the best practices.

- **Py-Evolve** automatically transplants input code change into a target code




## What Follows




## Contributions

- Repository of best practices of evolving ML codes are publicly available.

https://mlcodepatterns.github.io

- Our code analysis and automations tools are publicly available

https://pythoninfer.github.io/

- A thorough evaluation of how the proposed solution is useful for developers

## Evaluation

- We discovered 28K repetitive code changes in ML systems by analyzing 4M git commits.
- We found 4 best practices and 22 code change themes.

| Best practices | Q1 | Q2 | Q3 |
|---|---|---|---|
| Transform to Context Managers | Often - 90% | Often - 100% | Yes-74% |
| Dissolve “for” loops to domain specific abstractions | Often - 95% | Often - 100% | Yes-89% |
| Update API usage | Often - 85% | Often - 100% | Yes-70% |
| Use advanced language features | Often - 30% | Rarely - 69% | Yes-30% |

- 71% of survey respondents said they wanted the identified code patterns to be automated in their IDEs.

**Example code change pattern in project** NifTK/NiftyNet: **commit** c8b28432

```
for elem in elements:          ⇨   np.sum(elements)
    result += elem
```

- PyEvolve-generated 181 patches to famous projects, they accepted 90%, highlighting the usefulness of PyCraft.

*“Well done, your changes are cleaner and faster’’*

*“The changes look good; I am not sure why we didn’t write it that way before”*

Microsoft IBM TensorFlow Keras

## Executive Summary

1. Tools for evolving ML systems are significantly behind [1].
2. We developed a platform to tranfer Java code analysis tools for Python.
3. We developed tools to automatically transplant code changes.
4. We found 4 best practices and 22 code change themes.
5. We submitted patches to open-source repositories.
6. We released tools to introduce best practices in ML code [2].

[1] Malinda Dilhara, Ameya Ketkar, and Danny Dig. 2021. Understanding Software-2.0. ACM Transactions Software Engineering Methodology.
[2] Malinda Dilhara, Ameya Ketkar, Nikhith Sannidhi, and Danny Dig. 2022. Discovering repetitive code changes in Python ML systems. International Conference on Software Engineering (ICSE ’22)
[3] Malinda Dilhara, Danny Dig, and Ameya Ketkar. PYEVOLVE: Automating Frequent Code Changes in Python ML Systems (ICSE 2023)