Identifying Bugs Related to Non-Functional Requirements in Java and Python

Background: Non-Functional Bugs

- Non-Functional Bug (NFB) = a bug that does not affect what the program does, but **how** it does it

- Fixes include making programs more time efficient, memory efficient, secure, etc.

OBJECTIVE

To create a dataset of NFBs in open-source software projects

Related Work

1. **MuBench**: dataset of 89 API misuses from 33 open source projects and survey

2. **Defects4J**: 357 real Java bugs from 5 open source projects

3. **Bugs.jar**: 1,158 Java bugs from 8 open source Apache projects

4. **Ohira et al**: dataset of 4000 functional and non-functional bugs from 4 open source Apache projects

5. **iBugs**: dataset of 369 realistic Java bugs from AspectJ

Our Dataset:

- 138 bugs from 67 open-source projects
- exclusive to non-functional bugs
- Projects written in Java or Python

[1]S. Amann, S. Nadi, HA. Nguyen, TN. Nguyen, and M. Mezini. MUBench: a benchmark for API-misuse detectors. MSR '16, pages 464-467. ACM, 2016.

[2]R. Just, D. Jalali, and M. D. Ernst. Defects4J: A Database of Existing Faults to Enable Controlled Testing Studies for Java Programs. ISSTA'14, pages 437–440. ACM, 2014

[3]RK. Saha, Y. Lyu, W. Lam, H. Yoshida, and MR. Prasad. Bugs.jar: a large-scale, diverse dataset of real-world Java bugs. MSR '18, pages 10-13. ACM, 2018.

[4]M. Ohira, Y. Kashiwa, Y. Yamatani, H. Yoshiyuki, Y. Maeda, N. Limsettho, K. Fujino, H. Hata, A. Ihara, and K. Matsumoto. A dataset of high impact bugs: manually-classified issue reports. MSR '15, pages 518-521. IEEE Press, 2015.

[5]V. Dallmeier and T. Zimmermann. Extraction of Bug Localization Benchmarks from History. ASE'07, pages 433–436. ACM, 2007.

Methodology

```yaml
source:
    name: github-search
project:
    name: Fowler
    url: https://github.com/TheFreshMango/Fowler
fix:
    tag: performance
    description: Replacing "+" with StringBuilder improves performance and memory space
    commit message: >
            refactoring: Customer now uses StringBuilder for statement()
    commit: https://github.com/TheFreshMango/Fowler/commit/d5f5290
location:
    file:
        Fowler/Fowler Refactor/src/de/dhbw/fowler/Customer.java
    method:
        public String statement()
api:
    java.lang.String
api change:
   java.lang.String -> java.lang.StringBuilder
rule:
    use a StringBuilder instead of adding strings to concatenate efficiently
```

Identify Candidate Repositories

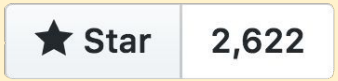
Extract commits using PyDriller

Manual Review

Documentation

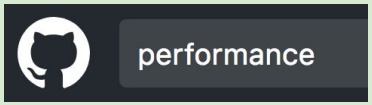
Identifying Candidate Repositories

Method One: Star Rating



- Github users “star” repositories they are interested in
- We chose the repositories with the highest number of stars
- 91 Java and 15 Python projects reviewed

Method Two: Github Search



- Github allows users to search for keywords in different domains (projects, commits, issues, etc.)
- 23 Java and 11 Python projects reviewed

Method Three: RepoReapers dataset

reaper

- Munaiah et al: set of “well-engineered” Github projects[1]
- we selected those with the most stars
- 17 Java and 10 Python projects reviewed

[1]Munaiah N, Kroh S, Cabrey C, and Nagappan M. Curating GitHub for engineered software projects. Empirical Software Engineering. 22(6): pages 3219-3253. Springer US. 2017.

Extracting Commits Using PyDriller

- PyDriller[1] is an open-source software-mining tool for Git
- Filtered the commit history of projects to messages containing **keywords** related to NFBs
- Restricted search to commits that changed .java or .py files

"fix"	"bug"	"error"
"refactor"	"secur[ity]"	"maint[enance]"
"stab[ility]"	"portab[ility]"	"efficien[cy]"
"usab[ility]"	"reliab[ility]"	"testab[ility]"
"changeab[ility]"	"replac[e]"	"memory"
"resource"	"runtime"	"crash"
"leak"	"attack"	"authenticat[ion]"
"authoriz[ation]"	"cipher"	"crack"
"decrypt"	"encrypt"	"vulnerab[ility]"
"minimiz[e]"	"optimiz[e]"	"slow"
"#"	"fast"	"perform[ance]

terms filtered out		
"typo"	"npe"	"spell"

These keywords include original words relevant to the study, as well as subsets from Hindle et al.[2] and de la Mora and Nadi[3].

[1]D. Spadini, M. Aniche, and A. Bacchelli. PyDriller: Python Framework for Mining Software Repositories. ESEC/FSE. 2018. https://github.com/ishepard/pydriller
[2]A. Hindle, NA. Ernst, MW. Godfrey, and J.Mylopoulos. Automated topic naming to support cross-project analysis of software maintenance activities. MSR '11, pages 163-172. ACM, 2011
[3]FL. de la Mora and S. Nadi. Which library should I use? A metric-based comparison of software libraries. ICSE NIER '18, pages 37-40. ACM, 2018

Manual Review

- Weeding out false positives
- Example:

2 speed_bomb.py

```
@@ -245,7 +245,7 @@ def reset_bomb_length(self):
245 245            self.bomb_length = 5.0
246 246
247 247        def get_bomb_length(self):
248     -          self.bomb_length -= 0.2
    248 +          self.bomb_length -= 0.3
249 249            if self.bomb_length < 1:
250 250                self.bomb_length = 1
251 251            return self.bomb_length
```

Commit Message: “`Update speed_bomb.py made speed change faster`”

https://github.com/adangert/JoustMania/commit/a9711c2

Documenting Bugs

```
source:
    name: github-search
project:
    name: Fowler
    url: https://github.com/TheFreshMango/Fowler
fix:
    tag: performance
    description: Replacing "+" with StringBuilder improves performance and memory space
    commit message: >
            refactoring: Customer now uses StringBuilder for statement()
    commit: https://github.com/TheFreshMango/Fowler/commit/d5f5290
location:
    file:
        Fowler/Fowler Refactor/src/de/dhbw/fowler/Customer.java
    method:
        public String statement()
api:
    java.lang.String
api change:
   java.lang.String -> java.lang.StringBuilder
rule:
    use a StringBuilder instead of adding strings to concatenate efficiently
```

Documenting Bugs

```
source:
    name: github-search
project:
    name: Fowler
    url: https://github.com/TheFreshMango/Fowler
fix:
    tag: performance
    description: Replacing "+" with StringBuilder improves performance and memory space
    commit message: >
            refactoring: Customer now uses StringBuilder for statement()
    commit: https://github.com/TheFreshMango/Fowler/commit/d5f5290
location:
    file:
        Fowler/Fowler Refactor/src/de/dhbw/fowler/Customer.java
    method:
        public String statement()
api:
    java.lang.String
api change:
   java.lang.String -> java.lang.StringBuilder
rule:
    use a StringBuilder instead of adding strings to concatenate efficiently
```

bug category

Documenting Bugs

```
source:
    name: github-search
project:
    name: Fowler
    url: https://github.com/TheFreshMango/Fowler
fix:
    tag: performance
    description: Replacing "+" with StringBuilder improves performance and memory space
    commit message: >
            refactoring: Customer now uses StringBuilder for statement()
    commit: https://github.com/TheFreshMango/Fowler/commit/d5f5290
location:
    file:
        Fowler/Fowler Refactor/src/de/dhbw/fowler/Customer.java
    method:
        public String statement()
api:
    java.lang.String
api change:
   java.lang.String -> java.lang.StringBuilder
rule:
    use a StringBuilder instead of adding strings to concatenate efficiently
```

bug category

our interpretation of the fix

Documenting Bugs

```
source:
    name: github-search
project:
    name: Fowler
    url: https://github.com/TheFreshMango/Fowler
fix:
    tag: performance
    description: Replacing "+" with StringBuilder improves performance and memory space
    commit message: >
            refactoring: Customer now uses StringBuilder for statement()
    commit: https://github.com/TheFreshMango/Fowler/commit/d5f5290
location:
    file:
        Fowler/Fowler Refactor/src/de/dhbw/fowler/Customer.java
    method:
        public String statement()
api:
    java.lang.String
api change:
   java.lang.String -> java.lang.StringBuilder
rule:
    use a StringBuilder instead of adding strings to concatenate efficiently
```

bug category

our interpretation of the fix

related api (if applicable)

Documenting Bugs

```
source:
    name: github-search
project:
    name: Fowler
    url: https://github.com/TheFreshMango/Fowler
fix:
    tag: performance
    description: Replacing "+" with StringBuilder improves performance and memory space
    commit message: >
            refactoring: Customer now uses StringBuilder for statement()
    commit: https://github.com/TheFreshMango/Fowler/commit/d5f5290
location:
    file:
        Fowler/Fowler Refactor/src/de/dhbw/fowler/Customer.java
    method:
        public String statement()
api:
    java.lang.String
api change:
   java.lang.String -> java.lang.StringBuilder
rule:
    use a StringBuilder instead of adding strings to concatenate efficiently
```

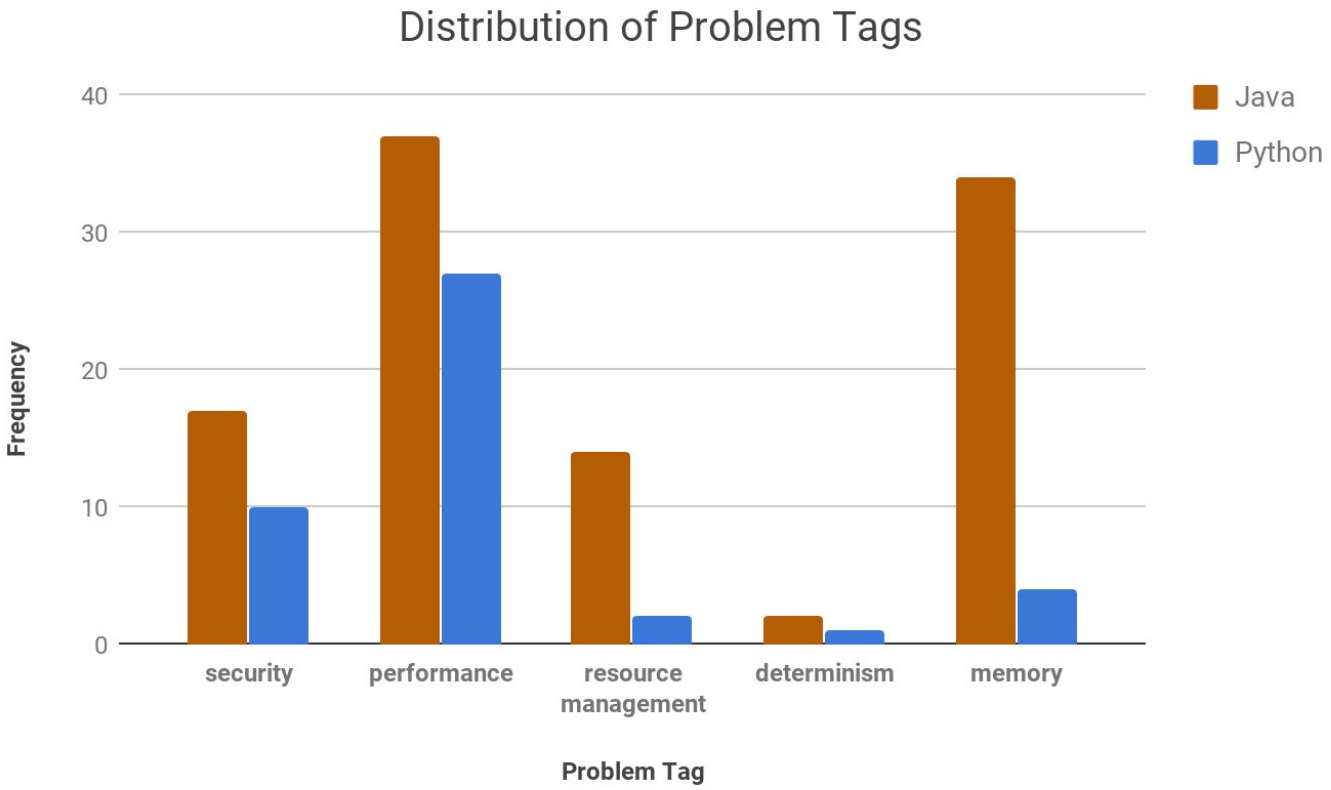
bug category

our interpretation of the fix

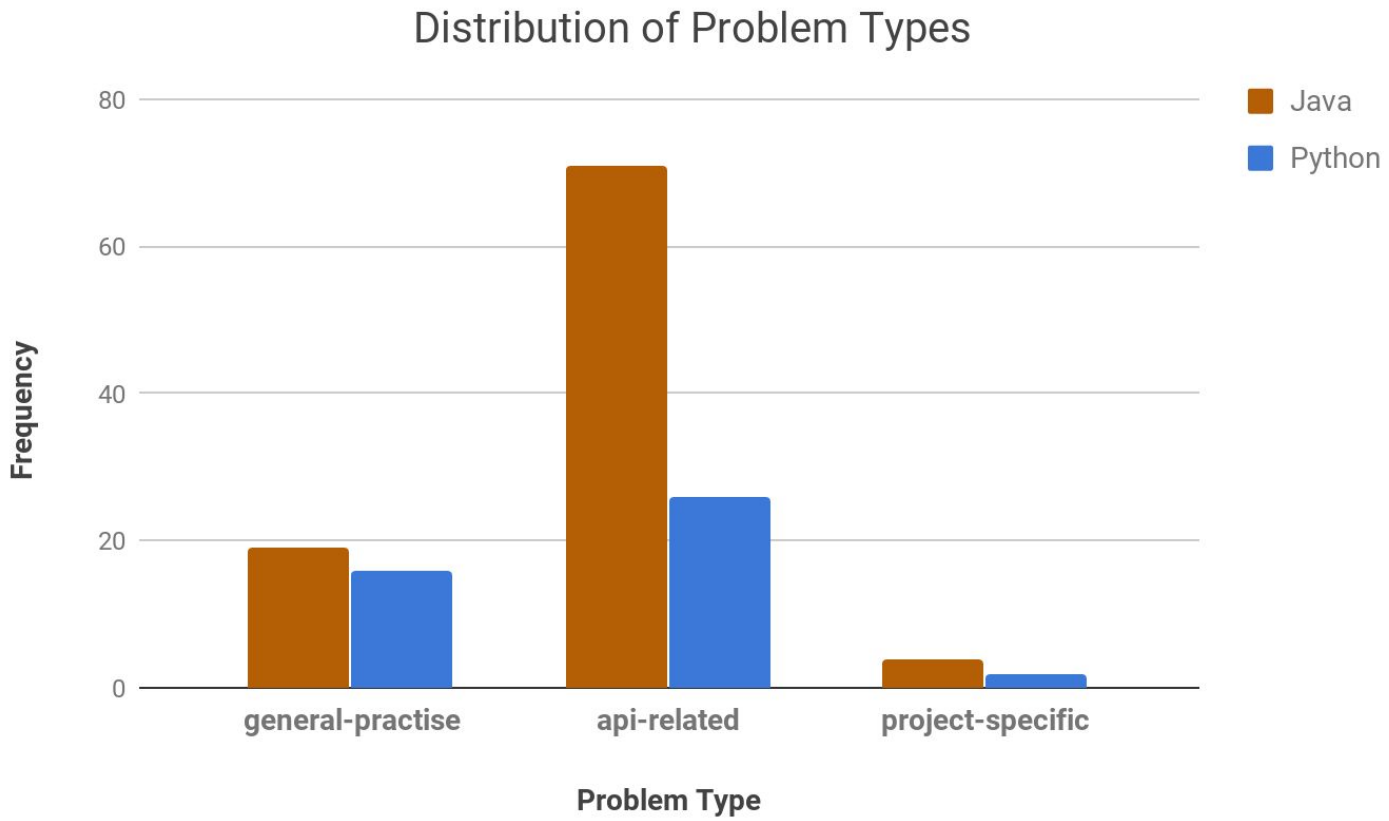
related api (if applicable)

conclusion

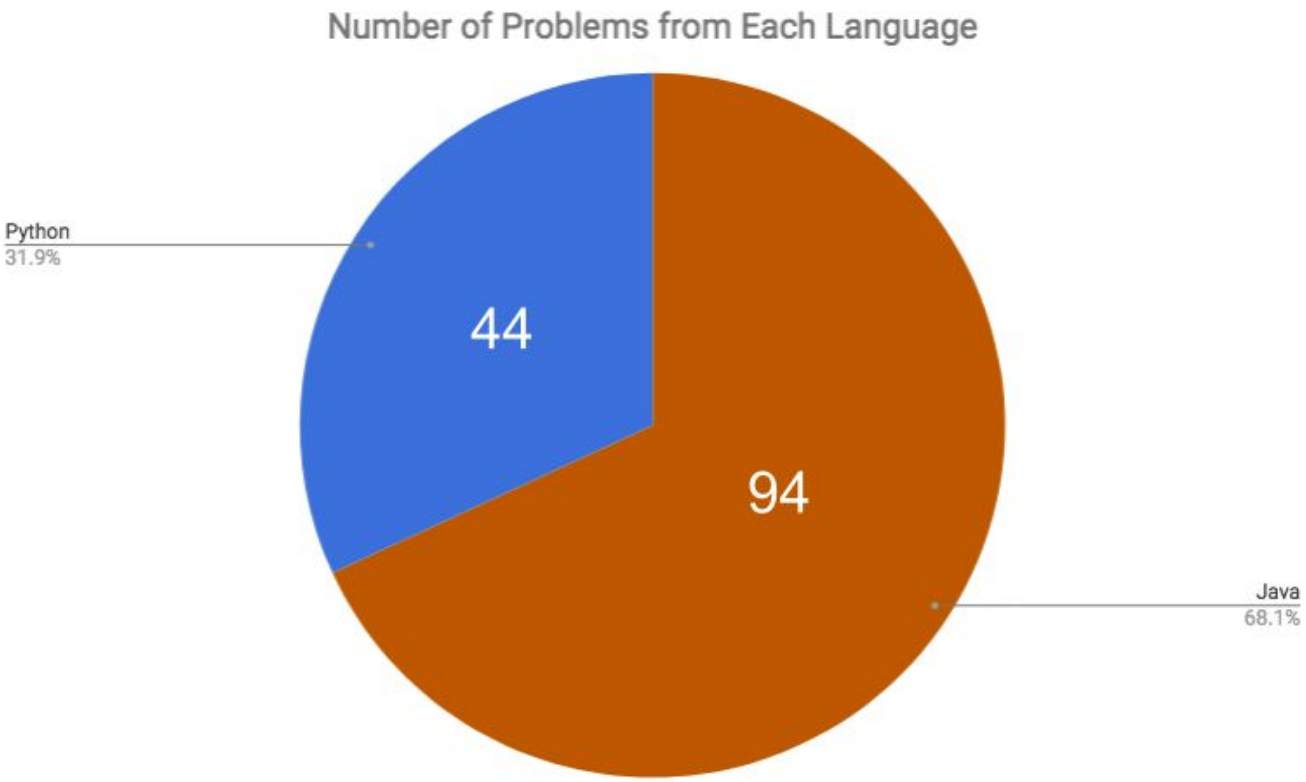
Data Distribution



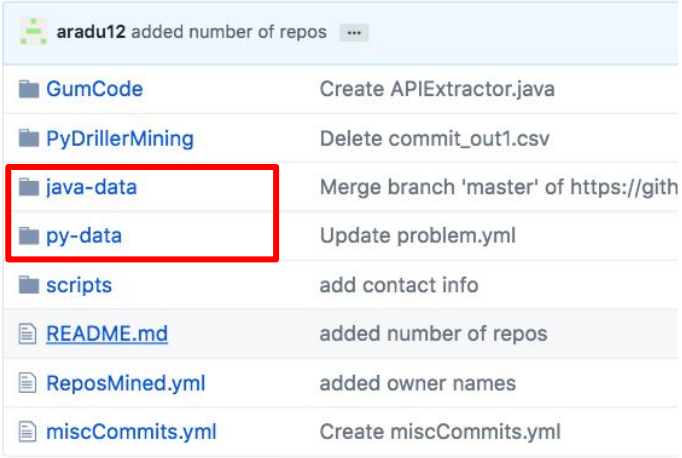
Data Distribution



Data Distribution

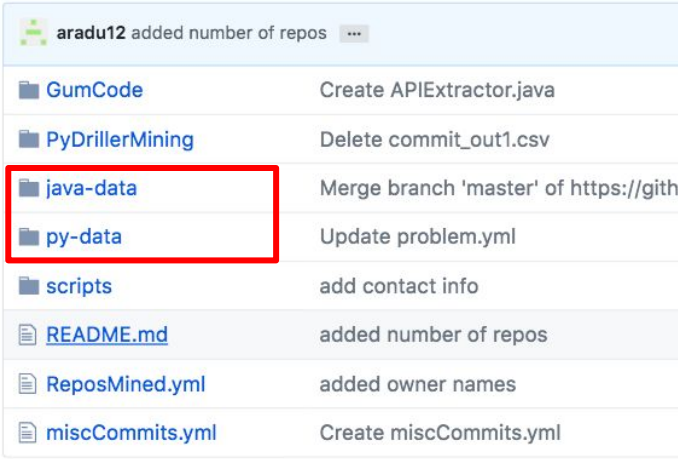


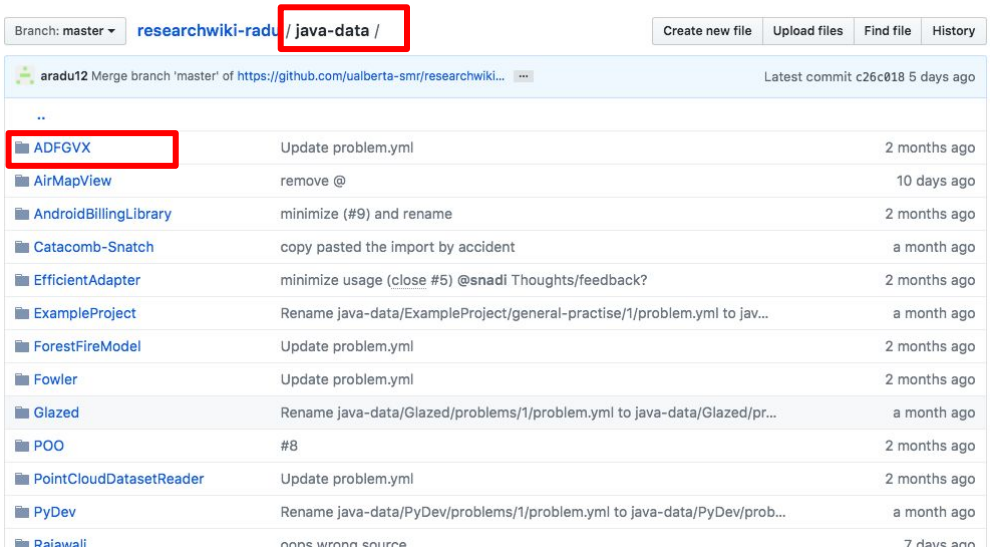
Repo Organization



- Separate folders for data in each language

Repo Organization





- Separate folders for data in each language
- Each project has its own folder

Repo Organization

these folders contain problems, problem version IDs, and project metadata

Repo Organization

these folders contain problems, problem version IDs, and project metadata

problems are grouped by type

Repo Organization

these folders contain problems, problem version IDs, and project metadata

problems are grouped by type

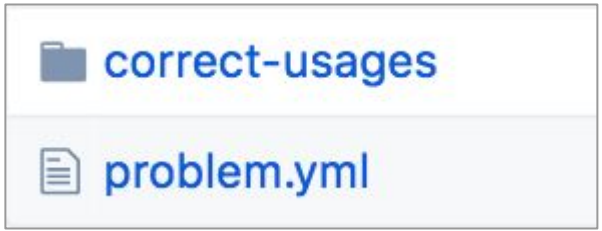
each problem has its own folder

Repo Organization

these folders contain problems, problem version IDs, and project metadata

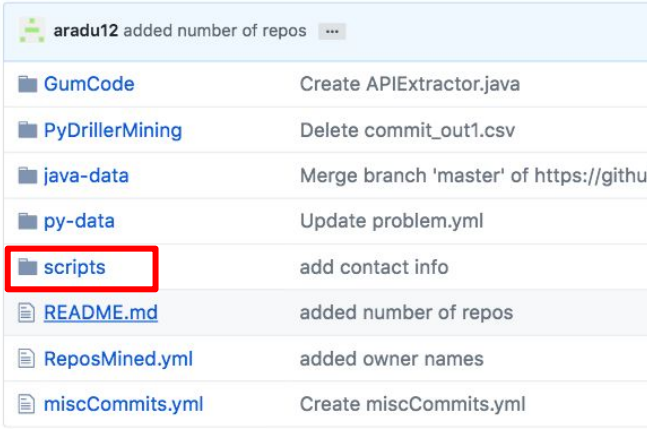
problems are grouped by type

each problem has its own folder



the problem details

Processing the Data



- We provide scripts to manage the data
- DataBox class allows users to filter problems by star rating, tag, problem type, etc.
- Users can write their own client script or use our example: DataBoxClient.py

Applications of the Dataset

- Code Recommenders
- Training set for bug detection tools
- Analysis of coding practices between veterans and beginners

OBJECTIVE

To create a dataset of NFBs in open-source software projects

OBJECTIVE

To create a dataset of NFBs in open-source software projects

3

Methodology

Identify Candidate Repositories

Extract commits using PyDriller

Manual Review

Documentation

5

OBJECTIVE

To create a dataset of NFBs in open-source software projects

3

Methodology

Identify Candidate Repositories

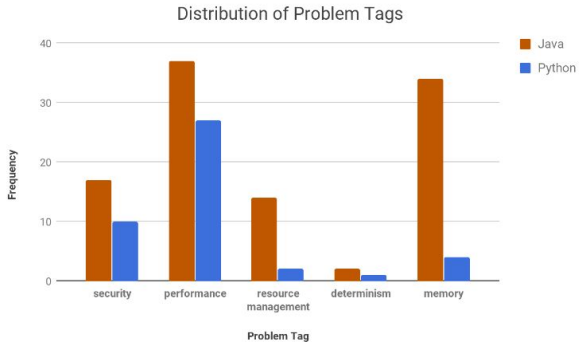
Extract commits using PyDriller

Manual Review

Documentation

5

Data Distribution



14

OBJECTIVE

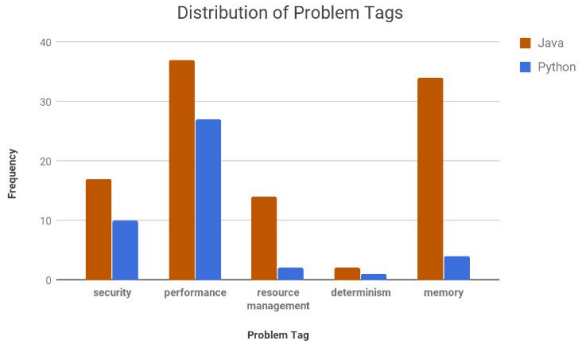
To create a dataset of NFBs in open-source software projects

3

Methodology

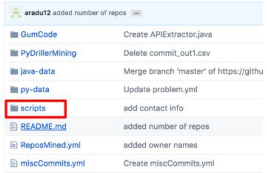
5

Data Distribution



14

Processing the Data



- We provide scripts to manage the data
- DataBox class allows users to filter problems by star rating, tag, problem type, etc.
- Users can write their own client script or use our example: DataBoxClient.py

23

OBJECTIVE

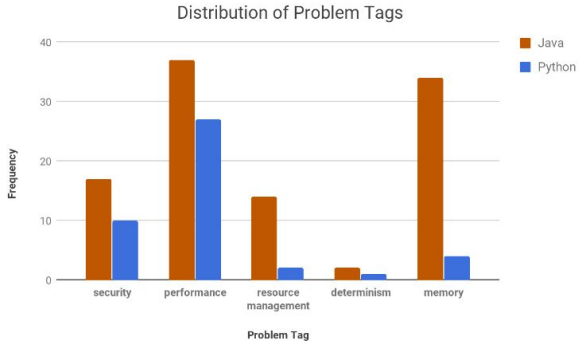
To create a dataset of NFBs in open-source software projects

3

Methodology

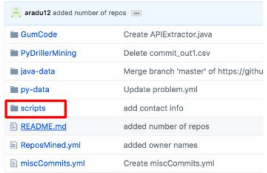
5

Data Distribution



14

Processing the Data



- We provide scripts to manage the data
- DataBox class allows users to filter problems by star rating, tag, problem type, etc.
- Users can write their own client script or use our example: DataBoxClient.py

23