Replication Utilities

This repository contains the resources and steps required to replicate our study.
The replication package consists of two major stages:

1. **Stage 1** – Evaluation and refinement of the multi-language design smell detector.
2. **Stage 2** – Re-analysis of smell prevalence and their relation to software fault- and change-proneness.

---

1. Detection of Multi-Language Design Smell Occurrences

Location: `Stage 1/`
Includes:

- `Dataset/` – Scripts and metadata to fetch and prepare the 60 open-source multi-language projects.
- `Ground Truth/` – Manual annotations for each smell, labeling guidelines, and validation data.
- `Detection Approach Revised/` – Refined implementation of the detector.
- `Results/` – Detection results and precision/recall comparison.

Getting Started

Running the Detector

1. Navigate to:

```
cd Stage\ 1/Detection\ Approach\ Revised/mlssdd/
```

2. Open `DetectCodeSmellsAndAntiPatterns.java`.
3. Comment out all smell detectors except the one to be executed.
4. Compile and run:

```
javac mlssdd/DetectCodeSmellsAndAntiPatterns.java
java mlssdd.DetectCodeSmellsAndAntiPatterns
```

5. The output CSV files are generated under:

```
Stage 1/Results/results_of_revised_technique_on_new_data/[SmellName]/
```

Example:
To detect *Unused Parameter*, uncomment:

```
new UnusedParameterDetector().runDetection();
```

and rerun the above commands.

Comparison:
Baseline and refined detector accuracy are summarized in
`precision_recall_baseline.csv` and `precision_recall_refined.csv`.

---

2. Prevalence Analysis

Location: `Stage 2/Prevalence Results/`
Includes:

- `general_smelly/` – Files affected by ≥ 1 smell (per release).
- `smelly/` – 15 subfolders, one per smell type.
- Aggregated results:
  - `percentage_of_general_smelly_files_releasewise.csv`
  - `percentage_of_smelly_files_affected_by_each_smell.csv`
  - `difference_of_prevalence_by_both_techniques.csv`

Getting Started

Each CSV file reports prevalence metrics per release and per smell.
These can be analyzed directly using R or Python (e.g., `pandas`, `matplotlib`).

---

3. Change-Proneness Analysis

Location: `Stage 2/Change-Proneness Results/`
Includes:

- `changedJNI/` – Lists of changed JNI files per release.
- `general_smelly/` and `JNIfiles/` – Supporting data.
- `scripts/` – Scripts for extracting changes and running Fisher's exact tests.
- `fisher_results_change.csv` – Final computed test results.

Getting Started

1. Ensure Python ≥ 3.8 is installed.
2. Run the provided scripts sequentially:

```
python scripts/extract_changed_files.py
python scripts/fisher_change_analysis.py
```

3. The resulting CSV provides odds ratios, p-values, and confidence intervals.

---

4. Fault-Proneness Analysis

Location: `Stage 2/Fault-Proneness Results/`
Includes:

- `faultyJNI/` – Lists of faulty JNI files (per release).
- `scripts/` – Scripts for fault extraction and Fisher tests.
- `fisher_results_fault.csv` – Final computed test results.

Getting Started

```
python scripts/extract_faulty_files.py
python scripts/fisher_fault_analysis.py
```

Outputs mirror the structure of the change-proneness results.

---

5. Dependencies

- **Java 8** +
- **Python 3.8** + with `pandas`, `numpy`, `scipy`
- **srcML** (for code parsing)
- **Apache Commons Compress** (for archive handling)

Download srcML: https://www.srcml.org