PyTraceBugs: A Large Python Code Dataset for Supervised Machine Learning in Software Defect Prediction

E. N. Akimova, A. Yu. Bersenev, A. A. Deikov, K. S. Kobylkin, A. V. Konygin, I. P. Mezentsev, V. E. Misilov

Krasovskii Institute of Mathematics and Mechanics,

Ural Federal University, Ekaterinburg, Russia

The 28th Asia-Pacific Software Engineering Conference, online, 6-9 December, 2021

BUGS IN THE SOFTWARE DEVELOPMENT

Bugs in complex software projects have a variety of undesirable consequences, including:

- data loss;
- program crashes;
- hardware failures.

All these issues could increase cost of software development and lead to significant money losses.

The task of automatically detecting bugs is a longstanding problem in the software engineering.

SPECIFICS OF PYTHON LANGUAGE

Python is a language of choice for many developers working in a variety of domains, including web development, data science, and machine learning.

PYTHON SPECIFICS

- dynamic typing;
- its interpreter and existing static analysis tools do not provide any thorough checks of source code.

This postpones revealing of bugs to the runtime stage and leads to the need of expensive debugging stage during the development.

DEEP LEARNING METHODS FOR BUG PREDICTION IN PYTHON SOURCE CODE

- applying deep learning methodologies originally intended for natural language processing tasks (say, Transformers) to the software engineering leads to dramatic improvements;
- using those methodologies (e.g. graph neural networks) for finding bugs in Python source code also gives promising results (Allamanis et. al, 2021).

SPECIFICS OF DEEP LEARNING MODELS

1. have lots of parameters;
2. require large datasets for their training.

CODE ANALYSIS TASKS, UNDERLYING KNOWN DATASETS

Not every known bug dataset is suitable for training and evaluating deep learning models. Some of these datasets are intended for other tasks.

Task	Key Feature of Dataset	Description
Automatic Test Generation	Reproducibility	Consistent results: tests should fail on buggy code and pass on the fixed one.
Program Repair	Isolation	Buggy and fixed code should differ only by a bug fix.
Bug Prediction	Representativeness Sufficient size	The properties of samples should correspond to the properties of parent classes (buggy and correct code).

CONTENT OF THE KNOWN DATASETS

Tasks, underlying datasets, restrict them to contain a specific type of information. For example, datasets aimed to either automatic test generation or program repair usually contain the so called bugfixes.

Each bugfix contains two consequtive versions of the same source code, where the first piece contains bugs whereas the second one is an immediate fix of those bugs.

Fixed parts of bugfix pairs can not be guaranteed to be correct. Therefore, they do not fully represent the class of correct code.

SUMMARY OF THE KNOWN DATASETS

Name	Language	Granularity	Size	Task	Content type
ManyBugs	C	module	185	test generation	bugfix pairs
Defects4J	Java	module	357	test generation	bugfix pairs
Bugs.jar	Java	module	1158	test generation	bugfix pairs
BugsInPy	Python	file	493	test generation	bugfix pairs
GHPR	Java	file	3026	bug prediction	bugfix pairs
BugHunter	Java	file/class/snippet[1]	159k	bug prediction	bugfix pairs
PyPiBugs	Python	snippet	2374	program repair	bugfix pairs
CodRep	Java	1 line	800k	program repair	bugfix pairs
ManySStuBs4J	Java	1 line	153k	program repair	bugfix pairs

[1] snippet means either a function or a method

INTRODUCING THE PYTRACEBUGS DATASET

Name	Language	Granularity	Size	Task	Content type
ManyBugs	C	module	185	test generation	bugfix pairs
Defects4J	Java	module	357	test generation	bugfix pairs
Bugs.jar	Java	module	1158	test generation	bugfix pairs
BugsInPy	Python	file	493	test generation	bugfix pairs
GHPR	Java	file	3026	bug prediction	bugfix pairs
BugHunter	Java	file/class/snippet[2]	159k	bug prediction	bugfix pairs
PyPiBugs	Python	snippet	2374	program repair	bugfix pairs
CodRep	Java	1 line	800k	program repair	bugfix pairs
ManySStuBs4J	Java	1 line	153k	program repair	bugfix pairs
PyTraceBugs	**Python**	**snippet**	**24k buggy 5.7M correct**	**bug prediction**	**buggy code correct code**

[2] snippet means either a function or a method

PURPOSE OF THE PYTRACEBUGS DATASET

AIM OF THE DATASET

In distinction to the known datasets, PyTraceBugs dataset is collected specifically for pre-training and fine-tuning the deep learning models for the bug prediction problem at the granularity of snippets (implementations of functions or methods).
This problem is considered in the form of the binary classification one with two classes of snippets:

- buggy code;
- error-free code.

PyTraceBugs contains a large number of Python source code snippets labeled either buggy or correct.

PRINCIPLES, GUIDING LABELING OF SNIPPETS

- PyTraceBugs contains automatically labeled source code snippets from well-maintained Github repositories for real-world software projects.
- Buggy code is extracted by processing the bug reporting issues and corresponding bugfixing commits.
- Error-free code is obtained from stable code. That means the code was not changed for a long time up to the current state of the repository.

BUGS IN THE PYTRACEBUGS DATASET

SPECIFICS OF BUGS FROM THE DATASET

The dataset only includes those bugs which cause program to stop and throw an error exception report, containing the traceback information.

```
a = [1, 2, 3, 4]

def getItem(index, array):
    return array[index]

getItem(5, a)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-1-43e12b2f0d17> in <module>
      4     return array[index]
      5
----> 6 getItem(5, a)

<ipython-input-1-43e12b2f0d17> in getItem(index, array)
      2
      3 def getItem(index, array):
----> 4     return array[index]
      5
      6 getItem(5, a)

IndexError: list index out of range
```

The most present bugs are errors related to missing object attributes and empty objects.

DESCRIPTION OF THE PYTRACEBUGS DATASET

STRUCTURE AND CONTENT OF THE DATASET

It is split into training, validation, and test samples.

- training and validation samples contain automatically labeled source code snippets;
- test sample of 330 snippets contains manually curated buggy code and selected stable code;
- training and test samples are taken from distinct repositores.

CONFIDENCE OF LABELING IN TRAINING AND VALIDATION SAMPLES

To evaluate the quality of the dataset, the amount of noise (persentage of mislabeled buggy snippets) is estimated. The mistakenly labeled buggy snippet is that for which its corresponding fix changes the test code, comments, docstrings, or is confined to code refactoring.

WAYS TO EVALUATE AMOUNT OF NOISE

- the lower bound is the percentage of the changes bound to docstrings and comments (2.6%);
- the percentage of refactoring changes is estimated by manually observing a random sample of several hundreds of snippets (10–15%).

CONFIDENCE OF LABELING IN THE TEST SAMPLE

The test sample was validated manually by Python experts. Thus, the confidence of labeling is almost 100%.

PRINCIPLES GUIDING THE MANUAL VALIDATION

- a bug reported on the web page of the corresponding issue is simple to understand;
- a fix of the bug introduced into a buggy snippet is also simple;
- the reported bug is not dependency, compatibility, or the regression bug;
- correct snippets are chosen from stable snippets the restriction that the snippet should be called many times from other snippets.

TRAINING PREDICTIVE MODELS ON THE DATASET

An alternative way to demonstrate quality of the dataset consists in building predictive models using its data.

DETAILS OF TRAINING PREDICTIVE MODEL

- multi-language pretrained CodeBERT model is applied to compute embeddings of source code from the dataset;
- LightGBM classifier is trained on the computed embeddings.

Results of the prediction experiments on the test sample

	Precision	Recall	F_1-measure
Correct	0.61	0.99	0.76
Buggy	0.96	0.34	0.5

CONCLUSION

- PyTraceBugs is a large labeled Python source code dataset intended for both **training and evaluating of deep learning models for software bug prediction**.
- It contains **24 thousands of examples of real bugs** at the granularity of snippets.
- It contains **5.7 millions** of correct code snippets.
- The manually curated test sample consists of **330 snippets** (both buggy and correct).
- **Confidence in labeling** of buggy snippets is 85% for the training and validation samples, and almost 100% for the test sample.
- **An accurate bug prediction model** is trained on the dataset.
- The dataset is available at `https://github.com/acheshkov/pytracebugs`.

THANK YOU FOR YOUR ATTENTION!