TabFuzz: High-level mutations for tabular data

Author: Martijn Smits

Supervisor: Burcu Ozkan

Computer Science and Technology, TU Delft

CSE3000 Research Project

29-06-2021

1 Research question

How can we provide users to enter input specifications and implement mutations in a generic way for all kinds of tabular data?

2 Background

Fuzz testing is discovering faults in software by automatically providing unexpected inputs

Data-Intensive Scalable Computing (DISC) programs are used for parallel computing of large volumes of data

BigFuzz

- Transforms DISC programs into Java programs
- Applies fuzz testing to the transformed program
- Uses high-level mutations
- TabFuzz replicates and improves the BigFuzz solution

3 Method

Input specification

The programmer can specify the following five properties:

- Datatype
- Range
- Special values
- Column Name
- Repeat

Input generation

A valid input file is generated based on the given input specification

Input mutation

One mutation per fuzzing cycle

High level mutations:

- Data Distribution Mutation
- Data Type Mutation
- Data Column Mutation
- Null Data Mutation
- Empty Data Mutation
- Special Value Mutation

Columnname: Zipcode
Datatype: String
Range: 900[0-9]{2}
Special: 90042

Columnname: Number
Datatype: int
Range: >0

Datatype: char
Repeat: 5-10

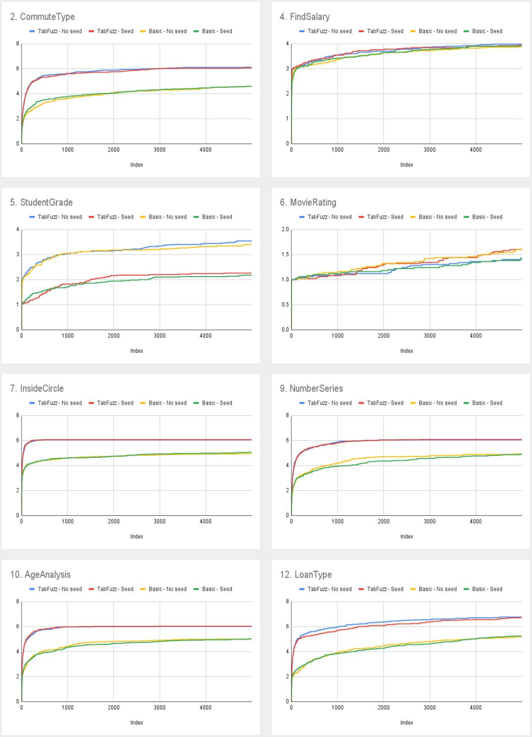
4 Evaluation

- 12 Benchmarks
- Seed generation vs provided seed
- BigFuzz replica vs TabFuzz

5 Conclusion

- Seed generation is just as effective as using a provided seed
- TabFuzz outperforms the BigFuzz replica

Results



2. CommuteType

4. FindSalary

5. StudentGrade

6. MovieRating

7. InsideCircle

9. NumberSeries

10. AgeAnalysis

12. LoanType