Student Programming Strategies

Projected by Kin Leong Lee, Lei Qian Supervised by Dr. Claudia Szabo
School of Computer Science
THE UNIVERSITY of ADELAIDE

Project Introduction

In introductory computer science courses, novice students always struggle handling compilation errors and using data structure.

This project aims to analyse collected data about student programming assignments and determine what programming patterns students employ.

In this work, the **dataset** is obtained from the BlueJ Blackbox project, which has recorded over 150,000 users, their over 10,000,000 compilations and tens of gigabytes of source code.

The source codes are recorded between 2013 and 2014. There are total 546,188 projects in the dataset.

Project Scope

Research Questions:

- **RQ1:** How to select and manipulate a representative subset of the database?
- **RQ2:** What measurement can assess student error handling abilities?
- **RQ3:** What metric determines students' improvement on programming skills ?

Sample Selection Criteria:

To identify student programming patterns we analyse data taken from the academic programming projects without considering test cases.

Proposed Analysis Methodology

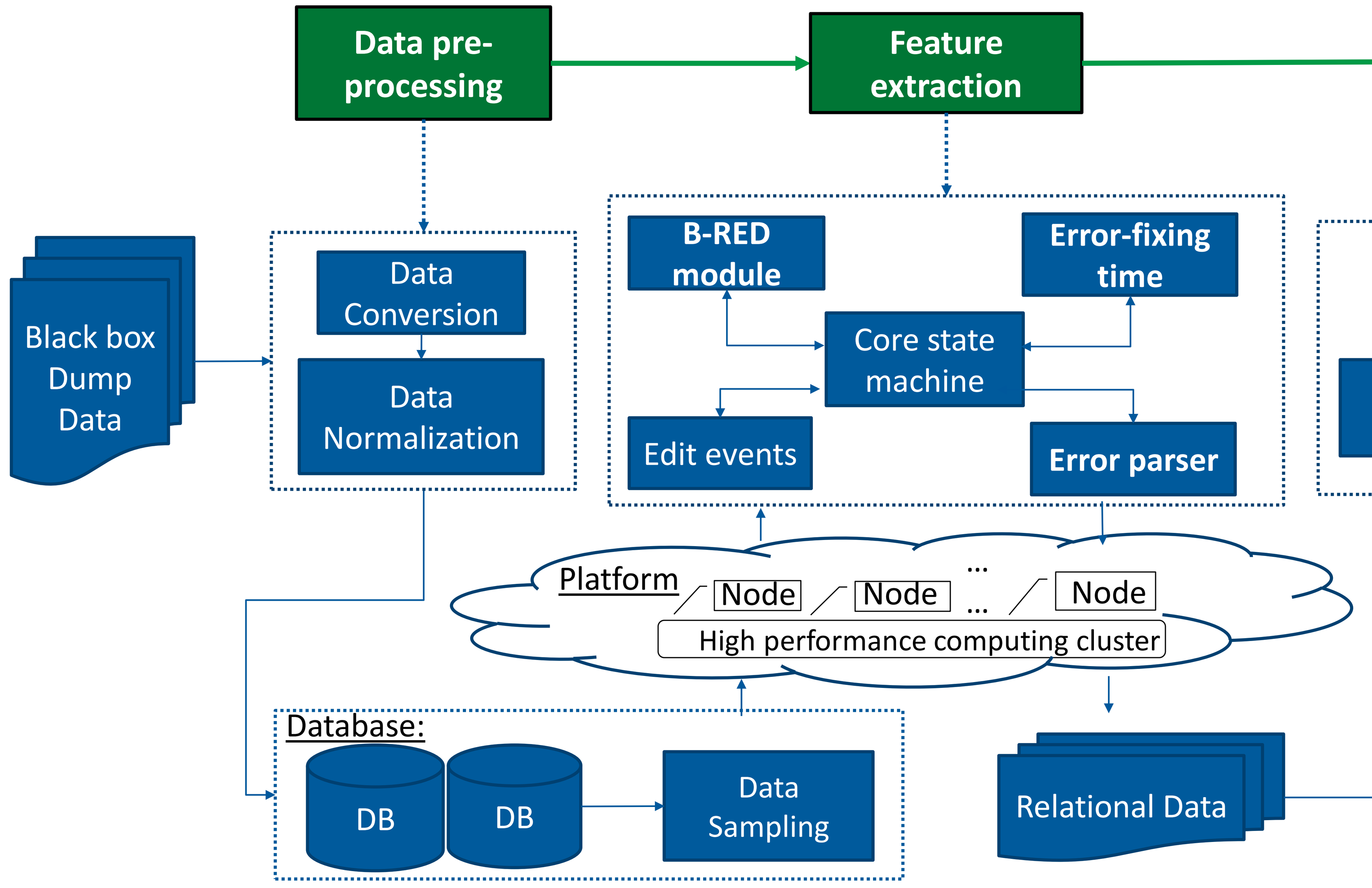


Figure 1: System Architecture

B-RED:

Modified Repeated Error Density (B-RED) algorithm to fit into BlueJ project: $RED = \sum_{i=1}^{n} \frac{r_i^2}{r_i+1}$

Event	Sequence s	x	r	RED
A	… x x …	2	1	0.5
B	… x x x …	3	2	1.33
C	… x x … x x	4	2	1

Table 1: RED Metric

Error Parser:

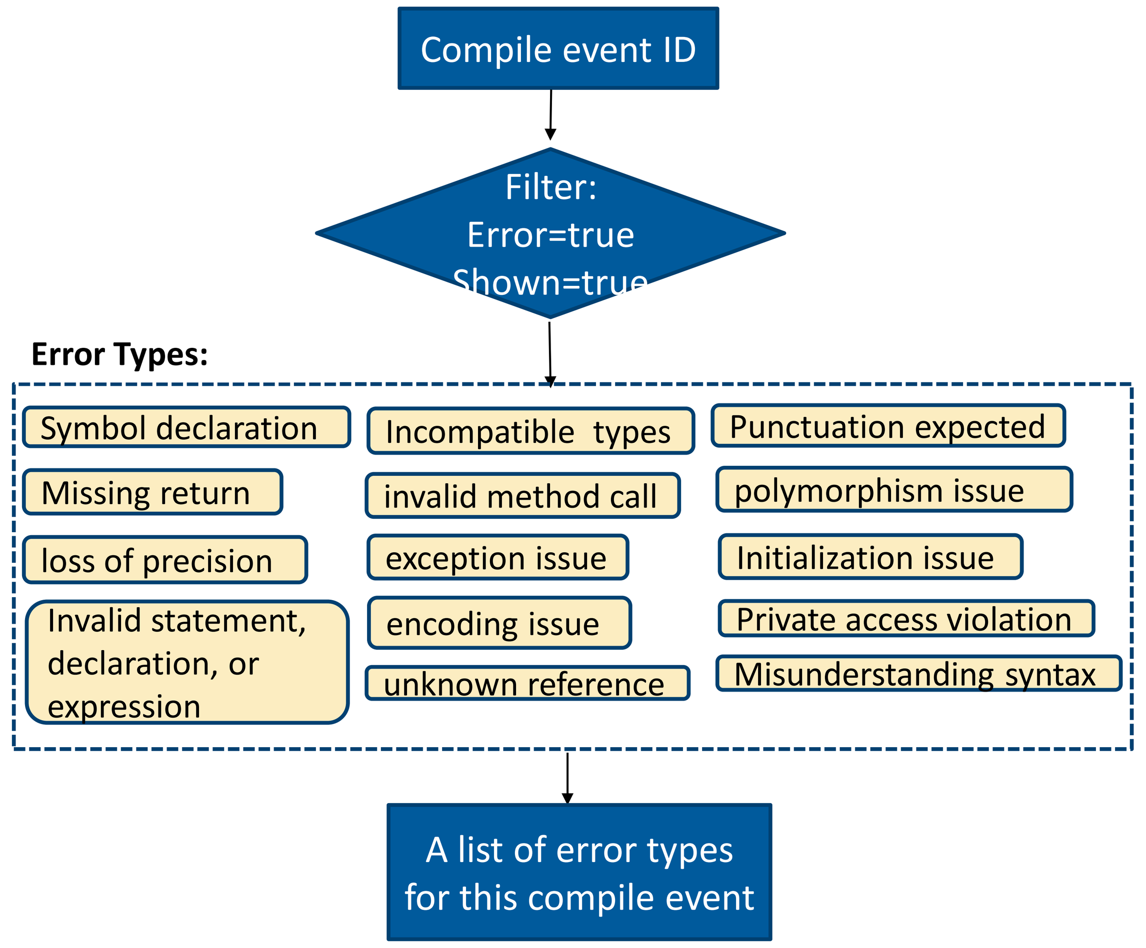


Figure 2: Error Parser Structure

Error-fixing time:

$T = \frac{\sum_{i=1}^{n}(t_s - t_e)}{n}$

Where t_s is error occurrence time, t_e is error session end time and n is the number of error types

Statistical model:

- Correlation coefficient:

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$

- Regression coefficient:

$$r = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2}$$

- Hypothesis test:

$$\text{T-value} = \frac{\bar{x}-\mu}{std\sqrt{n}}$$

Experimental Results

Cluster	ID	Significance level	Regression coefficient	Number of users	Percentage
P	P1	High	(-∞, -1)	557	55.53%
	P2	Medium	[-0.5, -1]	250	24.22%
	P3	Low	(-0.5, 0)	218	21.40%
N	N1	High	[1, ∞)	614	55.53%
	N2	Medium	[0.5, 1)	251	22.64%
	N3	Low	(0, 0.5)	245	22.10%

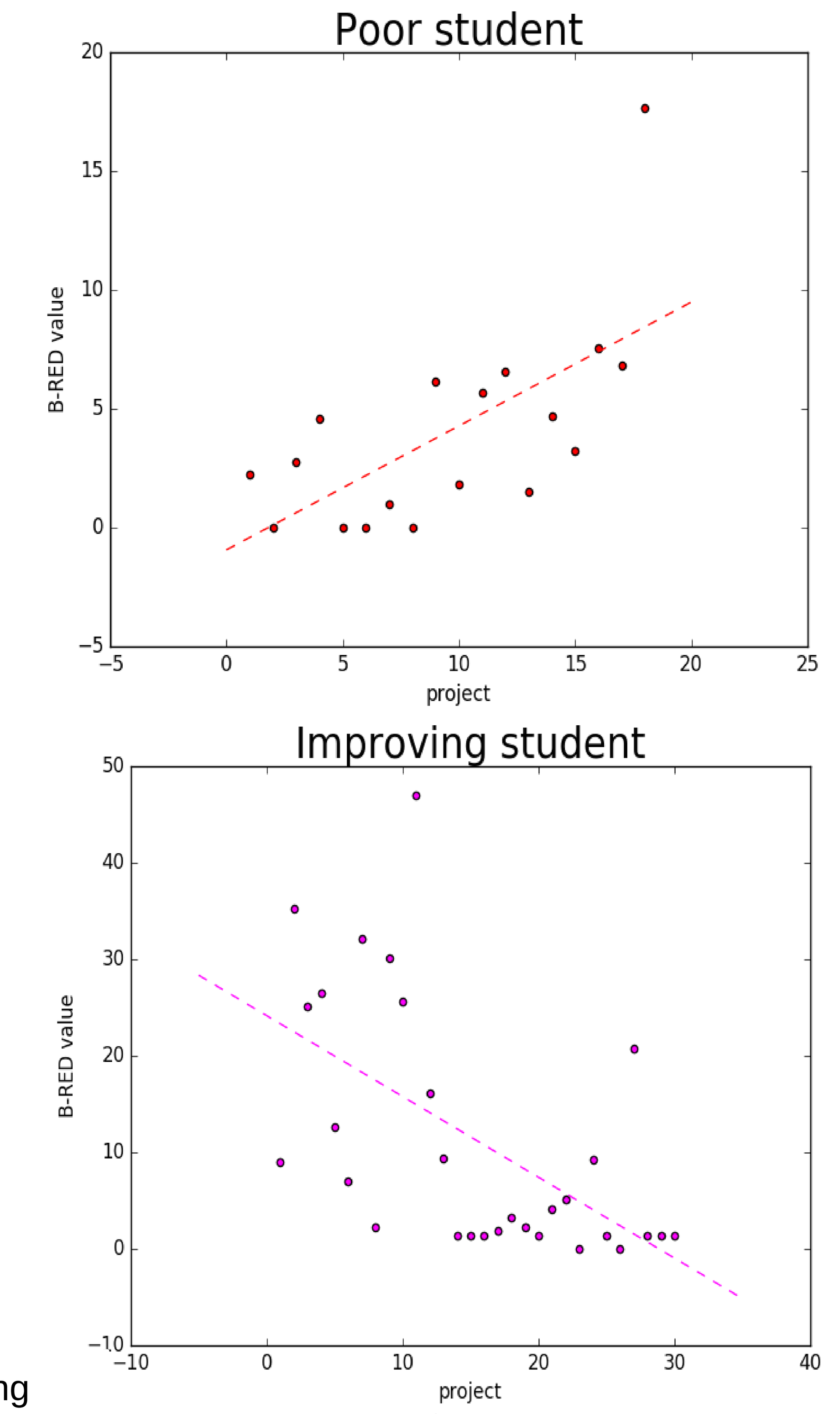
Table 3: Student Clusters

- P group: students with great improvement on their handling skills over project;
- N group: students with declining handling trend.

Error Type	Student Group	Fixing Time	P-value
Invalid statement, declaration or expression	P	75.346	<0.05
	N	116.054	
Private access violation	P	25.664	<0.05
	N	95.152	
Misunderstanding syntax	P	37.424	<0.05
	N	69.386	
Invalid method call	P	55.113	<0.05
	N	251.926	

Table 4: Analysis Result of Student Fixing Time

The differences of fixing times for the error types in table 4 are the underlying reasons why P group outperforms N group.



Figurer 3: Examples of the Students in P and N groups

Project Conclusion

- Analyse users' continuous behaviours with learning mechanism
- Identify the patterns of problem-solving in the student group with high performance
- Validate that the good students handle OOP errors better than other group
- Propose the data analysis methodology for BlueJ project
- The first detailed analysis of student error-handling abilities with Blackbox database

Reference

1. Becker, B. A. (2016b). A new metric to quantify repeated compiler errors for novice programmers. Proceedings of the 21st Annual Conference on Innovation and Technology in Computer Science Education. ACM.