

Training a Machine-Learning Model for Optimal Fitness Function Selection with the Aim of Finding Bugs

Introduction

- Testing plays a **key role** in software development
- **EvoSuite** - test case generating tool
- **Coverage criteria** – Branch, Line, Input, Output, Exception, etc.
- **Input Diversity** – measures the diversity of the inputs used by the test cases [1].
- **Branch Coverage** – measures how close a test is to covering a branch [1].
- **Default coverage criteria** – combination of Line, Branch, Output, Weak Mutation, CBranch, Exception, Method and Methods without Exception Coverage

Research Question

When and how does Input Diversity affect the number of bugs detected when combined with Branch Coverage?

Methodology

- 346 classes taken from **SF-110 Corpus of Classes** [2] and **Apache Commons**
- Data Filter – **statistical significance, Vargha and Delaney effect size measurement**
- Feature selection – **Random Forest and MRMR**

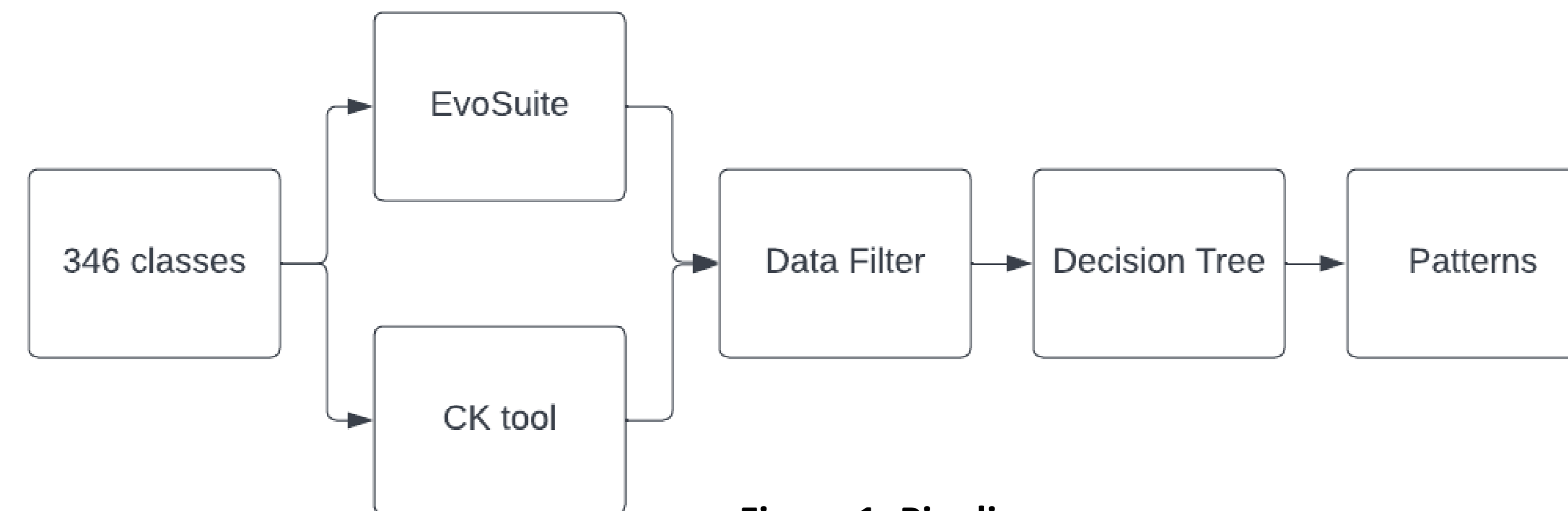


Figure 1: Pipeline

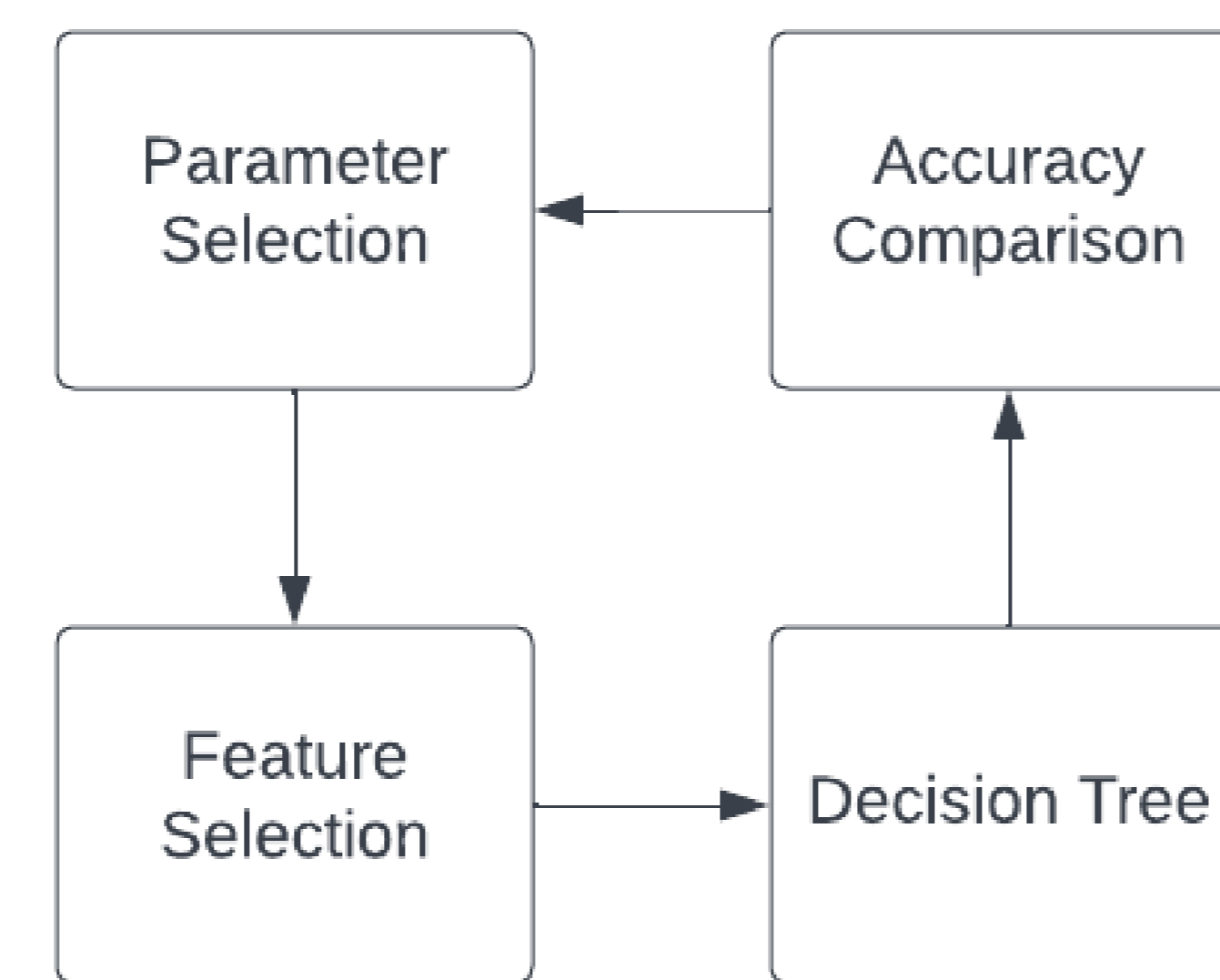


Figure 2: Model Optimization

Results

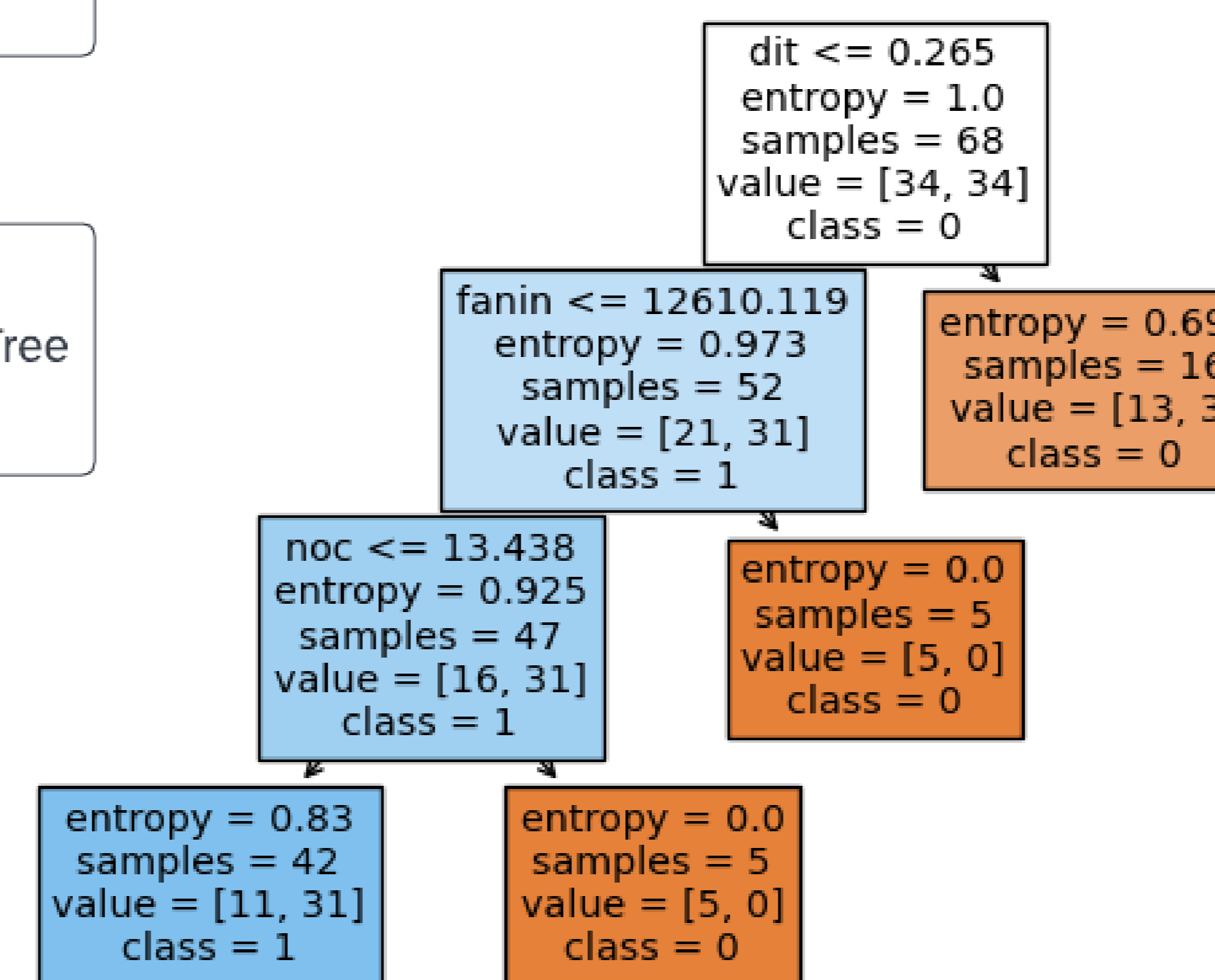


Figure 3: Decision tree for Input Diversity with Branch Coverage against Branch Coverage for 60 seconds when looking at mutation score

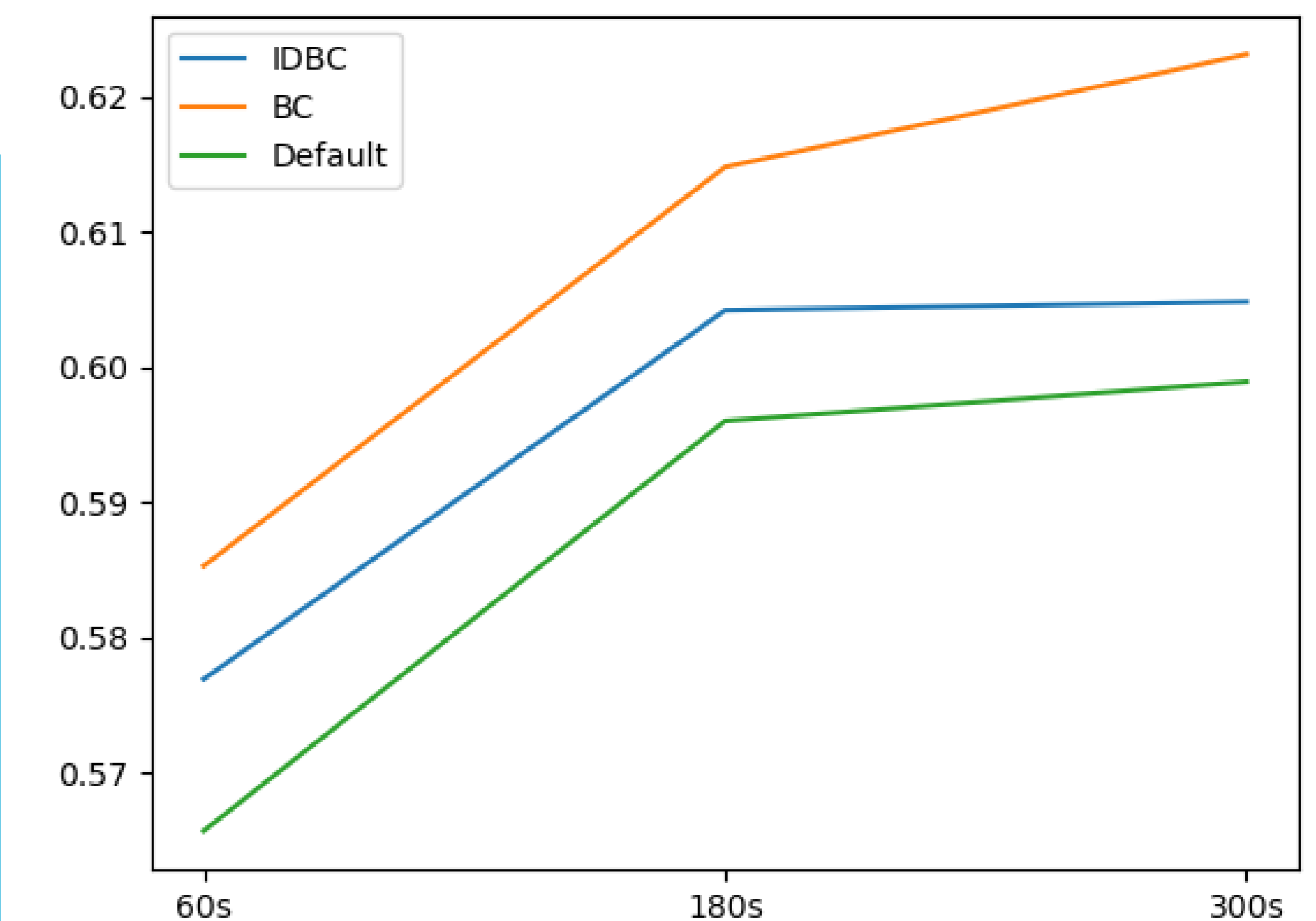
	Time budget	#Win				#Lose			
		Large	Medium	Small	Negligible	Large	Medium	Small	Negligible
IDBC vs BC	60s	4	2	0	0	16	2	0	0
	180s	8	1	0	0	17	3	0	0
	300s	7	1	0	0	22	4	0	0
IDBC vs DCC	60s	55	1	0	0	8	1	0	0
	180s	60	2	0	0	9	2	1	0
	300s	52	3	1	0	13	2	0	0

Table 1: Vargha and Delaney effect size measurement for branch coverage

	Time budget	#Win				#Lose			
		Large	Medium	Small	Negligible	Large	Medium	Small	Negligible
IDBC vs BC	60s	51	2	0	0	11	2	0	0
IDBC vs DCC	60s	12	0	0	0	94	2	0	0

Table 2: Vargha and Delaney effect size measurement for mutation scores

Figure 4: Change in the average branch coverage with respect to time



Conclusion

- Input Diversity with Branch Coverage is more effective than the Default coverage criteria for branch coverage
- Input Diversity with Branch Coverage is more effective than Branch Coverage for fault detection capabilities
- Increase in branch coverage diminishes as the time budget is increased.

Future Work

- Include more data
- Try different feature selection methods

References

- [1] Annibale Panichella, Fitsum Meshesha Kifetew, and Paolo Tonella. 2018. Incremental Control Dependency Frontier Exploration for Many-Criteria Test Case Generation. In Search-Based Software Engineering, Thelma Elita Colanzi and Phil McMinn (Eds.). Springer International Publishing, Cham, 309–324.
- [2] Gordon Fraser and Andrea Arcuri. 2014. A Large Scale Evaluation of Automated Unit Test Generation Using EvoSuite. ACM Transactions on Software Engineering and Methodology (TOSEM) 24, 2 (2014), 8.