

Analysis of Machine Learning Approaches

Bauhaus Uni Weimar

Jula McGibbon

jula.mcgibbon@uni-weimar.de

Approaches

Analyzed ML Approaches

- CART (R)
- SARKAR (R)
- SPL Conqueror (C#)
- SPL Conqueror Active Learning (C#)

Cross Analysis program

- Written in python, running on OSX or Windows
- Runs for xml or csv files
- **Script Configuration options:**
 - interactive script configuration
 - all script configurations in one input file
 - automatic script configuration generation (running all possible combinations)
- **Outputs:**
 - All tested configurations and results in one log file
 - Plots: Graphs, Boxplots, Bar charts
 - Overview: Mean, Median, Standard Deviation, Variance

Analysis

Machines:

- **ASUS GeForce GTX 1080 Ti STRIX OC GAMING**

- GPU: 1594 MHz

- RAM: 11.1 GHz, 11264 MB

- **MacBook Air**

- GPU: 1,6 GHz Intel Core i5

- RAM: 1600 MHz, 8 GB

Tested for all approaches:

- Minimum Improvement Per Round: [0.001, 0.01]
- Number Of Rounds: [10, 30, 60,80]
- Number Of Repetitions Per Round: [1,2,3]

Extented Configurations for SPL Conqueror:

- Negative featurewise
- featurewise
- pairwise
- random (only tested on Mac Book Air)

Extented Configurations for SPL Conqueror Active Learning:

- **explorer-random** batchSize: [5,15,30] sleepCycles: [1,5,10]
- **explorer-max-distance** batchSize:[5,15,30] sleepCycles:[1,5,10]
- **explorer-max-error** internalRoundsPerCycle:[1,5,15]
batchSizeExploit:[5,15] batchSizeExplore:[5,15] sleepCycles:[1]
sleepRoundsExplore:[5,10]
- **explorer-combi** batchSize:[5,15,30] sleepCycles:[1,5,10]
- **explorer-omni**

For one data set tested configurations:

- CART: 24
- SARKAR: 24
- SPL Conqueror (random excluded): 64
- SPL Conqueror Active Learning: 432

Analysis results: Apache Data

For each approach:

- For each execution one fault rate graph is generated
- For all executions one fault rate graph is generated
- For all top/flop fault rates one summarizing graph is generated

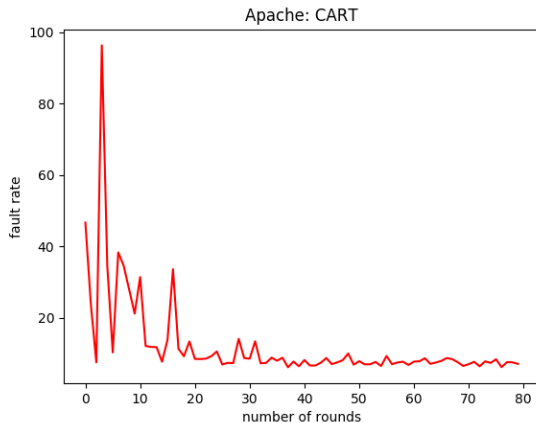


Figure 1: Iteration 24, first of 2 repetitions, minIPR: 0.01, numberOfRounds:80 (info: logAll_CART.R)

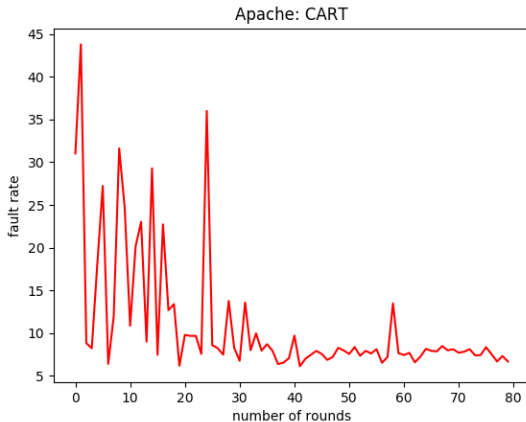


Figure 2: Iteration 24, second of 2 repetitions, minIPR: 0.01, numberOfRounds:80 (info: logAll_CART.R)

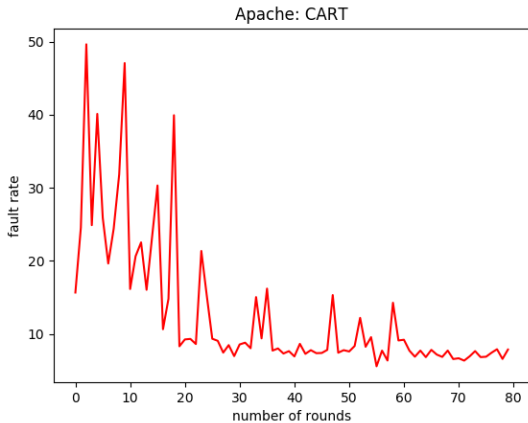


Figure 3: Iteration 24, third of 2 repetitions, minIPR: 0.01, numberOfRounds:80 (info: logAll_CART.R)

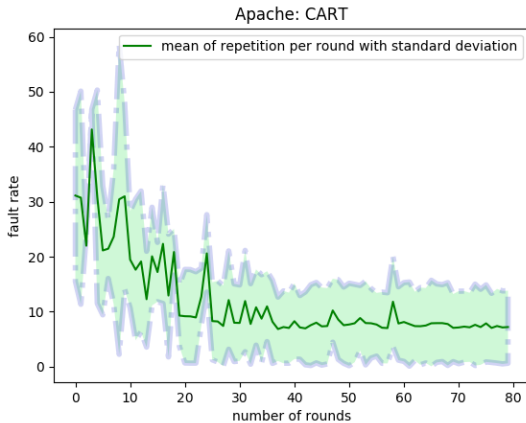


Figure 4: Iteration 24, mean of all 3 executions

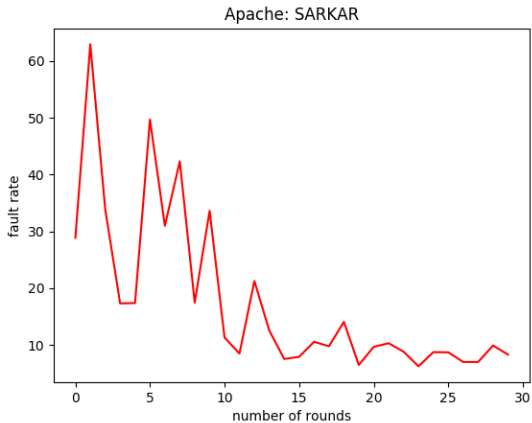


Figure 5: Iteration 22, first of 2 repetitions, minIPR: 0.01, numberOfRounds:30 (info: logAll_SARKAR.R)

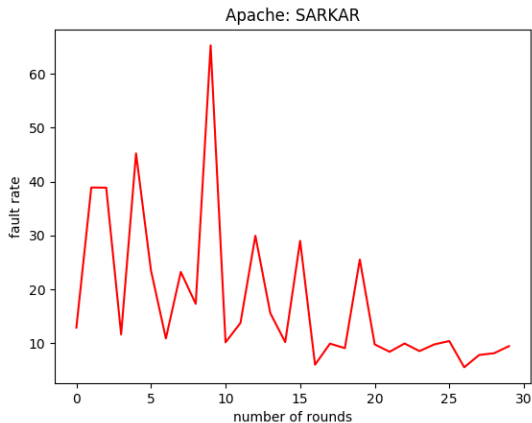


Figure 6: Iteration 22, second of 2 repetitions, minIPR: 0.01, numberOfRounds:30 (info: logAll_SARKAR.R)

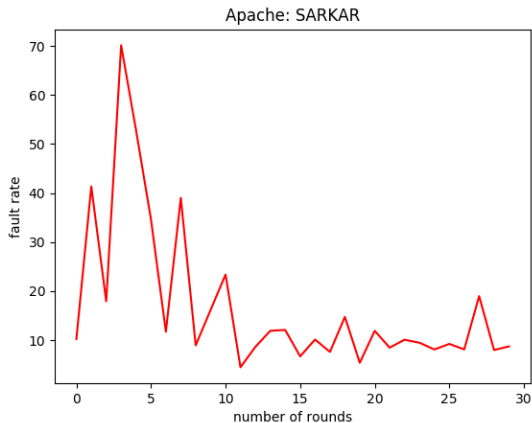


Figure 7: Iteration 22, third of 2 repetitions, minIPR: 0.01, numberOfRounds:30 (info: logAll_SARKAR.R)

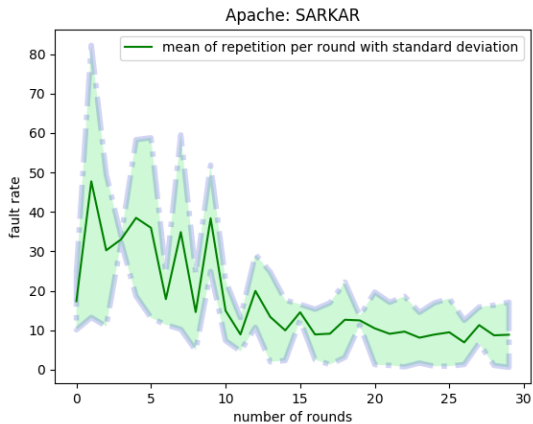


Figure 8: Iteration 22, mean of all 3 executions

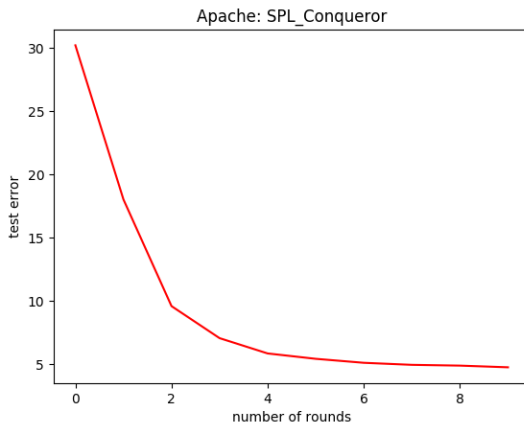


Figure 9: Iteration 25, minIPR: 0.001, numberOfRounds:10, featureWise+pairWise (info: logAll_SPLC.R)

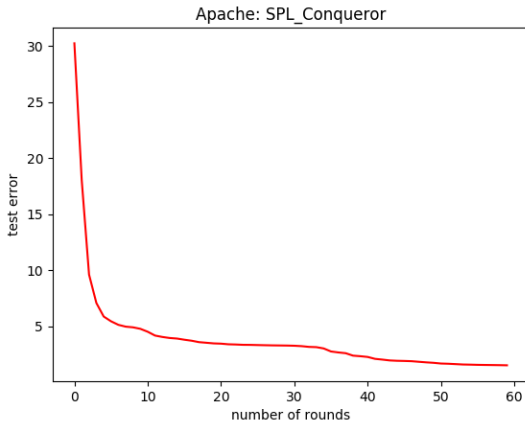


Figure 10: Iteration 27, minIPR: 0.001, numberOfRounds:60, featureWise+pairWise (info: logAll_SPLC.R)

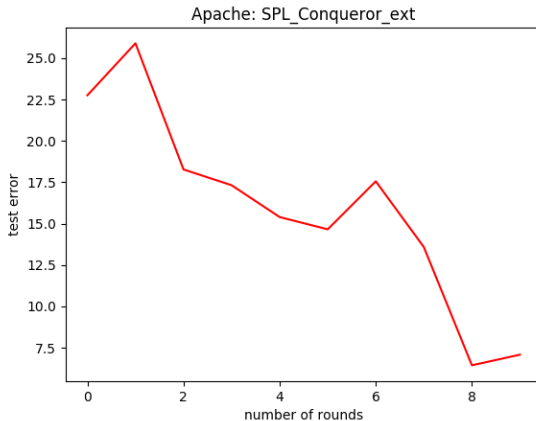


Figure 11: Iteration 153, minIPR: 0.001, numberOfRounds:10, explorer-combi (info: logAll_SPLCext.R)

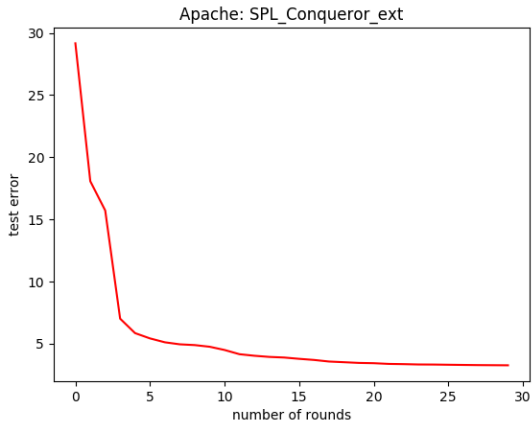


Figure 12: Iteration 146, minIPR: 0.001, numberOfRounds:30, explorer-max-error (info: logAll_SPLCext.R)

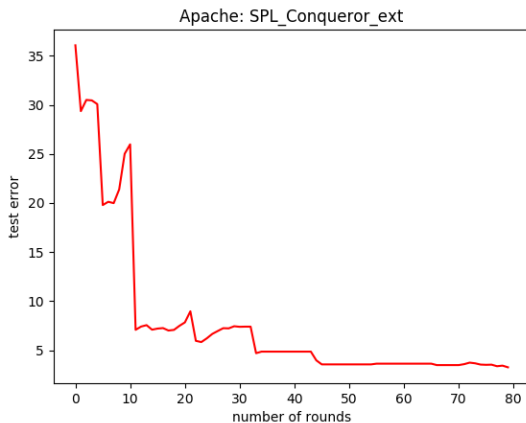


Figure 13: Iteration 164, minIPR: 0.001, numberOfRounds:80, explorer-random (info: logAll_SPLCext.R)

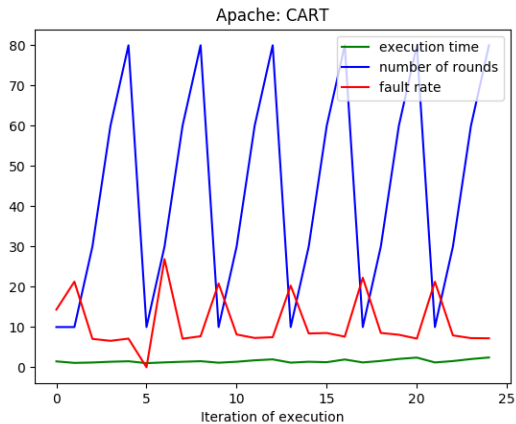


Figure 14: All iterations

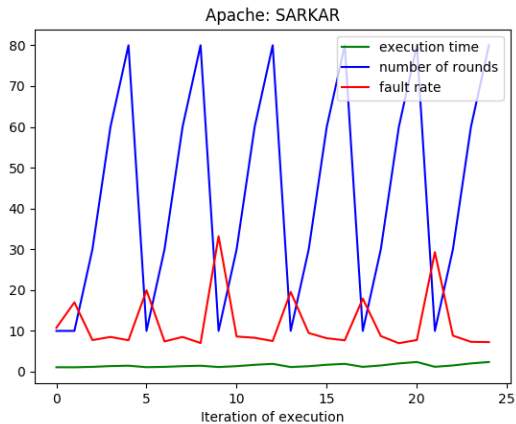


Figure 15: All iterations

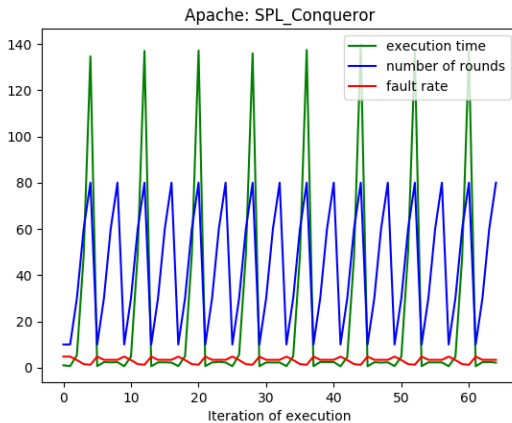


Figure 16: All iterations

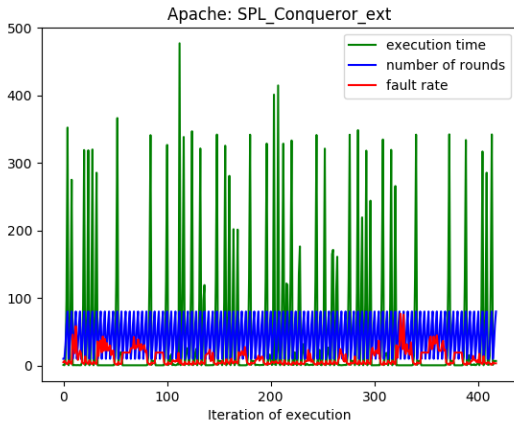
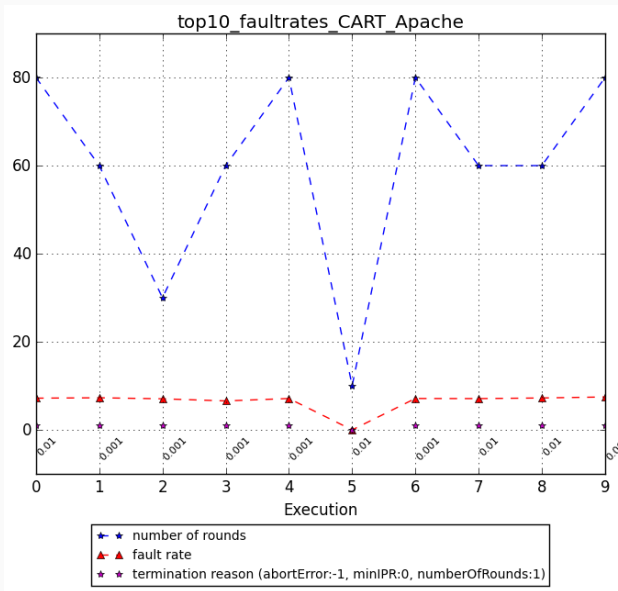


Figure 17: All iterations

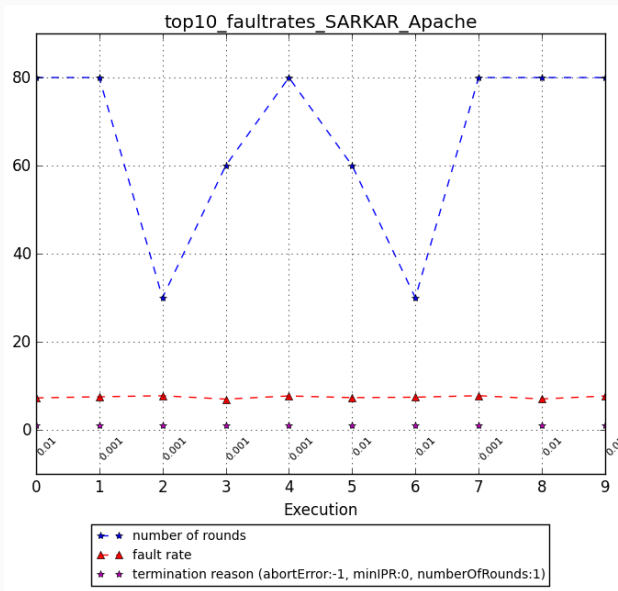
In all Top/Flop results:

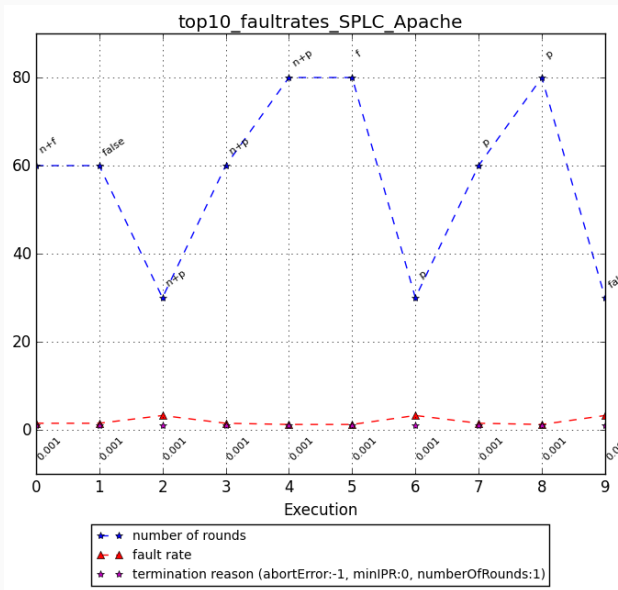
- If the same value (e.g. fault rate) is included in the data, it is only shown once in the plot

Graphs

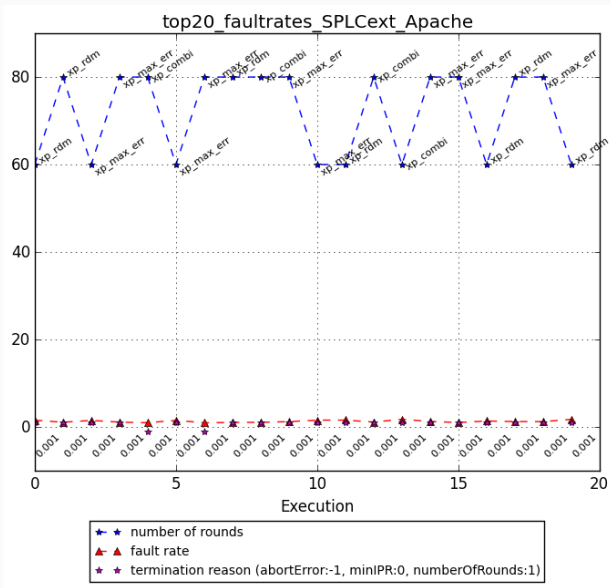


Graphs

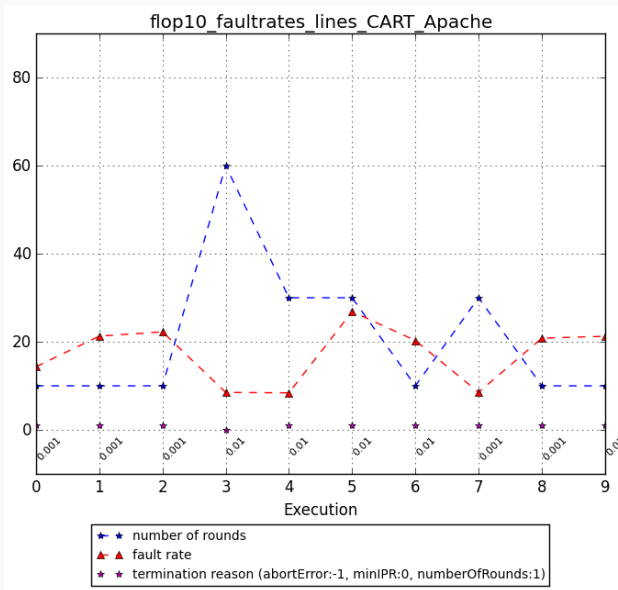




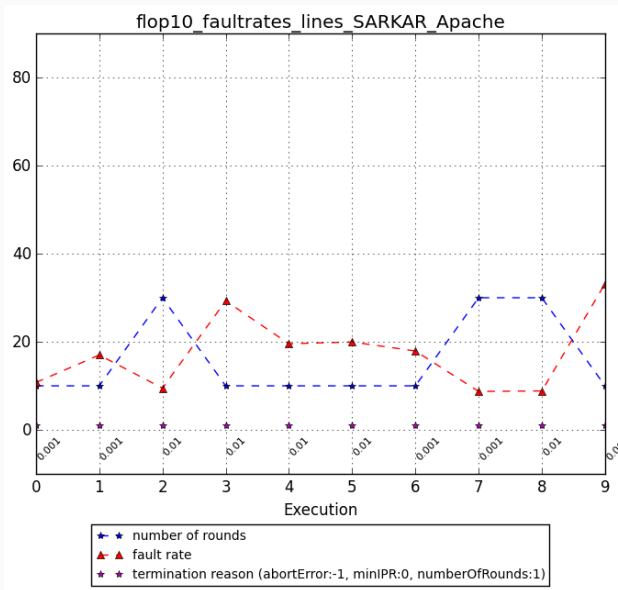
Graphs



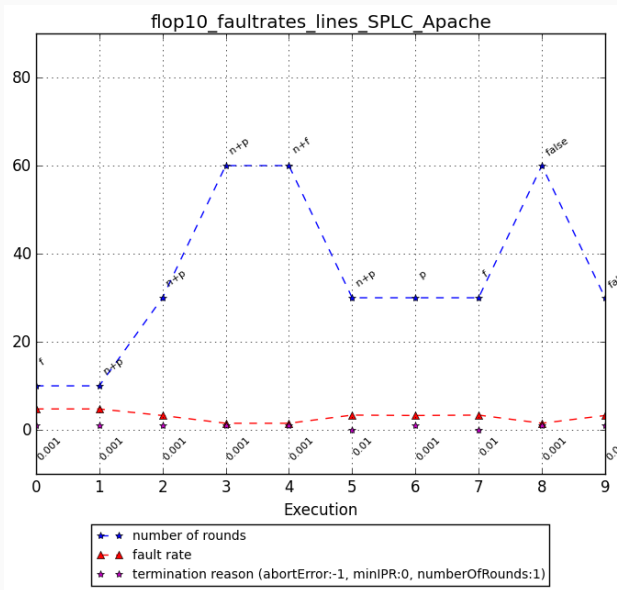
Graphs



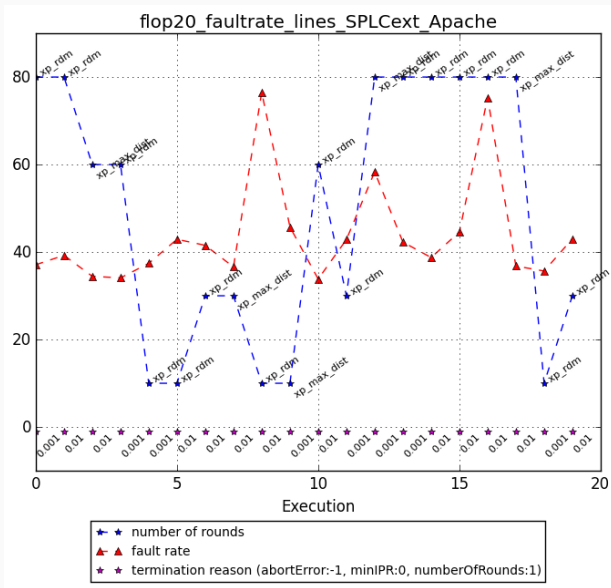
Graphs



Graphs



Graphs



For each approach:

- All execution times compared in two box plots
- Top/Flop execution times in two box plots
- All fault rates of all approaches compared in one boxplot
- Fault rates of Top/Flop execution times compared in two box plots
- Fault rates of Top/Flop fault rates compared in one box plot

Boxplots

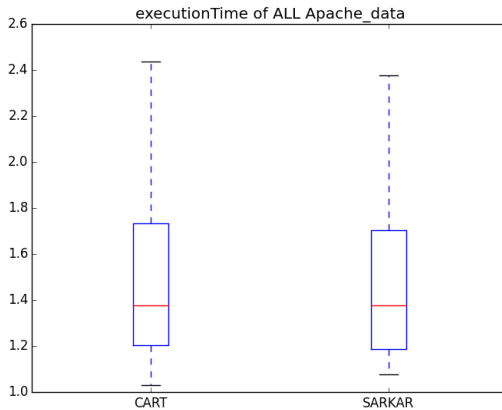


Figure 18: All iterations

Boxplots

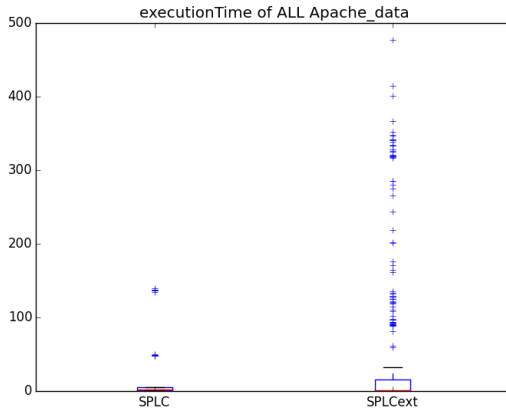


Figure 19: All iterations

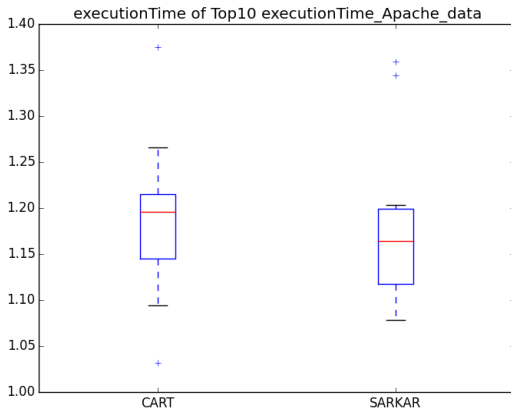


Figure 20: Top 10 execution times

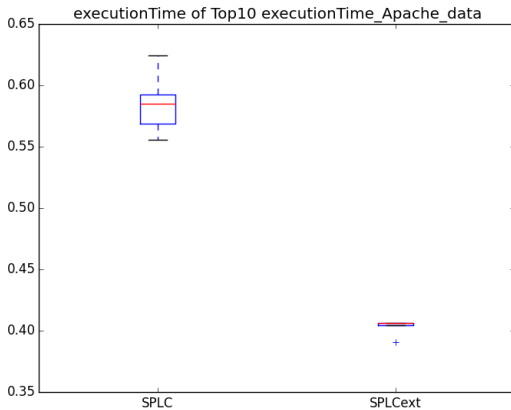


Figure 21: Top 10 execution times

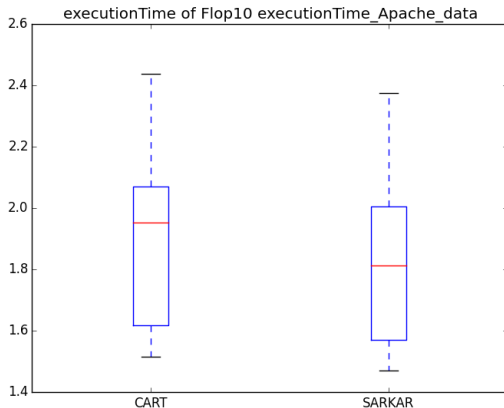


Figure 22: Flop 10 execution times

Boxplots

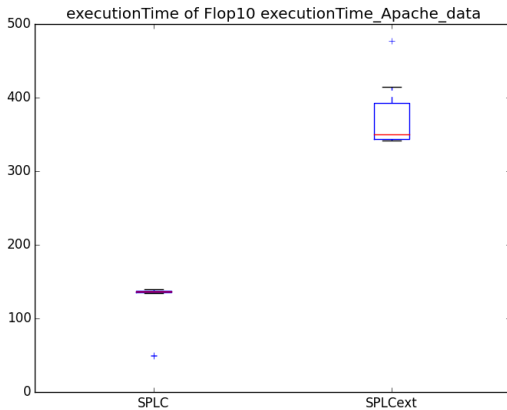


Figure 23: Flop 10 execution times

Boxplots

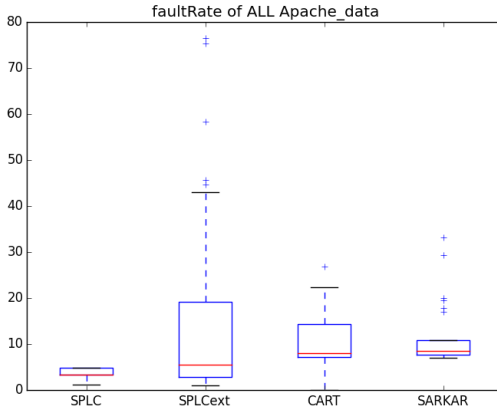


Figure 24: All iterations

Boxplots

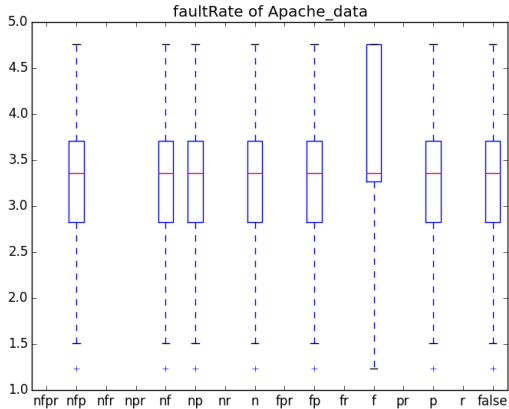


Figure 25: SPLC Conqueror; All iterations

Boxplots

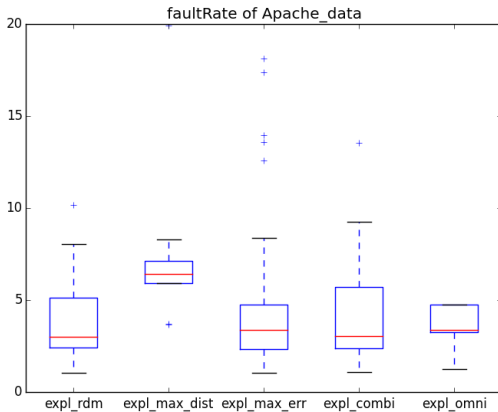


Figure 26: SPLC Conqueror Active Learning; All iterations

Boxplots



Figure 27: Fault rates of Top 10 execution times

Boxplots

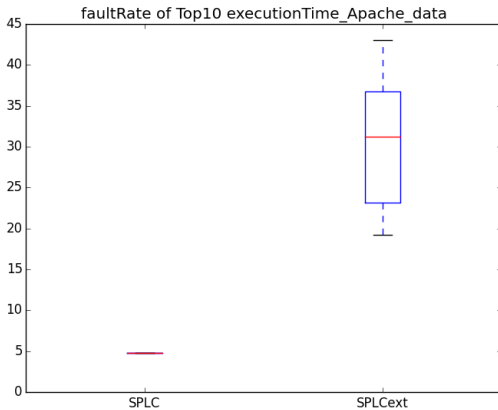


Figure 28: Fault rates of Top 10 execution times

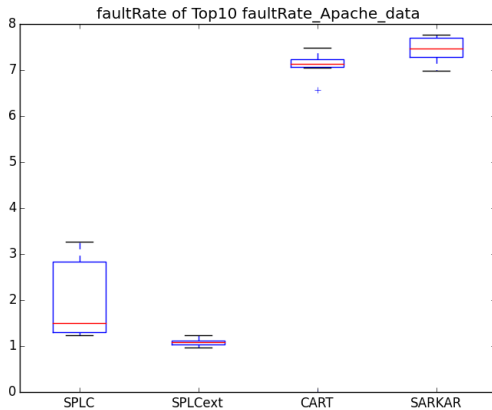


Figure 29: Fault rates of Top 10 fault rates

Boxplots

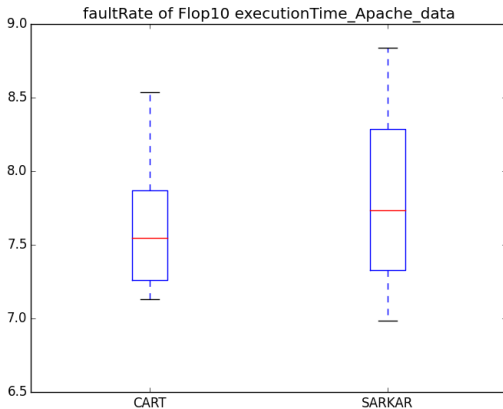


Figure 30: Fault rates of Flop 10 execution times

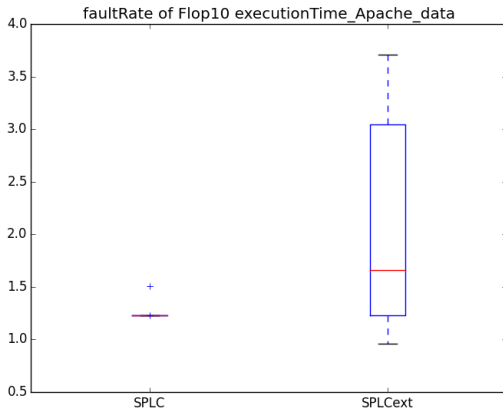


Figure 31: Fault rates of Flop 10 execution times

Boxplots

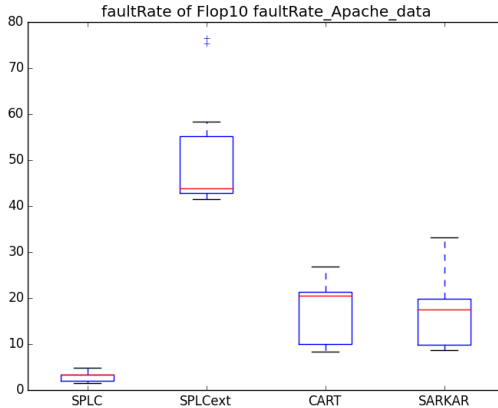


Figure 32: Fault rates of Flop 10 fault rates

For each approach:

- Frequency of all SPL Conqueror configurations in Top/Flop Fault rates in one bar chart
- Frequency of all SPL Conqueror mixed configurations in Top/Flop Fault rates in one bar chart
- Frequency of all SPL Conqueror Active Learning approaches in Top/Flop Fault rates in one bar chart

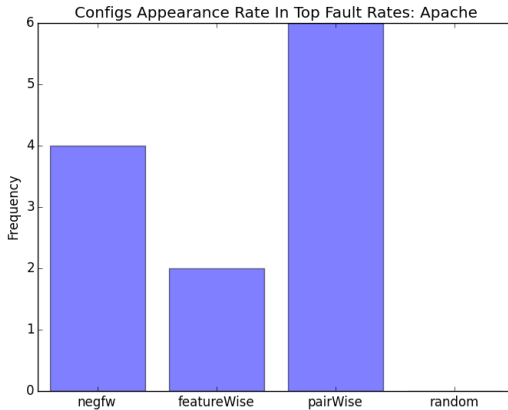


Figure 33: SPL Conqueror

Bar chart

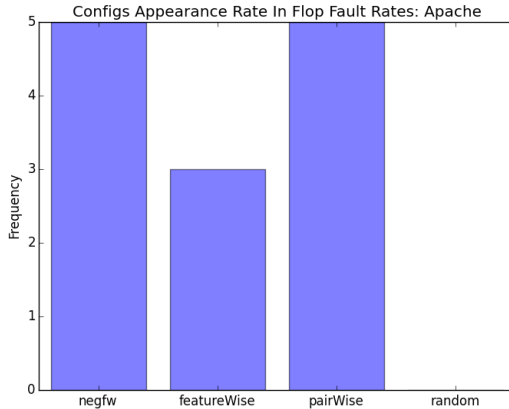


Figure 34: SPL Conqueror

Bar chart

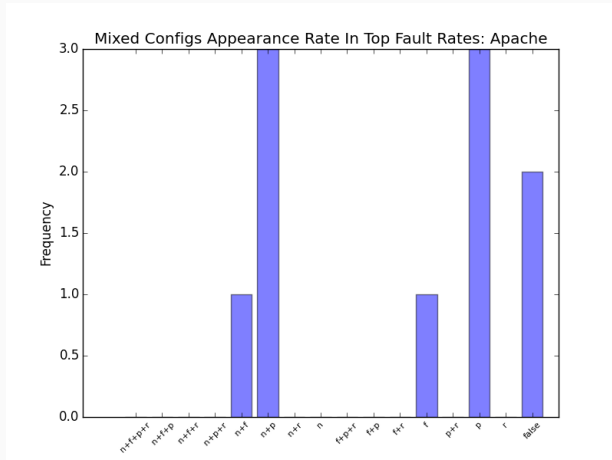


Figure 35: SPL Conqueror

Bar chart

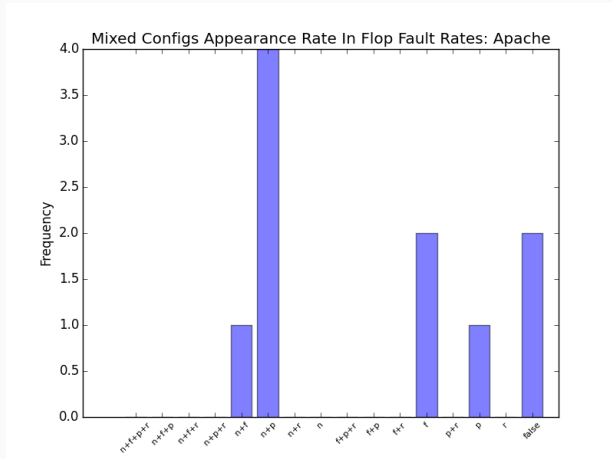


Figure 36: SPL Conqueror

Bar chart

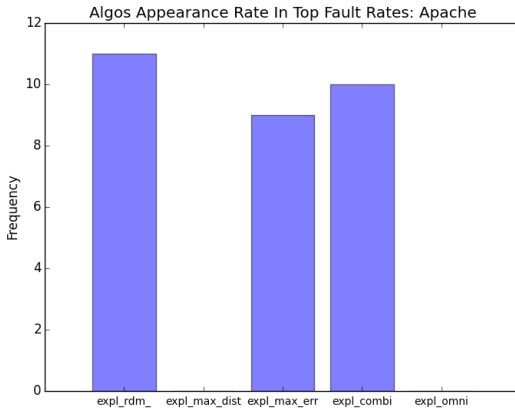


Figure 37: SPL Conqueror Active Learning

Bar chart

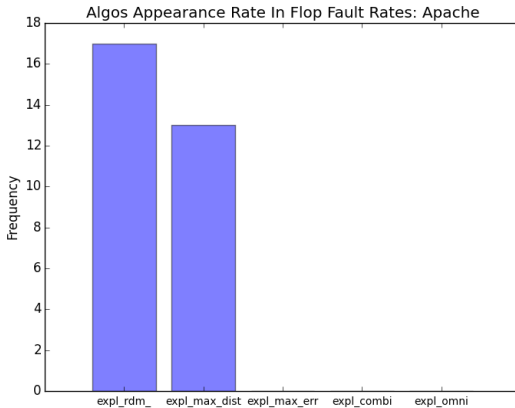
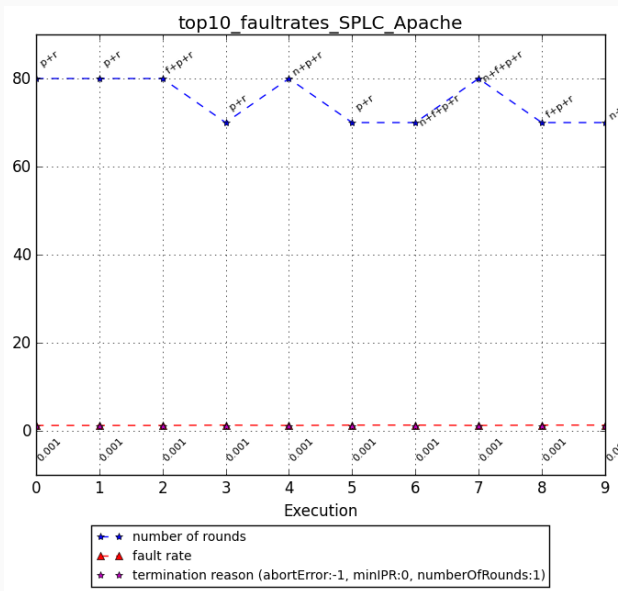


Figure 38: SPL Conqueror Active Learning

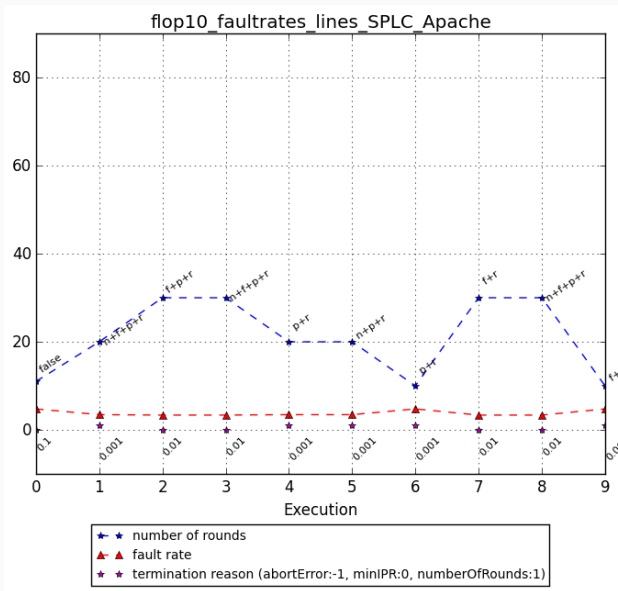
Since the random function of SPL Conqueror didn't work on the main analysis Windows machine (ASUS GeForce GTX), some plots were generated for the data generated on a OSX system (Mac Book Air).

1040 configurations were tested.

Mac data: Graphs



Mac data: Graphs



Mac data: Bar chart

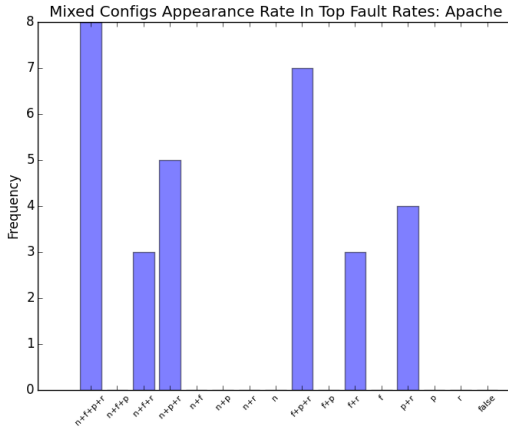


Figure 41: SPL Conqueror

Mac data: Bar chart

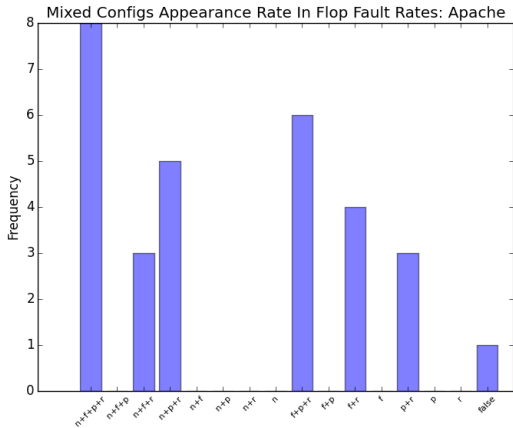


Figure 42: SPL Conqueror

Analysis results: Apache and H264 data

Apache_data				
faultRate	CART	SARKAR	SPLC	SPLC AL
-----	-----	-----	-----	-----
mean	11.051	11.656	3.224	10.826
median	7.926	8.522	3.361	5.467
standard dev	6.791	7.148	1.224	11.356
variance	46.118	51.09	1.498	128.95
H264_data				
faultRate	CART	SARKAR	SPLC	SPLC AL
-----	-----	-----	-----	-----
mean	9.955	8.622	3.369	11.182
median	7.804	7.262	3.214	6.175
standard dev	4.759	3.57	0.344	14.288
variance	22.645	12.745	0.119	204.147

Figure 43: Overview; Fault rates

Apache_data

executionTime	CART	SARKAR	SPLC	SPLC AL
mean	1.529	1.513	24.464	46.622
median	1.375	1.375	2.305	1.039
standard dev	0.403	0.395	45.15	98.025
variance	0.162	0.156	2038.48	9608.83

H264_data

executionTime	CART	SARKAR	SPLC	SPLC AL
mean	1.66	1.631	36.743	20.027
median	1.485	1.469	14.844	3.579
standard dev	0.538	0.489	36.77	72.252
variance	0.29	0.239	1352.03	5220.41

Figure 44: Overview; Execution times

Conclusion

CART and SARKAR

- Don't converge fast towards a very small fault rate
- But reach quite good fault rates with high number of rounds
- Very fast execution times

SPL Conqueror

- Converges fast towards a very small fault rate
- Good execution times with a few outliers
- Reaches best results with combinations of featurewise, negativewise, pairwise and random sampling

SPL Conqueror

- Converges quite fast towards a very small fault rate
- Good execution times with a some outliers
- All explorer approaches reach quite the same results, only explorer max distance is a bit worse

Thanks for your attention.

References



https://www.github.com/yoola/Cross_ML