Solving machine learning with machine learning: Exploiting Very Large-Scale Neighbourhood Search for synthesizing machine learning pipelines

Author
Auke Sonneveld
a.c.sonneveld@student.tudelft.nl

Professor
Sebastijan Dumančić
s.dumancic@tudelft.nl

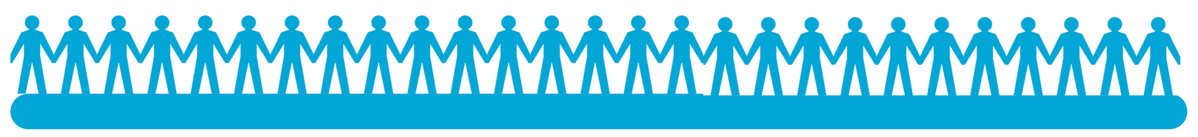
Supervisor
Tilman Hinnerichs
T.R.Hinnerichs@tudelft.nl

1. Motivation

ML is very powerful!

Only 300.000 know how to use ML

AutoML aims to make that 7.8 BILLION



2. Research Question

How good is Very Large-Scale Neighbourhood search (VLSN) at searching for high-performing pipelines in a grammar, compared to other algorithms?

4. Experimental Setup

1. Training:validation:test = 70:15:15
2. Limited training set during search
3. Stop after 100 pipelines
4. 10 runs per algorithm per dataset

5. Results

Metrics

- a. Average Accuracy
- b. Standard Deviation

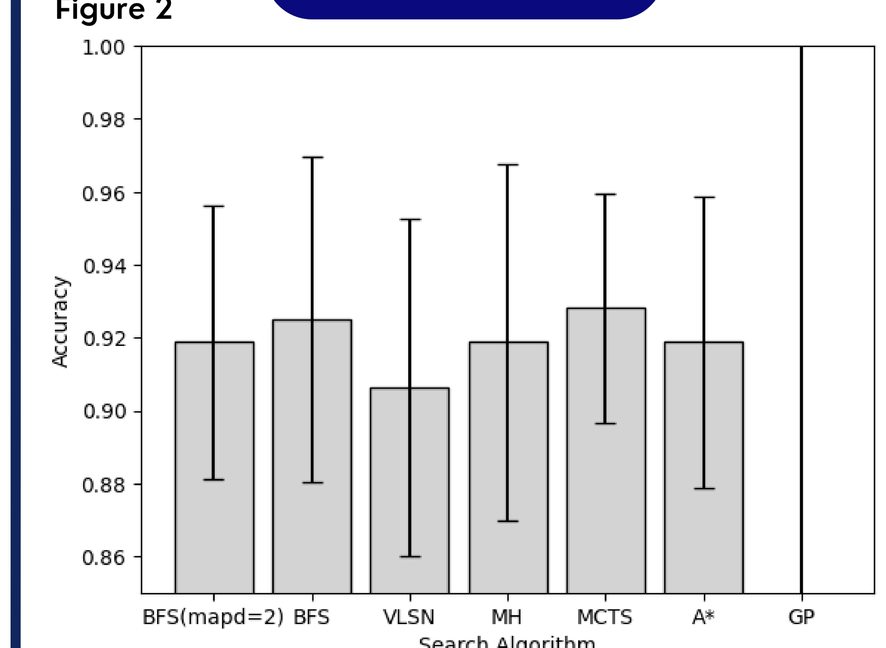
Algorithms

- a. BFS = Breadth-First Search
- b. MH = Metropolis-Hastings
- c. MCTS = Monte-Carlo Tree Search
- d. A* = A* Search
- e. GP = Genetic Programming
- f. BFS2 = BFS with depth limit 2

Average accuracy and standard deviation achieved on ten runs by different search algorithms on three datasets

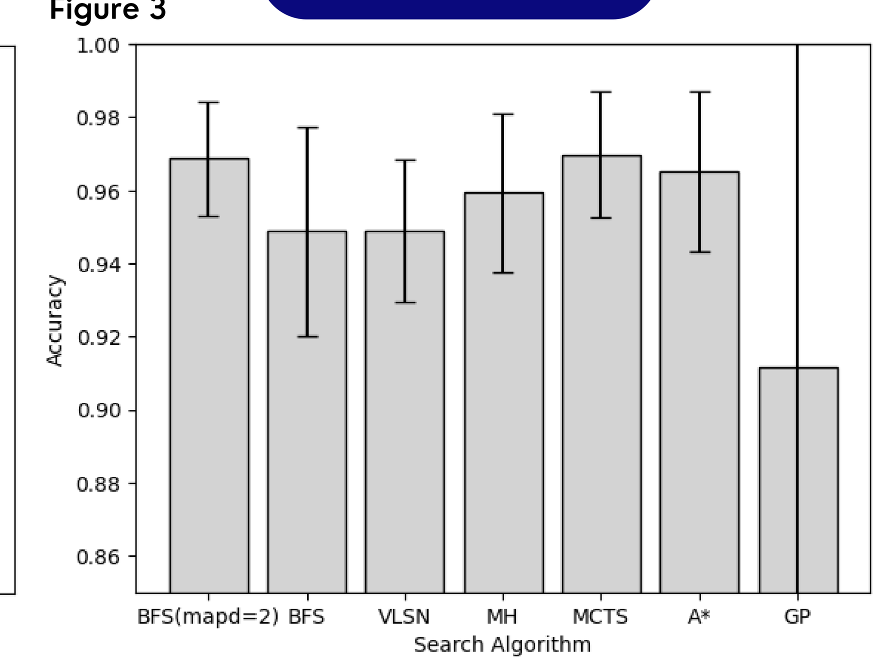
seeds dataset

Figure 2



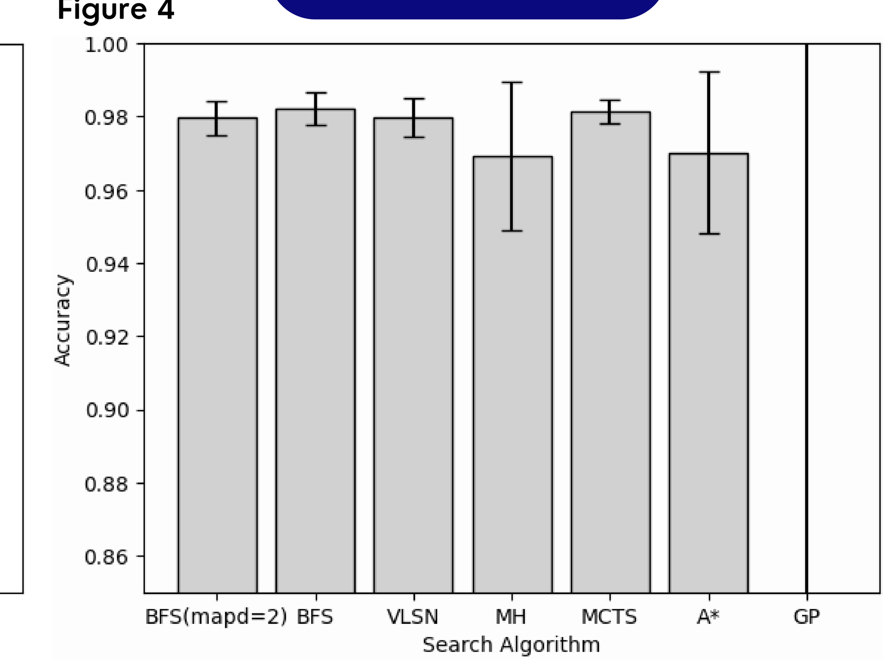
wdbc dataset

Figure 3



har dataset

Figure 4



3. Methodology

Datasets

Source

OpenML

Simple datasets

Table 1

Name	ID	Entries	Features	Target Classes
diabetes	37	768	8	2
qsar-biodeg	1494	1055	42	2
seeds	1499	210	7	3
iris	61	150	4	3
blood-transfusion	1464	748	4	2
monks-problems-2	334	601	6	2
ilpd	1480	583	5	2
tic-tac-toe	50	958	9	2

Complex datasets

Table 2

Name	ID	Entries	Features	Target Classes
har	1478	10299	561	6
gisette	41026	7000	5000	2
madelon	1485	2600	501	2
musk	1116	6598	167	2
gas-drift	1476	13910	128	6

Datasets from related research

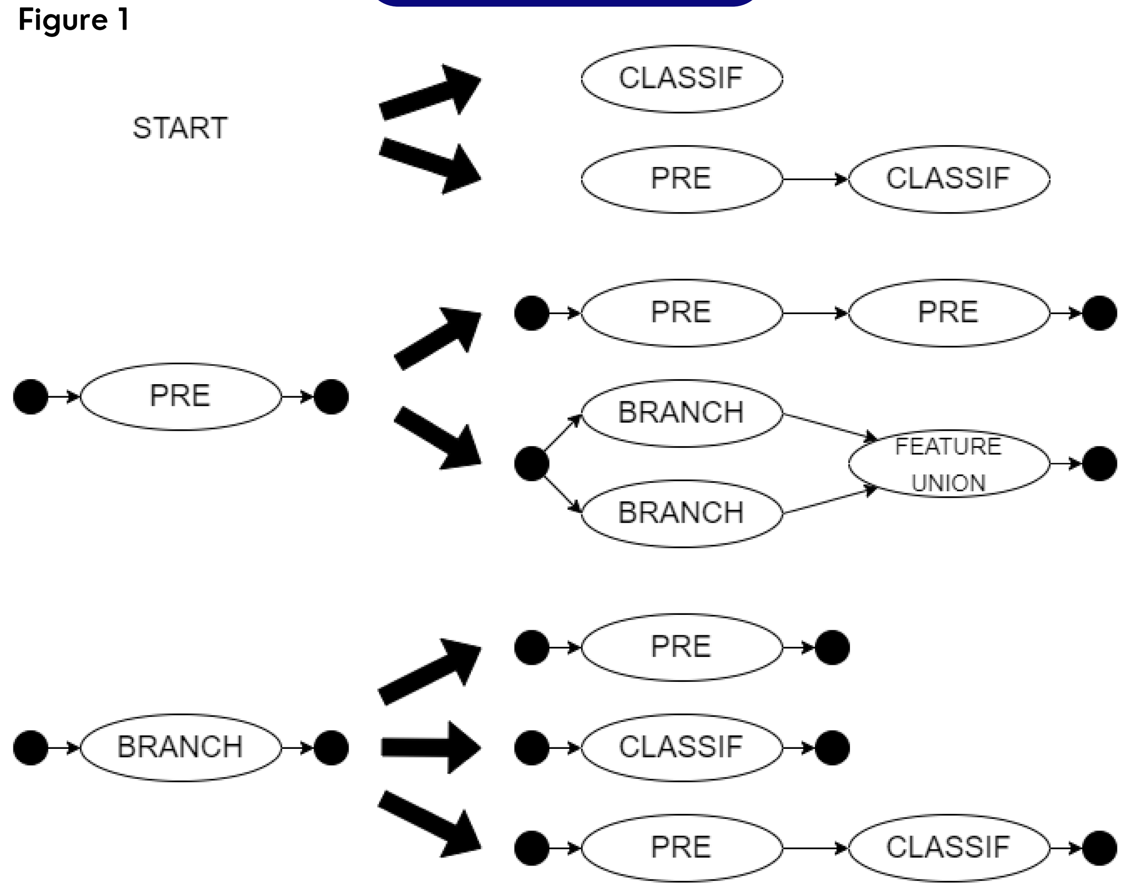
Table 3

Name	ID	Entries	Features	Target Classes
wdbc	1510	569	30	2
glass	41	214	9	6
car-evaluation	40664	1728	21	4
spambase	44	461	57	2
wine-quality-red	40691	1599	11	6
wine-quality-white	40498	4898	11	7

Grammar

Structure

Figure 1



Operators

PREPOC = StandardScalar | RobustScaler | MinMaxScaler | MaxAbsScaler | PCA | Binarizer | PolynomialFeatures

FSELECT = VarianceThreshold | SelectKBest | SelectPercentile | SelectFwe | RFE

CLASSIF = DecisionTreeClassifier | RandomForestClassifier | GradientBoostingClassifier | LogisticRegression | KNeighborsClassifier

6. Conclusions & 7. Future Work

1. Performance of:
   - a. VLSN is adequate
   - b. MCTS is best

   } Small differences and large standard deviation
2. Preprocessing and Feature Selection operators don't improve performance, possibly because:
   - a. Used datasets don't need preprocessing and feature selection
   - b. Default hyperparameters negate preprocessing and feature selection
   - c. Limited training set during search creates bias towards BFS

1. Extend experiment using:
   - a. Broader range of datasets
   - b. Larger number of runs
2. Investigate the:
   - a. Impact of adding hyperparameter tuning for the operators,
   - b. Reliability of pipeline evaluation using a limited training set