



Universiteit  
Leiden  
The Netherlands

## Efficient tuning of automated machine learning pipelines

Nguyen, D.A.

### Citation

Nguyen, D. A. (2024, October 9). *Efficient tuning of automated machine learning pipelines*. Retrieved from <https://hdl.handle.net/1887/4094132>

Version: Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/4094132>

**Note:** To cite this publication please use the final published version (if applicable).

# Efficient Tuning of Automated Machine Learning Pipelines

Proefschrift

ter verkrijging van  
de graad van doctor aan de Universiteit Leiden,  
op gezag van rector magnificus prof.dr.ir. H. Bijl,  
volgens besluit van het college voor promoties  
te verdedigen op woensdag 9 oktober 2024  
klokke 11.30 uur

door

Duc Anh Nguyen  
geboren te Thai Binh, Vietnam  
in 1987

**Promotores:**

Prof.dr. T.H.W. Bäck

Prof.dr. B. Sendhoff (Technical University Darmstadt, Germany)

**Co-promotor:**

Dr. A.V. Kononova

**Promotiecommissie:**

Prof.dr. M.M. Bonsangue

Prof.dr. A. Plaat

Dr. Jan N. van Rijn

Prof.dr. D. Zaharie (West University of Timisoara, Romania)

Prof.dr. G. Ochoa (University of Sterling, Scotland)

Prof.dr. J. Sun (Xi'an Jiaotong University, China)

Copyright © 2024 Duc Anh Nguyen All Rights Reserved.

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 766186.

---

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Problem definition . . . . .	9
1.1.1	Machine Learning Pipeline Optimization . . . . .	10
1.1.2	Combined Algorithm Selection and Hyperparameter Optimization . . . . .	12
1.2	Research Questions . . . . .	13
1.3	Outline of the Dissertation . . . . .	16
1.4	Publications and software packages . . . . .	18
<b>2</b>	<b>Automated Machine Learning: An Overview</b>	<b>21</b>
2.1	Life Cycle of Machine Learning Development . . . . .	21
2.1.1	Data preparation . . . . .	23
2.1.2	Automated Machine Learning Pipeline . . . . .	24
2.1.2.1	Machine learning pipeline . . . . .	24
2.1.2.2	Evaluation measurement metrics . . . . .	26
2.1.3	Over-fitting and under-fitting . . . . .	32
2.2	ML pipeline architecture search . . . . .	35
2.2.1	Fixed ML pipeline architecture . . . . .	35
2.2.2	Flexible ML pipeline architecture . . . . .	36
2.3	Meta Learning . . . . .	37
2.4	Explainable and low-code for AutoML . . . . .	39
2.4.1	Stakeholders of AutoML . . . . .	40
2.4.2	Components of an explainable AutoML . . . . .	41
2.4.3	Maturity Levels of Automation Tools . . . . .	42
2.4.3.1	Tools for data scientist . . . . .	42
2.4.3.2	Tools for software engineer . . . . .	44

## CONTENTS

---

<b>3</b>	<b>An In-Depth Review of AutoML Optimization Approaches</b>	<b>45</b>
3.1	Black-box optimization approaches . . . . .	45
3.1.1	Grid Search . . . . .	46
3.1.2	Random Search . . . . .	47
3.1.3	Bayesian Optimization . . . . .	49
3.1.3.1	Probabilistic Regression Models . . . . .	49
3.1.3.2	Acquisition Function . . . . .	53
3.2	Multi-fidelity approaches . . . . .	54
3.2.1	Racing procedure . . . . .	54
3.2.1.1	Iterated racing (irace) . . . . .	55
3.2.2	Bandit-based approaches . . . . .	55
3.2.2.1	Successive Halving . . . . .	57
3.2.2.2	Hyperband . . . . .	57
<b>4</b>	<b>Setup of Benchmark Experiments</b>	<b>61</b>
4.1	Benchmarking methodology . . . . .	62
4.2	First experiment: class-imbalanced classification problems with two operators . . . . .	64
4.2.1	Datasets . . . . .	65
4.2.2	Resampling Algorithms . . . . .	65
4.2.3	Implementation details . . . . .	67
4.3	Second experiment: AutoML benchmark with up to six operators .	70
4.3.1	Datasets . . . . .	70
4.3.2	Implementation details . . . . .	70
4.3.3	Parameter setting . . . . .	72
<b>5</b>	<b>An Empirical Investigation Comparing CASH Optimization Approaches for Class Imbalance Problems</b>	<b>73</b>
5.1	Introduction . . . . .	74
5.2	Related Works . . . . .	75
5.2.1	Imbalanced Classification . . . . .	75
5.2.2	The Combined Algorithm Selection and Hyperparameter Optimization (CASH) Approach . . . . .	76
5.3	Experimental Setup . . . . .	77
5.4	Results and discussion . . . . .	78
5.5	Conclusions and Future Work . . . . .	83

<b>6</b>	<b>On the use of AutoML optimization in real-world applications</b>	<b>85</b>
6.1	Introduction . . . . .	86
6.2	Background . . . . .	88
6.2.1	Multi-Class Imbalance Learning . . . . .	88
6.2.1.1	One vs. Rest approach . . . . .	89
6.2.1.2	One vs. One approach . . . . .	89
6.2.1.3	Multi-class direct classification . . . . .	90
6.2.2	Performance Metrics . . . . .	90
6.3	Experiments . . . . .	91
6.3.1	Datasets . . . . .	92
6.3.2	Experimental procedure . . . . .	93
6.3.3	Results . . . . .	95
6.4	Conclusion . . . . .	100
<b>7</b>	<b>Efficient AutoML via Combinational Sampling</b>	<b>103</b>
7.1	Introduction . . . . .	103
7.2	The Proposed Approaches for Automated Machine Learning . . . . .	106
7.2.1	Novel combination-based initial sampling for Bayesian optimization for AutoML optimization . . . . .	106
7.2.2	A New Optimization Library for AutoML Optimization . . . . .	110
7.3	Experimental Setup . . . . .	110
7.4	Results and Discussion . . . . .	111
7.4.1	First experimental results . . . . .	111
7.4.2	Results of second experiment . . . . .	114
7.5	Conclusions and Future Work . . . . .	118
<b>8</b>	<b>An Efficient Contesting Procedure for AutoML Optimization</b>	<b>121</b>
8.1	Introduction . . . . .	122
8.2	Background . . . . .	123
8.2.1	Contesting procedure for AutoML optimization . . . . .	124
8.2.2	Early-stop strategies . . . . .	125
8.3	Proposed approach . . . . .	126
8.3.1	Algorithm description . . . . .	126
8.3.1.1	Elimination criteria based on the highest performances . . . . .	126
8.3.1.2	Elimination criteria based on a statistical procedure	128
8.3.2	The Splitting approach . . . . .	132

## CONTENTS

---

8.3.3	Fixing the gap between serial and parallel BO . . . . .	134
8.4	Experimental Setup . . . . .	135
8.5	Results and Discussion . . . . .	137
8.5.1	First experiment results . . . . .	139
8.5.2	Second experimental results . . . . .	144
8.6	Application on Surface Defect Classification in Steel Manufacturing	146
8.6.1	Experimental setup . . . . .	147
8.6.2	Experimental results and discussion . . . . .	149
8.7	Conclusions and future work . . . . .	153
<b>9</b>	<b>Conclusions</b>	<b>155</b>
9.1	Summary . . . . .	155
9.2	Future work . . . . .	160
9.2.1	Combination-based sampling . . . . .	161
9.2.2	Contesting procedures . . . . .	161
9.2.3	Benchmarking methods and application domains . . . . .	162
<b>A</b>	<b>Appendix</b>	<b>165</b>
A.1	Additional information for the first experiment . . . . .	165
A.1.1	Parameter setting . . . . .	165
A.2	Imbalance datasets . . . . .	165
A.3	Additional information for the second experiment . . . . .	170
A.3.1	Datasets used in the second experiment . . . . .	170
A.3.2	Search space . . . . .	170
	<b>Bibliography</b>	<b>179</b>
	<b>Index</b>	<b>199</b>
	<b>English Summary</b>	<b>203</b>
	<b>Nederlandse Samenvatting</b>	<b>207</b>
	<b>Acknowledgments</b>	<b>211</b>
	<b>About the Author</b>	<b>213</b>