Exploring Hyperparameter Usage and Tuning in Machine Learning Research

Sebastian Simon, Nikolay Kolyada, Christopher Akiki, Martin Potthast, Benno Stein, Norbert Siegmund

UNIVERSITÄT LEIPZIG
Bauhaus-Universität Weimar
UNIVERSITÄT LEIPZIG
ScaDS.AI DRESDEN LEIPZIG
UNIVERSITÄT LEIPZIG
ScaDS.AI DRESDEN LEIPZIG
Bauhaus-Universität Weimar
UNIVERSITÄT LEIPZIG
ScaDS.AI DRESDEN LEIPZIG

CAIN, May 15th-16th, 2023
Online

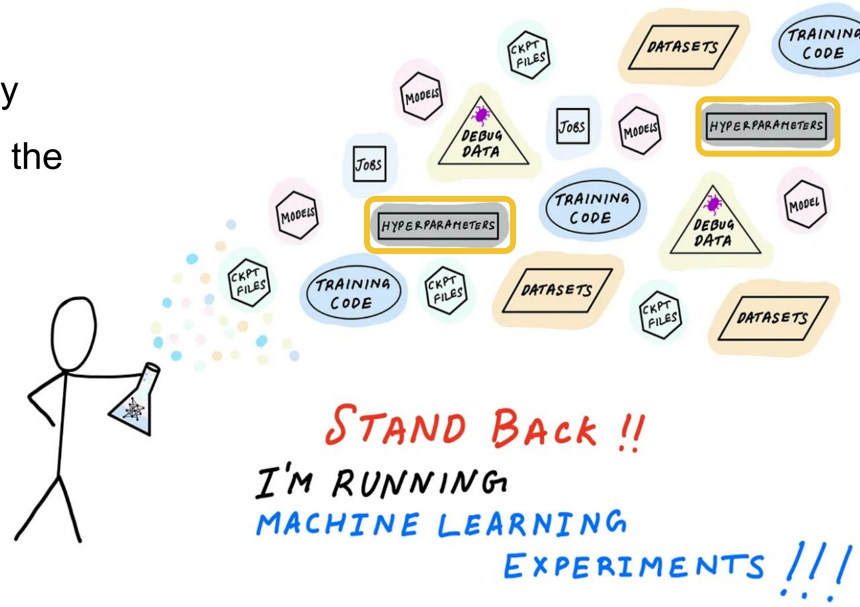
Success Story of Machine Learning

[2]

[1]

[3]

- Highly experiment-driven development
- Goal: obtain ML model with a desired quality
- Hyperparameter Tuning significantly affects the quality



[4]

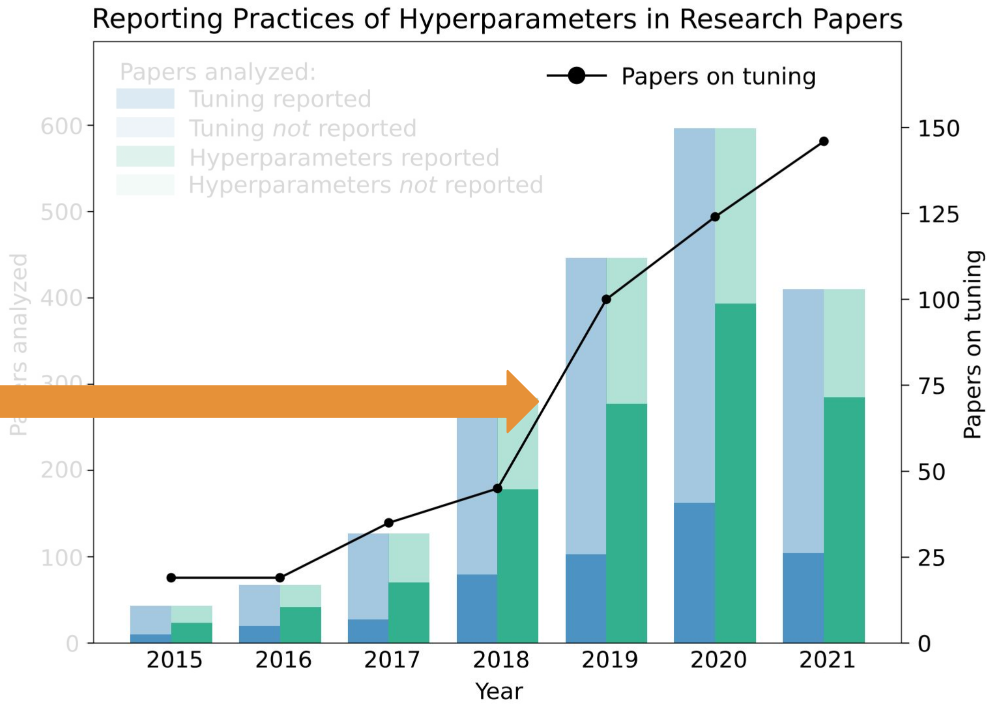
State of Hyperparameter Tuning Research

Papers about tuning:

Source: DBLP
Time: 2015-2021
Keywords: hyperparameter importance, hyperparameter tuning, and hyperparameter optimization

Observation:
7-fold increase in number of papers about a hyperparameter approach

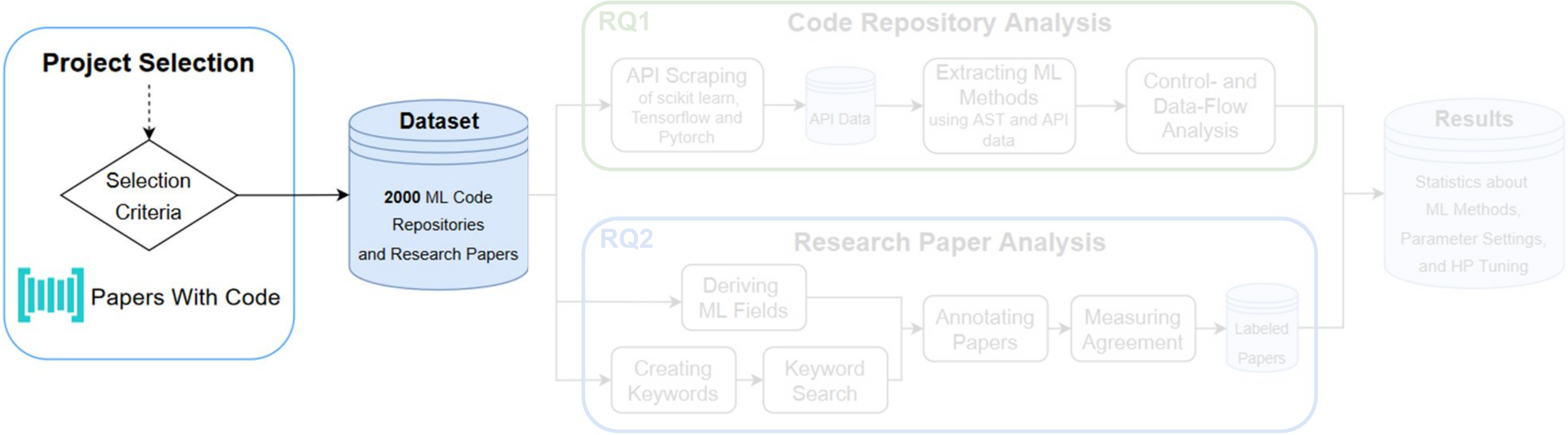
Not a single paper about whether and how hyperparameter are used and tuned at all



Research Methodology

RQ1 Which, how, and to what extent are ML methods configured w.r.t. their hyperparameter settings?

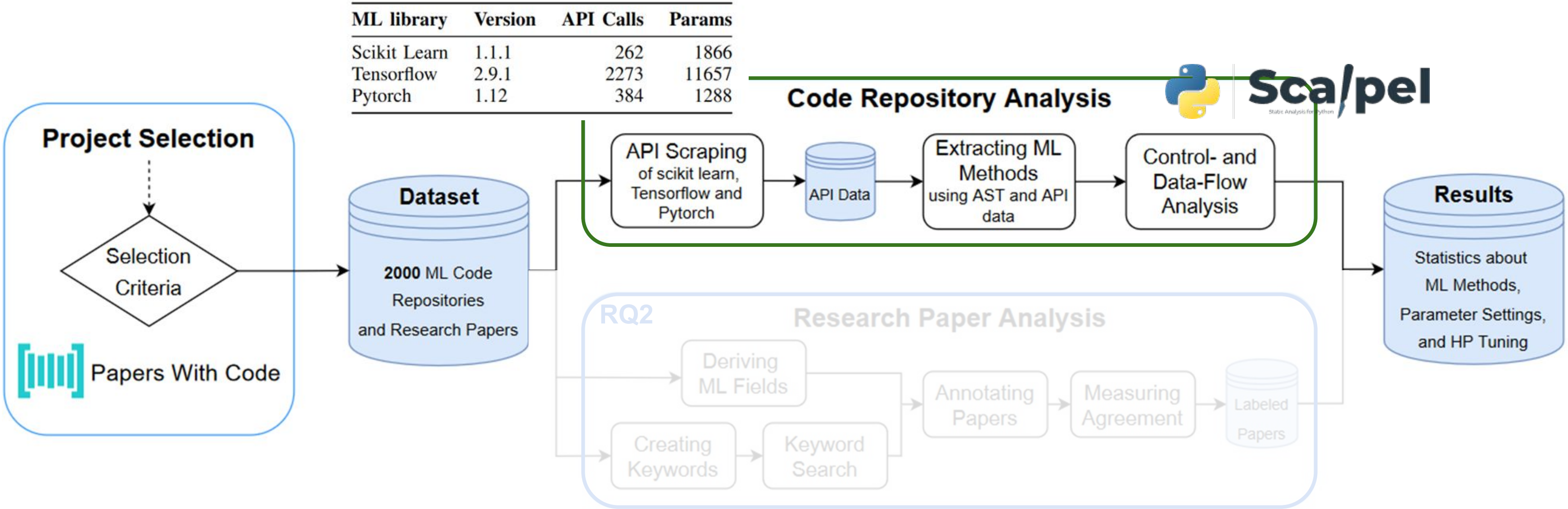
RQ2 How are hyperparameter configurations reported in the accompanied paper?


Project Selection
Selection Criteria
Papers With Code
Dataset
2000 ML Code Repositories and Research Papers
RQ1
Code Repository Analysis
API Scraping of scikit learn, Tensorflow and Pytorch
API Data
Extracting ML Methods using AST and API data
Control- and Data-Flow Analysis
RQ2
Research Paper Analysis
Deriving ML Fields
Creating Keywords
Keyword Search
Annotating Papers
Measuring Agreement
Labeled Papers
Results
Statistics about ML Methods, Parameter Settings, and HP Tuning

Research Methodology

RQ1 Which, how, and to what extent are ML methods configured w.r.t. their hyperparameter settings?

RQ2 How are hyperparameter configurations reported in the accompanied paper?



ML library	Version	API Calls	Params
Scikit Learn	1.1.1	262	1866
Tensorflow	2.9.1	2273	11657
Pytorch	1.12	384	1288

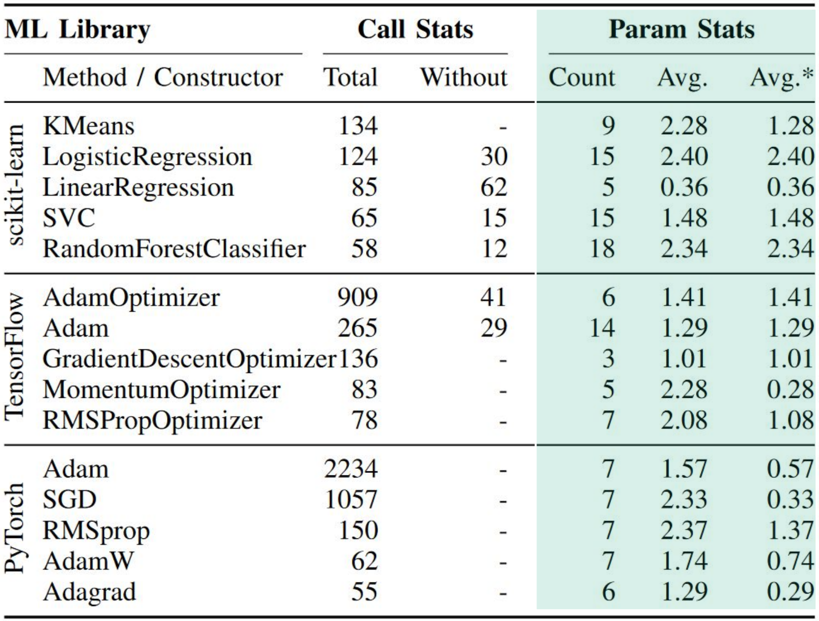
Project Selection
Selection Criteria
Papers With Code
Dataset
2000 ML Code Repositories and Research Papers
Code Repository Analysis
Scalpel
API Scraping of scikit learn, Tensorflow and Pytorch
API Data
Extracting ML Methods using AST and API data
Control- and Data-Flow Analysis
Results
Statistics about ML Methods, Parameter Settings, and HP Tuning
RQ2
Research Paper Analysis
Deriving ML Fields
Creating Keywords
Keyword Search
Annotating Papers
Measuring Agreement
Labeled Papers

Results RQ1: Configuration of ML Methods

How often are hyperparameters actually configured in the analyzed libraries?

Observation:

Only a few hyperparameter of ML methods are set, while the majority remain untouched. Consequently, most hyperparameters retain their default values.



ML Library		Call Stats		Param Stats		
	Method / Constructor	Total	Without	Count	Avg.	Avg.*
scikit-learn	KMeans	134	-	9	2.28	1.28
	LogisticRegression	124	30	15	2.40	2.40
	LinearRegression	85	62	5	0.36	0.36
	SVC	65	15	15	1.48	1.48
	RandomForestClassifier	58	12	18	2.34	2.34
TensorFlow	AdamOptimizer	909	41	6	1.41	1.41
	Adam	265	29	14	1.29	1.29
	GradientDescentOptimizer	136	-	3	1.01	1.01
	MomentumOptimizer	83	-	5	2.28	0.28
	RMSPropOptimizer	78	-	7	2.08	1.08
PyTorch	Adam	2234	-	7	1.57	0.57
	SGD	1057	-	7	2.33	0.33
	RMSprop	150	-	7	2.37	1.37
	AdamW	62	-	7	1.74	0.74
	Adagrad	55	-	6	1.29	0.29

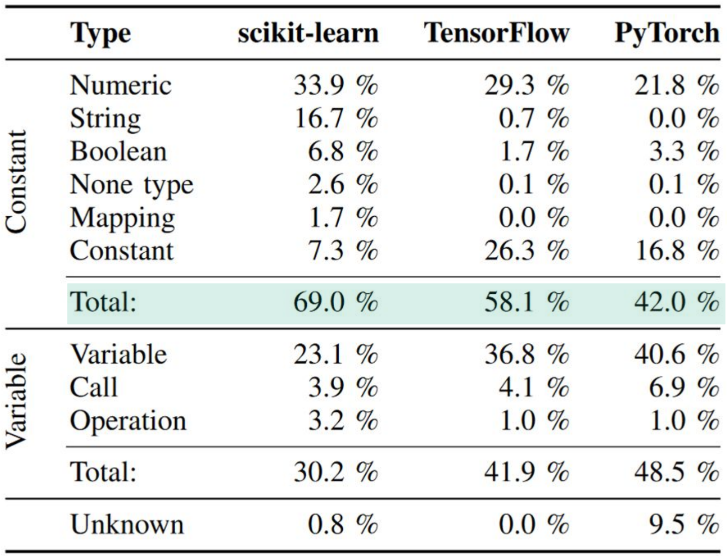
Table: Top 5 most used ML methods per Library with their call and parameter statistics. (* without mandatory parameters)

Results RQ1: Configuration of ML Methods

Are hyperparameters configured dynamically or set with a constant value?

Observation:

Hyperparameters are set by a large fraction with a constant value, ranging from 42 % up to 69 % depending on the framework. It is unclear how these values have been obtained.



	Type	scikit-learn	TensorFlow	PyTorch
Constant	Numeric	33.9 %	29.3 %	21.8 %
	String	16.7 %	0.7 %	0.0 %
	Boolean	6.8 %	1.7 %	3.3 %
	None type	2.6 %	0.1 %	0.1 %
	Mapping	1.7 %	0.0 %	0.0 %
	Constant	7.3 %	26.3 %	16.8 %
	Total:	69.0 %	58.1 %	42.0 %
Variable	Variable	23.1 %	36.8 %	40.6 %
	Call	3.9 %	4.1 %	6.9 %
	Operation	3.2 %	1.0 %	1.0 %
	Total:	30.2 %	41.9 %	48.5 %
	Unknown	0.8 %	0.0 %	9.5 %

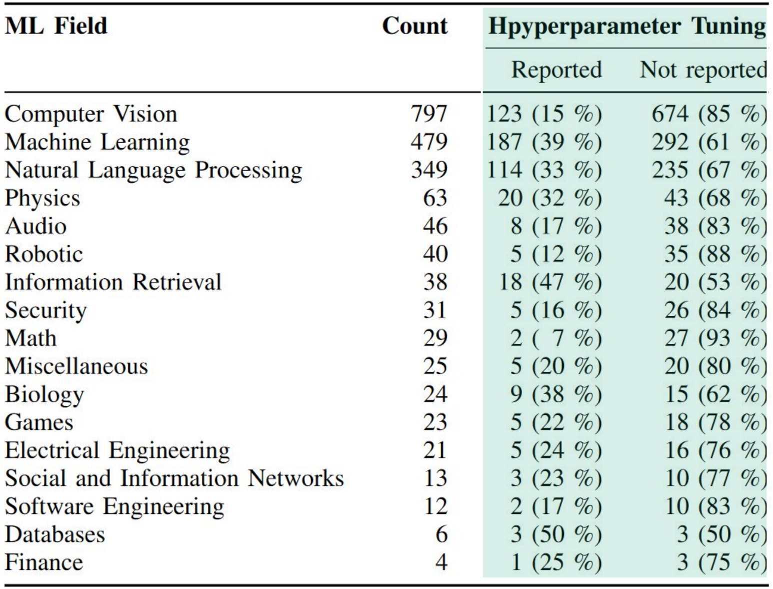
Table: Distribution of Python AST-Types passed as hyperparameters to ML methods.

Results RQ2: Reporting of Hyperparameter Configurations

How many papers report hyperparameter tuning per research field?

Observation:

Regardless the research field, most papers do not explicitly report hyperparameter tuning.



ML Field	Count	Hpyperparameter Tuning	
		Reported	Not reported
Computer Vision	797	123 (15 %)	674 (85 %)
Machine Learning	479	187 (39 %)	292 (61 %)
Natural Language Processing	349	114 (33 %)	235 (67 %)
Physics	63	20 (32 %)	43 (68 %)
Audio	46	8 (17 %)	38 (83 %)
Robotic	40	5 (12 %)	35 (88 %)
Information Retrieval	38	18 (47 %)	20 (53 %)
Security	31	5 (16 %)	26 (84 %)
Math	29	2 (7 %)	27 (93 %)
Miscellaneous	25	5 (20 %)	20 (80 %)
Biology	24	9 (38 %)	15 (62 %)
Games	23	5 (22 %)	18 (78 %)
Electrical Engineering	21	5 (24 %)	16 (76 %)
Social and Information Networks	13	3 (23 %)	10 (77 %)
Software Engineering	12	2 (17 %)	10 (83 %)
Databases	6	3 (50 %)	3 (50 %)
Finance	4	1 (25 %)	3 (75 %)

Table: Number of research papers of ML field that reported and did not reported hyperparameter tuning.

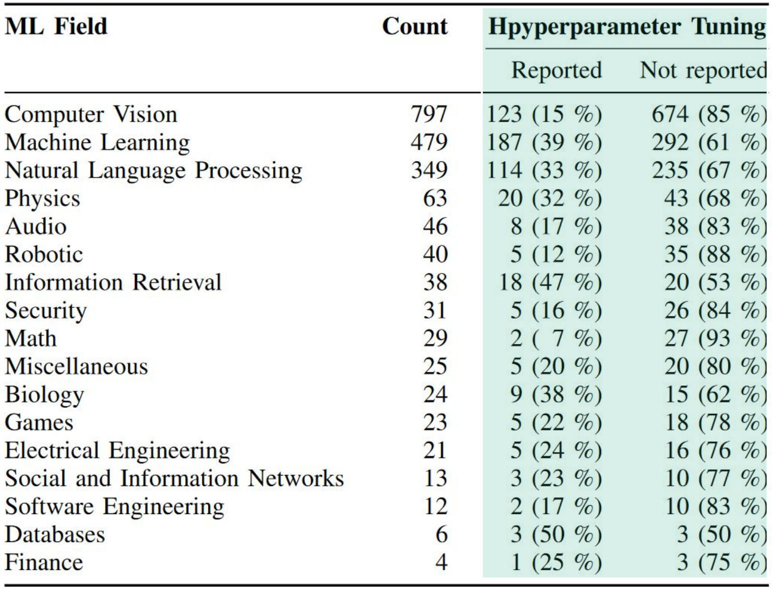
Results RQ2: Reporting of Hyperparameter Configurations

From papers that report tuning, what tuning technique did they use?

Observation:

281 (55 %) papers did not mention a concrete tuning technique. Remaining papers mainly use conservative techniques:

- 133 grid search
- 53 manual tuning
- 20 random search
- 20 Bayesian optimization

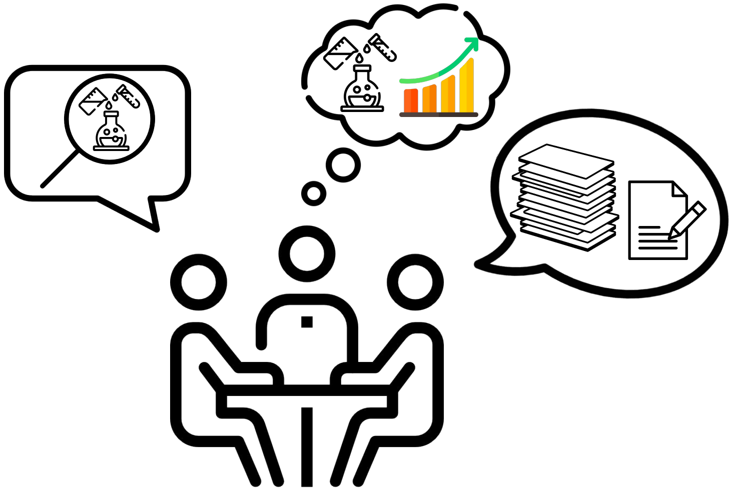


ML Field	Count	Hpyperparameter Tuning	
		Reported	Not reported
Computer Vision	797	123 (15 %)	674 (85 %)
Machine Learning	479	187 (39 %)	292 (61 %)
Natural Language Processing	349	114 (33 %)	235 (67 %)
Physics	63	20 (32 %)	43 (68 %)
Audio	46	8 (17 %)	38 (83 %)
Robotic	40	5 (12 %)	35 (88 %)
Information Retrieval	38	18 (47 %)	20 (53 %)
Security	31	5 (16 %)	26 (84 %)
Math	29	2 (7 %)	27 (93 %)
Miscellaneous	25	5 (20 %)	20 (80 %)
Biology	24	9 (38 %)	15 (62 %)
Games	23	5 (22 %)	18 (78 %)
Electrical Engineering	21	5 (24 %)	16 (76 %)
Social and Information Networks	13	3 (23 %)	10 (77 %)
Software Engineering	12	2 (17 %)	10 (83 %)
Databases	6	3 (50 %)	3 (50 %)
Finance	4	1 (25 %)	3 (75 %)

Table: Number of research papers of ML field that reported and did not reported hyperparameter tuning.

Summary

Striking difference between research on and research with hyperparameter tuning.
Lack of experimentation and reporting practices.

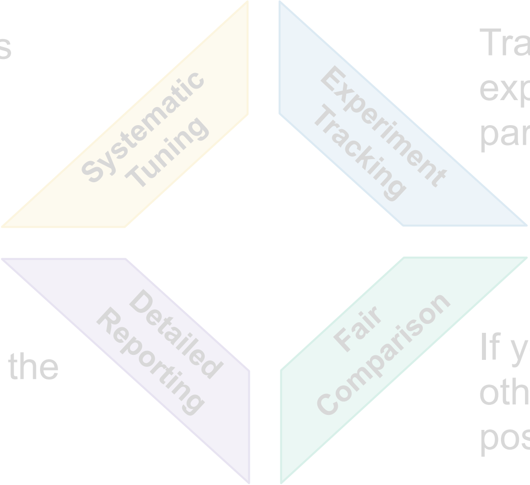


Call to Action

polyaxon W&B TensorBoard comet mlflow

Tune your hyperparameters (with modern techniques)

Track the (meta-) data of your experiments (e.g., metrics, artifacts, parameters)



Report the final values and the tuning procedure

If you compare your approach against others, optimize them as well if possible preprint

https://sws.informatik.uni-leipzig.de/wp-content/uploads/2023/03/CAIN_2023.pdf

Thank you for your attention!

✉ ssimon@informatik.uni-leipzig.de

References

[1] https://www.theatlantic.com/sponsored/microsoft-2016/a-revolution-in-the-automotive-industry/849/
[2] https://elearningindustry.com/why-is-elearning-significant-in-finance-industry

[3] https://www.elastic.co/de/industries/healthcare

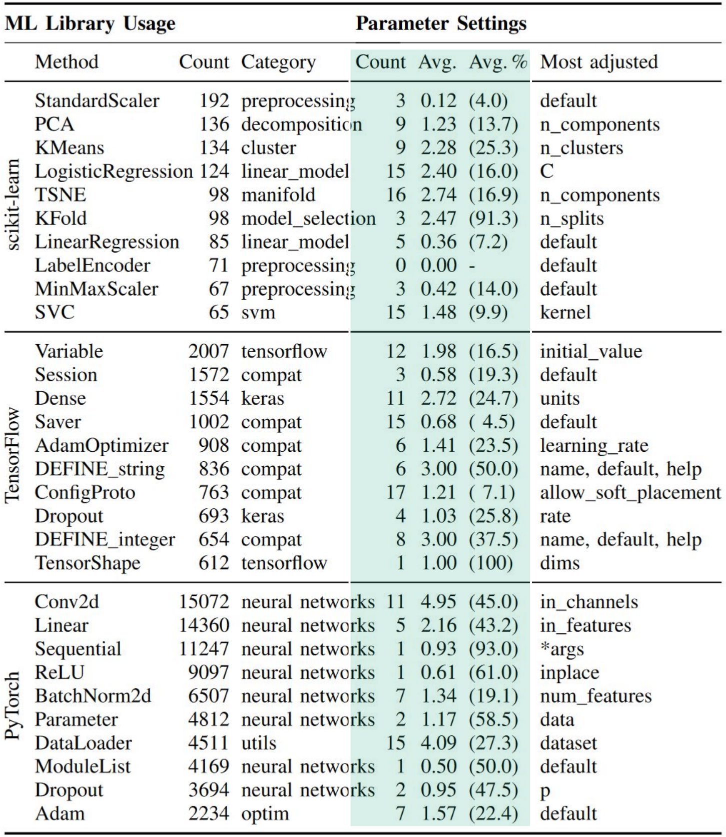
[4] https://medium.com/towards-data-science/a-quick-guide-to-managing-machine-learning-experiments-af84da6b060b

Icons: https://www.flaticon.com/

Results RQ1: Configuration of ML Methods

What are the most commonly used methods of these ML libraries?

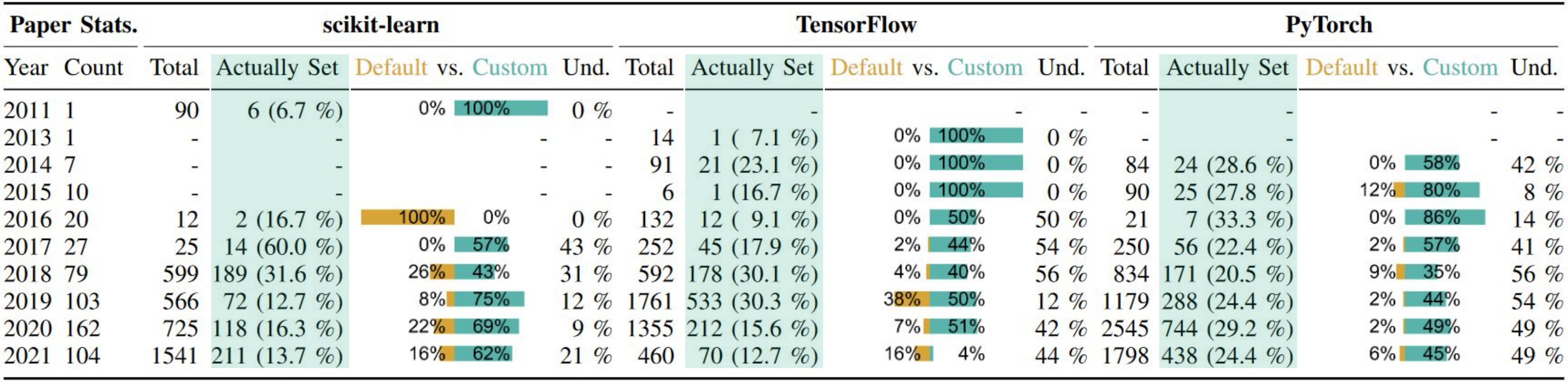
Observation:
Most commonly used methods are neural network building block provided by PyTorch and TensorFlow. Only few methods from scikit-learn are ML and experimental methods.



ML Library Usage				Parameter Settings			
	Method	Count	Category	Count	Avg.	Avg. %	Most adjusted
scikit-learn	StandardScaler	192	preprocessing	3	0.12	(4.0)	default
	PCA	136	decomposition	9	1.23	(13.7)	n_components
	KMeans	134	cluster	9	2.28	(25.3)	n_clusters
	LogisticRegression	124	linear_model	15	2.40	(16.0)	C
	TSNE	98	manifold	16	2.74	(16.9)	n_components
	KFold	98	model_selection	3	2.47	(91.3)	n_splits
	LinearRegression	85	linear_model	5	0.36	(7.2)	default
	LabelEncoder	71	preprocessing	0	0.00	-	default
	MinMaxScaler	67	preprocessing	3	0.42	(14.0)	default
	SVC	65	svm	15	1.48	(9.9)	kernel
TensorFlow	Variable	2007	tensorflow	12	1.98	(16.5)	initial_value
	Session	1572	compat	3	0.58	(19.3)	default
	Dense	1554	keras	11	2.72	(24.7)	units
	Saver	1002	compat	15	0.68	(4.5)	default
	AdamOptimizer	908	compat	6	1.41	(23.5)	learning_rate
	DEFINE_string	836	compat	6	3.00	(50.0)	name, default, help
	ConfigProto	763	compat	17	1.21	(7.1)	allow_soft_placement
	Dropout	693	keras	4	1.03	(25.8)	rate
	DEFINE_integer	654	compat	8	3.00	(37.5)	name, default, help
	TensorShape	612	tensorflow	1	1.00	(100)	dims
PyTorch	Conv2d	15072	neural networks	11	4.95	(45.0)	in_channels
	Linear	14360	neural networks	5	2.16	(43.2)	in_features
	Sequential	11247	neural networks	1	0.93	(93.0)	*args
	ReLU	9097	neural networks	1	0.61	(61.0)	inplace
	BatchNorm2d	6507	neural networks	7	1.34	(19.1)	num_features
	Parameter	4812	neural networks	2	1.17	(58.5)	data
	DataLoader	4511	utils	15	4.09	(27.3)	dataset
	ModuleList	4169	neural networks	1	0.50	(50.0)	default
	Dropout	3694	neural networks	2	0.95	(47.5)	p
	Adam	2234	optim	7	1.57	(22.4)	default

Table: Top 10 most commonly used methods per Library with their call and parameter statistics.

Hyperparameter Usage in Code Repositories



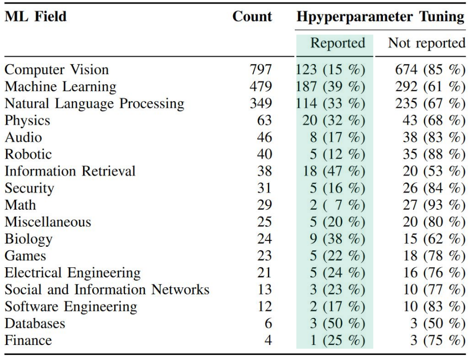
Paper Stats.		scikit-learn				TensorFlow				PyTorch			
Year	Count	Total	Actually Set	Default vs. Custom	Und.	Total	Actually Set	Default vs. Custom	Und.	Total	Actually Set	Default vs. Custom	Und.
2011	1	90	6 (6.7 %)	0% 100%	0 %	-	-	-	-	-	-	-	-
2013	1	-	-	-	-	14	1 (7.1 %)	0% 100%	0 %	-	-	-	-
2014	7	-	-	-	-	91	21 (23.1 %)	0% 100%	0 %	84	24 (28.6 %)	0% 58%	42 %
2015	10	-	-	-	-	6	1 (16.7 %)	0% 100%	0 %	90	25 (27.8 %)	12% 80%	8 %
2016	20	12	2 (16.7 %)	100% 0%	0 %	132	12 (9.1 %)	0% 50%	50 %	21	7 (33.3 %)	0% 86%	14 %
2017	27	25	14 (60.0 %)	0% 57%	43 %	252	45 (17.9 %)	2% 44%	54 %	250	56 (22.4 %)	2% 57%	41 %
2018	79	599	189 (31.6 %)	26% 43%	31 %	592	178 (30.1 %)	4% 40%	56 %	834	171 (20.5 %)	9% 35%	56 %
2019	103	566	72 (12.7 %)	8% 75%	12 %	1761	533 (30.3 %)	38% 50%	12 %	1179	288 (24.4 %)	2% 44%	54 %
2020	162	725	118 (16.3 %)	22% 69%	9 %	1355	212 (15.6 %)	7% 51%	42 %	2545	744 (29.2 %)	2% 49%	49 %
2021	104	1541	211 (13.7 %)	16% 62%	21 %	460	70 (12.7 %)	16% 4%	44 %	1798	438 (24.4 %)	6% 45%	49 %

Table: Statistics on hyperparameter usage in code repositories where the associated research paper reported hyperparameter tuning sorted by year.

Observation:

Configuration settings of ML methods do not receive the attention they actually need. Only a few of the available hyperparameters are set across all libraries, while the majority remain untouched.

Results RQ2: Reporting of Hyperparameter Configurations



ML Field	Count	Hpyperparameter Tuning	
		Reported	Not reported
Computer Vision	797	123 (15 %)	674 (85 %)
Machine Learning	479	187 (39 %)	292 (61 %)
Natural Language Processing	349	114 (33 %)	235 (67 %)
Physics	63	20 (32 %)	43 (68 %)
Audio	46	8 (17 %)	38 (83 %)
Robotic	40	5 (12 %)	35 (88 %)
Information Retrieval	38	18 (47 %)	20 (53 %)
Security	31	5 (16 %)	26 (84 %)
Math	29	2 (7 %)	27 (93 %)
Miscellaneous	25	5 (20 %)	20 (80 %)
Biology	24	9 (38 %)	15 (62 %)
Games	23	5 (22 %)	18 (78 %)
Electrical Engineering	21	5 (24 %)	16 (76 %)
Social and Information Networks	13	3 (23 %)	10 (77 %)
Software Engineering	12	2 (17 %)	10 (83 %)
Databases	6	3 (50 %)	3 (50 %)
Finance	4	1 (25 %)	3 (75 %)

From papers that report tuning, what was their tuning technique?

Observation:

281 (55 %) papers did not mention a concrete tuning technique.
Remaining papers mainly use conservative techniques:

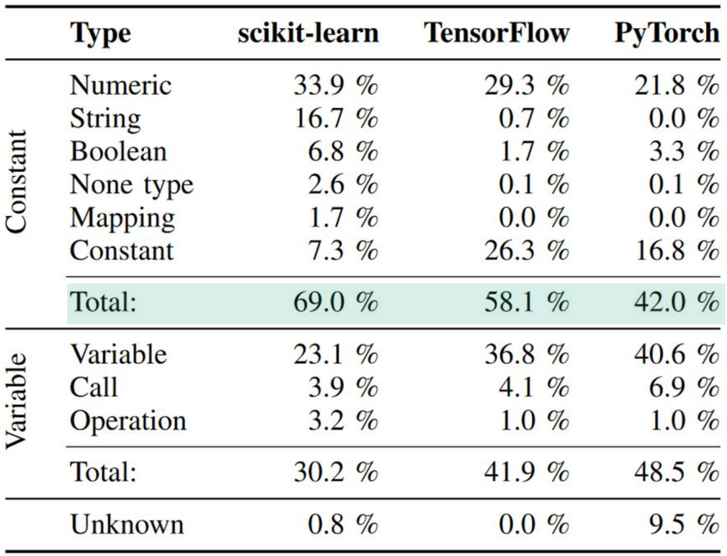
- 133 grid search
- 53 manual tuning
- 20 random search
- 20 Bayesian optimization

Answer RQ2: We found a stark discrepancy between applying hyperparameter tuning and reporting it. Overall, tuning seems to be not a common practice and it often remains unclear how parameter values have been obtained, hampering reproducibility of results.

Results RQ1: Configuration of ML Methods

Are hyperparameters configured dynamically or set with a constant value?

Observation:
Hyperparameters are set by a large fraction with a constant value, ranging from 42 % up to 69 % depending on the framework. It is unclear how these values have been obtained.



	Type	scikit-learn	TensorFlow	PyTorch
Constant	Numeric	33.9 %	29.3 %	21.8 %
	String	16.7 %	0.7 %	0.0 %
	Boolean	6.8 %	1.7 %	3.3 %
	None type	2.6 %	0.1 %	0.1 %
	Mapping	1.7 %	0.0 %	0.0 %
	Constant	7.3 %	26.3 %	16.8 %
	Total:	69.0 %	58.1 %	42.0 %
Variable	Variable	23.1 %	36.8 %	40.6 %
	Call	3.9 %	4.1 %	6.9 %
	Operation	3.2 %	1.0 %	1.0 %
	Total:	30.2 %	41.9 %	48.5 %
	Unknown	0.8 %	0.0 %	9.5 %

Table: Distribution of Python AST-Types passed as hyperparameters to ML methods.

Answer RQ1: Only a fraction of available tuning parameters are actually set. Most retain their default values. If hyperparameters are set, the majority are constant values without the possibility for tracking and automated tuning.

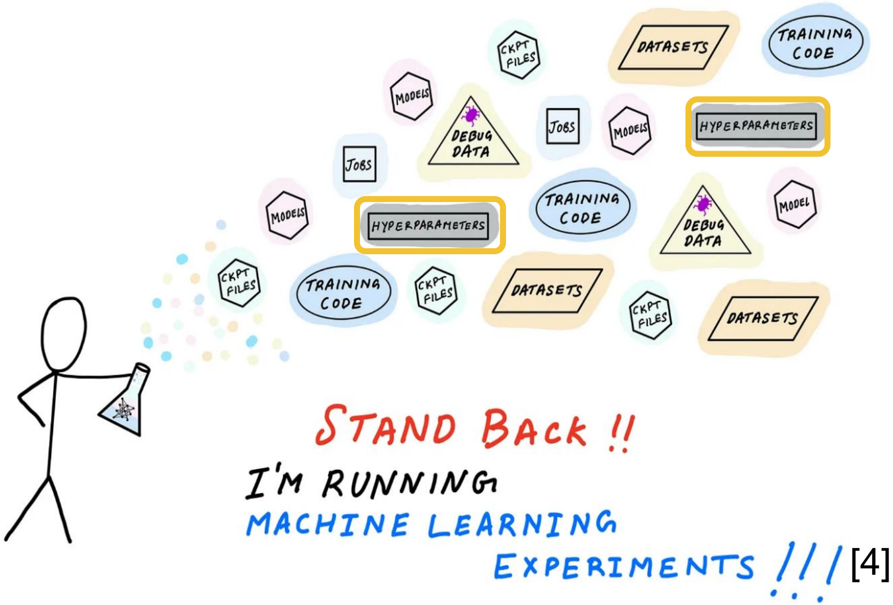
Success Story of Machine Learning

Experiment-driven development enables evaluation of:

- modeling techniques
- ML configuration
- data slices

Hyperparameter Tuning significantly affects:

- accuracy
- robustness
- reliability
- generalizability
- …



Summary: RQ1 and RQ2

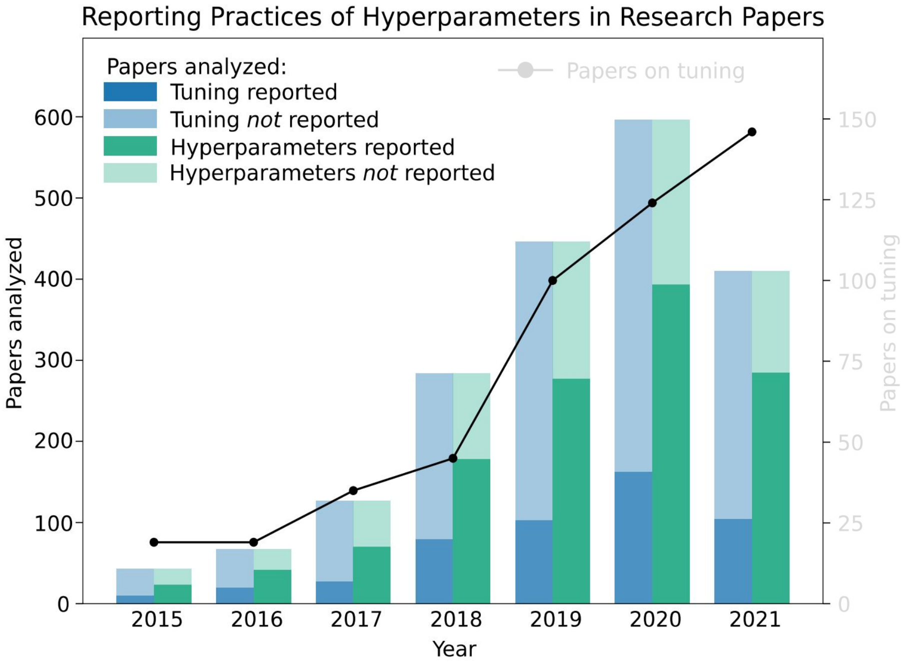
Observations:

Only a few hyperparameters are set, while the majority remains untouched.

If hyperparameters are set, most of them are constant values.

Across all years, about 75% of papers do not report hyperparameter tuning, only about 50% of papers state chosen values.

Hyperparameter tunings seems to be not a common practice.



Striking difference between research on and research with hyperparameter tuning

Summary

Striking difference between research on and research with hyperparameter tuning