Supplementary Material for Efficient AutoML via Combinational Sampling

Duc Anh Nguyen*, Anna V. Kononova*, Stefan Menzel§, Bernhard Sendhoff§, and Thomas Bäck*
*Leiden Institute of Advanced Computer Science (LIACS), Leiden University, The Netherlands
Email: {d.a.nguyen,a.kononova,t.h.w.baeck}@liacs.leidenuniv.nl
§Honda Research Institute Europe GmbH (HRI-EU), Offenbach/Main, Germany
Email: {stefan.menzel,bernhard.sendhoff}@honda-ri.de

I. Our AutoML

The overall structure of our AutoML framework is summarized in Fig. S- 1. The process begins with metadata extraction on the input dataset, then the metadata uses to generate a suitable search space χ by the Auto-sklearn search space generator. Next the search space χ is converted to our search space $\mathcal{M}$. Note that, in addition to hyperparameter classes used in Auto-sklearn, our search space includes the new hyperparameter class for modelling the choice of algorithms and an attribute for grouping options. In the meantime, the input dataset splited into two independent sets $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$. Our framework taking dataset $\mathcal{D}_{train}$ and search space $\mathcal{M}$. Then, we optimizes the search space $\mathcal{M}$ until the wall-time reaches 1 hour and returns the best found pipeline setting λ^*, consisting of a sequence of operators and their optimized hyperparameter settings. Once the optimization process is done, a machine learning pipeline is initialized using values of the best found pipeline setting λ^*. The best found pipeline learn $\mathcal{D}_{train}$ and predict $\mathcal{D}_{test}$. Then the test accuracy is calculated.

In summary, the main parts of our framework is highlighted in the blue rounded rectangle, consists of the following three main parts:

- **Search space**: we re-use the search space of Auto-sklearn since it supports both classification and regression problems. For a given dataset input, Auto-sklearn dynamically generates a suitable search space based on the dataset input itself; thus, the search spaces for machine learning problems are different in terms of the selected algorithms and hyperparameters. In practice, this search space generator is based on two aspects: the machine learning problem itself, i.e., binary classification, multi-class classification, multi-label classification, regression, multi-output regression, and the dataset representation, i.e.,either dense or sparse dataset. The resulting search space for a single problem is large and commonly having up to 153 hyperparameters and 6 operators, i.e., categorical encoder, numerical transformer, imputation transformer, rescaling, feature preprocessor, and learning operator.
- **Optimizer**: our framework use TPE as the underlying optimizer. Both the Hyperopt sampling approach and our sampling approach are implemented, controlling by a parameter, called Type, "Hyperopt" for the hyperopt sampling approach and "full" for our approach.

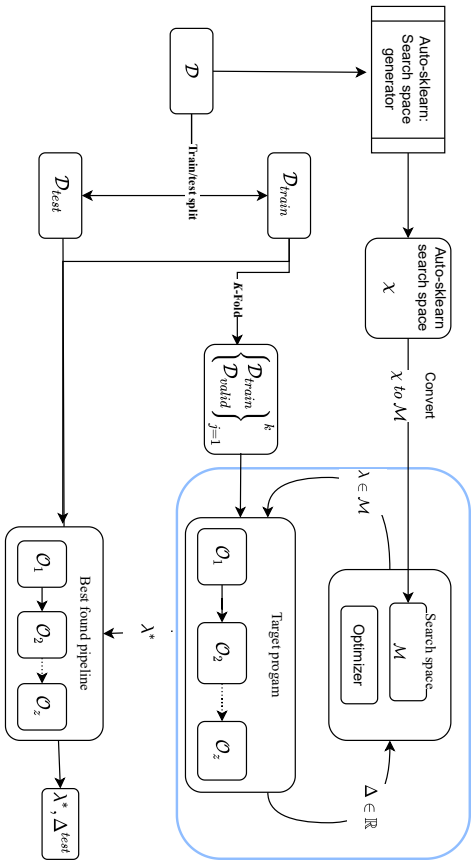

Fig. S-1. Flowchart of the experimental setup.

- **Target program**: initialize a machine learning pipeline based on the sequence of operators and hyperparameter settings, which are generated by the optimizer and performs the k-folds cross validation, resulting in a real-valued.

II. Experimental Setup: parameter setting

A. Datasets used in the first experiment

In this section, we present 44 examined datasets taken from the KEEL repository [1] in Table S- 1. For each dataset, we include the *Imbalance Ratio* (IR), which is the ratio of the number of majority class instances to that of minority class instances.

TABLE S-1
THE NUMBER OF POSITIVE, NEGATIVE CLASSES, ATTRIBUTES (#ATT) AND THE IMBALANCE RATIO (IR) OF THE KEEL DATASETS, ORDERED BY INCREASING IR VALUE.

Data Sets	# Negative	# Positive	#Att	**IR**
glass1	138	76	9	1.82
ecoli-0_vs_1	77	143	7	1.86
wisconsin	444	239	9	1.86
pima	500	268	8	1.87
iris0	100	50	4	2
glass0	144	70	9	2.06
yeast1	1055	429	8	2.46
haberman	225	81	3	2.78
vehicle2	628	218	18	2.88
vehicle1	629	217	18	2.9
vehicle3	634	212	18	2.99
glass-0-1-2-3_vs_4-5-6	163	51	9	3.2
vehicle0	647	199	18	3.25
ecoli1	259	77	7	3.36
new-thyroid1	180	35	5	5.14
new-thyroid2	180	35	5	5.14
ecoli2	284	52	7	5.46
segment0	1979	329	19	6.02
glass6	185	29	9	6.38
yeast3	1321	163	8	8.1
ecoli3	301	35	7	8.6
page-blocks0	4913	559	10	8.79
yeast-2_vs_4	463	51	8	9.08
yeast-0-5-6-7-9_vs_4	477	51	8	9.35
vowel0	898	90	13	9.98
glass-0-1-6_vs_2	175	17	9	10.29
glass2	197	17	9	11.59
shuttle-c0-vs-c4	1706	123	9	13.87
yeast-1_vs_7	429	30	7	14.3
glass4	201	13	9	15.46
ecoli4	316	20	7	15.8
page-blocks-1-3_vs_4	444	28	10	15.86
abalone9-18	689	42	8	16.4
glass-0-1-6_vs_5	175	9	9	19.44
shuttle-c2-vs-c4	123	6	9	20.5
yeast-1-4-5-8_vs_7	663	30	8	22.1
glass5	205	9	9	22.78
yeast-2_vs_8	462	20	8	23.1
yeast4	1433	51	8	28.1
yeast-1-2-8-9_vs_7	917	30	8	30.57
yeast5	1440	44	8	32.73
ecoli-0-1-3-7_vs_2-6	274	7	7	39.14
yeast6	1449	35	8	41.4
abalone19	4142	32	8	129.44

B. Datasets used in the second experiment

In this section, we present 73 examined datasets taken from OpenML repository [2] in Table S- 2. For each task, we include the *OpenML ID* (#task id) and the corresponding dataset (#ID, Name) together with the number of classes (#Classes), the number of instances(#Instances), the number of features for one instance (Total features, number of numeric and categorical features), the number of missing values (#Missing values), the number of instances with missing value (#Incomplete instances).

III. ADDITIONAL PLOTS ON INITIAL SAMPLING APPROACHES

This section provides 2 figures on the number of samples allocated to different combination of methods in random sampling implemented in Hyperopt vs. our proposed approach. Fig.S- 2 shows those over three independent runs. Fig.S- 3 presents those of the first run of Fig.S- 2 in the level of individual methods.

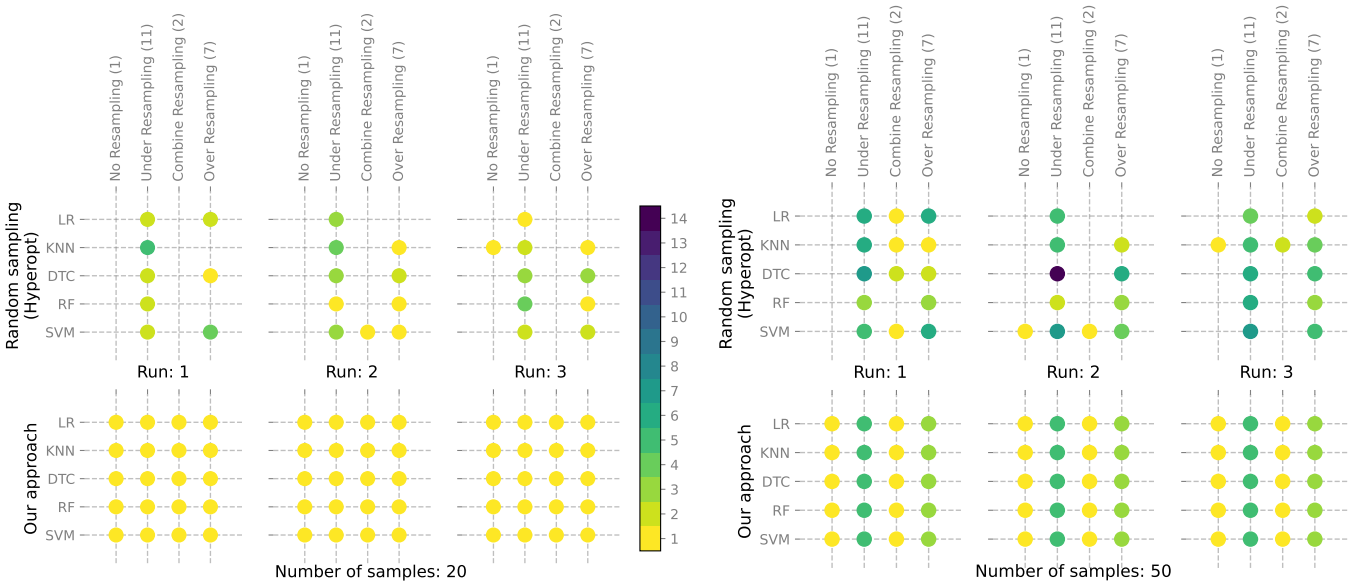

Fig. S-2. Illustration on the number of samples allocated to different combination of methods in random sampling implemented in Hyperopt (top) vs. our proposed approach (bottom) over three independent runs. Cases with 20 (left) and 50 (right) samples are shown here. Figure best viewed in color.

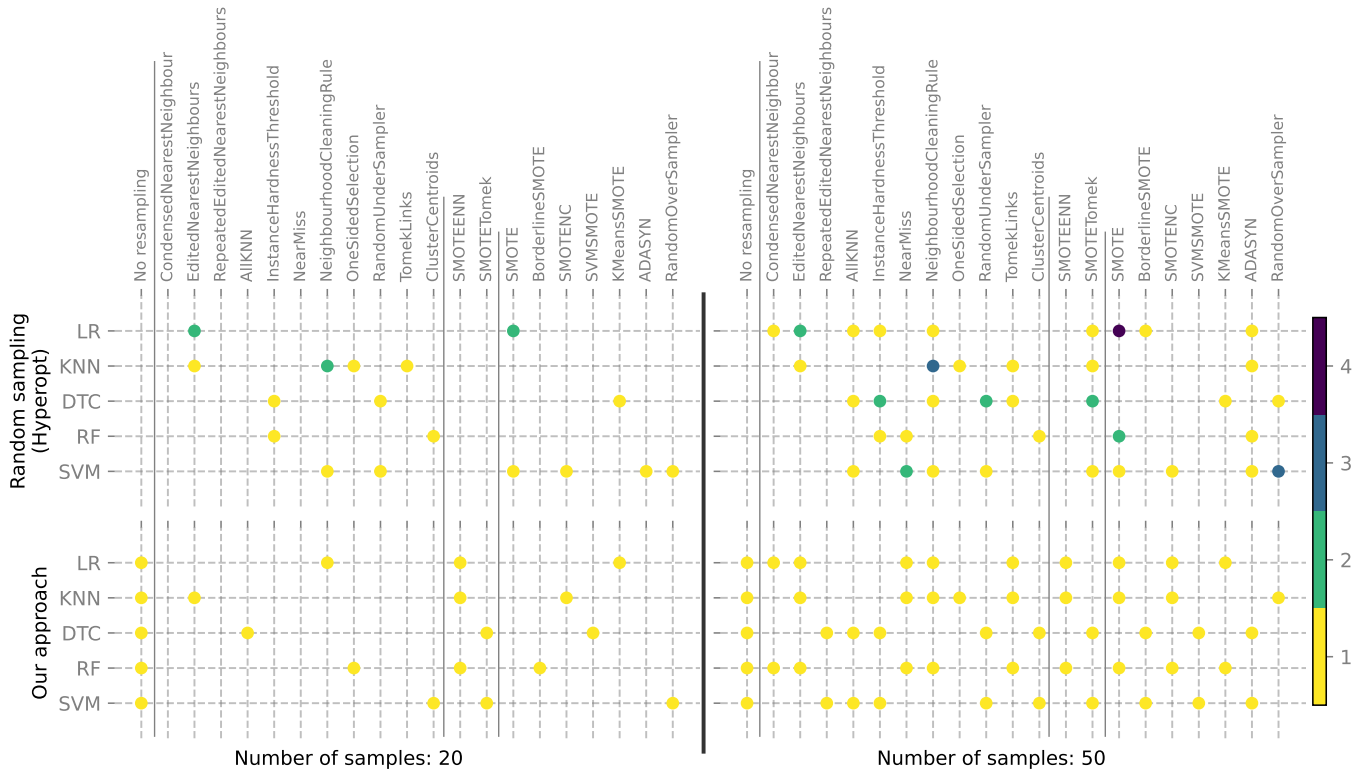

Fig. S-3. Illustration on the distribution of samples obtained via initial sampling methods on the level of individual methods. The left part shows the case with 20 samples, while the case with 50 samples is shown on the right. The methods that belong to different groups are separated by a grey line.

TABLE S-2
LIST OF 73 DATASETS USED IN OUR SECOND EXPERIMENT, ORDERED BY INCREASING OPENML ID.

Task ID	Data set		#Classes	#Instances	Features			#Missing	#Incomplete
	#ID	Name			# Total	#Numeric	#Categorical	Values	instances
3	3	kr-vs-kp	2	37	3196	0	37	0	0
12	12	mfeat-factors	10	217	2000	216	1	0	0
15	15	breast-w	2	10	699	9	1	16	16
23	23	cmc	3	10	1473	2	8	0	0
24	24	mushroom	2	23	8124	0	23	2480	2480
29	29	credit-approval	2	16	690	6	10	67	37
31	31	credit-g	2	21	1000	7	14	0	0
3021	38	sick	2	30	3772	7	23	6064	3772
41	42	soybean	19	36	683	0	36	2337	121
53	54	vehicle	4	19	846	18	1	0	0
2079	188	eucalyptus	5	20	736	14	6	448	95
3543	451	irish	2	6	500	2	4	32	32
3560	469	analcatdata_dmf	6	5	797	0	5	0	0
3561	470	profb	2	10	672	5	5	1200	666
3904	1053	jm1	2	22	10885	21	1	25	5
3917	1067	kc1	2	22	2109	21	1	0	0
3945	1111	KDDCup09_appete	2	231	50000	192	39	8024152	50000
3946	1112	KDDCup09_churn	2	231	50000	192	39	8024152	50000
3948	1114	KDDCup09_upsell	2	231	50000	192	39	8024152	50000
189354	1169	airlines	2	8	539383	3	5	0	0
14965	1461	bank-marketing	2	17	45211	7	10	0	0
10101	1464	blood-transfusi	2	5	748	4	1	0	0
9981	1468	cnae-9	9	857	1080	856	1	0	0
9985	1475	first-order-the	6	52	6118	51	1	0	0
9977	1486	nomao	2	119	34465	89	30	0	0
9952	1489	phoneme	2	6	5404	5	1	0	0
9955	1492	one-hundred-pla	100	65	1600	64	1	0	0
7592	1590	adult	2	15	48842	6	9	6465	3620
7593	1596	covertype	7	55	581012	10	45	0	0
9910	4134	Bioresponse	2	1777	3751	1776	1	0	0
34539	4135	Amazon_employee	2	10	32769	0	10	0	0
14952	4534	PhishingWebsite	2	31	11055	0	31	0	0
14969	4538	GesturePhaseSeg	5	33	9873	32	1	0	0
34538	4550	MiceProtein	8	82	1080	77	5	1396	528
14954	6332	cylinder-bands	2	40	540	18	22	999	263
14968	6332	cylinder-bands	2	40	540	18	22	999	263
14967	23380	cjs	6	35	2796	32	3	68100	2795
125920	23381	dresses-sales	2	13	500	1	12	835	401
146606	23512	higgs	2	29	98050	28	1	9	1
167120	23517	numerai28.6	2	22	96320	21	1	0	0
146607	40536	SpeedDating	2	123	8378	59	64	18372	7330
146195	40668	connect-4	3	43	67557	0	43	0	0
167140	40670	dna	3	181	3186	0	181	0	0
146212	40685	shuttle	7	10	58000	9	1	0	0
167141	40701	churn	2	21	5000	16	5	0	0
167121	40923	Devnagari-Scrip	46	1025	92000	1024	1	0	0
167124	40927	CIFAR_10	10	3073	60000	3072	1	0	0
146800	40966	MiceProtein	8	82	1080	77	5	1396	528
146821	40975	car	4	7	1728	0	7	0	0
167125	40978	Internet-Advert	2	1559	3279	3	1556	0	0
146824	40979	mfeat-pixel	10	241	2000	240	1	0	0
146818	40981	Australian	2	15	690	6	9	0	0
146817	40982	steel-plates-fa	7	28	1941	27	1	0	0
146820	40983	wilt	2	6	4839	5	1	0	0
146822	40984	segment	7	20	2310	19	1	0	0
146819	40994	climate-model-s	2	21	540	20	1	0	0
146825	40996	Fashion-MNIST	10	785	70000	784	1	0	0
167119	41027	jungle_chess_2p	3	7	44819	6	1	0	0
168868	41138	APSFailure	2	171	76000	170	1	1078695	75244
168908	41142	christine	2	1637	5418	1599	38	0	0
168911	41143	jasmine	2	145	2984	8	137	0	0
168912	41146	sylvine	2	21	5124	20	1	0	0
189356	41147	albert	2	79	425240	26	53	2734000	425159
168335	41150	MiniBooNE	2	51	130064	50	1	0	0
168337	41159	guillermo	2	4297	20000	4296	1	0	0
168338	41161	riccardo	2	4297	20000	4296	1	0	0
168909	41163	dilbert	5	2001	10000	2000	1	0	0
168910	41164	fabert	7	801	8237	800	1	0	0
168332	41165	robert	10	7201	10000	7200	1	0	0
168331	41166	volkert	10	181	58310	180	1	0	0
189355	41167	dionis	355	61	416188	60	1	0	0
168330	41168	jannis	4	55	83733	54	1	0	0
168329	41169	helena	100	28	65196	27	1	0	0

ACKNOWLEDGMENT

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

REFERENCES

[1] J. Alcalá-Fdez, L. Sánchez, S. Garcia, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, *et al.*, "Keel: a software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2009.

[2] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "Openml: Networked science in machine learning," *SIGKDD Explorations*, vol. 15, no. 2, pp. 49–60, 2013.