[Supplementary Document] AutoWeka4MCPS-AVATAR: Accelerating Automated Machine Learning Pipeline Composition and Optimisation

Tien Dung Nguyen, Katarzyna Musial, Bogdan Gabrys

Advanced Analytics Institute, University of Technology Sydney

15 Broadway, Sydney, Australia

Abstract

This supplementary document provides the experiment results that are not available in the paper.

Keywords: automated machine learning, pipeline composition and optimisation, machine learning pipeline evaluation

1. Experiments and Discussion

1.1. Experiments to investigate the time required to evaluate invalid pipelines

Table 1 and 2 present the number of invalid pipelines and the wasted time used to evaluate these invalid pipelines in ML pipeline composition and optimisation of AutoWeka4MCPS and Auto-sklearn. Firstly, these tables show that not all of the constructed pipelines are valid. We can see the presence of invalid pipelines for both AutoML tools across different datasets and iterations. For example, there are 9 invalid and 54 valid pipelines in the case of using AutoWeka4MCPS for the dataset *abalone* in *Iter 1*. Secondly, the evaluation time of these invalid pipelines may be significant in several cases. For example, the wasted evaluation time is 75.48% in the case of using the dataset *car* and *Iter 5*. We can see that changing seed numbers has a strong impact on the wasted evaluation time in the case of AutoWeka4MCPS. For example, the experiments with the dataset *abalone* show that the wasted evaluation time is in the range between 0.21% and 99.10%. The reason is that Weka libraries themselves can evaluate the compatibility of a single component pipeline without execution. If the initialisation of the pipeline composition and optimisation with a specific seed number results in pipelines consisting of only one predictor, and these pipelines are well-performing, it tends to exploit similar ML pipelines. As a result, the wasted evaluation time is low. However, if the initialisation results in a complex pipeline which is invalid, there is no guidance for the next promising pipelines. Therefore, the next promising pipelines are randomly selected. As a result, the upcoming pipelines that will be evaluated may also be invalid. This impact is negligible in the case of Auto-sklearn. In other words, the impact of changing seed numbers on the variance of the wasted evaluation in the case of Auto-sklearn is less than in the case of AutoWeka4MCPS. For example, standard deviation of the number of invalid pipelines in the case of

Table 1: Negative impacts of invalid pipelines using AutoWeka4MCPS. (1): the number of invalid/ valid pipelines, (2): the total evaluation time of invalid/ valid pipelines (s), (3): the wasted evaluation time (%).

Dataset	**Criteria**	**Iter 1**	**Iter 2**	**Iter 3**	**Iter 4**	**Iter 5**
abalone	**(1)**	9/54	25/66	16/53	29/97	3/13
	(2)	16/7,284	8,639/322	7,571/286	3,361/2,351	8,070/74
	(3)	0.21	96.40	96.36	58.84	99.10
adult	**(1)**	5/25	4/23	22/23	8/25	4/19
	(2)	3,982/4,308	12/8,188	5,493/805	7,321/356	3,656/6,229
	(3)	48.04	0.14	87.22	95.36	36.98
amazon	**(1)**	12/23	6/4	26/0	17/19	1/0
	(2)	5,413/3,056	7,662/66	4,928/0	7,701/1,689	3,603/0
	(3)	63.92	99.67	100.00	82.01	100.00
car	**(1)**	22/94	26/123	31/133	71/263	32/128
	(2)	4,353/1,416	8,239/224	4,206/5,039	2,745/3,614	13,092/1,443
	(3)	75.48	97.35	45.49	43.17	90.07
cifar10small	**(1)**	2/6	11/30	16/0	1/6	10/17
	(2)	56/9,700	6,376/854	6,590/0	1,474/4,563	2,006/7,602
	(3)	0.57	88.19	100.00	24.41	20.88
convex	**(1)**	5/24	4/40	5/2	2/15	10/15
	(2)	3,625/4,309	5,640/3,787	4,124/2,129	2,229/4,968	6,622/3,364
	(3)	45.69	59.83	65.95	30.97	66.31
dexter	**(1)**	14/55	3/18	4/0	0/4	6/13
	(2)	1,827/4,246	3,604/3,852	7,205/0	0/8,796	7,210/89
	(3)	30.08	48.34	100.00	0.00	98.78
dorothea	**(1)**	5/17	9/3	4/0	1/0	5/16
	(2)	3,627/1,477	231/7,261	7,208/0	3,602/0	3,639/3,511
	(3)	71.05	3.09	100.00	100.0	50.89
gcredit	**(1)**	75/314	52/209	86/378	24/165	139/706
	(2)	479/5,513	5,233/941	1,387/5,908	3,363/1,975	1,827/2,664
	(3)	8.00	84.77	19.01	63.00	40.68
gisette	**(1)**	5/22	3/16	5/6	2/5	11/28
	(2)	930/7,853	638/6,218	5,239/3,522	4,119/6,116	4,533/3,185
	(3)	10.59	9.31	59.80	40.24	58.74
kddcup	**(1)**	8/18	46/22	11/32	4/0	5/2
	(2)	4,294/2,870	3,778/3,739	3,927/6,250	7,309/0	71/7,219
	(3)	59.94	50.26	38.58	100.00	0.97
krvskp	**(1)**	10/46	31/142	22/117	20/74	28/99
	(2)	3,614/1,223	4,292/1,506	8,788/766	5,609/224	3,792/1,243
	(3)	74.73	74.03	91.99	96.16	75.32
madelon	**(1)**	11/47	12/38	6/4	11/33	20/65
	(2)	6,474/3,034	7,170/3,257	5,108/514	3,705/2,469	6,022/3,151
	(3)	68.09	68.77	90.86	60.01	65.65
mnist	**(1)**	2/11	5/5	11/0	0/13	0/2
	(2)	29/7,171	10,641/3,660	5,780/0	0/7,228	0/7,214
	(3)	0.40	74.41	100.00	0.00	0.00
secom	**(1)**	19/89	2/14	17/97	17/118	3/31
	(2)	3,974/4,236	2,534/5,676	6,382/2,404	4,405/5,523	4/9,414
	(3)	48.40	30.86	72.64	44.37	0.04
semeion	**(1)**	27/84	5/0	22/23	20/51	14/23
	(2)	1,935/6,224	9,103/0	2,759/3,459	3,165/3,647	3,311/1,731
	(3)	23.71	100.00	44.37	46.46	65.67
shuttle	**(1)**	16/57	2/0	11/0	10/49	1/10
	(2)	3,727/1,620	4,873/0	7210/0	7,220/2,276	1/7,915
	(3)	69.71	100.00	100.00	76.03	0.01
waveform	**(1)**	18/42	5/5	11/0	25/84	20/33
	(2)	4,639/173	4,158/410	5,511/0	4,496/1,058	5,146/2,714
	(3)	96.41	91.02	100.00	80.95	65.47
wineqw	**(1)**	27/90	2/0	12/0	15/62	20.32
	(2)	6,082/725	4,444/0	8,376/0	4,573/306	8,526/1,412
	(3)	89.35	100.00	100.00	98.73	85.79
yeast	**(1)**	39/178	5/0	49/83	49/173	28/77
	(2)	5,828/325	7,033/0	8,821/224	10,630/1,172	7,501/660
	(3)	94.73	100.00	97.52	86.13	91.92

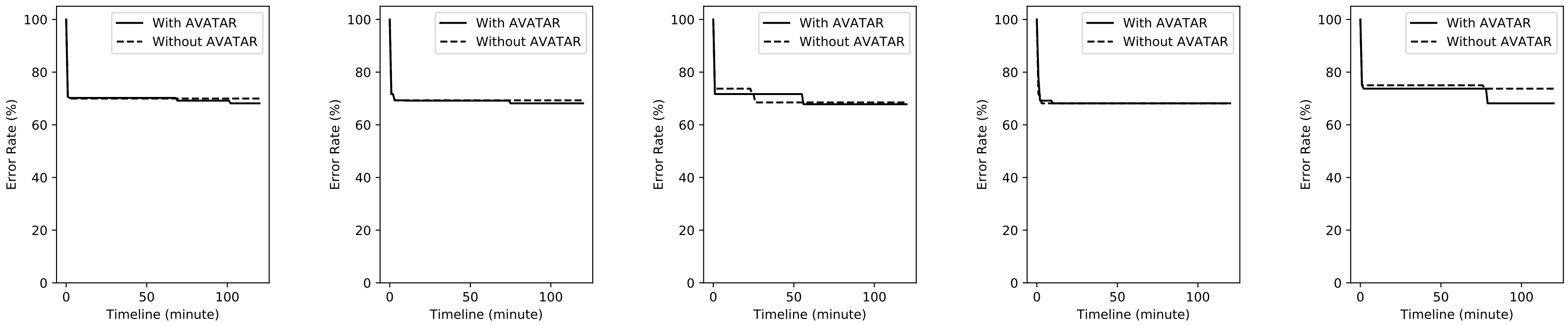
Table 2: Negative impacts of invalid pipelines using Auto-sklearn. (1): the number of invalid/ valid pipelines, (2): the total evaluation time of invalid/ valid pipelines (s), (3): the wasted evaluation time (%).

Dataset	**Criteria**	**Iter 1**	**Iter 2**	**Iter 3**	**Iter 4**	**Iter 5**
abalone		crashed	crashed	crashed	crashed	crashed
adult		crashed	crashed	crashed	crashed	crashed
amazon		crashed	crashed	crashed	crashed	crashed
car		crashed	crashed	crashed	crashed	crashed
cifar10small	**(1)**	4/4	7/1	6/4	6/7	6/2
	(2)	4579/2457	6819/263	5885/1194	3867/3253	6613/508
	(3)	65.08	96.29	83.14	54.31	92.86
convex	**(1)**	9/19	9/18	9/16	8/18	9/14
	(2)	2692/4475	2625/4539	2880/4291	2646/4524	3121/4052
	(3)	37.57	36.65	40.16	36.90	43.51
dexter	**(1)**	9/44	9/42	9/44	9/44	9/44
	(2)	4917/2177	3773/3333	4742/2366	4704/2415	4028/3079
	(3)	69.31	53.10	66.72	66.08	56.68
dorothea		crashed	crashed	crashed	crashed	crashed
gcredit		crashed	crashed	crashed	crashed	crashed
gisette	**(1)**	6/7	5/6	4/3	5/4	3/5
	(2)	5227/1920	4380/2723	6734/387	6125/1002	5156/1988
	(3)	73.13	61.67	94.56	85.94	72.17
kddcup		crashed	crashed	crashed	crashed	crashed
krvskp		crashed	crashed	crashed	crashed	crashed
madelon	**(1)**	9/50	11/47	9/49	9/49	9/49
	(2)	4215/2925	4763/2374	4914/2225	4774/2366	3718/3423
	(3)	59.03	66.74	68.84	66.86	52.07
mnist	**(1)**	3/8	6/10	6/10	7/5	5/9
	(2)	2802/4355	3379/3795	4013/3160	5837/1337	5516/1658
	(3)	39.15	47.10	55.94	81.37	76.89
secom		crashed	crashed	crashed	crashed	crashed
semeion	**(1)**	4/32	12/55	7/51	5/37	4/26
	(2)	6122/1042	5906/1228	5412/1731	5715/1444	5943/1230
	(3)	85.46	82.79	75.77	79.83	82.85
shuttle	**(1)**	3/10	3/10	2/18	2/15	3/19
	(2)	4329/2853	4312/2870	2087/5087	1897/5282	1957/5215
	(3)	60.27	60.03	29.09	26.42	27.28
waveform	**(1)**	3/21	4/27	3/27	3/30	3/22
	(2)	4081/3098	4295/2873	3679/3489	3890/3277	3876/3299
	(3)	56.84	59.92	51.32	54.28	54.02
wineqw	**(1)**	3/54	2/39	3/55	2/57	2/43
	(2)	1891/5244	2845/4309	1815/5322	9/7127	2140/5012
	(3)	26.50	39.77	25.44	0.12	29.92
yeast		crashed	crashed	crashed	crashed	crashed

using Auto-sklearn and the dataset *cifar10small, convex and dexter* are 1, 0 and 0 respectively. However, standard deviation of the number of invalid pipelines in the case of using AutoWeka4MCPS and the dataset *cifar10small, convex and dexter* are 6, 3, 5 respectively. The reason is that Auto-sklearn uses meta-learning to initialise with promising ML pipelines. The experiments with the datasets *abalone*, *adult*, *amazon*, *car*, *dorothea*, *gcredit*, *kddcup*, *krvskp*, *secom*, and *yeast* show that Auto-sklearn limits the generation of invalid pipelines by making assumption about cleaned input datasets. The experiments crash if the input datasets have data quality issues (i.e. missing values) or not transformed into a specific, required format (i.e. all attributes must be in numeric format.). Similar to AutoWeka4MCPS, the Auto-sklearn can not handle invalid pipelines effectively even with the initialisation using the meta-learning. In conclusion, we empirically prove the presence of invalid pipelines and that the wasted time used to evaluate these invalid pipelines can be significant.

1.2. Experiments to compare the performance of SMAC with and without the AVATAR

Figures 1 to 20 show the convergence of error rate of the iterations. Because the exploitation and exploration of SMAC may only evaluate several folds of the dataset with a configuration to decide the next configurations, we use the error rate of the folds of the datasets for the visualisation of the convergence. We can see that in 98 out of 100 iterations the convergence of the AVATAR is faster than or equal without the AVATAR. There are 2 out of 100 the convergence of the AVATAR is slower than without the AVATAR, (convex, Iter4) and (kddcup, Iter2). In 39 out of 100 iterations which have the same error rate with and without the AVATAR, there 5 iterations which have the faster convergence with the AVATAR after 2 hours optimisation, (abalone, Iter2), (convex, Iter3), (dorothea, Iter3), (abalone, Iter5) and (gcredit, Iter5). In additions, there 7 iterations which have the faster convergence with the AVATAR before the end of the optimisation time, but they have the same convergence at the end of the optimisation time. They are (winequality, Iter1), (cifar10small, Iter2), (amazon, Iter2), (mnist, Iter2), (gisette, Iter3), (cifar10small, Iter4) and (gisette, Iter4). Even in the iterations which have the same error rate with and without the AVATAR, we can also see that SMAC with the AVATAR converges faster than without the AVATAR.



(a) Iter 1

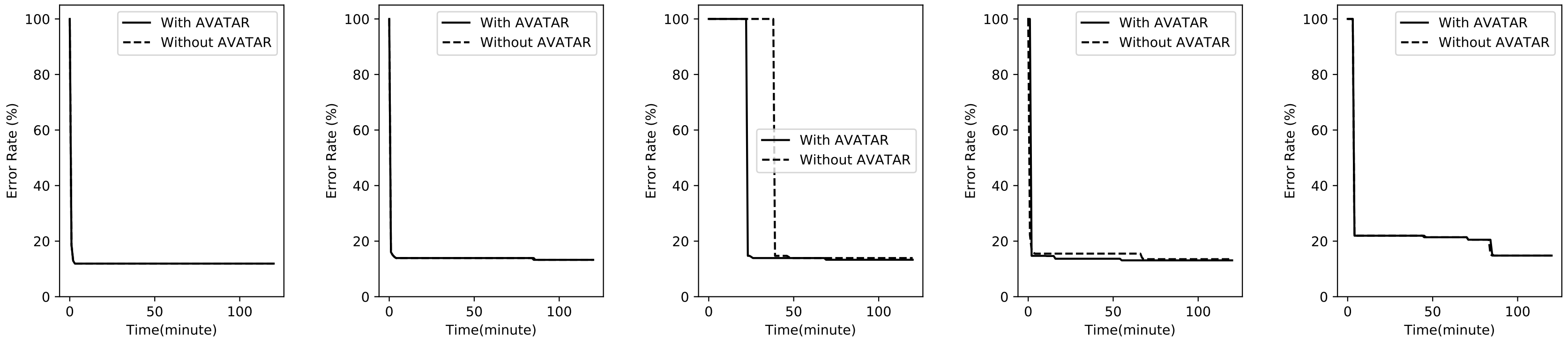
(b) Iter 2

(c) Iter 3

(d) Iter 4

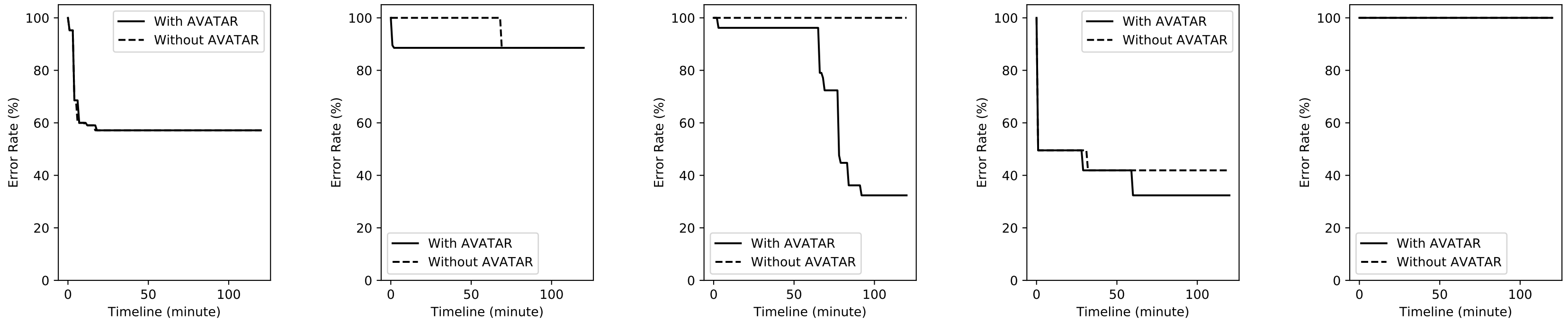
(e) Iter 5

Figure 1: The convergence of the most promising pipelines of the dataset abalone.



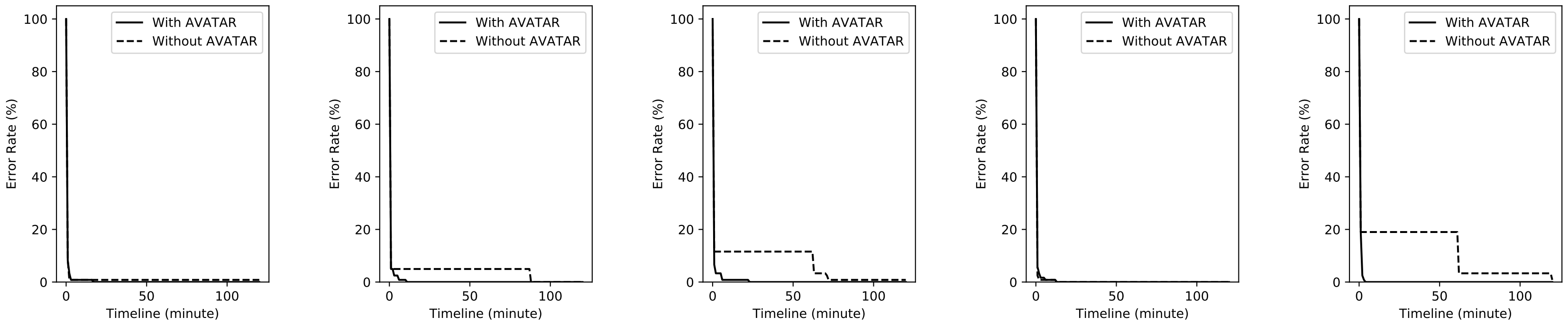
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 2: The convergence of the most promising pipelines of the dataset adult.



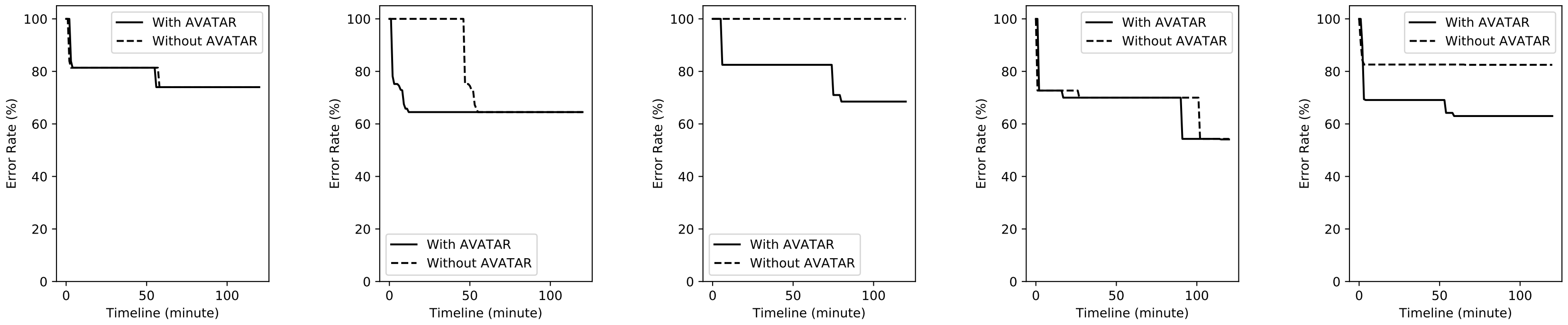
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 3: The convergence of the most promising pipelines of the dataset amazon.



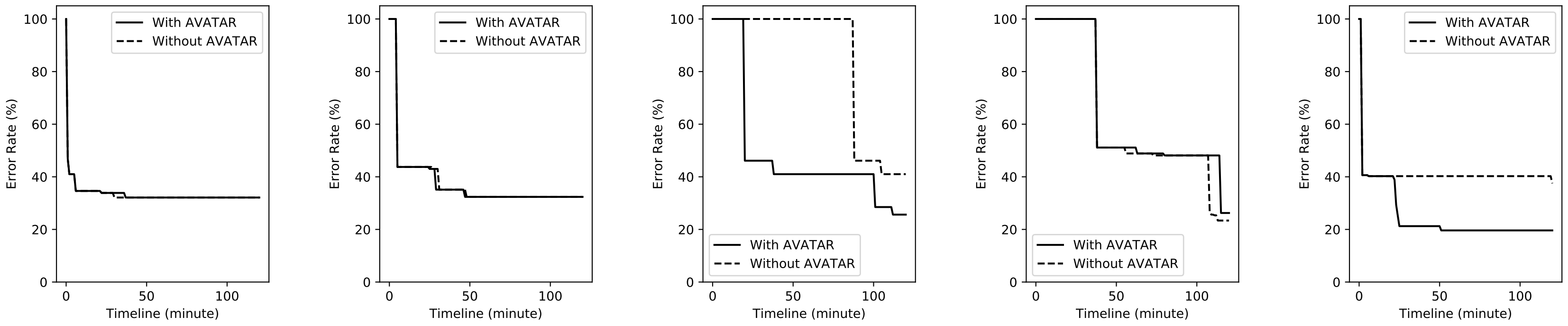
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 4: The convergence of the most promising pipelines of the dataset car.



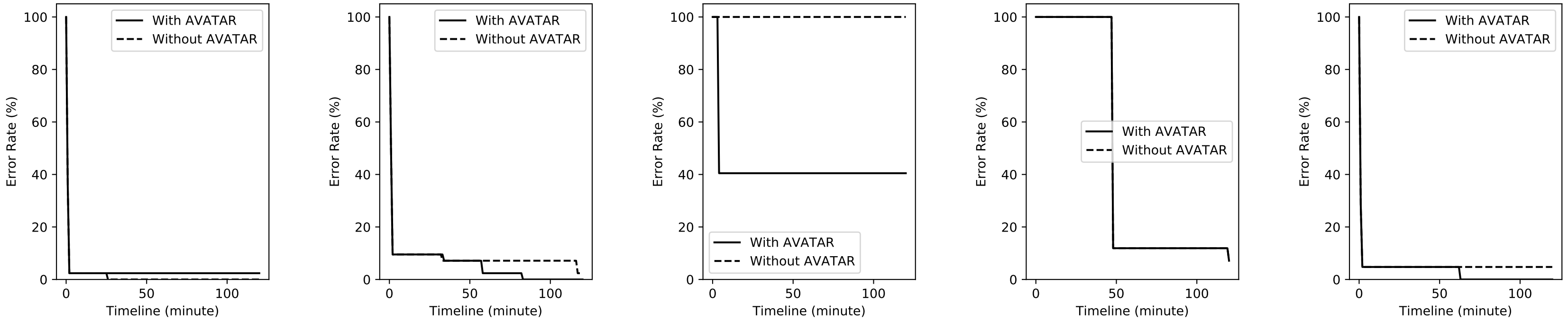
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 5: The convergence of the most promising pipelines of the dataset cifar10small.



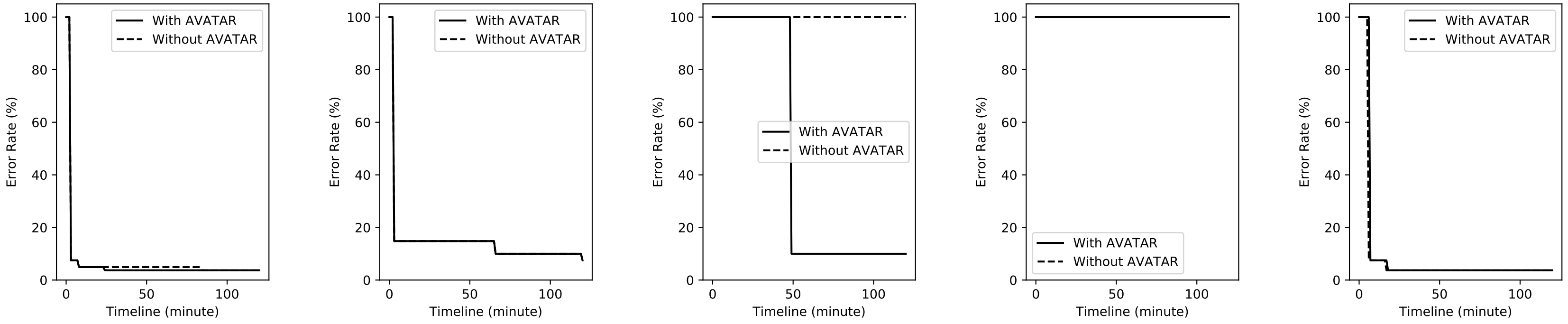
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 6: The convergence of the most promising pipelines of the dataset convex.



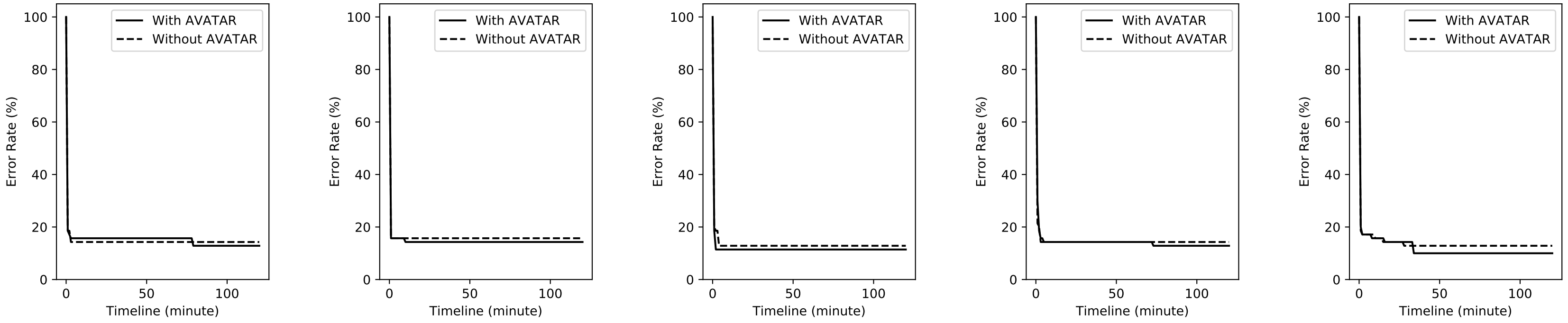
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 7: The convergence of the most promising pipelines of the dataset dexter.



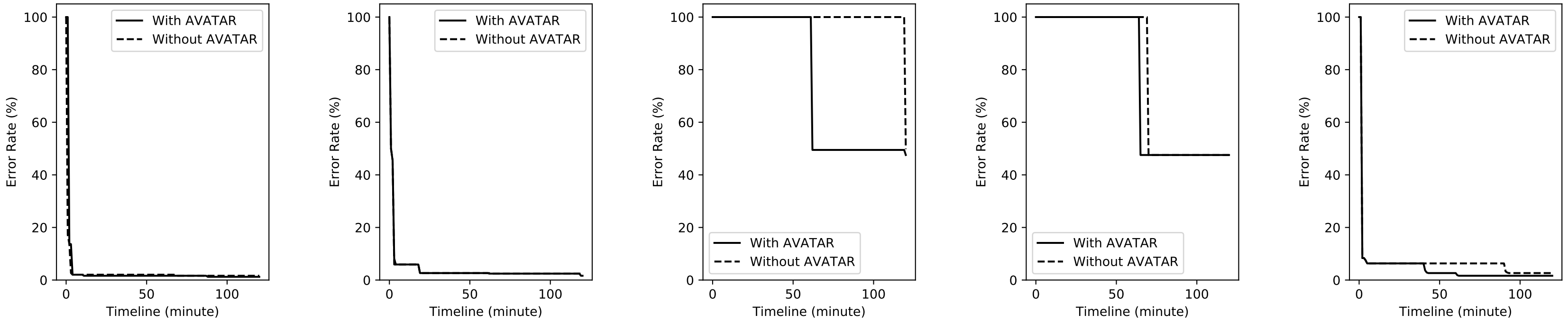
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 8: The convergence of the most promising pipelines of the dataset dorothea.



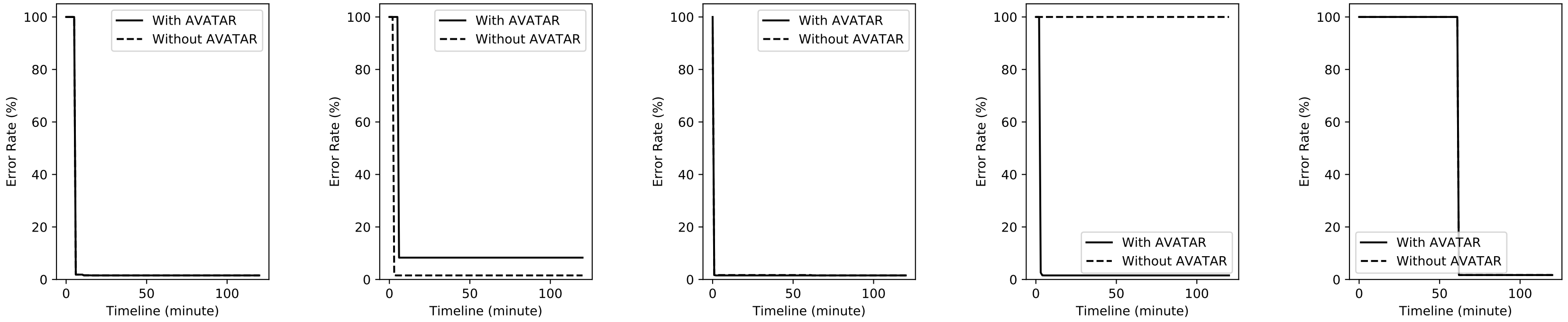
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 9: The convergence of the most promising pipelines of the dataset gcredit.



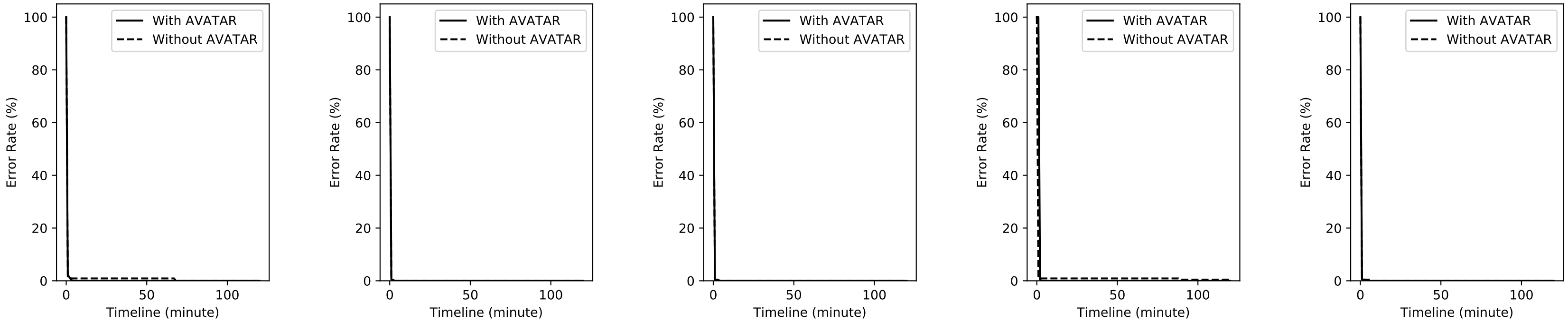
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 10: The convergence of the most promising pipelines of the dataset gisette.



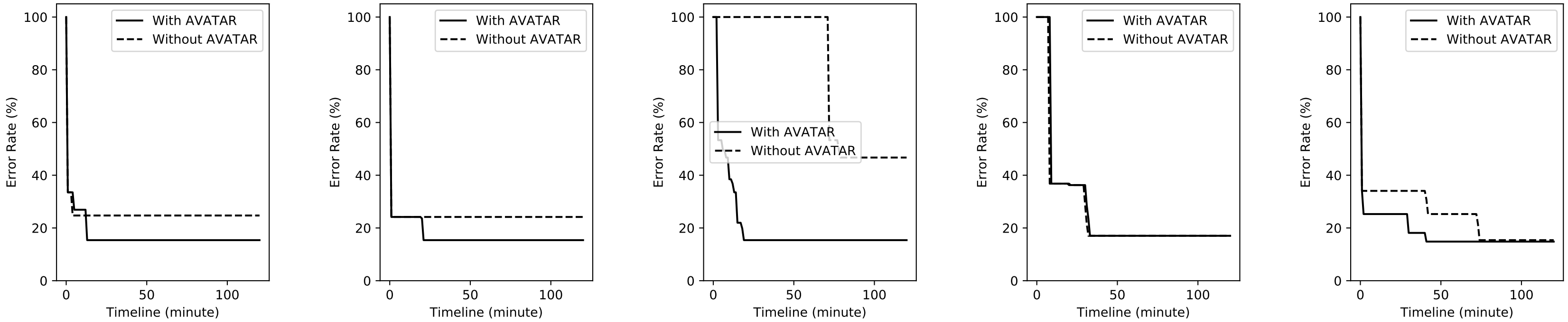
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 11: The convergence of the most promising pipelines of the dataset kddcup.



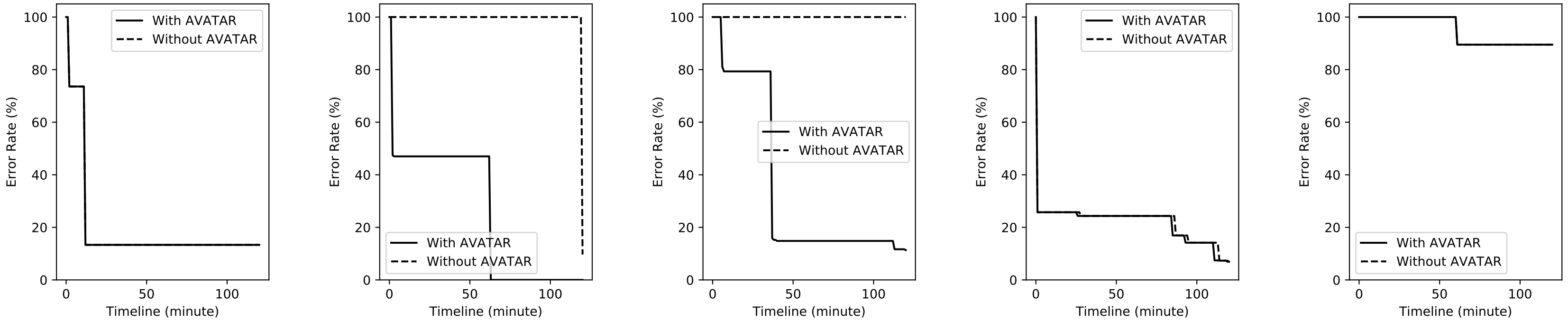
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 12: The convergence of the most promising pipelines of the dataset krvskp.



(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 13: The convergence of the most promising pipelines of the dataset madelon.



(a) Iter 1

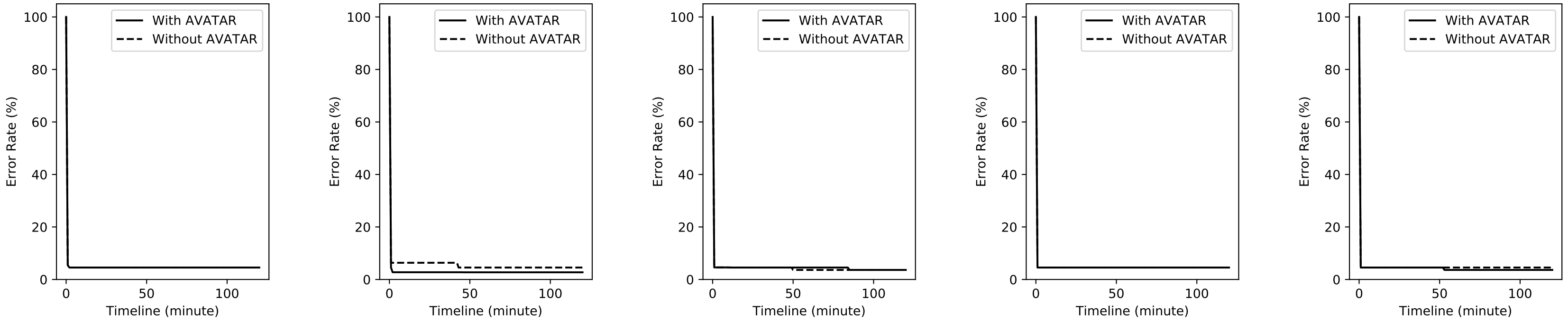
(b) Iter 2

(c) Iter 3

(d) Iter 4

(e) Iter 5

Figure 14: The convergence of the most promising pipelines of the dataset mnist.



(a) Iter 1

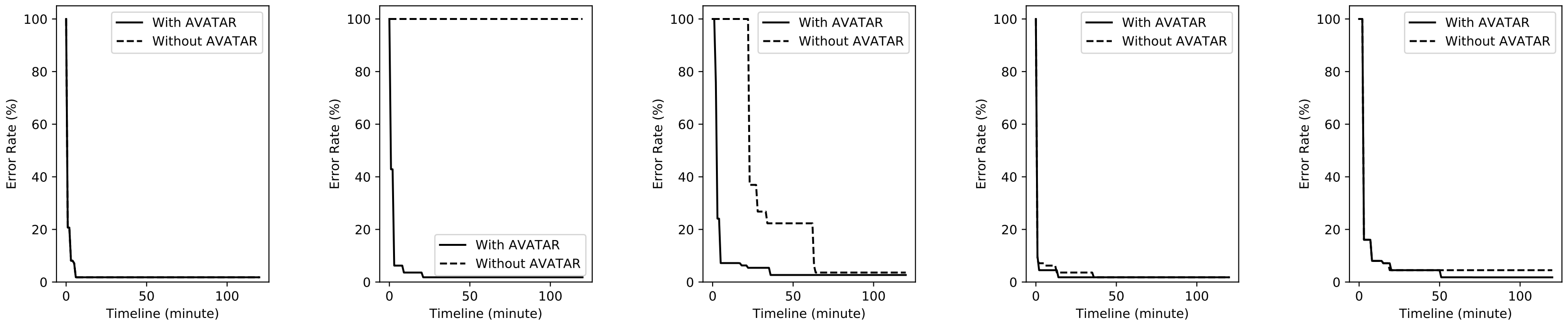
(b) Iter 2

(c) Iter 3

(d) Iter 4

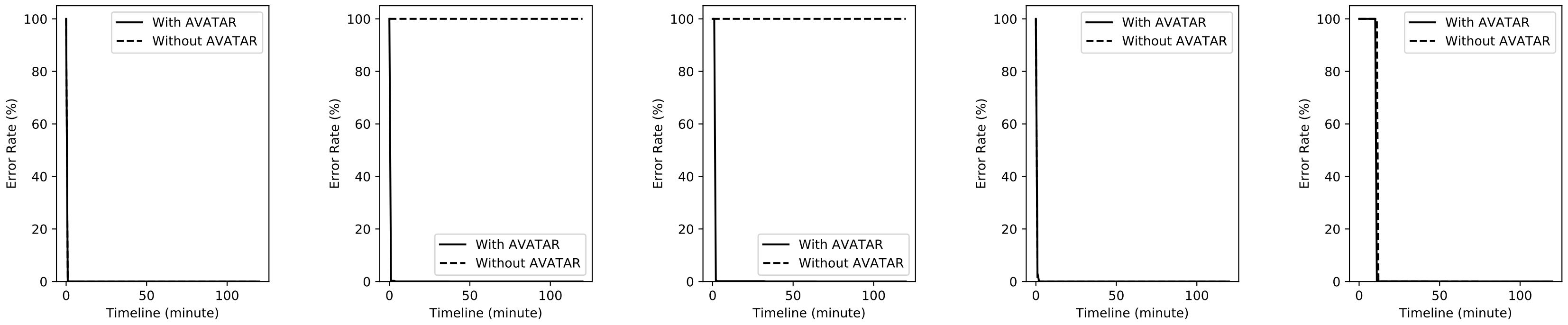
(e) Iter 5

Figure 15: The convergence of the most promising pipelines of the dataset secom.



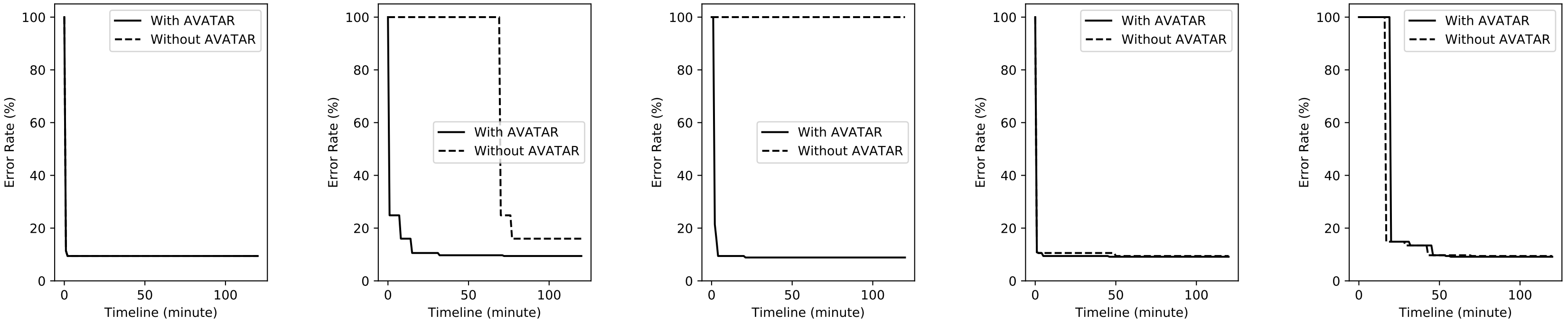
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 16: The convergence of the most promising pipelines of the dataset semeion.



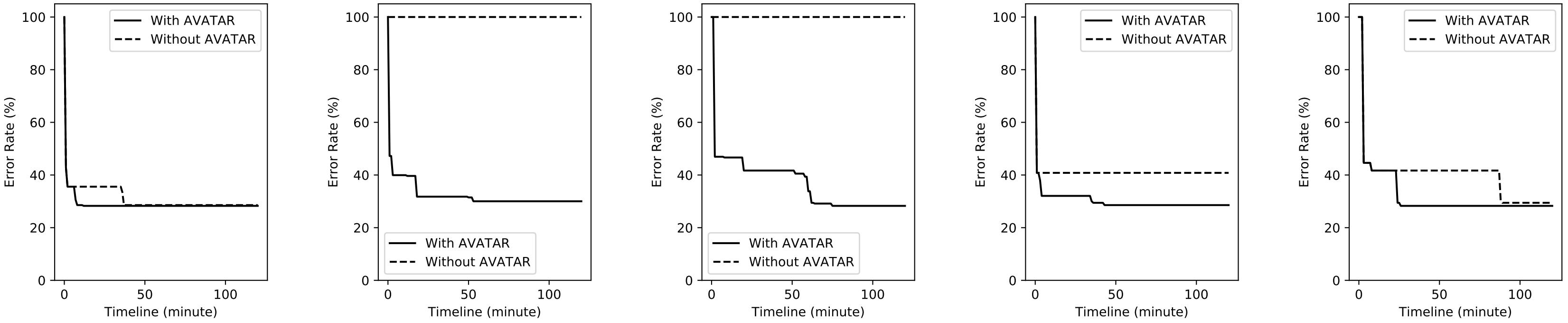
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 17: The convergence of the most promising pipelines of the dataset shuttle.



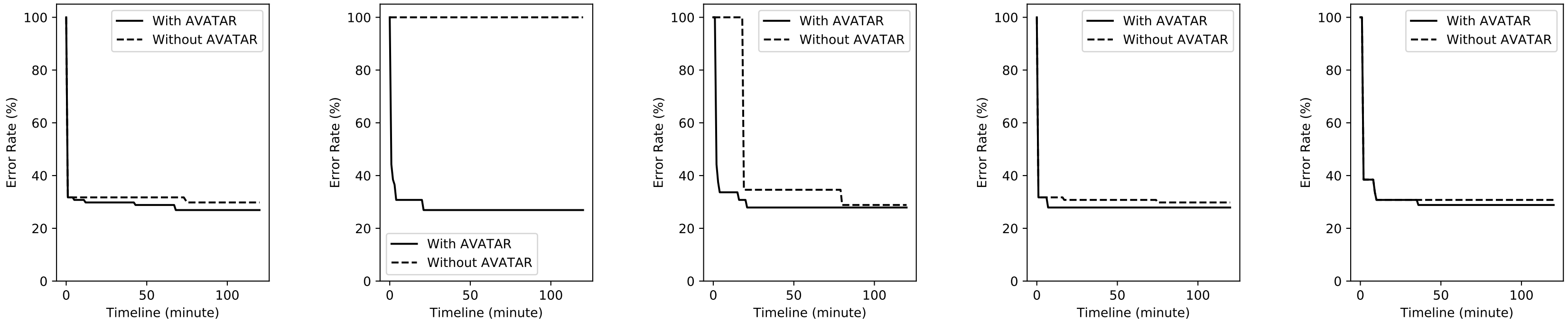
(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 18: The convergence of the most promising pipelines of the dataset waveform.



(a) Iter 1 (b) Iter 2 (c) Iter 3 (d) Iter 4 (e) Iter 5

Figure 19: The convergence of the most promising pipelines of the dataset winequality.



(a) Iter 1

(b) Iter 2

(c) Iter 3

(d) Iter 4

(e) Iter 5

Figure 20: The convergence of the most promising pipelines of the dataset yeast.