Online Appendix for AIJ Article
"Algorithm Runtime Prediction: Methods & Evaluation"

Frank Hutter, Lin Xu, Holger H. Hoos, and Kevin Leyton-Brown

Abstract

In this online appendix, we provide supplementary material for our AIJ article "Algorithm Runtime Prediction: Methods & Evaluation". Specifically, we provide the proof of Proposition 1, details on ridge regression variant RR-el, details on the data used, and additional experimental results.

Appendix B. Proof of Proposition 1

$$K_{\text{cat}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\sum_{l=1}^{p}(-\lambda_l \cdot \mathbb{I}_{x_{i,l} \neq x_{j,l}})\right). \tag{B.1}$$

$$K_{\text{mixed}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\sum_{l \in \mathcal{P}_{\text{cont}}} \left(-\lambda_l \cdot (x_{i,l} - x_{j,l})^2\right) + \sum_{l \in \mathcal{P}_{\text{cat}}} \left(-\lambda_l \cdot \mathbb{I}_{x_{i,l} \neq x_{j,l}}\right)\right).$$

Proposition 1 (K_{mixed} is positive definite)**.** *For any combination of continuous and categorical input dimensions $\mathcal{P}_{cont}$ and $\mathcal{P}_{cat}$, K_{mixed} is a positive definite kernel function.*

Proof. We will use the facts that any constant is a positive definite kernel function, that the space of positive definite kernel functions is closed under addition and multiplication, and that k is a positive definite kernel function if there exists an embedding ϕ into some (potentially infinite-dimensional) space such that $k(x, z) = \phi(x)^\intercal \cdot \phi(z)$ [see, *e.g.*, 3]. First, consider a one-dimensional input with finite domain $\mathcal{I}$; we will begin by showing that K_{cat} is a positive definite kernel function for this domain. Let $a_1, \ldots, a_m$ denote the finitely many distinct elements of $\mathcal{I}$, and define an embedding ϕ into an m-dimensional space, mapping each element a_j to the m-dimensional indicator vector $\boldsymbol{v}_{a_j}$ that is zero everywhere except at position j, where it is one. Then define a kernel function $k_1(x, z)$ for $x, z \in \mathcal{I}$ as $k_1(x, z) = \phi(x)^\intercal \cdot \phi(z) = \boldsymbol{v}_x^\intercal \cdot \boldsymbol{v}_z = \sum_{j=1}^{m} \boldsymbol{v}_x(j) \cdot \boldsymbol{v}_z(j) = \mathbb{I}_{x=z}$. To bring this in the form of Equation (B.1), we add the constant kernel function $k_2(x, z) = c = \exp(-\lambda)/(1 - \exp(-\lambda))$, and then multiply by the constant kernel function $k_3(x, z) = 1/(1 + c) = 1 - \exp(-\lambda)$. We can thus rewrite function K_{cat} as the product of positive definite kernels, thereby establishing that it, too, is positive definite:

$$\begin{aligned} K_{\text{cat}}(x,z) &= (k_1(x,z)+k_2(x,z))\cdot k_3(x,z) \\ &= \begin{cases} 1 & \text{if } x = z \\ \exp(-\lambda) & \text{otherwise} \end{cases} \\ &= \exp(-\lambda \cdot \mathbb{I}(x \neq z)). \end{aligned}$$

It remains to show that K_{mixed} is positive definite. This follows immediately: K_{mixed} is a product of positive definite kernels (one K_{cat} kernel per categorical input and one K_{cont} kernel per continuous input), and the space of positive definite kernel functions is closed under multiplication. □

Appendix C. Ridge Regression Variant RR-el: Eliminating redundant features

We now describe ridge regression variant RR-el [1], mentioned in footnote of the main article. That method started with a quadratic feature expansion (similar to variant RR in the main article) and then performed *backward selection*, iteratively eliminating, one at a time, the features whose values could accurately be predicted by a linear combination of the other features. This approach is infeasible if the number of features p is too large: the quadratic feature expansion yields $q = \binom{p}{2} + p \in \Theta(p^2)$ features, and the iterative backward selection requires time proportional to $q^5 = p^{10}$. (To see this, first note that eliminating the first feature requires building q models, each of which takes time cubic in q (rank-one updates do not apply since each of the models has a different response variable). Backward selection down to one feature then requires time proportional to $\sum_{i=1}^{q} i \cdot \Theta(i^3) \in \Theta(q^5)$.) This approach would thus have been infeasible on our data, since we use substantially larger feature sets than in [1]. Nevertheless, it is one of the aims of this work to compare all previous approaches for building EPMs, and so we followed this method as faithfully as possible, given resource constraints. Specifically, we performed a quadratic feature expansion and then checked for each resulting feature f_i whether it could be predicted by a linear combination of up to 10 other features (selected in turn by greedy forward selection), dropping f_i under the same conditions as in the original work: if the adjusted R^2 value of a linear model predicting f_i exceeds 0.99999. Also following the original work, we set the ridge penalizer to $\epsilon = 10^{-6}$.

Ridge regression variant RR-el required extremely long training times (up to 3 days for the 7 012 instances in COMPETITON, compared to 47 seconds for the second-slowest method, projected processes) and we thus only ran 2-fold cross-validation with it. For comparability, we reran all other methods with 2-fold cross-validation as well; the results are given in Table C.1.

We also include RR-el in Figures E.1–E.10, which show raw runtime predictions of all models for unseen instances of all benchmarks. RR-el performed very poorly in most cases, not eliminating enough features and suffering from overfitting. (The one interesting exception was CPLEX-RCW (see Column 3 of Figure E.8), where RR-el performed substantially better than the other two ridge regression methods.) Due to these drawbacks, we did not consider RR-el in our further experiments.

	RMSE							Time to learn model (s)						
Domain	RR	RR-el	SP	NN	PP	RT	RF	RR	RR-el	SP	NN	PP	RT	RF
Minisat 2.0-COMPETITON	1.1	2.9E2	1.3	0.7	0.83	0.75	**0.55**	**4.9**	2.9E5	15	12	46	12	11
Minisat 2.0-HAND	1.2	3.2	1.3	0.77	0.87	0.92	**0.67**	3.7	2.6E5	8.4	3.8	49	3.5	**2.9**
Minisat 2.0-RAND	0.64	3.4	0.77	**0.41**	0.58	0.57	**0.41**	**4**	1.3E5	5.8	6.2	47	4.1	4.3
Minisat 2.0-INDU	0.97	86	1.0	0.99	0.86	0.93	**0.69**	3.6	2.3E5	7	3.2	44	3.8	**2.4**
Minisat 2.0-SWV-IBM	0.54	6.8	0.75	0.4	0.61	0.29	**0.21**	3.4	2E5	7.3	2.6	45	2.9	**1.5**
Minisat 2.0-IBM	0.63	76	12	0.38	0.45	0.38	**0.25**	3.4	2E5	4.5	1.6	43	1.6	**0.85**
Minisat 2.0-SWV	0.48	0.54	0.22	0.18	0.11	0.12	**0.1**	3.5	1.8E5	6.2	1.4	62	1.5	**0.62**
CryptoMinisat-INDU	0.97	20	1.0	1.2	1.0	1.0	**0.82**	3.5	2.3E5	6.8	3.2	44	2.9	**2.1**
CryptoMinisat-SWV-IBM	0.9	7.7	0.93	0.76	0.89	0.69	**0.55**	3.6	2E5	5.1	2.8	44	2.8	**1.5**
CryptoMinisat-IBM	0.74	1.4E2	0.95	0.63	0.64	0.6	**0.49**	3.4	1.9E5	5.5	1.6	43	1.6	**0.85**
CryptoMinisat-SWV	0.94	2.4	1.1	0.74	0.68	0.7	**0.57**	3.4	1.8E5	4.4	1.3	63	1.5	**0.62**
SPEAR-INDU	1.0	47	1.0	1.0	0.89	0.87	**0.73**	3.6	2.3E5	6.4	3.1	47	3.2	**2.3**
SPEAR-SWV-IBM	0.7	9.4	0.83	0.58	0.82	0.58	**0.44**	3.5	2E5	7.9	2.6	49	2.8	**1.5**
SPEAR-IBM	0.79	4.3E2	3.6	0.6	0.71	0.52	**0.45**	3.4	1.9E5	5.4	1.6	48	1.6	**0.89**
SPEAR-SWV	0.49	3.0	0.68	0.52	0.5	0.58	**0.44**	3.4	1.8E5	6.4	1.4	52	1.5	**0.64**
tnm-RANDSAT	1.0	8.6	1.1	1.1	0.95	1.2	**0.92**	3.7	1.2E5	7.8	3.8	49	4.3	**2.8**
SAPS-RANDSAT	0.97	4.9	1.1	0.83	0.77	1.0	**0.7**	3.6	1.2E5	7	3.8	44	3.8	**2.6**
CPLEX-BIGMIX	9E11	1.4E18	1.1E7	1.1	1.1	0.99	**0.72**	3.3	1.4E5	8.5	2.7	38	3.1	**1.9**
Gurobi-BIGMIX	3.1E12	3.5E19	6.2E2	1.8	1.3	1.5	**1.2**	3.2	1.4E5	3.9	2.7	38	3.2	**2**
SCIP-BIGMIX	2.6	1.9E19	1.3	0.99	0.99	0.76	**0.63**	3.4	1.4E5	6.2	2.7	37	2.9	**2.1**
lp_solve-BIGMIX	3.6	7.7E18	1.6	0.93	1.1	0.69	**0.57**	3.3	1.3E5	4	2.7	44	**1.6**	2.5
CPLEX-CORLAT	0.49	13	0.53	0.61	**0.46**	0.65	0.5	3.0	1.9E4	5.6	3.1	26	2.8	**1.8**
Gurobi-CORLAT	0.4	7.5	0.44	0.46	**0.38**	0.49	**0.38**	3.1	1.8E4	4.3	3.2	28	2.8	**1.8**
SCIP-CORLAT	0.39	12	0.42	0.47	**0.37**	0.51	0.39	3.1	1.8E4	7	3.2	27	3.0	**1.9**
lp_solve-CORLAT	**0.43**	14	0.47	0.49	0.49	0.56	**0.43**	3.0	1.8E4	3.8	3.1	29	**1.7**	2.3
CPLEX-RCW	0.26	0.1	0.31	0.12	0.05	0.09	**0.03**	3.1	1.2E4	4.2	3.2	26	2.8	**1.5**
CPLEX-REG	**0.39**	1.0	**0.39**	0.52	**0.39**	0.57	0.44	2.4	5.7E3	5.4	3.0	24	2.8	**1.9**
CPLEX-CR	0.46	2.3	0.58	0.47	**0.43**	0.61	0.45	**3.6**	2.7E4	8.6	6.1	32	6.5	4.3
CPLEX-CRR	0.44	0.74	0.54	0.42	0.38	0.49	**0.37**	**4.6**	4.8E4	12	9.7	36	12	6.9
LK-H-RUE	**0.62**	9.5E2	0.63	0.66	**0.62**	0.89	0.68	3.5	1.4E4	**1.0**	7.3	23	6.3	5.7
LK-H-RCE	0.72	6.5E2	0.72	0.83	**0.71**	1.0	0.77	3.7	1.6E4	**2.5**	7.5	23	6.3	5.5
LK-H-TSPLIB	6.5	16	80	1.8	1.8	1.4	**1.0**	1.7	1.4E4	2.8	0.51	4	0.31	**0.12**
Concorde-RUE	0.54	6.1E4	**0.43**	0.46	**0.43**	0.6	0.45	3.6	1.4E4	**2.5**	7.3	29	6	5.2
Concorde-RCE	**0.33**	28	0.34	0.36	0.34	0.46	0.36	3.4	1.6E4	**2**	7.4	26	6.2	5.2
Concorde-TSPLIB	3.3	32	14	1.1	1.1	0.64	**0.57**	1.7	1.4E4	3.0	0.59	4.5	0.48	**0.15**

Table C.1: Quantitative comparison of models for runtime predictions on previously unseen instances. We report 2-fold cross-validation performance. Lower RMSE values are better (0 is optimal). Note the very large RMSE values for the ridge regression variants on some data sets (we use scientific notation, denoting "$\times 10^x$" as "Ex"); these large errors are due to extremely small/large predictions for a few data points.

Appendix D. Details on the Data Used

Table D.2 provides statistics of the log runtime distributions for all our benchmarks.

Appendix E. Additional Experimental Results

Appendix E.1. Additional Results for Section 6: Performance Predictions for Unseen Instances

Figures E.1–E.10 show raw runtime predictions of all models for unseen instances of all benchmarks, providing additional details for Figure 4 in the main article. Table

Domain	min	q10	q25	median	q75	q90	max	q75-q25	q90-q10	mean	std
Minisat 2.0-COMPETITON	-2.3	-0.85	1.3	3.6	3.6	3.6	3.6	2.2	4.4	2.3	1.8
Minisat 2.0-HAND	-2.3	-1	0.8	3.3	3.6	3.6	3.6	2.8	4.6	2.1	1.9
Minisat 2.0-RAND	-2.3	-0.45	2.4	3.6	3.6	3.6	3.6	1.1	4	2.6	1.7
Minisat 2.0-INDU	-2.3	-1	0.71	2.1	3.6	3.6	3.6	2.9	4.6	1.8	1.7
Minisat 2.0-SWV-IBM	-2.3	-2	-0.61	0.037	0.74	3.6	3.6	1.3	5.6	0.25	1.6
Minisat 2.0-IBM	-2.3	-1.2	-0.14	0.55	2.1	3.6	3.6	2.3	4.7	0.92	1.7
Minisat 2.0-SWV	-2.3	-2.3	-1	-0.35	0.11	0.36	0.73	1.2	2.7	-0.6	0.9
CryptoMinisat-INDU	-2.3	-0.85	1.2	3.6	3.6	3.6	3.6	2.3	4.4	2.2	1.8
CryptoMinisat-SWV-IBM	-2.3	-1.4	-0.27	0.53	2.4	3.6	3.6	2.7	5	0.86	1.8
CryptoMinisat-IBM	-2.3	-0.82	0.14	0.79	3.6	3.6	3.6	3.4	4.4	1.3	1.8
CryptoMinisat-SWV	-2.3	-1.7	-0.96	0.14	0.96	3.6	3.6	1.9	5.3	0.28	1.6
SPEAR-INDU	-2.3	-1.2	1.2	2.9	3.6	3.6	3.6	2.4	4.8	2.1	1.8
SPEAR-SWV-IBM	-2.3	-1.7	-0.64	0.12	1.6	3.6	3.6	2.2	5.3	0.5	1.8
SPEAR-IBM	-2.3	-1	-0.16	0.72	3.6	3.6	3.6	3.7	4.6	1.1	1.8
SPEAR-SWV	-2.3	-2	-0.77	-0.18	0.24	1.2	3.6	1	3.2	-0.3	1.2
tnm-RANDSAT	-2.3	-2.3	-1.5	0.031	2.3	3.6	3.6	3.8	5.9	0.34	2.1
SAPS-RANDSAT	-2.3	-2.3	-1.7	0.18	3.6	3.6	3.6	5.3	5.9	0.59	2.3
CPLEX-BIGMIX	-2.3	-0.81	0.18	1.1	2.3	3.6	3.6	2.2	4.4	1.2	1.6
Gurobi-BIGMIX	-2.3	-0.62	0.32	1.5	3.6	3.6	3.6	3.2	4.2	1.5	1.6
SCIP-BIGMIX	-2.3	0.13	0.98	2	3.6	3.6	3.6	2.6	3.4	2	1.4
lp_solve-BIGMIX	-1.9	1.8	3.6	3.6	3.6	3.6	3.6	1.6e-05	1.7	3.2	1.1
CPLEX-CORLAT	-2.3	-2	-1	0.88	2.2	3.2	3.6	3.3	5.2	0.7	1.9
Gurobi-CORLAT	-2.3	-1.7	-0.85	0.64	1.5	2.2	3.3	2.4	3.9	0.38	1.4
SCIP-CORLAT	-2	-1.2	-0.4	0.88	1.7	2.3	3.6	2.1	3.5	0.7	1.3
lp_solve-CORLAT	-1.6	-1	1.1	3.6	3.6	3.6	3.6	2.4	4.6	2.4	1.8
CPLEX-RCW	0.63	1	1.4	1.9	2.4	3.1	3.6	0.98	2.1	2	0.73
CPLEX-REG	0.18	0.96	1.6	2.2	2.7	3	3.6	1	2.1	2.1	0.76
CPLEX-CR	-2.3	-1.5	0.7	1.9	2.6	3.1	3.6	1.9	4.6	1.4	1.6
CPLEX-CRR	-2.3	-0.51	1	1.9	2.5	3.1	3.6	1.5	3.6	1.6	1.4
LK-H-RUE	-1.7	-0.62	-0.018	0.61	1.1	1.9	3.6	1.1	2.5	0.63	0.98
LK-H-RCE	-1.5	-0.39	0.41	1.1	1.8	2.5	3.6	1.3	2.9	1.1	1.1
LK-H-TSPLIB	-2.3	-2.3	-2.3	-0.4	0.97	3.6	3.6	3.3	5.9	-0.17	1.9
Concorde-RUE	-0.77	0.45	1.3	2.1	2.7	3.2	3.6	1.4	2.7	2	1
Concorde-RCE	-0.7	0.58	1.2	1.7	2.1	2.5	3.6	0.96	1.9	1.6	0.73
Concorde-TSPLIB	-0.74	-0.41	0.048	1.2	2.7	3.6	3.6	2.6	4	1.4	1.4

Table D.2: Statistics of logarithmic algorithm runtime distributions for our benchmarks. qX stands for the q-th percentile. Note that q75-q25 is the equivalent of the interquartile *ratio* since we are working in the log domain; likewise for q90-q10.

E.3 provides correlation coefficients and log likelihoods for predictions in the feature space, complementing RMSEs and training times in Table 2 of the main article.

	Spearman rank correlation coefficient						Log likelihood	
Domain	RR	SP	NN	PP	RT	RF	PP	RF
Minisat 2.0-COMPETITON	0.69	0.57	0.86	0.79	0.83	**0.9**	-4.78	**−0.33**
Minisat 2.0-HAND	0.69	0.59	0.87	0.81	0.84	**0.91**	-2.65	**−0.43**
Minisat 2.0-RAND	0.79	0.74	**0.82**	0.8	0.78	**0.83**	-1.12	**−0.18**
Minisat 2.0-INDU	0.7	0.66	0.85	0.79	0.87	**0.92**	-5.72	**−0.43**
Minisat 2.0-SWV-IBM	0.95	0.89	0.97	0.96	0.98	**0.99**	-6.64	**0.12**
Minisat 2.0-IBM	0.94	0.91	0.97	0.97	0.98	**0.99**	-6.13	**0.06**
Minisat 2.0-SWV	0.94	0.95	0.97	**0.98**	**0.99**	**0.99**	-4.83	**0.2**
CryptoMinisat-INDU	0.66	0.59	0.72	0.71	0.76	**0.81**	-5.99	**−0.9**
CryptoMinisat-SWV-IBM	0.93	0.9	0.94	0.91	0.96	**0.97**	-6.91	**−0.37**
CryptoMinisat-IBM	0.93	0.85	0.94	0.94	**0.96**	**0.97**	-5.8	**−0.23**
CryptoMinisat-SWV	0.92	0.94	**0.95**	0.93	**0.97**	**0.97**	-6.88	**−0.59**
SPEAR-INDU	0.63	0.62	0.78	0.75	0.82	**0.88**	-6.66	**−0.59**
SPEAR-SWV-IBM	0.94	0.91	0.95	0.92	0.97	**0.98**	-13.6	**−0.22**
SPEAR-IBM	0.95	0.87	0.96	0.93	0.96	**0.98**	**−2.58**	**−0.18**
SPEAR-SWV	0.95	0.93	0.94	0.95	0.96	**0.97**	-7.33	**−0.19**
tnm-RANDSAT	0.87	0.86	0.9	0.89	0.83	**0.91**	-4.65	**−1.32**
SAPS-RANDSAT	0.9	0.86	0.93	0.92	0.91	**0.95**	-3.16	**−0.79**
CPLEX-BIGMIX	0.82	0.81	0.81	0.76	0.84	**0.9**	-8.06	**−0.7**
Gurobi-BIGMIX	**0.62**	**0.62**	0.57	0.57	0.54	**0.64**	-18.09	**−2.36**
SCIP-BIGMIX	0.81	0.76	0.81	0.73	0.84	**0.89**	-7.33	**−0.72**
lp_solve-BIGMIX	0.34	0.31	0.35	0.22	0.47	**0.6**	-13.22	**−0.24**
CPLEX-CORLAT	0.95	0.95	0.94	**0.96**	0.93	**0.95**	-4.46	**−0.53**
Gurobi-CORLAT	**0.95**	0.93	0.94	**0.95**	0.92	**0.95**	-3.12	**−0.38**
SCIP-CORLAT	0.94	0.94	0.93	**0.95**	0.91	**0.94**	-5.04	**−0.38**
lp_solve-CORLAT	0.76	0.75	0.75	0.75	**0.82**	0.76	-1.53	**−0.25**
CPLEX-RCW	0.94	0.92	0.99	**1**	1	**1**	**2**	0.23
CPLEX-REG	**0.87**	**0.87**	0.82	**0.87**	0.75	0.84	-8.52	**−0.59**
CPLEX-CR	0.9	0.86	0.9	**0.91**	0.85	0.9	-15.46	**−0.54**
CPLEX-CRR	0.89	0.85	0.9	**0.92**	0.88	**0.92**	-20.04	**−0.29**
LK-H-RUE	**0.82**	0.81	0.8	0.82	0.7	0.77	-46.04	**−1.16**
LK-H-RCE	**0.73**	0.72	0.69	**0.73**	0.55	0.68	-26.86	**−1.25**
LK-H-TSPLIB	**0.64**	**0.8**	**0.55**	**0.71**	**0.76**	**0.75**	-3.78	**−2**
Concorde-RUE	**0.88**	0.88	0.88	**0.88**	0.79	0.86	-34.47	**−0.66**
Concorde-RCE	**0.86**	0.85	0.85	0.85	0.76	0.84	-26.36	**−0.36**
Concorde-TSPLIB	**0.73**	**0.86**	0.72	0.67	**0.86**	**0.91**	**−1.44**	**−1.1**

Table E.3: Quantitative comparison of models for runtime predictions on unseen instances. We report 10-fold cross-validation performance. Higher rank correlations are better (1 is optimal); log-likelihoods are only defined for models that yield a predictive distribution (here: PP and RF); higher values are better. Boldface indicates results not statistically significantly from the best.

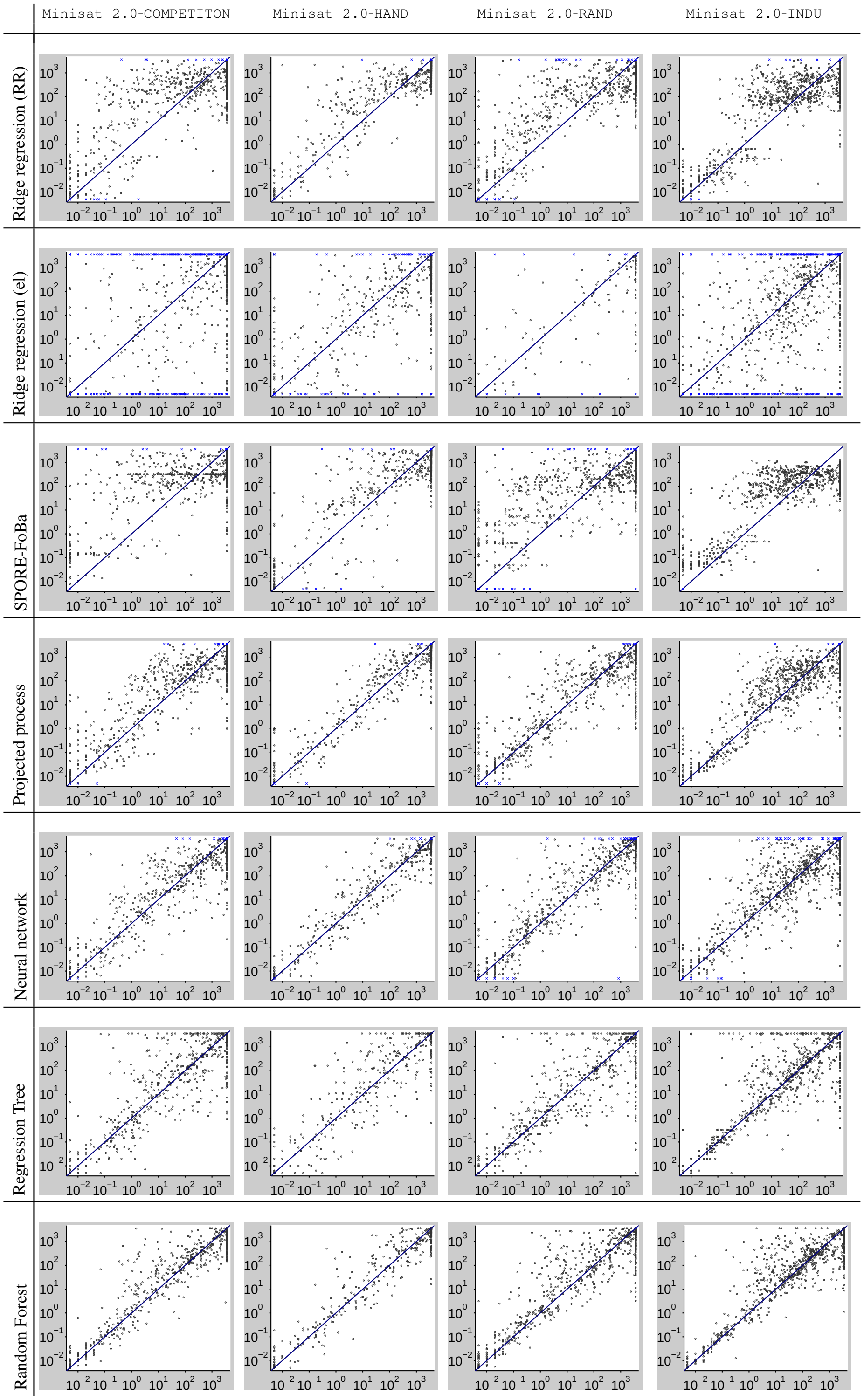


Figure E.1: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

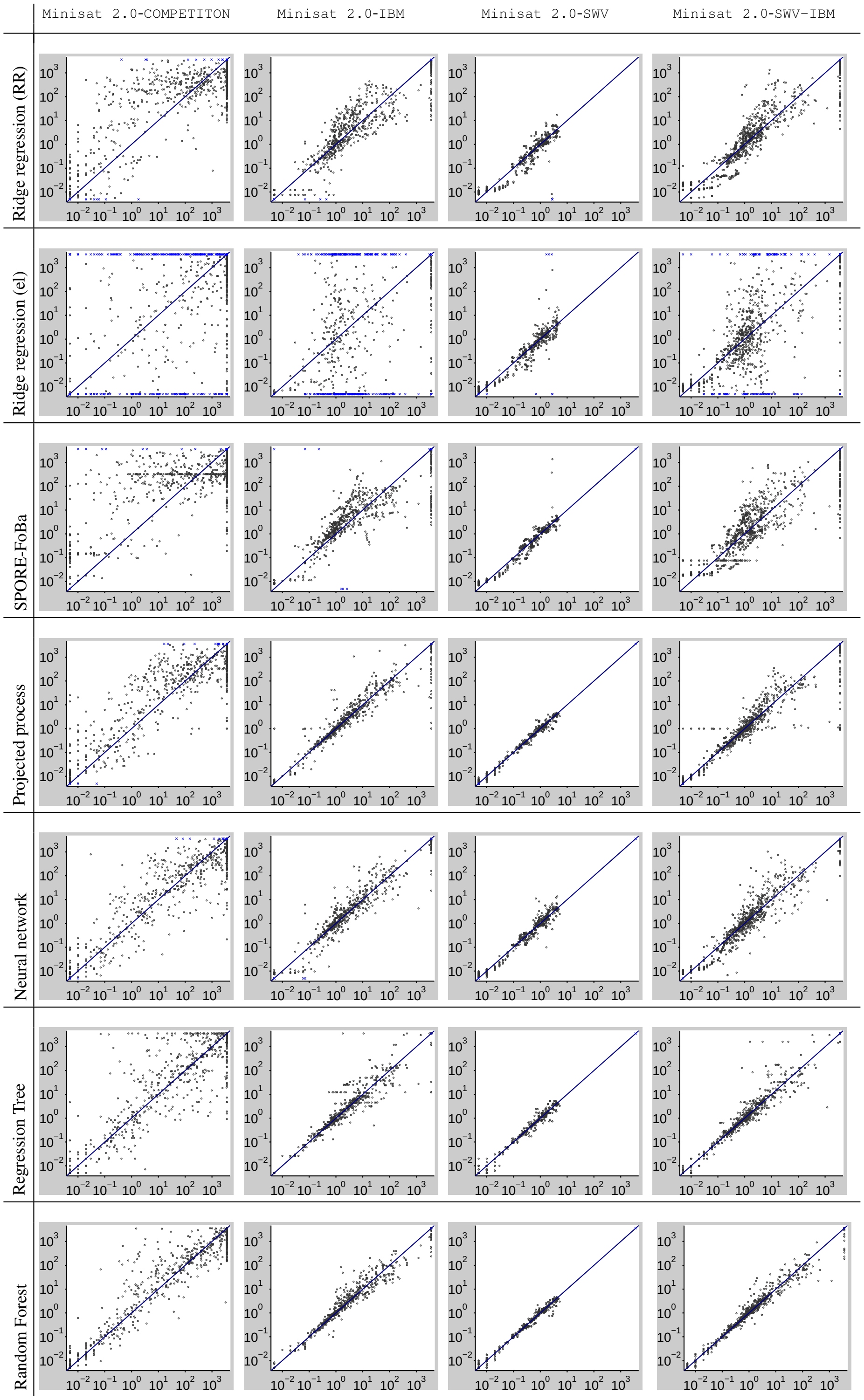


Figure E.2: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

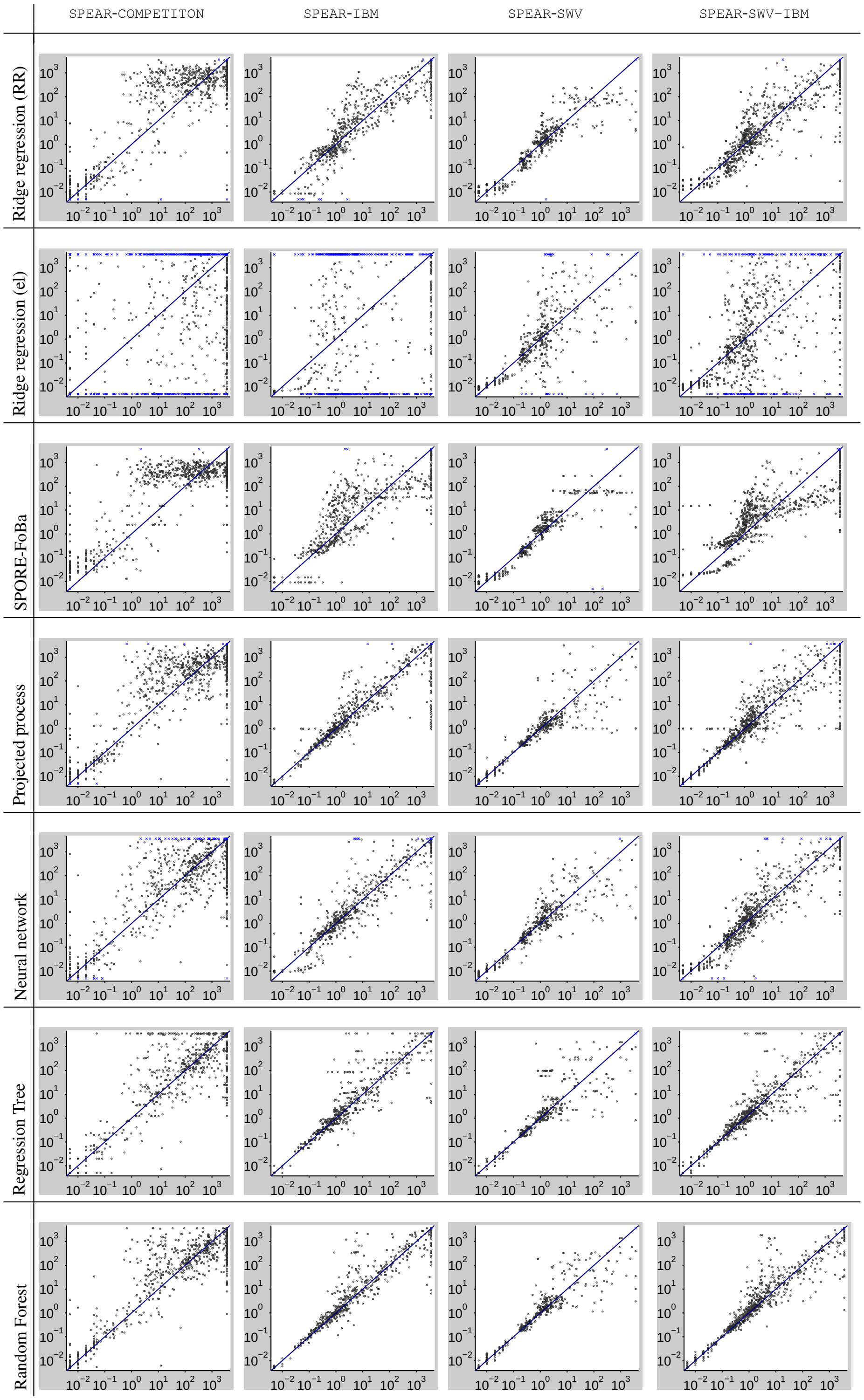


Figure E.3: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

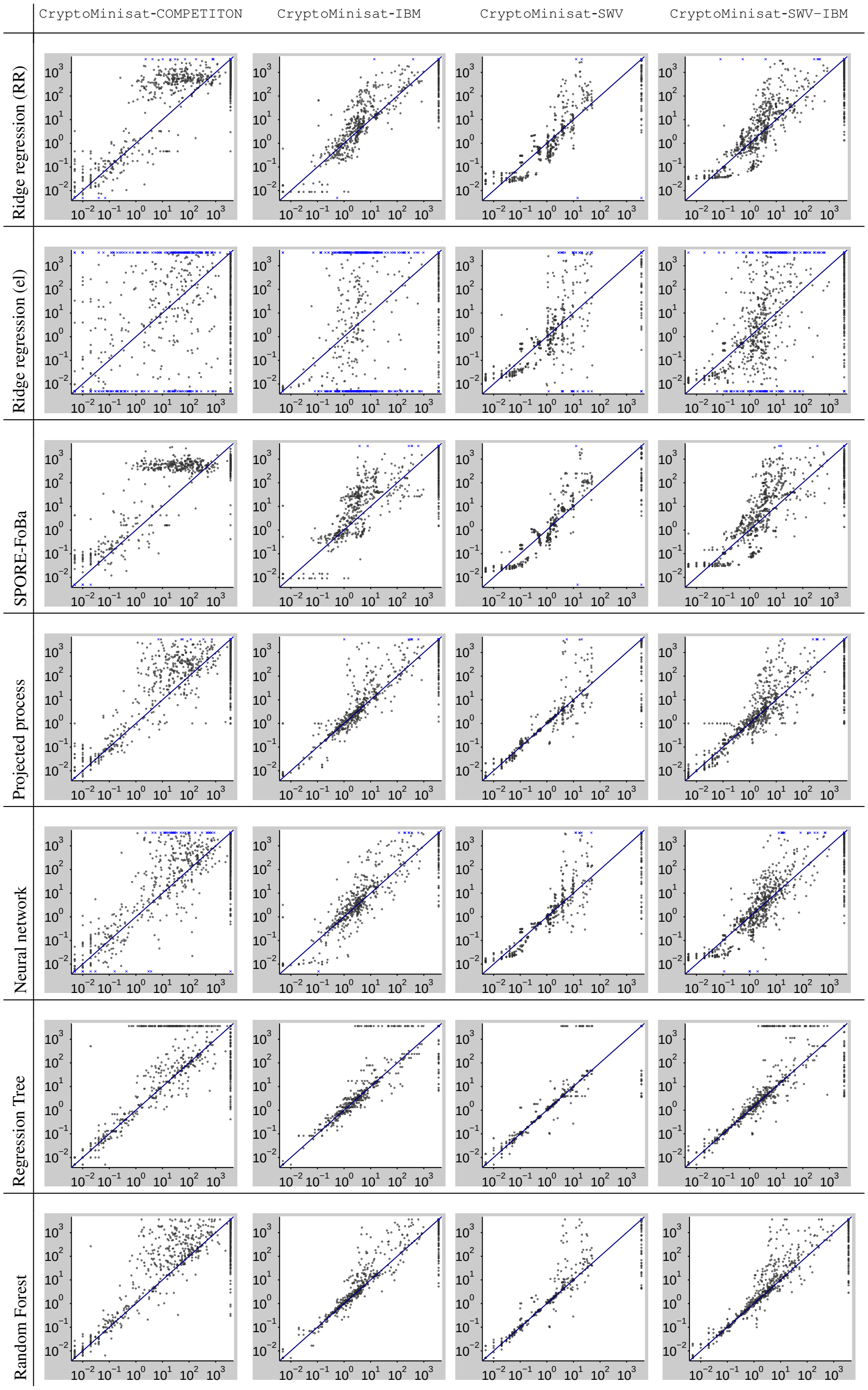


Figure E.4: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

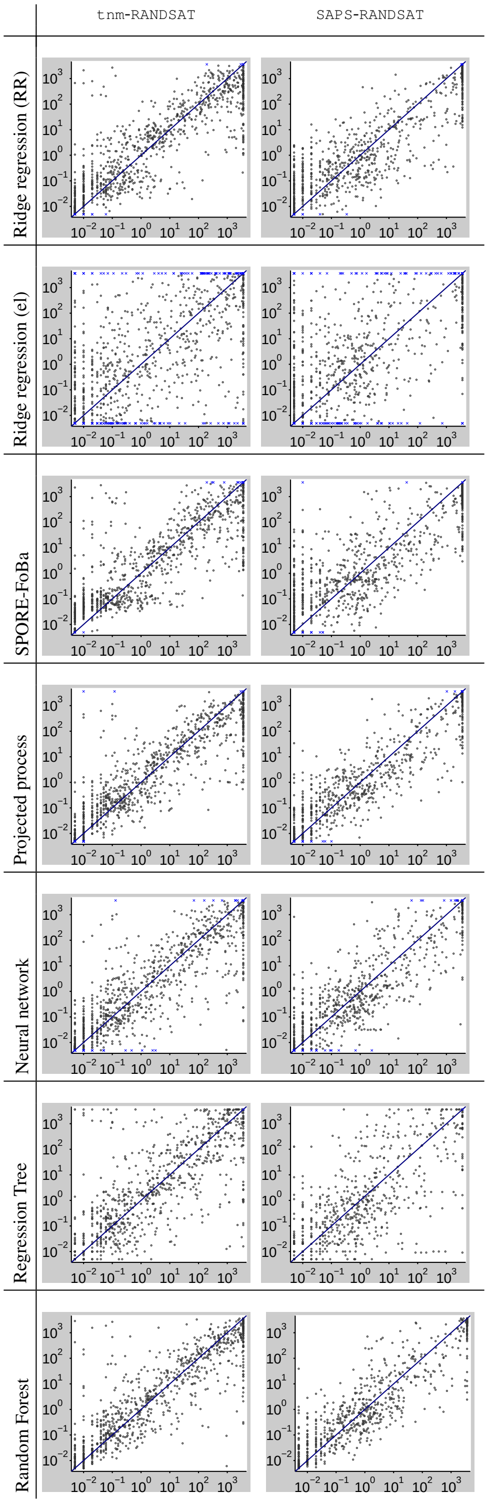


Figure E.5: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

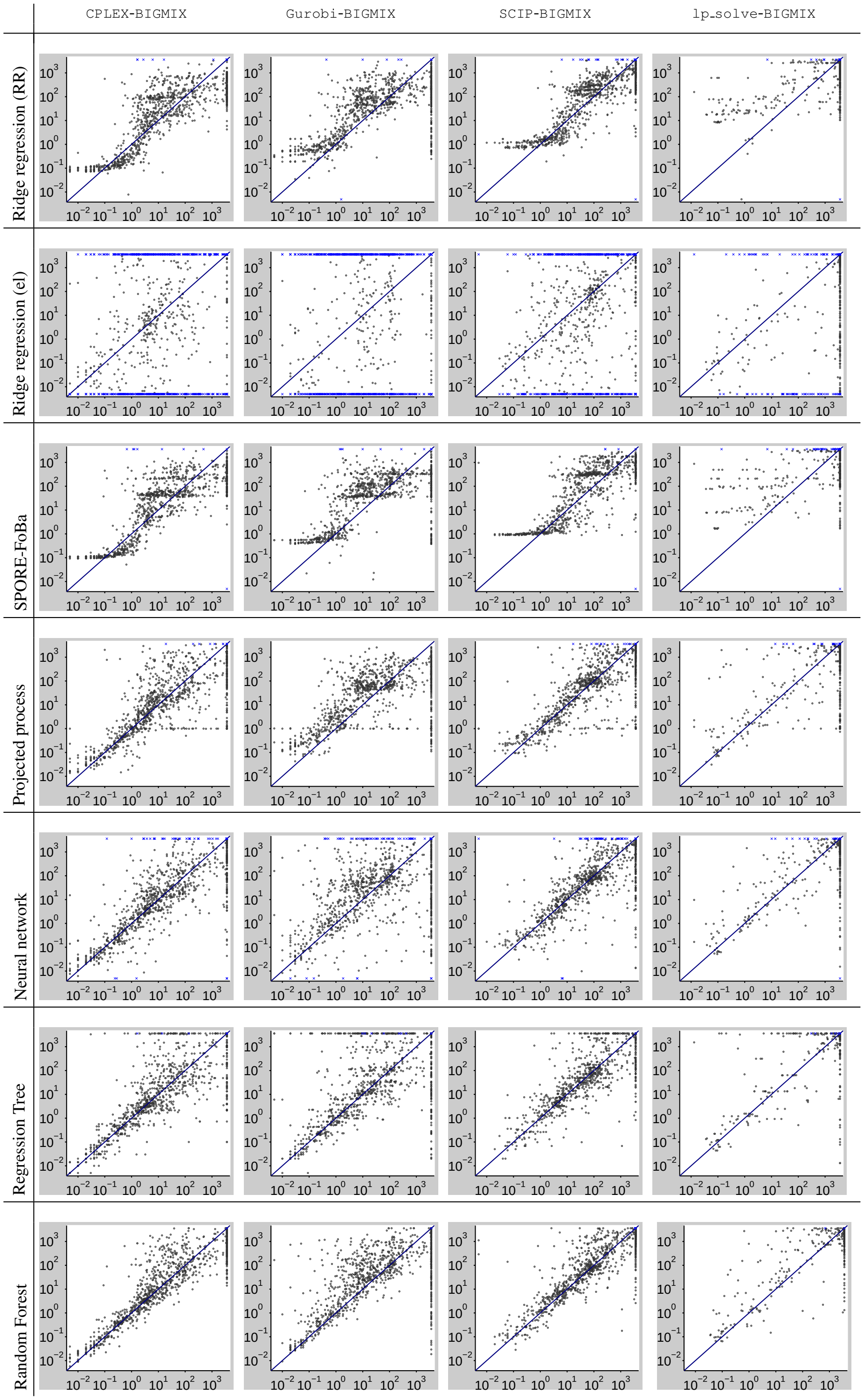


Figure E.6: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

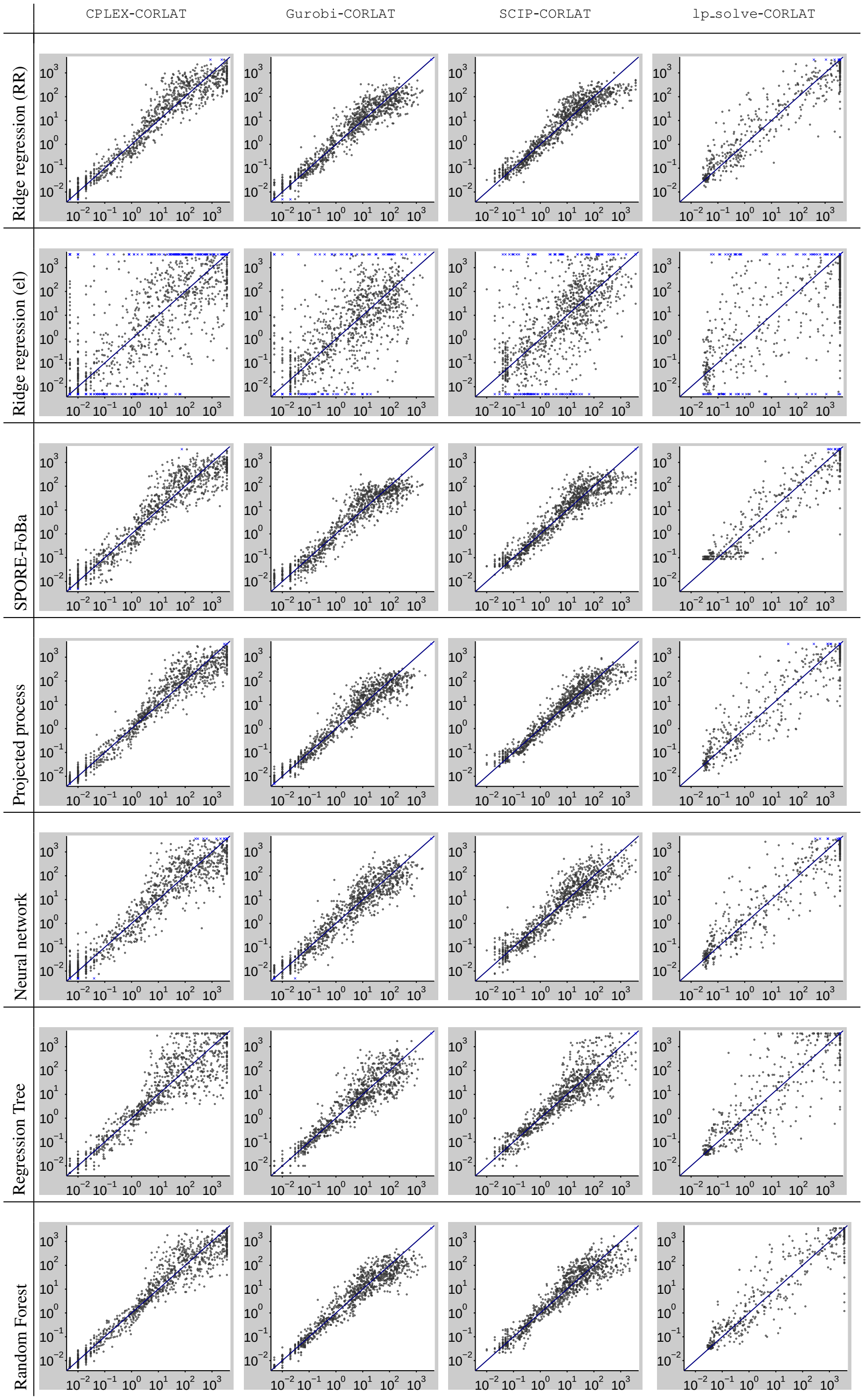


Figure E.7: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

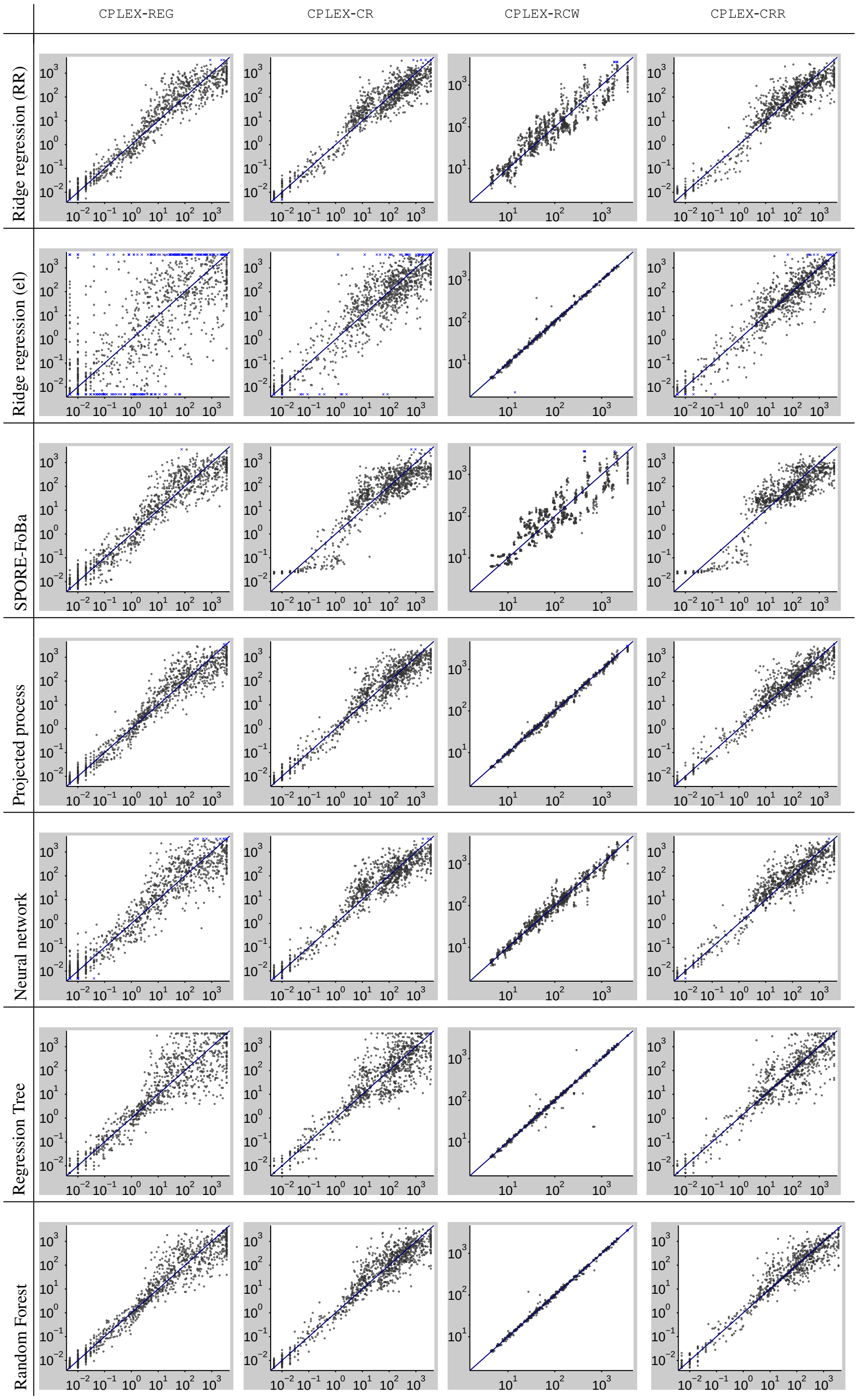


Figure E.8: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

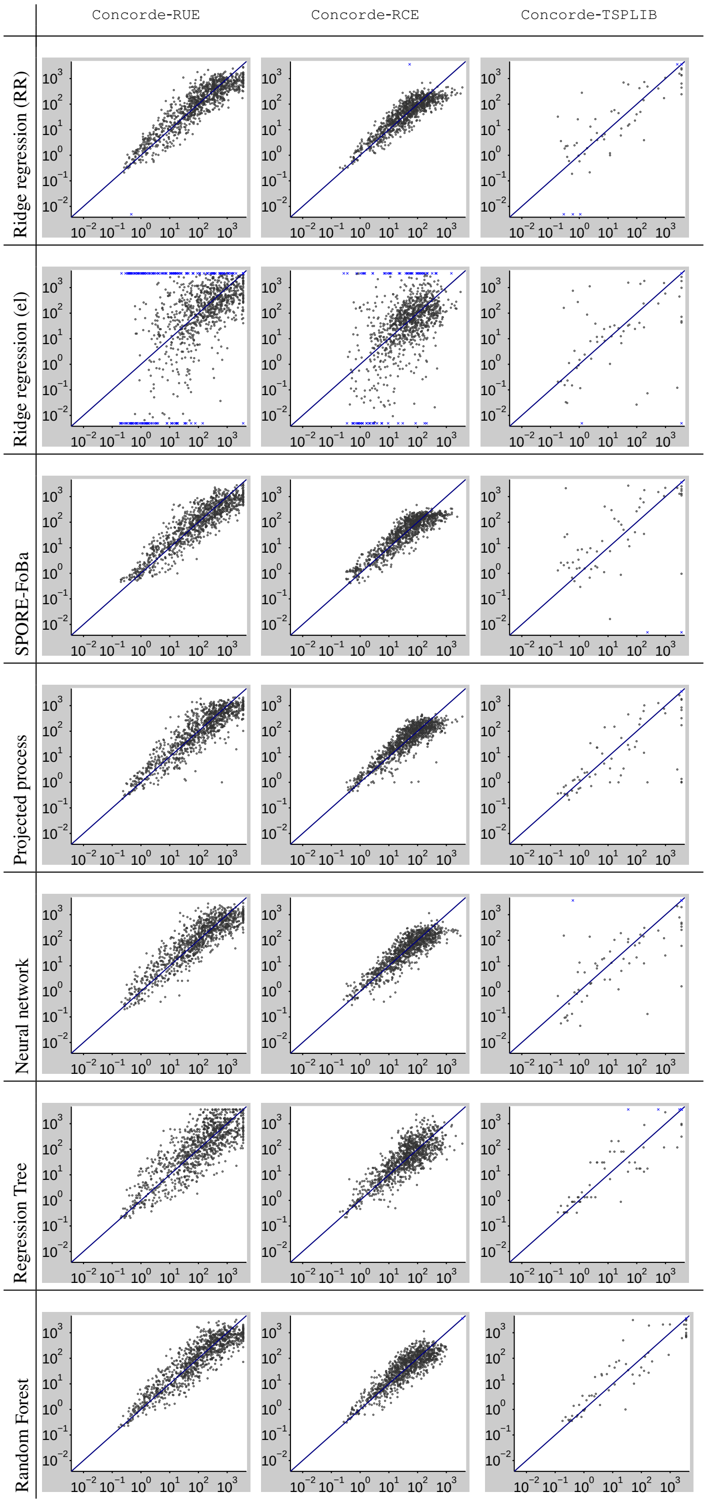


Figure E.9: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance.

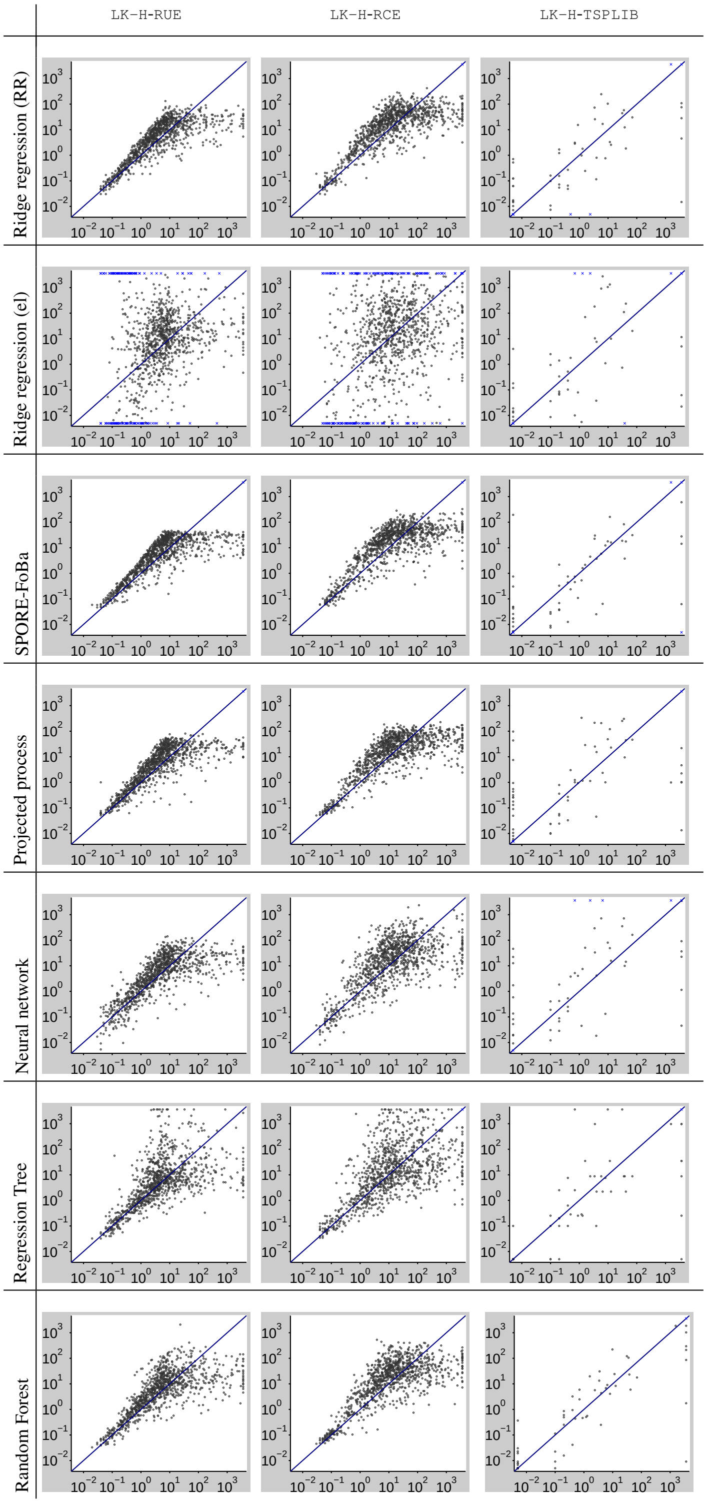


Figure E.10: Visual comparison of models for runtime predictions on unseen instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above 3 000 or below 0.001 are denoted by a blue cross rather than a black dot.

	RMSE								Time to learn model (s)							
	RR		SP		NN		RF		RR		SP		NN		RF	
Domain	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}
Minisat 2.0-COMPETITON	1.01	**0.93**	**1.25**	**1.12**	**0.62**	**0.61**	**0.47**	**0.47**	**6.8**	478	**28**	3.9E4	**22**	6717	**22**	631
Minisat 2.0-HAND	**1.05**	**0.97**	**1.34**	**1.18**	**0.63**	**0.62**	**0.51**	**0.5**	**3.7**	304	**7.9**	1.3E4	**6.2**	1857	**5.5**	154
Minisat 2.0-RAND	0.64	**0.56**	0.76	**0.48**	**0.38**	**0.39**	**0.37**	**0.36**	**4.5**	391	**8.0**	2.7E4	**11**	1498	**8.6**	199
Minisat 2.0-INDU	**0.94**	**0.93**	**1.01**	**1.01**	**0.78**	**0.79**	**0.52**	**0.54**	**3.7**	234	**7.8**	7326	**5.6**	915	**4.4**	135
Minisat 2.0-SWV-IBM	0.53	**0.47**	0.76	**0.4**	**0.32**	**0.33**	**0.17**	**0.15**	**3.5**	273	**6.4**	1.1E4	**4.7**	598	**2.7**	96
Minisat 2.0-IBM	**0.51**	**0.47**	0.71	**0.47**	**0.29**	**0.32**	**0.19**	**0.19**	**3.2**	218	**5.2**	1.1E4	**2.6**	362	**1.5**	53
Minisat 2.0-SWV	**0.35**	**0.34**	**0.31**	**1.18**	**0.16**	**0.18**	**0.08**	**0.08**	**3.1**	200	**4.9**	3468	**2.1**	156	**1.1**	36
CryptoMinisat-INDU	**0.94**	**0.94**	**0.99**	**0.97**	**0.94**	**0.9**	**0.72**	**0.71**	**3.7**	237	**7.9**	5372	**5.4**	527	**4.1**	105
CryptoMinisat-SWV-IBM	**0.77**	**0.75**	0.85	**0.67**	**0.66**	**0.69**	**0.48**	**0.48**	**3.5**	266	**11**	1.3E4	**4.5**	779	**2.8**	77
CryptoMinisat-IBM	**0.65**	**1.04**	**0.96**	**0.67**	**0.55**	**0.55**	**0.41**	**0.42**	**3.2**	215	**4.9**	8902	**2.6**	711	**1.5**	47
CryptoMinisat-SWV	**0.76**	0.89	**0.78**	**0.79**	**0.71**	**0.68**	**0.51**	**0.53**	**3.1**	181	**4.6**	3718	**2.1**	156	**1.0**	35
SPEAR-INDU	**0.95**	**0.96**	**0.97**	**29.5**	**0.85**	**0.89**	**0.58**	**0.6**	**3.6**	212	**9.5**	6402	**5.4**	1069	**4.3**	139
SPEAR-SWV-IBM	**0.67**	**0.63**	0.85	**0.63**	**0.53**	**0.5**	**0.38**	**0.36**	**3.5**	267	**7.0**	1.5E4	**4.3**	338	**2.8**	81
SPEAR-IBM	**0.6**	**0.6**	0.86	**0.55**	**0.48**	**0.48**	**0.38**	**0.36**	**3.2**	244	**5.8**	9550	**2.6**	220	**1.6**	49
SPEAR-SWV	**0.49**	**0.58**	**0.58**	**0.57**	**0.48**	**0.46**	**0.34**	**0.34**	**3.1**	183	**6.2**	2618	**2.1**	114	**1.1**	44
tnm-RANDSAT	1.01	**0.96**	1.05	**0.95**	**0.94**	**0.94**	**0.88**	**0.86**	**3.8**	269	**8.6**	1.1E4	**6.6**	527	**5.4**	138
SAPS-RANDSAT	0.94	**0.86**	1.09	**0.81**	**0.73**	**0.71**	**0.66**	**0.65**	**3.8**	307	**8.5**	1.6E4	**6.6**	370	**5.0**	136
CPLEX-BIGMIX	**3E8**	**0.91**	**0.93**	**0.93**	1.02	**0.91**	**0.64**	**0.64**	**3.4**	140	**8.3**	1257	**4.8**	213	**3.5**	111
Gurobi-BIGMIX	**1.51**	**1.21**	**1.23**	**1.22**	1.41	**1.23**	**1.17**	**1.15**	**3.4**	130	**5.1**	1127	**4.6**	210	**3.7**	89
SCIP-BIGMIX	**5E6**	**0.82**	0.88	**0.81**	0.86	**0.74**	**0.57**	**0.57**	**3.4**	148	**5.4**	1722	**4.5**	204	**3.8**	99
lp_solve-BIGMIX	**1.1**	**1.74**	**0.9**	**0.88**	**0.68**	**0.6**	**0.5**	**0.47**	**3.4**	131	**4.7**	1342	**4.6**	205	**4.9**	121
CPLEX-CORLAT	**0.49**	**0.48**	0.52	**0.46**	0.53	**0.5**	**0.47**	**0.47**	**3.2**	274	**7.6**	8185	**5.5**	459	**3.4**	108
Gurobi-CORLAT	**0.38**	**0.38**	0.44	**0.37**	**0.41**	**0.4**	**0.38**	**0.37**	**3.2**	254	**5.2**	1.0E4	**5.5**	408	**3.3**	101
SCIP-CORLAT	**0.39**	**0.38**	0.41	**0.38**	**0.42**	**0.4**	**0.38**	**0.37**	**3.2**	268	**8.0**	9769	**5.5**	431	**3.5**	108
lp_solve-CORLAT	**0.44**	**0.42**	0.48	**0.4**	**0.44**	**0.44**	**0.41**	**0.42**	**3.3**	281	**5.1**	4812	**5.5**	390	**4.4**	120
CPLEX-RCW	0.25	**0.19**	0.29	**0.14**	**0.1**	**0.11**	**0.02**	**0.02**	**3.1**	286	**7.5**	8474	**5.3**	495	**2.7**	91
CPLEX-REG	**0.38**	**0.38**	**0.39**	**0.38**	0.44	**0.38**	**0.42**	**0.42**	**3.1**	157	**6.5**	5586	**5.3**	459	**3.7**	112
CPLEX-CR	**0.46**	**0.45**	0.58	**0.43**	**0.46**	**0.47**	**0.45**	**0.44**	**4.3**	330	**12**	2.0E4	**11**	706	**8.4**	245
CPLEX-CRR	0.44	**0.41**	0.54	**0.39**	**0.42**	**0.42**	**0.36**	**0.36**	**5.4**	482	**18**	4.0E4	**17**	2130	**13**	396
LK-H-RUE	**0.61**	**0.61**	0.63	**0.61**	0.64	**0.61**	0.67	**0.64**	**4.1**	171	**1.1**	3128	**13**	628	**11**	270
LK-H-RCE	**0.71**	**0.71**	0.72	**0.7**	0.75	**0.71**	**0.76**	**0.75**	**4.2**	199	**2.7**	6775	**13**	1089	**11**	269
LK-H-TSPLIB	9.55	**1.09**	**1.11**	**0.93**	**1.77**	**1.67**	1.06	**0.88**	**1.6**	50	**3.0**	406	**0.5**	57	**0.1**	5.0
Concorde-RUE	**0.41**	**0.41**	**0.43**	**0.42**	0.43	**0.41**	**0.45**	**0.44**	**4.2**	243	**3.6**	7362	**13**	574	**9.9**	283
Concorde-RCE	**0.33**	**0.33**	0.34	**0.32**	**0.34**	**0.33**	**0.35**	**0.35**	**4.2**	221	**2.3**	1.0E4	**13**	576	**10**	249
Concorde-TSPLIB	**121**	**0.95**	**0.69**	**0.57**	**0.99**	**0.71**	**0.52**	**0.52**	**1.5**	52	**2.7**	375	**0.5**	32	**0.1**	5.0

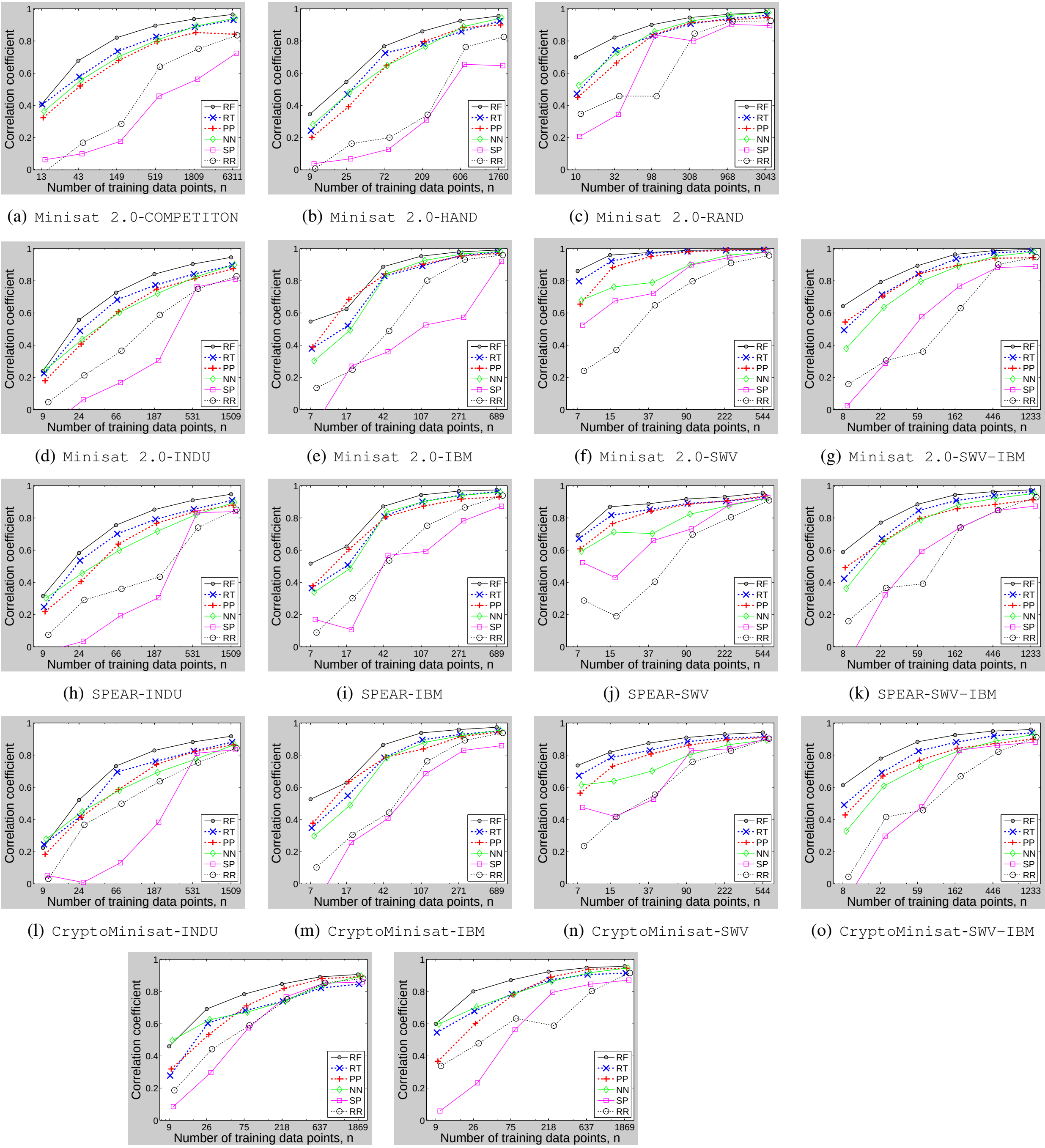
Table E.4: Quantitative evaluation of the impact of hyperparameter optimization on predictive accuracy. For each model family with hyperparameters, we report performance achieved with and without hyperparameter optimization. We compare 10-fold cross-validation performance for the default and for hyperparameters optimized using DIRECT with 2-fold cross-validation. For each dataset and model class, boldface denotes which of λ_{def} and λ_{opt} were not statistically significant from the better of the two (*e.g.*, bold-facing of 3E8 for RR and CPLEX-BIGMIX is correct since its poor mean performance stems from a single outlier).

Table 3 in the main article only provided representative results for hyperparameter optimization; here, we provide full details. Table E.4 quantifies the gains of ridge regression (RR), SPORE-Foba (SP), neural networks (NN), and random forests (RF) by hyperparameter optimization, showing small improvements of robustness across the board. Table E.5 shows that the effect of hyperparameter optimization on correlation coefficients and log likelihoods is similar.

	Spearman rank correlation coefficient								**Log likelihood**	
	RR		SP		NN		RF		RF	
Domain	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}	λ_{def}	λ_{opt}
Minisat 2.0-COMPETITON	0.69	**0.74**	0.57	**0.71**	**0.86**	**0.86**	**0.9**	**0.9**	**−0.33**	**−0.35**
Minisat 2.0-HAND	0.69	**0.74**	0.59	**0.72**	**0.87**	**0.88**	**0.91**	**0.91**	**−0.43**	**−0.41**
Minisat 2.0-RAND	**0.79**	**0.8**	0.74	**0.81**	**0.82**	**0.82**	**0.83**	0.83	**−0.18**	**−0.16**
Minisat 2.0-INDU	**0.7**	**0.71**	**0.66**	**0.68**	**0.85**	**0.83**	**0.92**	**0.92**	**−0.43**	**−0.46**
Minisat 2.0-SWV-IBM	**0.95**	**0.95**	0.89	**0.96**	**0.97**	**0.97**	**0.99**	**0.99**	**0.12**	**0.12**
Minisat 2.0-IBM	**0.94**	**0.95**	0.91	**0.96**	**0.97**	**0.97**	**0.99**	**0.99**	**0.06**	**0.07**
Minisat 2.0-SWV	**0.94**	**0.94**	**0.95**	**0.95**	**0.97**	**0.97**	**0.99**	**0.99**	**0.2**	**0.2**
CryptoMinisat-INDU	**0.66**	**0.67**	**0.59**	**0.6**	**0.72**	**0.71**	**0.81**	**0.81**	**−0.9**	**−0.86**
CryptoMinisat-SWV-IBM	**0.93**	**0.93**	0.9	**0.94**	**0.94**	**0.94**	**0.97**	**0.97**	**−0.37**	**−0.35**
CryptoMinisat-IBM	**0.93**	**0.92**	0.85	**0.92**	**0.94**	**0.94**	**0.97**	**0.96**	**−0.23**	**−0.3**
CryptoMinisat-SWV	**0.92**	**0.91**	**0.94**	**0.93**	**0.95**	**0.96**	**0.97**	**0.97**	**−0.59**	**−0.56**
SPEAR-INDU	**0.63**	**0.62**	**0.62**	**0.64**	**0.78**	**0.76**	**0.88**	**0.88**	**−0.59**	**−0.63**
SPEAR-SWV-IBM	**0.94**	**0.95**	0.91	**0.95**	**0.95**	**0.96**	**0.98**	**0.98**	**−0.22**	**−0.18**
SPEAR-IBM	**0.95**	**0.95**	0.87	**0.95**	**0.96**	**0.96**	**0.98**	**0.98**	**−0.18**	**−0.17**
SPEAR-SWV	**0.95**	**0.93**	**0.93**	**0.93**	**0.94**	**0.95**	**0.97**	**0.97**	**−0.19**	**−0.24**
tnm-RANDSAT	0.87	**0.89**	0.86	**0.89**	**0.9**	**0.89**	**0.91**	**0.91**	**−1.32**	**−1.09**
SAPS-RANDSAT	0.9	**0.91**	0.86	**0.92**	**0.93**	**0.93**	**0.95**	**0.95**	**−0.79**	**−0.74**
CPLEX-BIGMIX	**0.82**	**0.8**	**0.81**	**0.8**	**0.81**	**0.81**	**0.9**	**0.9**	**−0.7**	**−0.68**
Gurobi-BIGMIX	**0.62**	0.59	**0.62**	**0.61**	**0.57**	**0.6**	**0.64**	**0.65**	**−2.36**	**−2.09**
SCIP-BIGMIX	**0.81**	0.76	**0.76**	**0.78**	**0.81**	**0.82**	**0.89**	**0.89**	**−0.72**	**−0.63**
lp_solve-BIGMIX	**0.34**	**0.39**	0.31	**0.4**	**0.35**	**0.39**	**0.6**	0.53	**−0.24**	**−0.29**
CPLEX-CORLAT	**0.95**	**0.95**	0.95	**0.96**	0.94	**0.95**	**0.95**	**0.95**	**−0.53**	**−0.51**
Gurobi-CORLAT	**0.95**	**0.95**	0.93	**0.95**	**0.94**	**0.94**	**0.95**	**0.95**	**−0.38**	**−0.38**
SCIP-CORLAT	**0.94**	**0.94**	0.94	**0.95**	**0.93**	**0.94**	**0.94**	**0.95**	**−0.38**	**−0.37**
lp_solve-CORLAT	**0.76**	**0.76**	**0.75**	**0.76**	**0.75**	**0.77**	**0.76**	**0.77**	**−0.25**	**−0.23**
CPLEX-RCW	0.94	**0.97**	0.92	**0.98**	**0.99**	**0.99**	**1**	**1**	**0.23**	**0.23**
CPLEX-REG	**0.87**	**0.87**	**0.87**	**0.87**	0.82	**0.87**	**0.84**	**0.84**	**−0.59**	**−0.56**
CPLEX-CR	**0.9**	**0.91**	0.86	**0.91**	**0.9**	**0.9**	**0.9**	**0.91**	**−0.54**	**−0.53**
CPLEX-CRR	0.89	**0.9**	0.85	**0.91**	**0.9**	**0.9**	**0.92**	**0.92**	**−0.29**	**−0.27**
LK-H-RUE	**0.82**	**0.82**	0.81	**0.82**	0.8	**0.82**	0.77	**0.8**	-1.16	**−1**
LK-H-RCE	**0.73**	**0.73**	0.72	**0.73**	0.69	**0.73**	**0.68**	**0.68**	-1.25	**−1.13**
LK-H-TSPLIB	0.64	**0.83**	**0.8**	**0.83**	**0.55**	**0.59**	**0.75**	**0.76**	**−2**	**−1.5**
Concorde-RUE	**0.88**	**0.88**	**0.88**	0.88	**0.88**	0.88	**0.86**	**0.86**	**−0.66**	**−0.64**
Concorde-RCE	**0.86**	**0.86**	0.85	**0.86**	**0.85**	**0.86**	**0.84**	**0.84**	**−0.36**	**−0.36**
Concorde-TSPLIB	**0.73**	**0.86**	**0.86**	**0.88**	**0.72**	**0.8**	**0.91**	**0.88**	**−1.1**	**−0.95**

Table E.5: Quantitative evaluation of benefits of hyperparameter optimization. For each model class with hyperparameters to be set by optimization via cross-validation, we report performance of the default procedure and of the procedure preceded by a hyperparameter optimization stage using DIRECT with 2-fold cross-validation.

Figures E.11, E.12, and E.13 visualize scaling behaviour with the number of training instances for the SAT, MIP, and TSP domains, respectively, providing additional results complementing Figure 5 in the main article.



(a) `Minisat 2.0-COMPETITON`

(b) `Minisat 2.0-HAND`

(c) `Minisat 2.0-RAND`

(d) `Minisat 2.0-INDU`

(e) `Minisat 2.0-IBM`

(f) `Minisat 2.0-SWV`

(g) `Minisat 2.0-SWV-IBM`

(h) `SPEAR-INDU`

(i) `SPEAR-IBM`

(j) `SPEAR-SWV`

(k) `SPEAR-SWV-IBM`

(l) `CryptoMinisat-INDU`

(m) `CryptoMinisat-IBM`

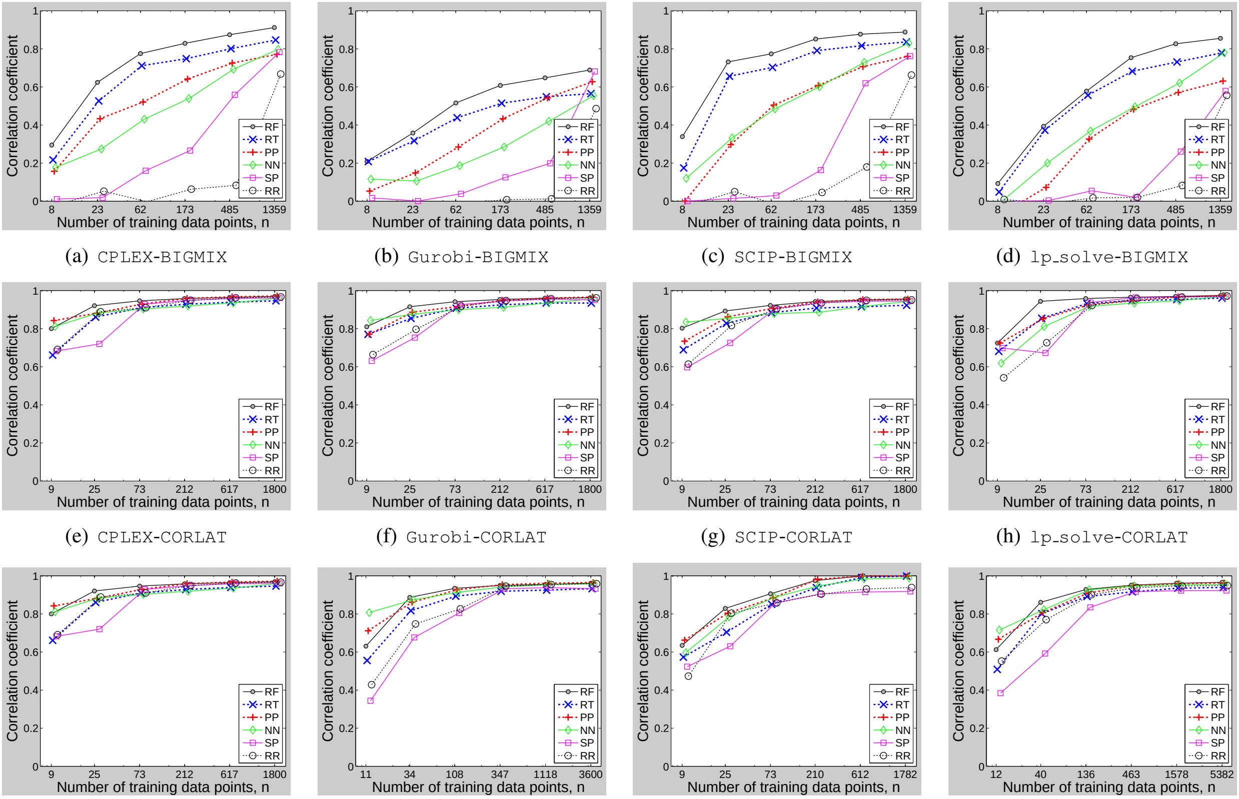
(n) `CryptoMinisat-SWV`

(o) `CryptoMinisat-SWV-IBM`

(p) `tnm-RANDSAT`

(q) `SAPS-RANDSAT`

Figure E.11: Quality of predictions in the instance space for SAT domains, as dependent on the number of training instances. For each model and number of training instances, we plot mean ± standard deviation of the correlation coefficient (CC) between true and predicted runtimes for new test instances; larger CC is better.



(a) CPLEX-BIGMIX

(b) Gurobi-BIGMIX

(c) SCIP-BIGMIX

(d) lp_solve-BIGMIX

(e) CPLEX-CORLAT

(f) Gurobi-CORLAT

(g) SCIP-CORLAT

(h) lp_solve-CORLAT

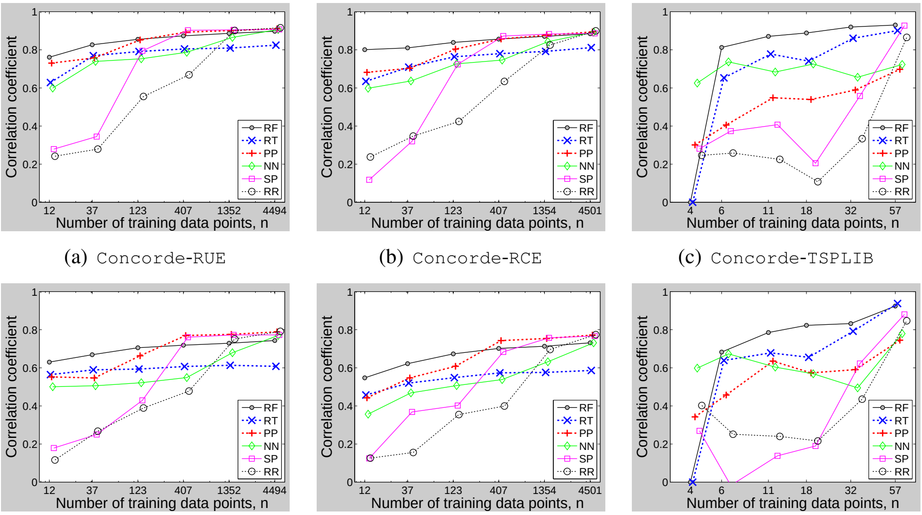
(i) CPLEX-REG

(j) CPLEX-CR

(k) CPLEX-RCW

(l) CPLEX-CRR

Figure E.12: Quality of predictions in the instance space for MIP domains, as dependent on the number of training instances. For each model and number of training instances, we plot mean ± standard deviation of the correlation coefficient (CC) between true and predicted runtimes for new test instances; larger CC is better.



(a) Concorde-RUE (b) Concorde-RCE (c) Concorde-TSPLIB

(d) LK-H-RUE (e) LK-H-RCE (f) LK-H-PORT-PORTC

Figure E.13: Quality of predictions in the instance space for TSP domains, as dependent on the number of training instances. For each model and number of training instances, we plot mean ± standard deviation of the correlation coefficient (CC) between true and predicted runtimes for new test instances; larger CC is better.

	Spearman rank correlation coefficient						Log likelihood	
	RR	SP	NN	PP	RT	RF	PP	RF
CPLEX-BIGMIX	**0.85**	0.67	0.63	**0.87**	0.74	0.84	**−0**	**−0.07**
CPLEX-CORLAT	**0.69**	0.49	0.41	**0.73**	0.55	**0.71**	**−0.82**	**−0.76**
CPLEX-REG	**0.68**	0.53	0.32	**0.7**	0.61	**0.75**	-0.57	**−0.43**
CPLEX-RCW	**0.64**	0.45	0.4	0.6	**0.6**	**0.62**	**0.1**	**0.02**
SPEAR-IBM	**0.93**	0.43	0.49	**0.93**	**0.92**	**0.92**	**−0.13**	**−0.18**
SPEAR-SWV	**0.8**	0.45	0.44	**0.82**	0.73	0.79	**−0.5**	**−0.35**

Table E.6: Quantitative comparison of models for runtime predictions on unseen configurations. We report 10-fold cross-validation performance. Higher rank correlations are better (1 is optimal); log-likelihoods are only defined for models that yield a predictive distribution (here: PP and RF); higher values are better. Boldface indicates results not statistically significantly from the best.

Appendix E.2. Additional Results for Section 7: Performance Predictions for Unseen Parameter Configurations

Table E.6 provides correlation coefficients and log likelihoods for predictions in the configuration space, complementing Table 5 in the main article.

Figures E.14 and E.15 visualize the raw runtime predictions of all models for previously untested parameter configurations, providing the full version of Figure 6 in the main article. Figure E.16 visualizes scaling behaviour with the number of training configurations, providing the full version of Figure 7 in the main article.

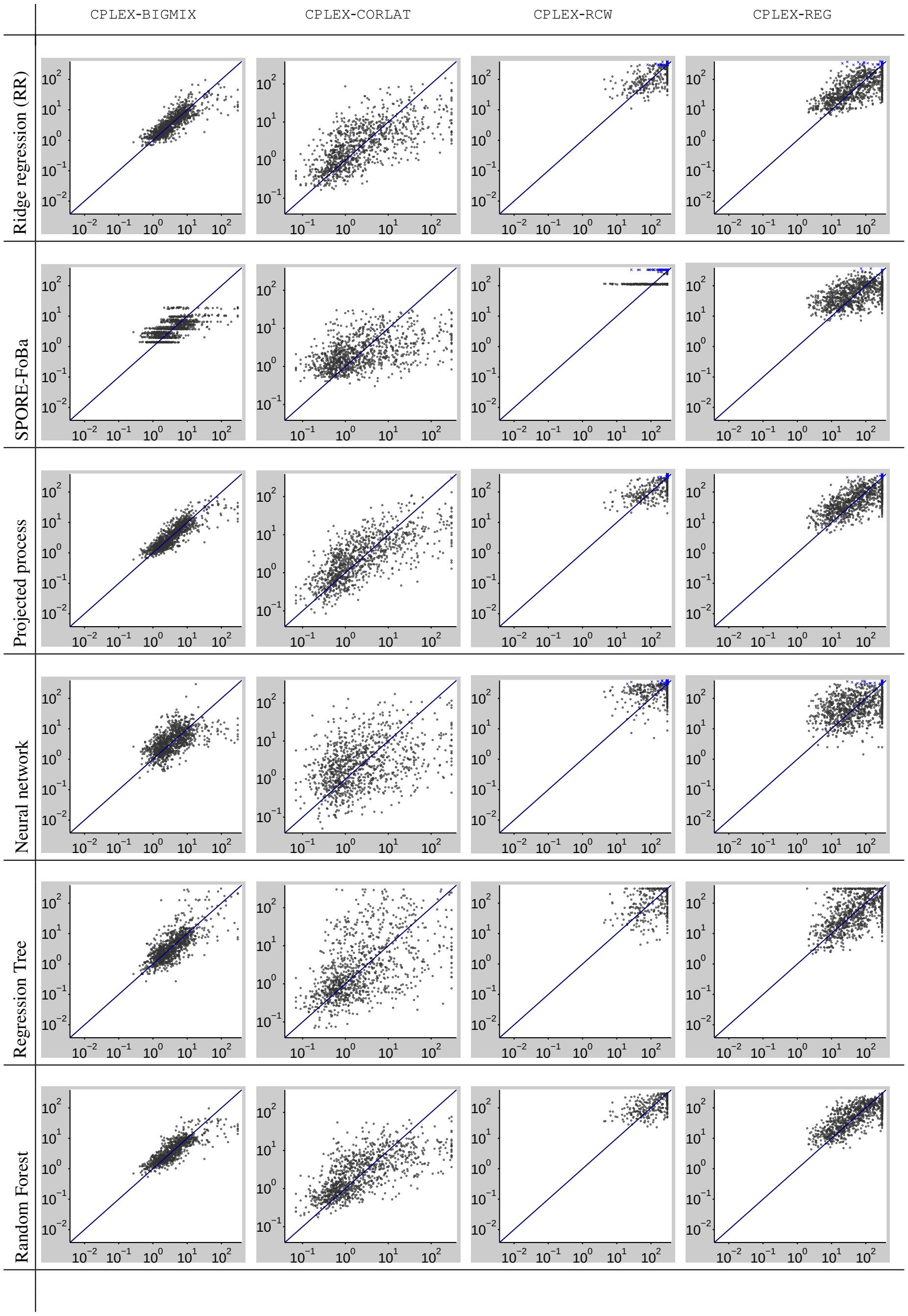


Figure E.14: Visual comparison of models for runtime predictions on previously untested parameter configurations. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one parameter configuration.

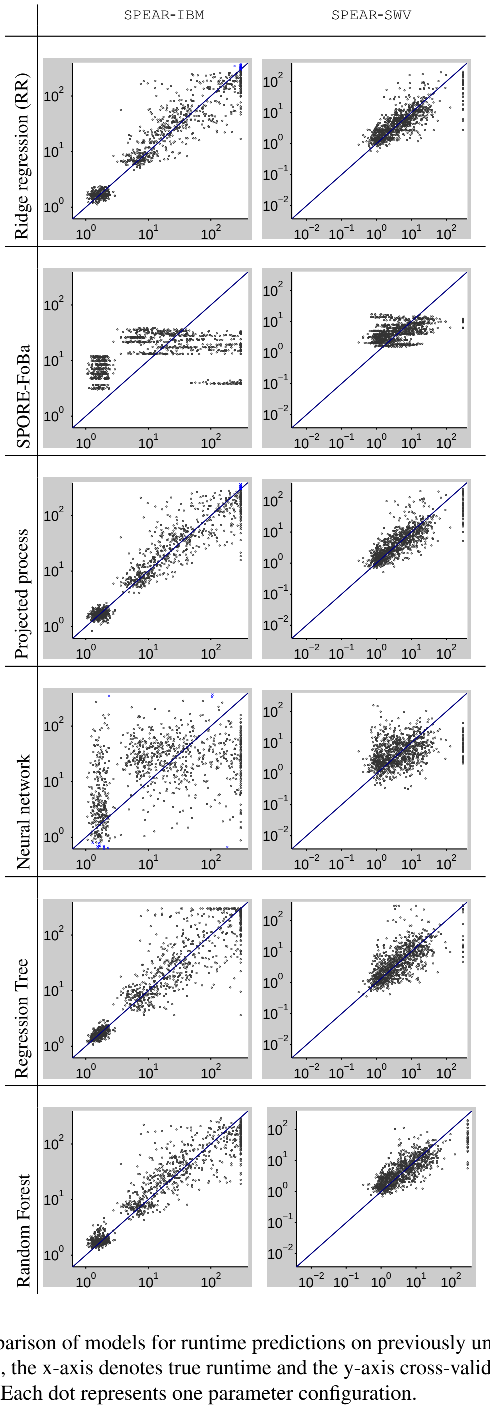
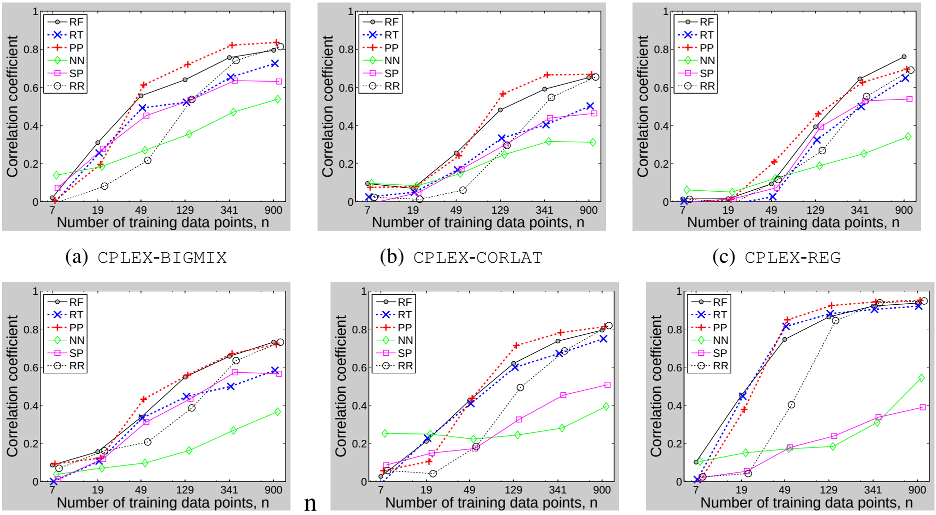


Figure E.15: Visual comparison of models for runtime predictions on previously untested parameter configurations. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one parameter configuration.



(a) `CPLEX-BIGMIX`

(b) `CPLEX-CORLAT`

(c) `CPLEX-REG`

(d) `CPLEX-RCW`

(e) `SPEAR-SWV`

(f) `SPEAR-IBM`

Figure E.16: Quality of predictions in the configuration space, as dependent on the number of training configurations. For each model and number of training instances, we plot mean ± standard deviation of the correlation coefficient (CC) between true and predicted runtimes for new test configurations.

	Spearman rank correlation coefficient						Log likelihood	
	RR	SP	NN	PP	RT	RF	PP	RF
CPLEX-BIGMIX	0.18	0.22	0.75	0.7	0.75	**0.85**	**−0.98**	-1
CPLEX-CORLAT	0.95	0.95	0.93	0.94	0.91	**0.96**	-0.73	**−0.27**
CPLEX-REG	**0.58**	0.48	0.44	0.49	0.38	0.54	0.33	**0.73**
CPLEX-RCW	0.57	0.54	0.24	0.33	0.48	**0.66**	0.75	**1.2**
CPLEX-CR	**0.95**	**0.95**	0.94	0.94	0.92	**0.95**	-0.36	**−0.02**
CPLEX-CRR	0.95	0.95	0.94	0.93	0.92	**0.96**	-0.1	**0.32**
SPEAR-IBM	0.92	0.88	0.94	0.94	0.93	**0.96**	-0.73	**0.07**
SPEAR-SWV	0.92	0.94	0.91	0.93	0.93	**0.96**	-0.72	**−0.23**
SPEAR-SWV-IBM	0.92	0.93	0.91	0.91	0.93	**0.96**	-0.88	**−0.08**

Table E.7: Quantitative comparison of models for runtime predictions on unseen instances and configurations. Models were based on 10 000 data points. Higher rank correlations are better (1 is optimal); log-likelihoods are only defined for models that yield a predictive distribution (here: PP and RF); higher values are better.

Appendix E.3. Additional Results for Section 8: Performance Predictions in the Joint Space of Instance Features and Parameter Configurations

Table E.6 provides correlation coefficients and log likelihoods for predictions in the configuration space, complementing Table 6 in the main article. Figures E.17 to E.19 show runtime predictions of all models for combinations of unseen instances and untested parameter configurations, providing the full version of Figure 8 in the main article. Figure E.20 visualizes scaling behaviour with the number of training data points, providing the full version of Figure 9 in the main article.

Figures E.21 to E.29 show true and predicted runtime matrices for our 9 scenarios, providing the full version of Figures 10 and 11 in the main article. Figure E.30 visualizes the predictive performance of random forests based on a varying number of training data points, providing the full version of Figures 12 in the main article.

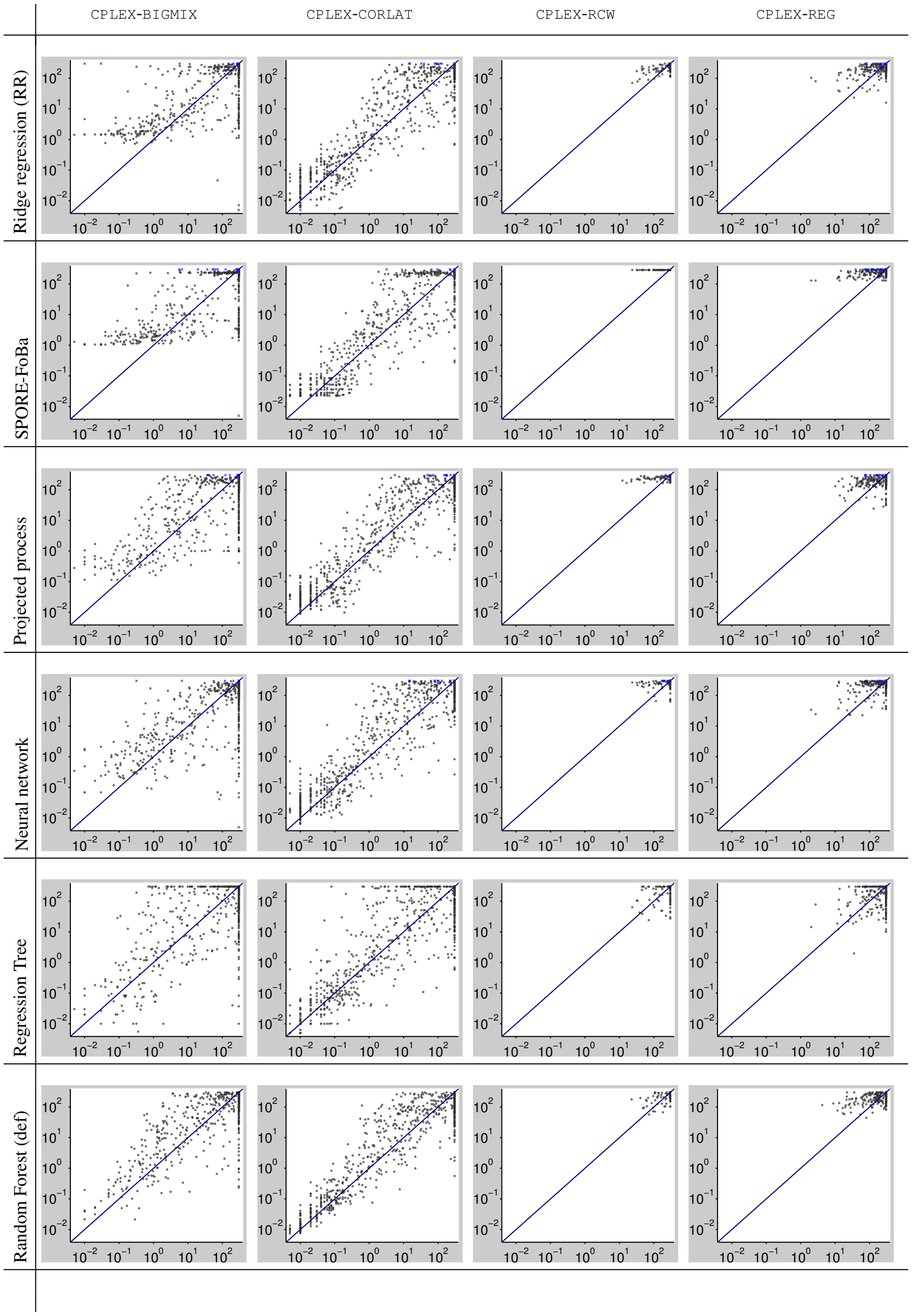


Figure E.17: Visual comparison of models for runtime predictions on pairs of previously unseen test configurations and instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one combination of an unseen instance and parameter configuration.

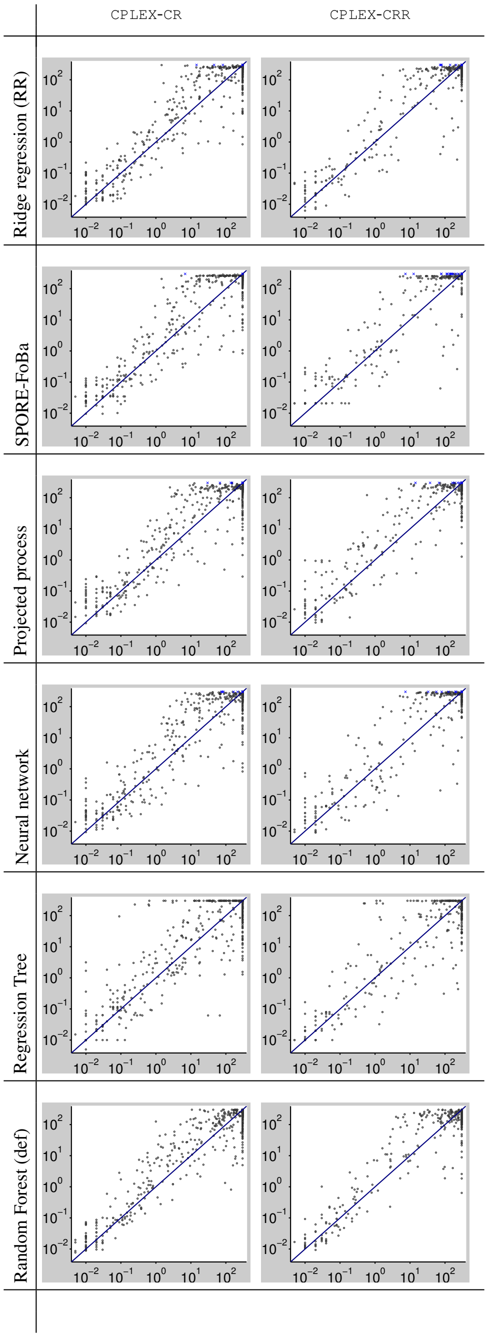


Figure E.18: Visual comparison of models for runtime predictions on pairs of previously unseen test configurations and instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one combination of an unseen instance and parameter configuration.

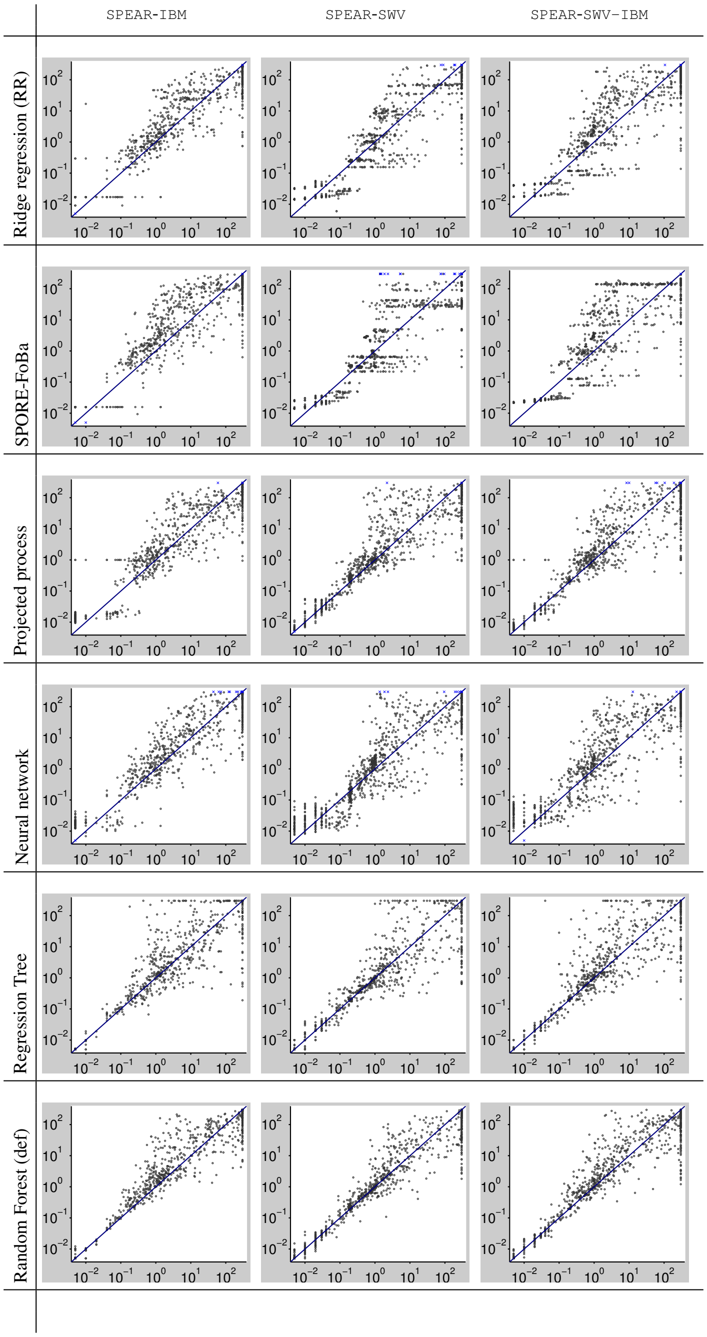
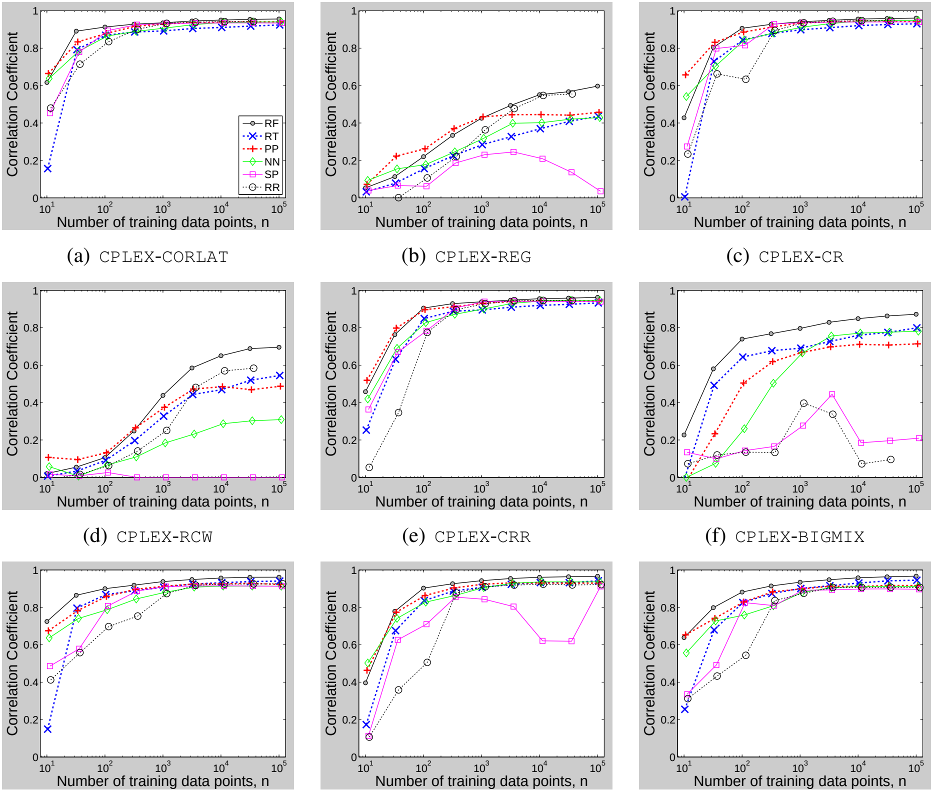


Figure E.19: Visual comparison of models for runtime predictions on pairs of previously unseen test configurations and instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one combination of an unseen instance and parameter configuration.



(a) CPLEX-CORLAT (b) CPLEX-REG (c) CPLEX-CR

(d) CPLEX-RCW (e) CPLEX-CRR (f) CPLEX-BIGMIX

(g) SPEAR-SWV (h) SPEAR-IBM (i) SPEAR-SWV-IBM

Figure E.20: Quality of predictions in the joint instance/configuration space, as dependent on the number of training data points. For each model and number of training data points, we plot the Pearson correlation coefficient (CC) between true and predicted runtimes for new test instances and configurations; larger CC is better. Note that for RCW and REG correlation coefficients are very low because most combinations of configurations and instances result in timeouts; the prediction error (in terms of RMSE) is very low.

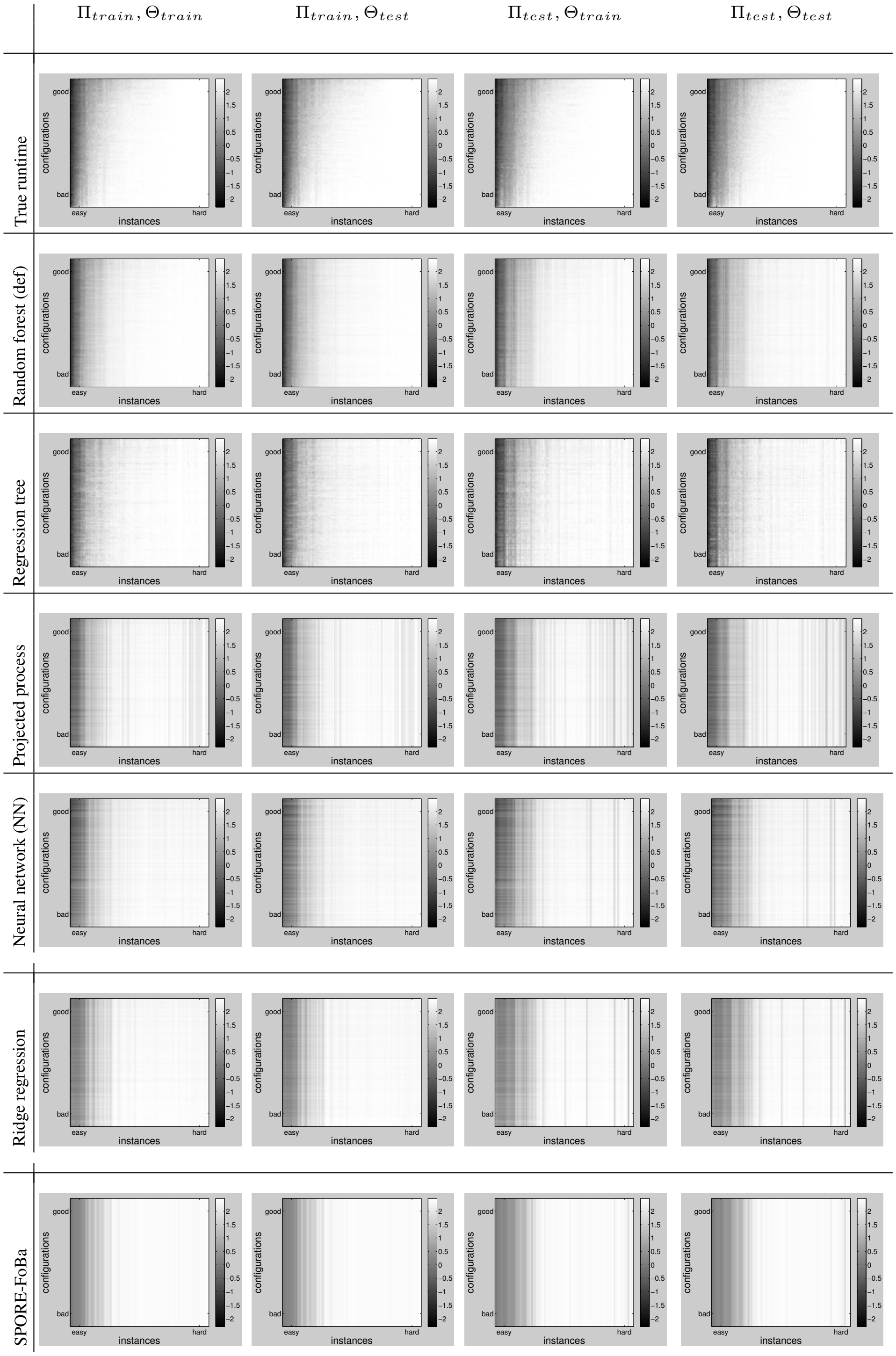


Figure E.21: True and predicted runtime matrices for scenario CPLEX-BIGMIX.

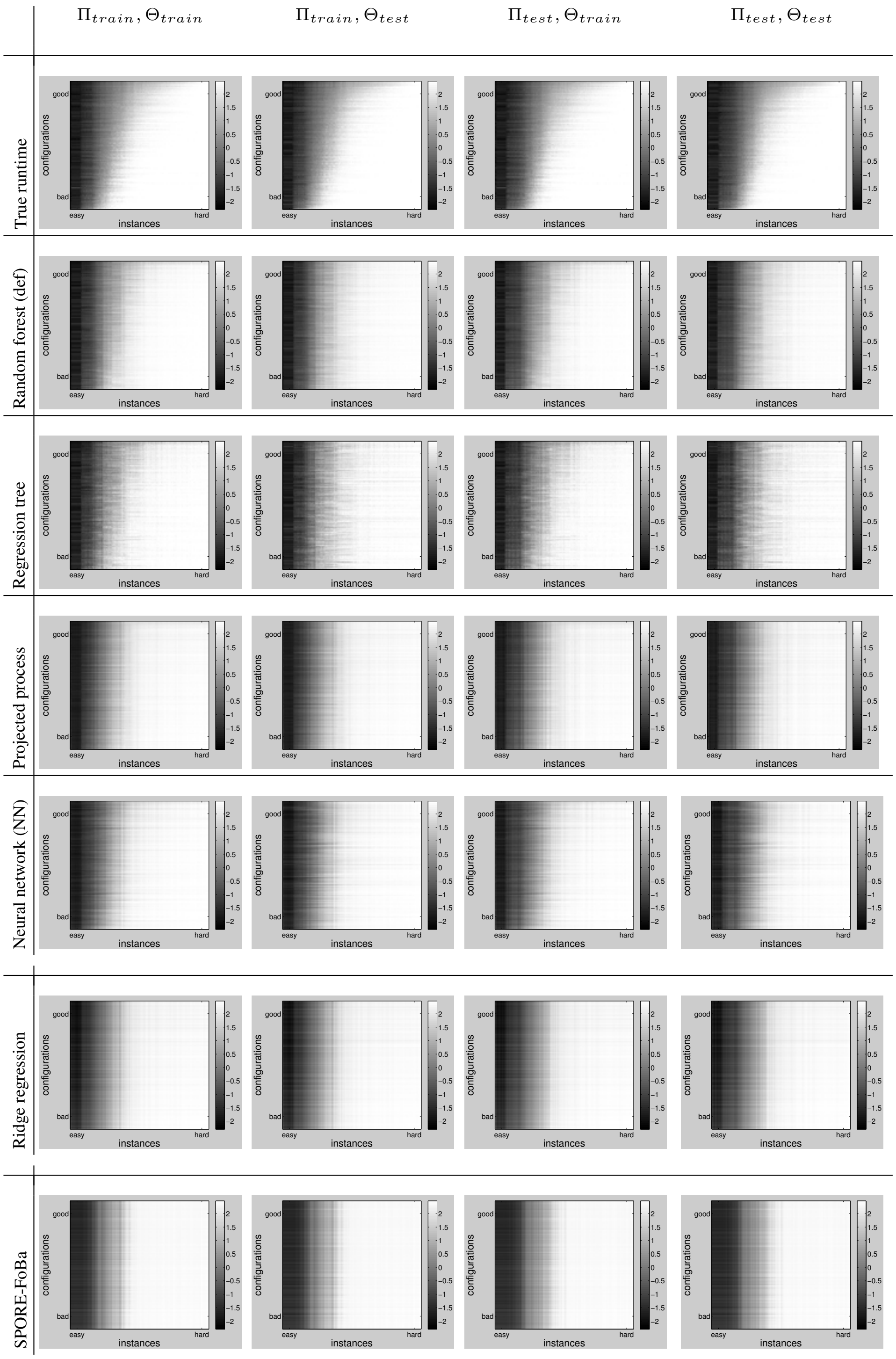


Figure E.22: True and predicted runtime matrices for scenario `CPLEX-CORLAT`.

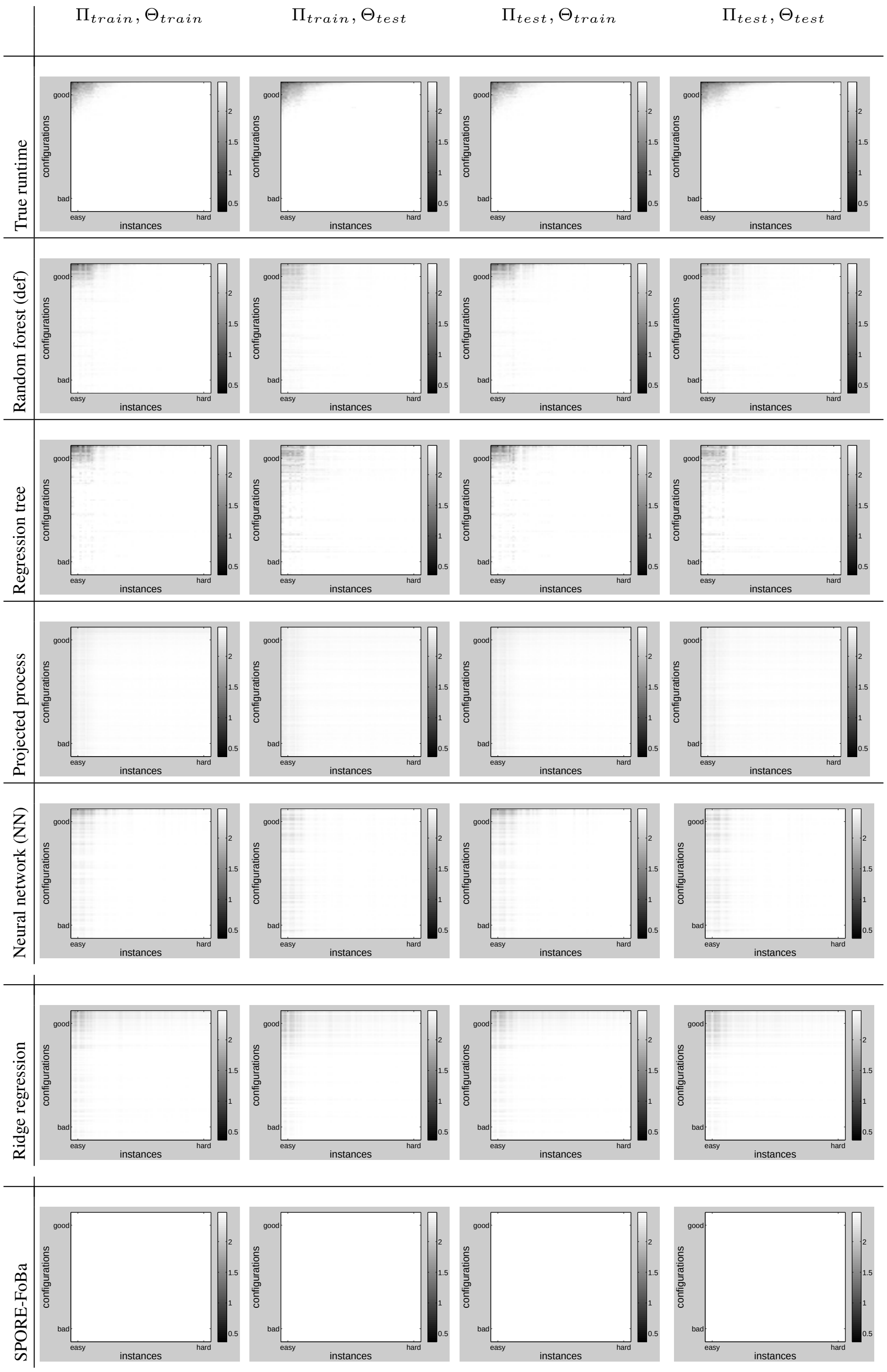


Figure E.23: True and predicted runtime matrices for scenario `CPLEX-RCW`.

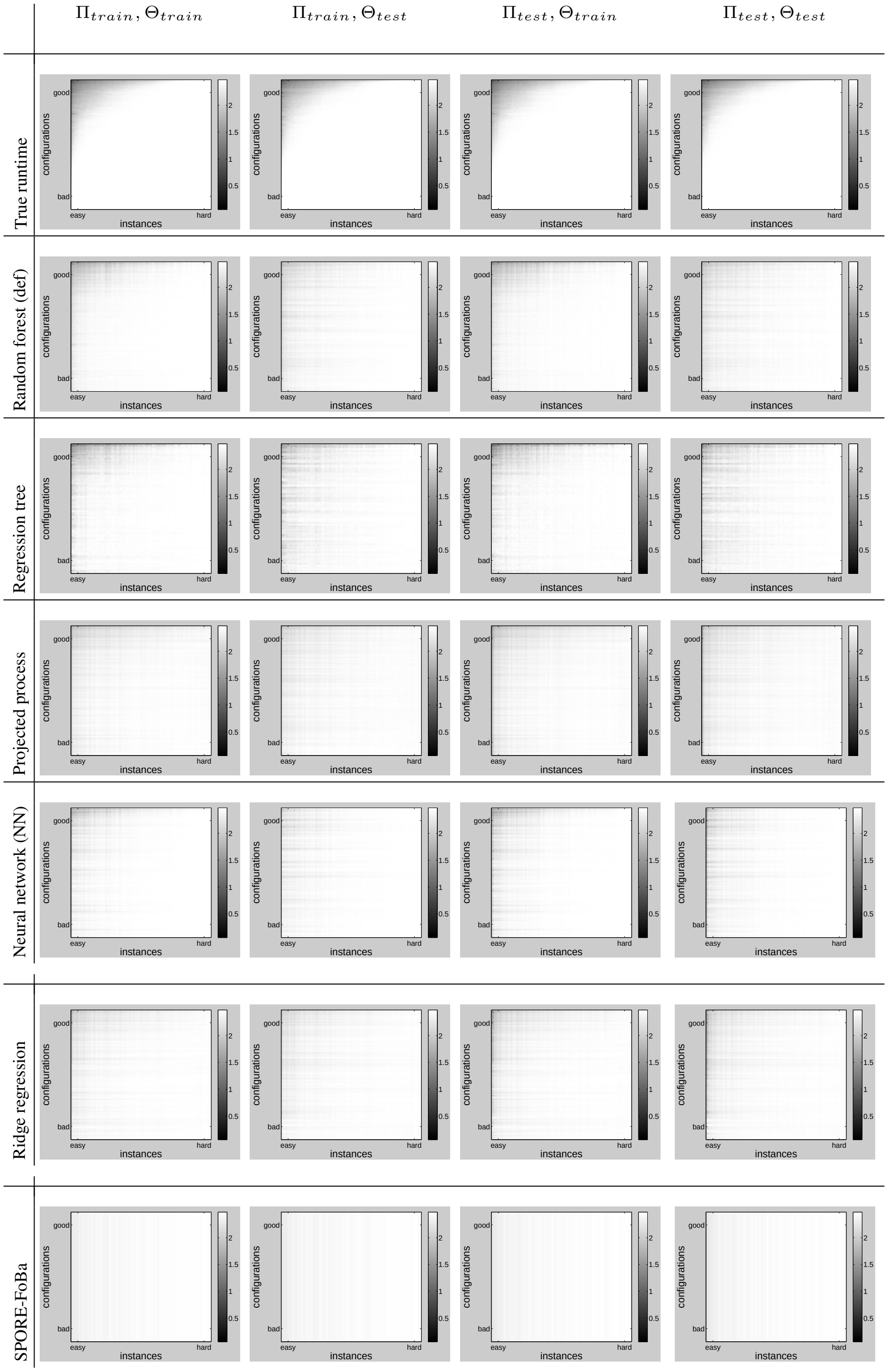


Figure E.24: True and predicted runtime matrices for scenario `CPLEX-REG`.

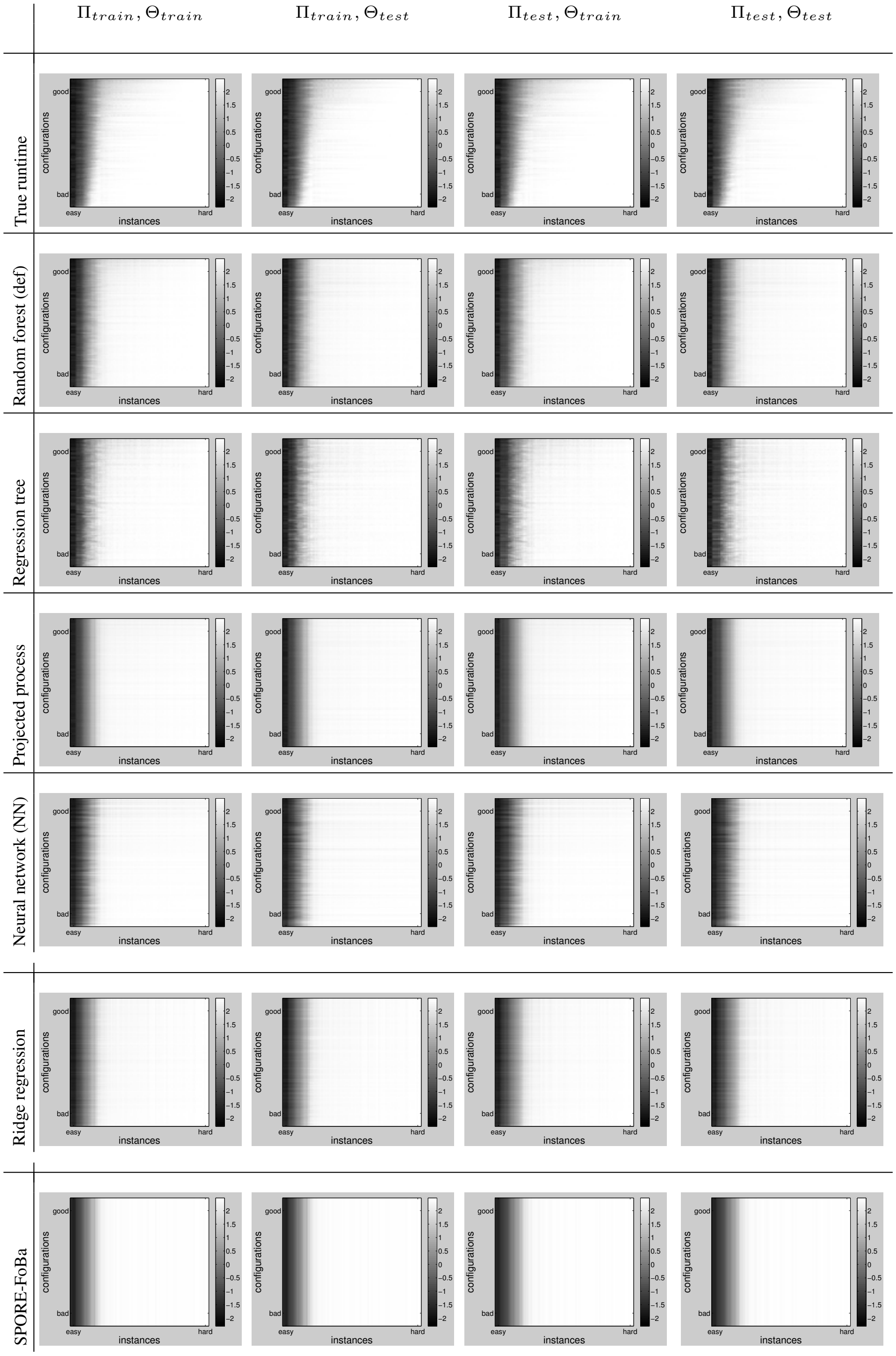


Figure E.25: True and predicted runtime matrices for scenario `CPLEX-CR`.

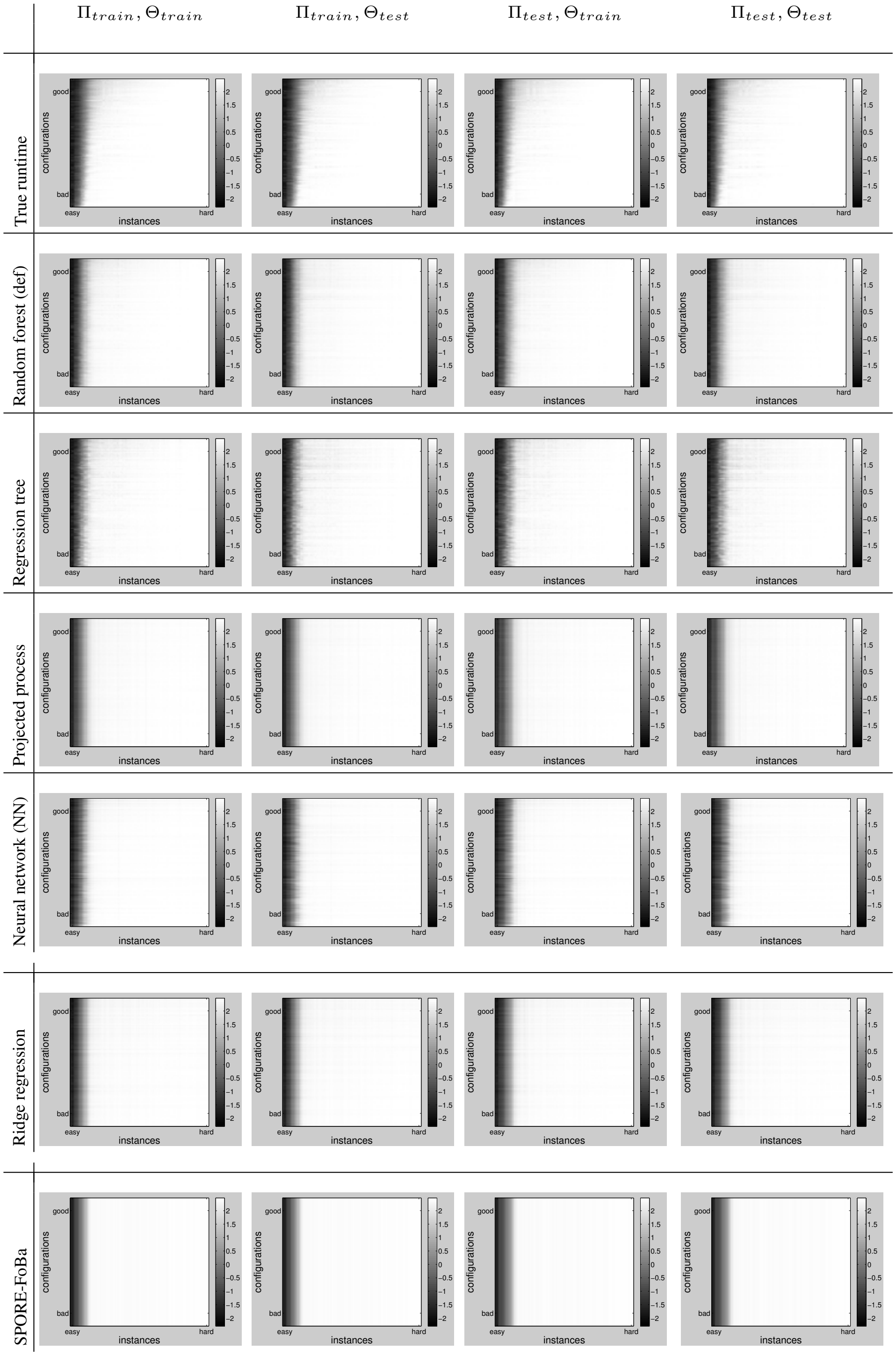


Figure E.26: True and predicted runtime matrices for scenario `CPLEX-CRR`.

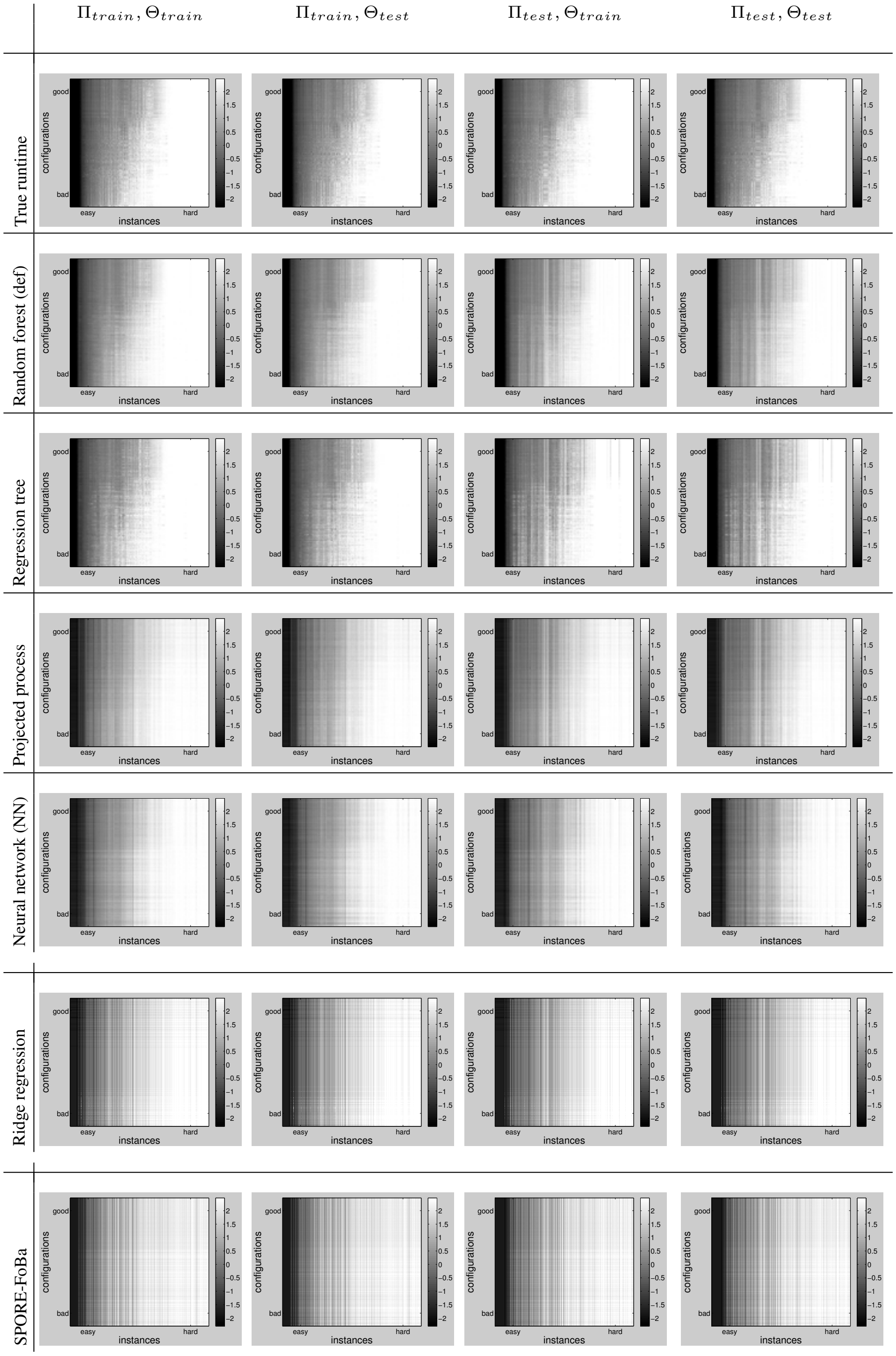


Figure E.27: True and predicted runtime matrices for scenario `SPEAR-IBM`.

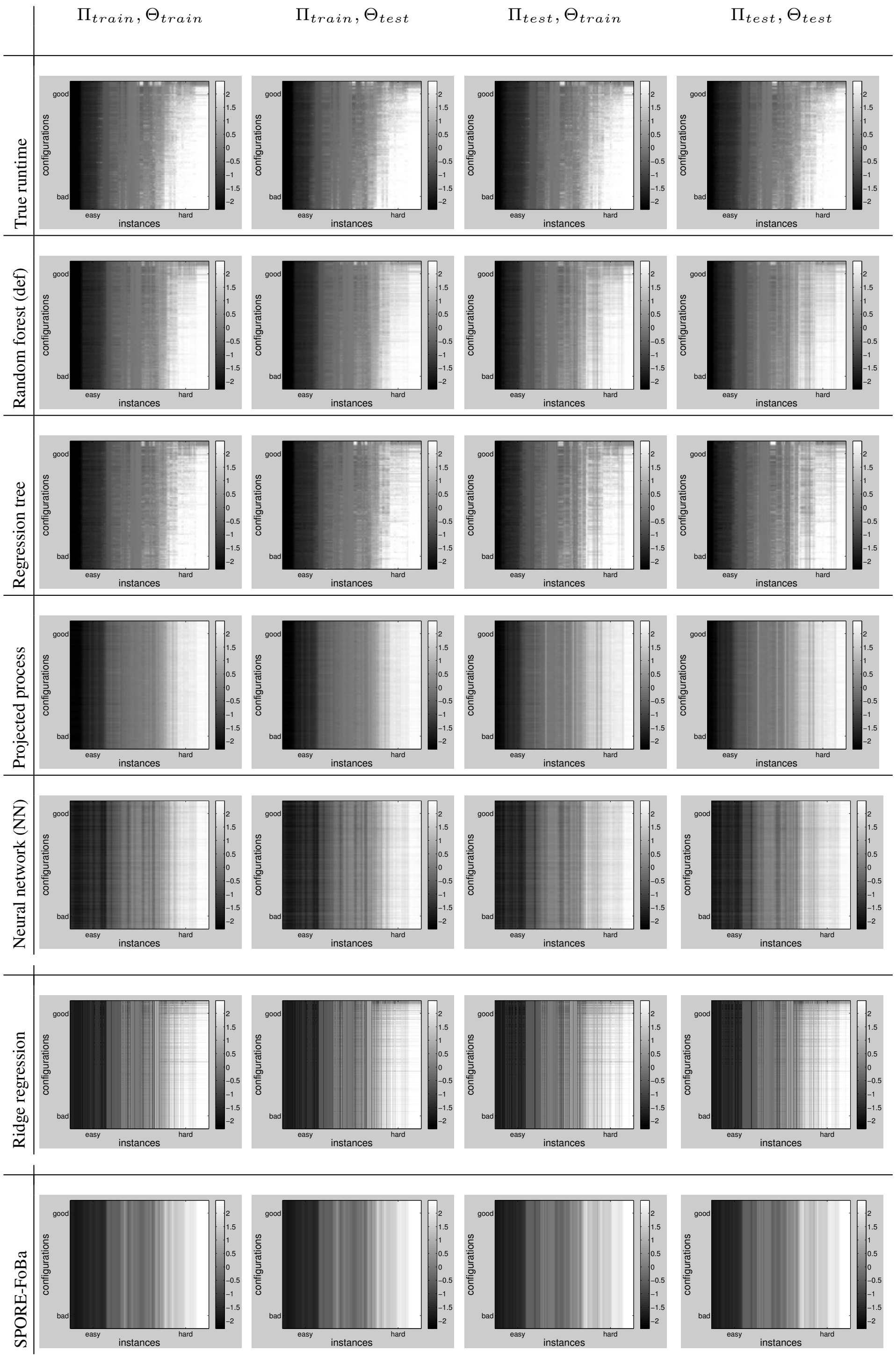


Figure E.28: True and predicted runtime matrices for scenario `SPEAR-SWV`.

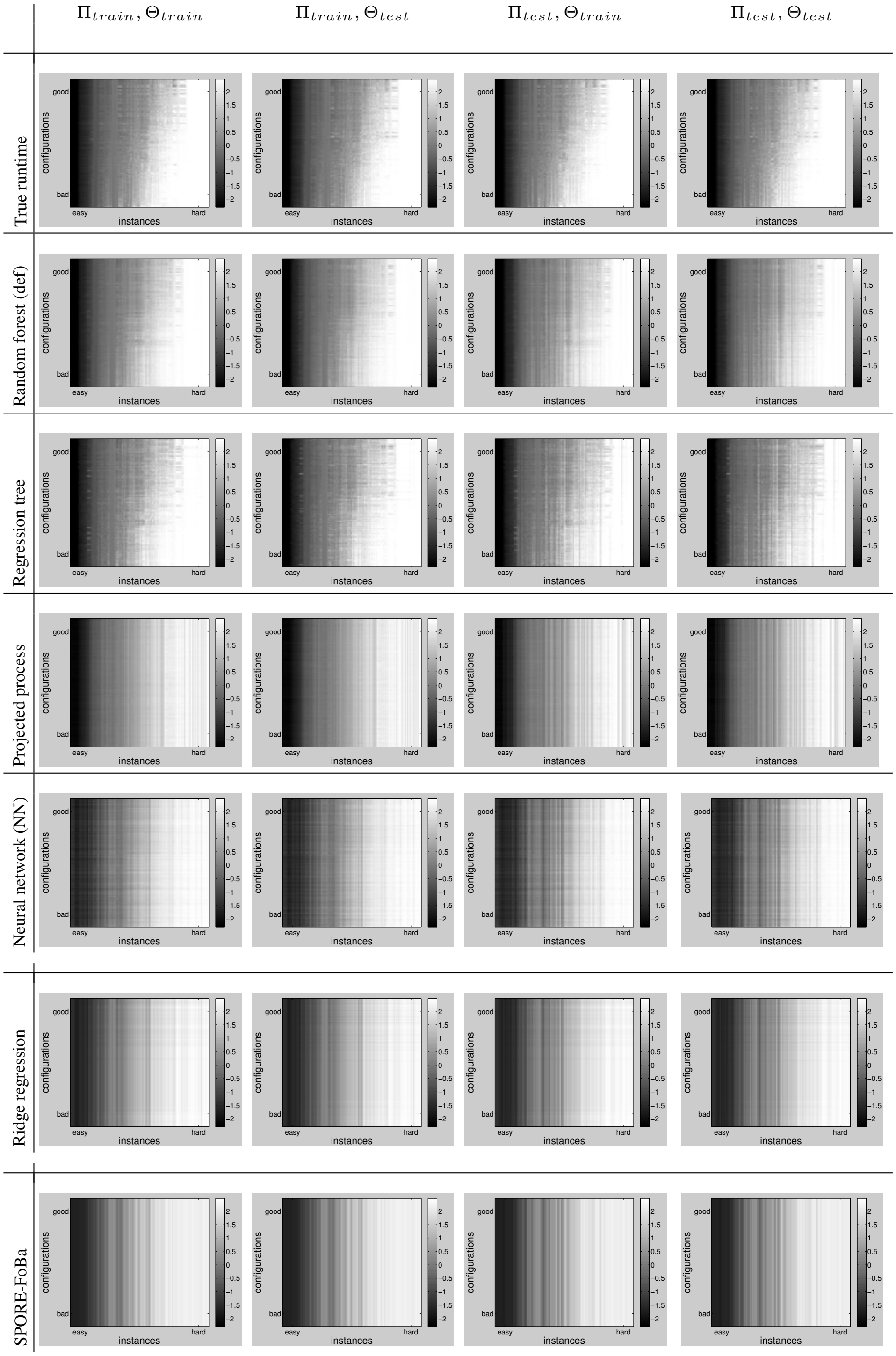


Figure E.29: True and predicted runtime matrices for scenario `SPEAR-SWV-IBM`.

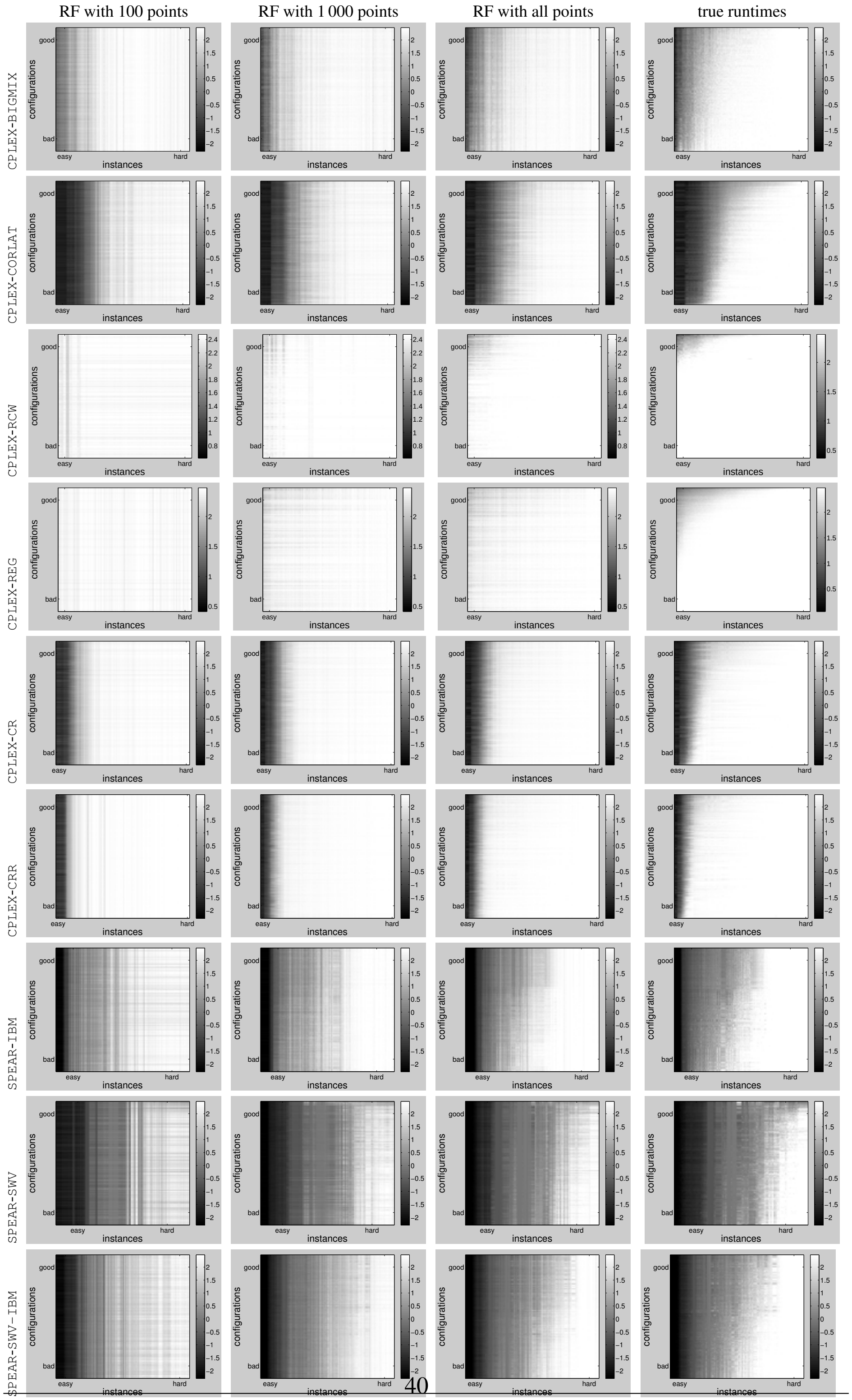


Figure E.30: Predicted runtime matrices with different number of training data points, compared to true matrix.

		RMSE				Log likelihood			
	Domain	Drop cens	Pretend uncens	S&H	Sampling S&H	Drop cens	Pretend uncens	S&H	Sampling S&H
fixed threshold of 1s	CPLEX-BIGMIX	2.6	2.2	**2**	**2**	**-57**	-230	-195	-192
	CPLEX-CORLAT	2.1	1.9	**1.8**	**1.8**	**-68**	-179	-151	-151
	CPLEX-CR	2.9	2.2	**2**	**2**	**-38**	-235	-192	-187
	CPLEX-CRR	3.1	2.3	**2.1**	**2.1**	**-42**	-254	-213	-211
	SPEAR-IBM	1.9	1.8	**1.6**	1.7	-141	-154	-132	**-131**
	SPEAR-SWV	1.4	**1.2**	**1.2**	**1.2**	**-37**	-75	-62	-61
	SPEAR-SWV-IBM	1.6	1.5	**1.4**	**1.4**	-108	-109	**-90**	-91
varying threshold	CPLEX-BIGMIX	0.95	0.90	0.67	**0.65**	**-1.9**	-8.7	-3.4	**-1.9**
	CPLEX-CORLAT	1.0	1.0	**0.72**	**0.72**	-4.3	-15	-6.3	**-4.2**
	CPLEX-REG	**0.20**	0.79	**0.20**	**0.20**	**-0.65**	-11	**-0.65**	-0.68
	CPLEX-RCW	0.34	0.65	**0.25**	**0.25**	**0.61**	-19	-1.4	0.38
	CPLEX-CR	0.64	0.76	**0.50**	**0.50**	-1.7	-6.8	-1.9	**-1.2**
	CPLEX-CRR	0.48	0.65	**0.39**	**0.39**	-0.73	-5.5	-1.1	**-0.56**
	SPEAR-IBM	0.73	0.67	0.58	**0.57**	-4.80	-11	-4.6	**-3.1**
	SPEAR-SWV	1.1	0.93	**0.82**	0.83	**-15**	-32	-19	-18
	SPEAR-SWV-IBM	0.83	0.78	0.65	**0.63**	-6.40	-15	-4.6	**-2.5**

Table E.8: Quantitative comparison of model performance with fixed censoring threshold of one second for training data (top half) and varying threshold (using the best achieved time for each instance, *i.e.*, slack factor 1; bottom half). "S&H" is the method by Schmee & Hahn [2], "Sampling S&H" is our modified version, "Drop cens" means dropping censored data, and "Pretend uncens" means treating all observations as uncensored. Boldface indicates the best average performance across methods. (For benchmarks CPLEX-REG and CPLEX-RCW, with the common fixed threshold of one second, *all* training data was censored and predictions were thus meaningless.)

Appendix E.4. Additional Results for Section 9: An Improved Mechanism to Handle Censored Runtimes in Random Forests

In the main article, we only studied the qualitative performance of the four strategies for handling censored data points, and only showed one benchmark (CPLEX-BIGMIX). Table E.8 quantifies performance for all 9 benchmarks. The top left part of this table shows that, for fixed censoring thresholds, dropping censored data yielded the worst prediction errors; treating this data as uncensored improved results; and using the Schmee & Hahn variants yielded further improvements. The top right part shows that for fixed thresholds, dropping censored values often yielded better uncertainty estimates than the other variants. This is because the Schmee & Hahn variants imputed similar values for all censored data points (close to the fixed threshold), yielding too little variation across trees, and thus overconfident predictions. In contrast, as shown in the bottom half of Table E.8, for data with a varying captime, the Schmee & Hahn variants (in particular our new variant) yielded both competitive uncertainty estimates and the lowest prediction error. Interestingly, in contrast to the fixed-threshold setting, under varying thresholds pretending censored data to be uncensored often performed worse than simply dropping censored data. Figures E.31 to E.39 visualize raw predictions of our four strategies for handling censored data for each benchmark.

Figures E.40 to E.48 illustrate how the quality of the different methods varied with how aggressively the data was capped. In brief, all results just presented were robust with respect to this level of aggressiveness; in particular, the Schmee & Hahn variants always yielded the lowest prediction error and, for instance-specific capping thresholds, typically the best uncertainty estimates.

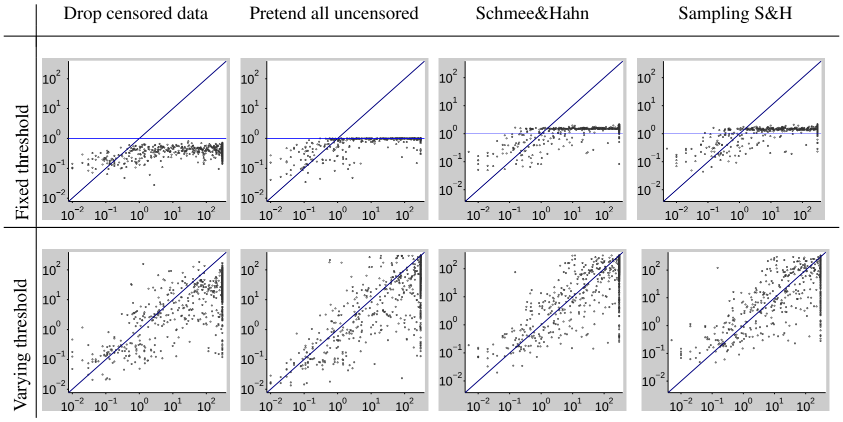


Figure E.31: True and predicted runtime of various ways of handling censored data in random forests, for scenario `CPLEX-BIGMIX` with fixed censoring threshold of one second during training (top) and varying threshold (bottom).

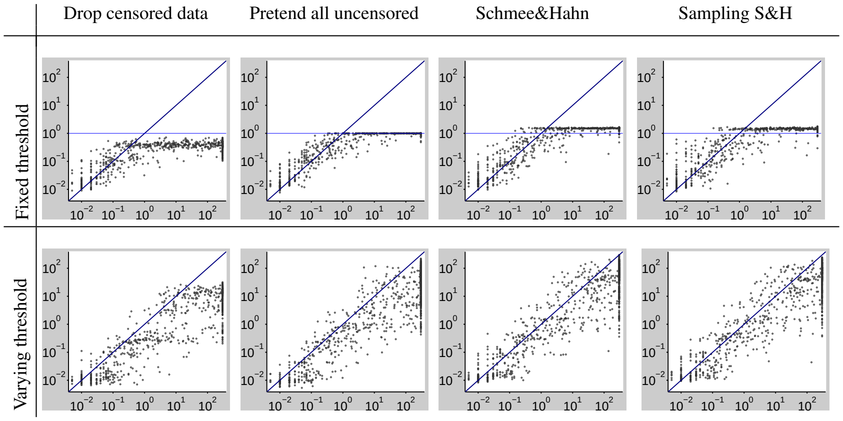


Figure E.32: True and predicted runtime of various ways of handling censored data in random forests, for scenario `CPLEX-CORLAT` with fixed censoring threshold of one second during training (top) and varying threshold (bottom).

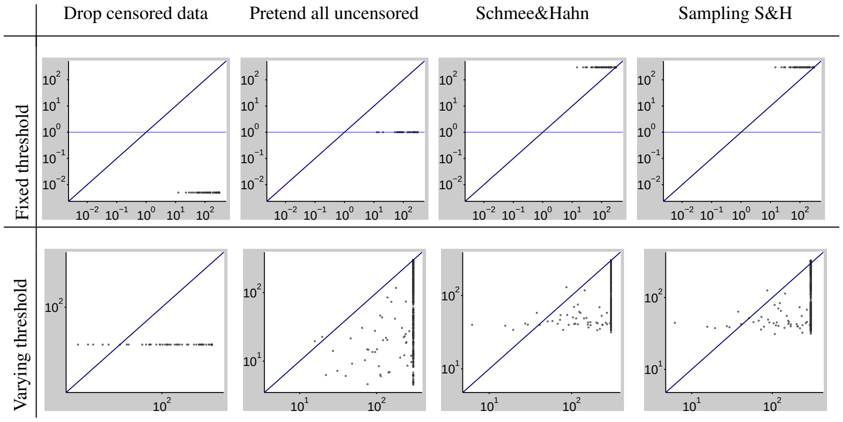


Figure E.33: True and predicted runtime of various ways of handling censored data in random forests, for scenario `CPLEX-RCW` with fixed censoring threshold of one second during training (top) and varying threshold (bottom). For the fixed threshold of one second, *all* training data was censored in this case. "Drop censored data" is not well defined in this case, and our implementation simply predicts the known minimal runtime, 0.05. Since no uncensored data points "pull down" predictions in this case, both S&H variants converge to the known maximal runtime.

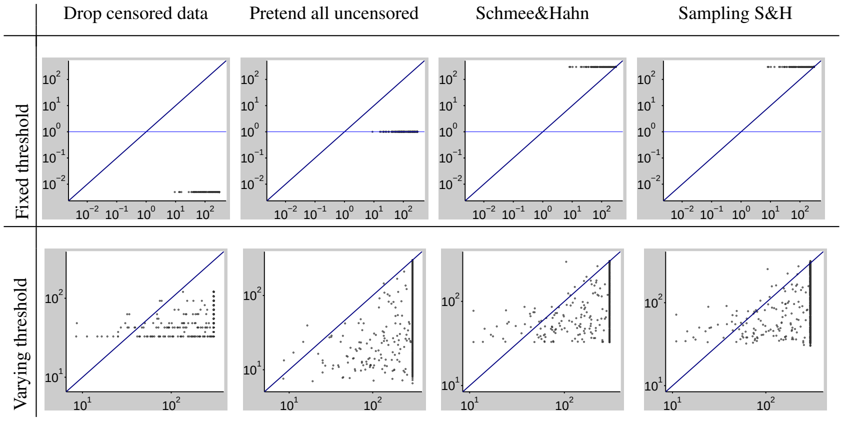


Figure E.34: True and predicted runtime of various ways of handling censored data in random forests, for scenario `CPLEX-REG` with fixed censoring threshold of one second during training (top) and varying threshold (bottom). For the fixed threshold of one second, *all* training data was censored in this case. "Drop censored data" is not well defined in this case, and our implementation simply predicts the known minimal runtime, 0.05. Since no uncensored data points "pull down" predictions in this case, both S&H variants converge to the known maximal runtime.

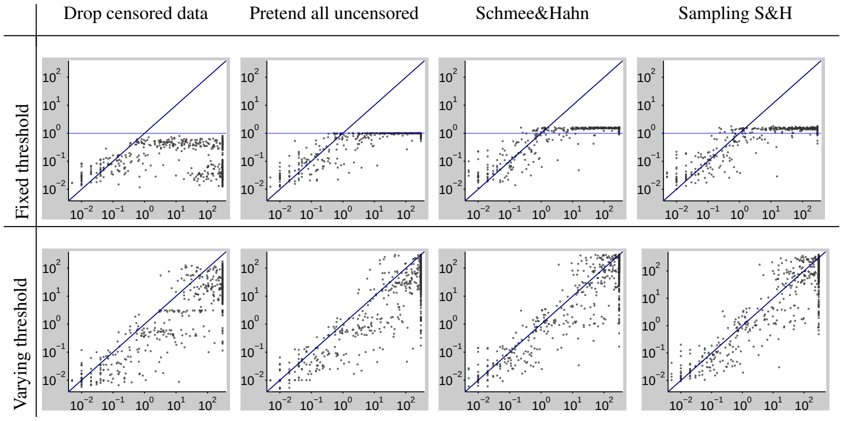


Figure E.35: True and predicted runtime of various ways of handling censored data in random forests, for scenario `CPLEX-CR` with fixed censoring threshold of one second during training (top) and varying threshold (bottom).

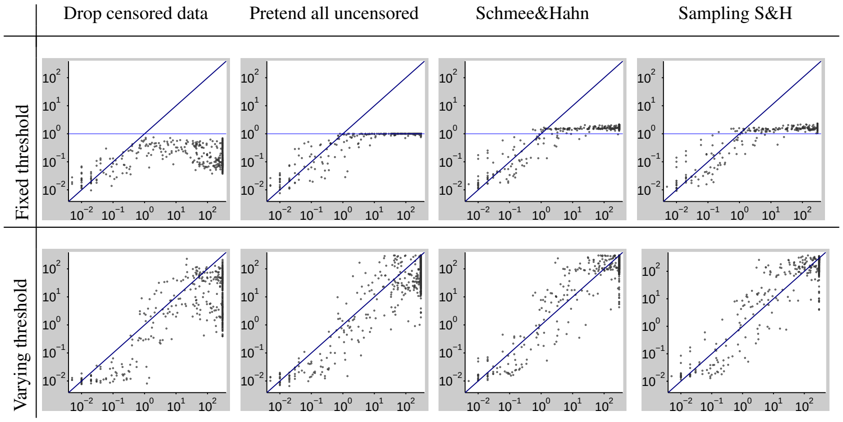


Figure E.36: True and predicted runtime of various ways of handling censored data in random forests, for scenario CPLEX-CRR with fixed censoring threshold of one second during training (top) and varying threshold (bottom).

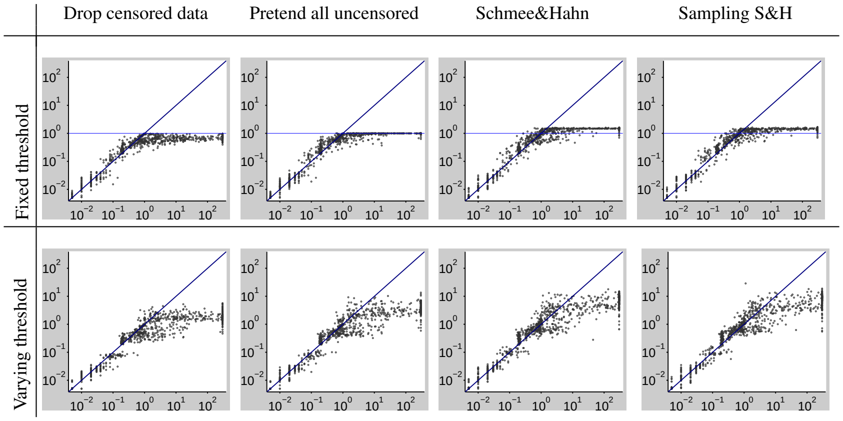


Figure E.37: True and predicted runtime of various ways of handling censored data in random forests, for scenario SPEAR-SWV with fixed censoring threshold of one second during training (top) and varying threshold (bottom).

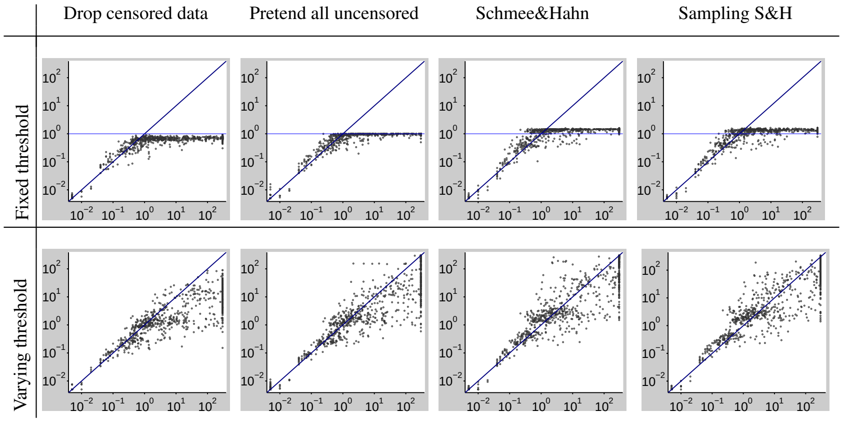


Figure E.38: True and predicted runtime of various ways of handling censored data in random forests, for scenario `SPEAR-IBM` with fixed censoring threshold of one second during training (top) and varying threshold (bottom).

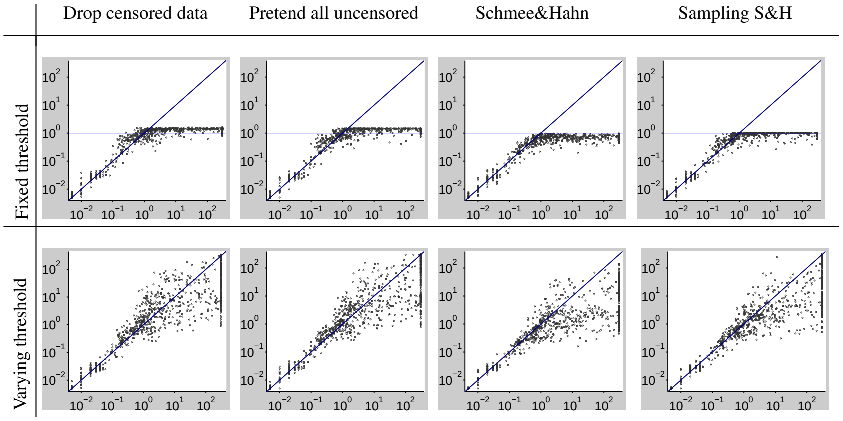


Figure E.39: True and predicted runtime of various ways of handling censored data in random forests, for scenario `SPEAR-SWV-IBM` with fixed censoring threshold of one second during training (top) and varying threshold (bottom).

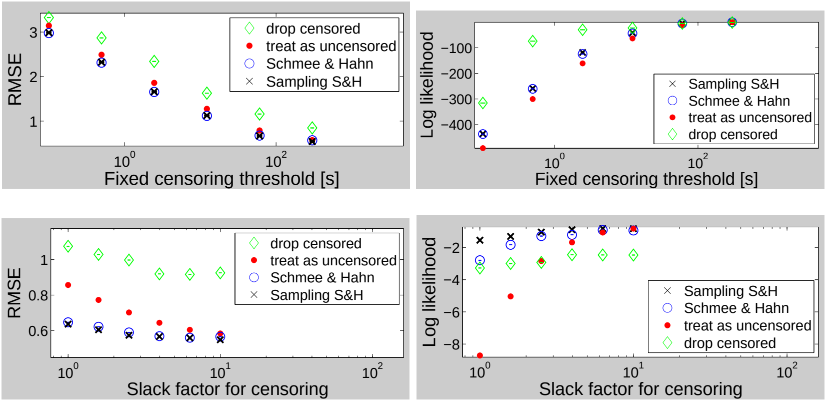


Figure E.40: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: `CPLEX-BIGMIX`.

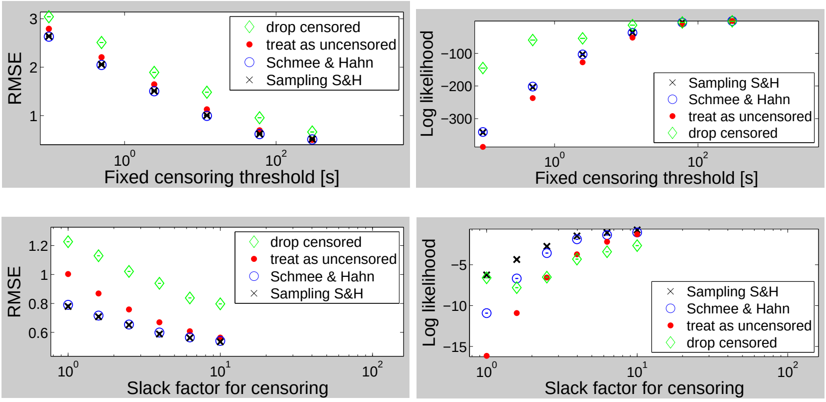


Figure E.41: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: `CPLEX-CORLAT`.

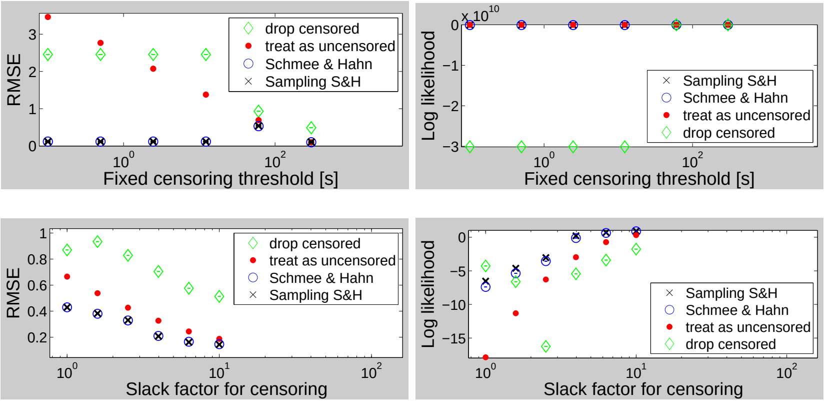


Figure E.42: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: `CPLEX-RCW`.

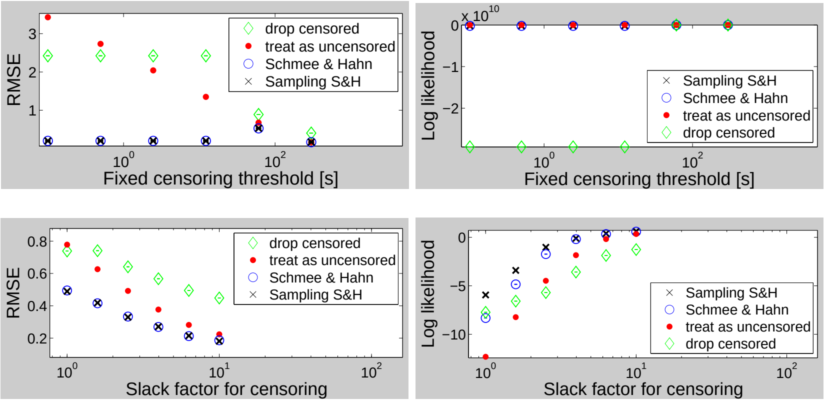


Figure E.43: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: `CPLEX-REG`.

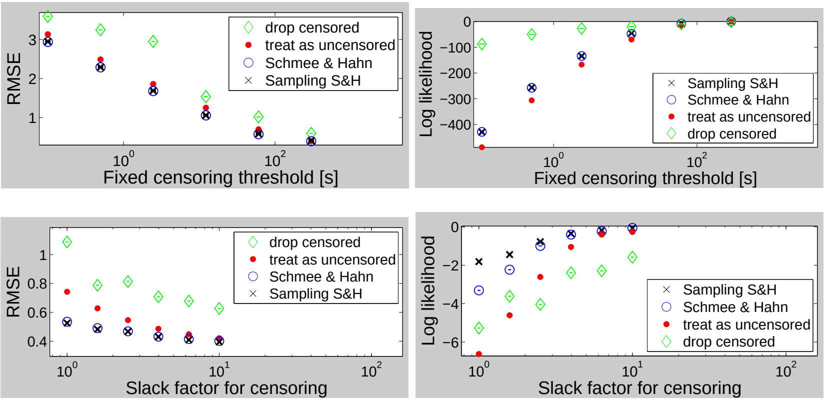


Figure E.44: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: CPLEX-CR.

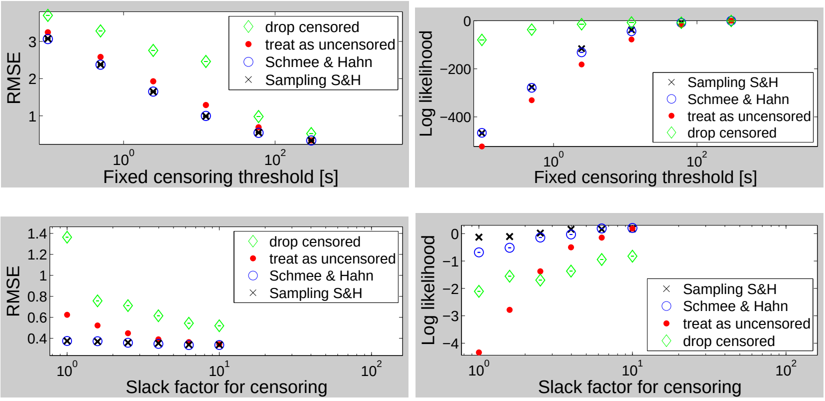


Figure E.45: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: CPLEX-CORLAT-REG-RCW.

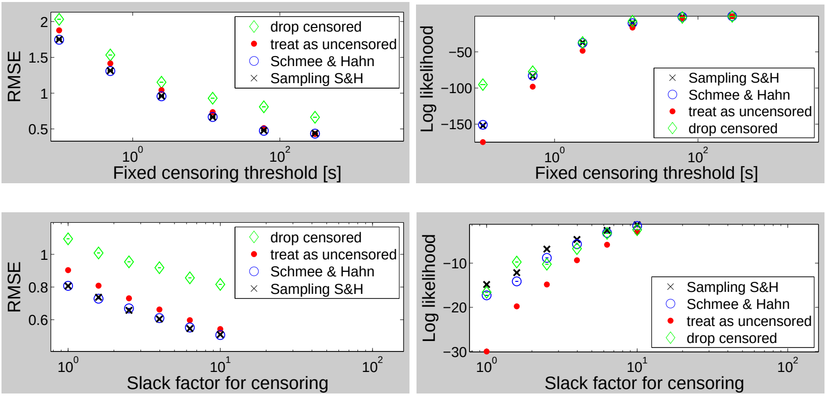


Figure E.46: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: `SPEAR-SWV`.

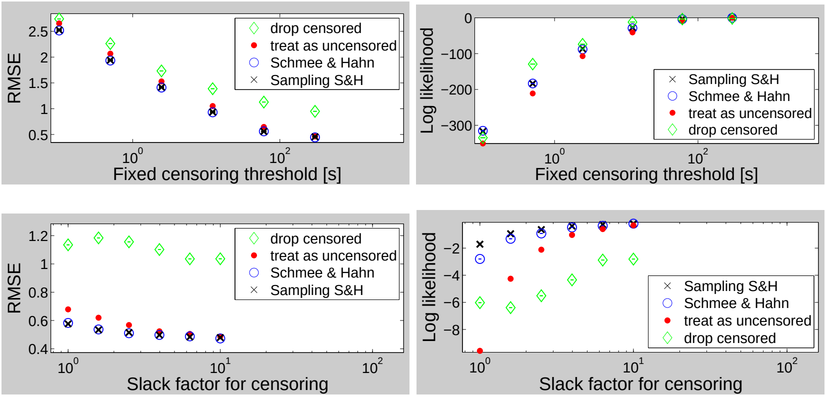


Figure E.47: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: `SPEAR-IBM`.

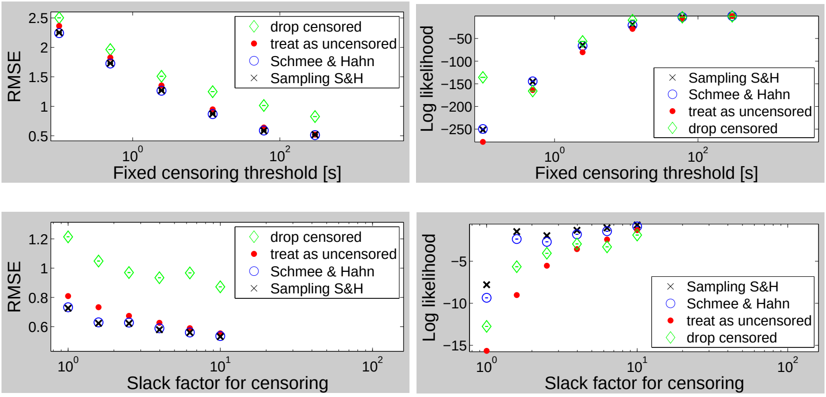


Figure E.48: RMSE and log likelihood of four ways of handling censored data with random forests, for various levels of aggressiveness in setting the censoring threshold. Top: fixed thresholds; bottom: instance-specific thresholds. In both cases, larger numbers mean less censoring. Benchmark: `SPEAR-SWV-IBM`.

	RMSE							Time to learn model (s)						
Domain	RR	SP	NN	PP	RT	RF-def	RF	RR	SP	NN	PP	RT	RF-def	RF
Minisat 2.0-COMPETITON	0.93	1.12	0.61	0.92	0.68	**0.47**	**0.47**	478	3.871e+04	6717	46.56	**20.96**	22.42	630.5
Minisat 2.0-HAND	0.97	1.18	0.62	0.85	0.75	**0.51**	**0.5**	303.9	1.269e+04	1857	44.14	6.15	**5.51**	153.7
Minisat 2.0-RAND	0.56	0.48	0.39	0.55	0.5	**0.37**	**0.36**	390.8	2.719e+04	1498	46.09	**7.15**	8.58	199.1
Minisat 2.0-INDU	0.93	1.01	0.79	0.86	0.71	**0.52**	**0.54**	233.8	7326	915.2	48.12	6.36	**4.42**	135.3
Minisat 2.0-SWV-IBM	0.47	0.4	0.33	0.52	0.25	**0.17**	**0.15**	272.5	1.104e+04	597.9	51.67	4.8	**2.74**	96.11
Minisat 2.0-IBM	0.47	0.47	0.32	0.34	0.3	**0.19**	**0.19**	217.9	1.068e+04	361.6	46.16	2.47	**1.5**	53.42
Minisat 2.0-SWV	0.34	1.18	0.18	0.1	0.1	**0.08**	**0.08**	199.9	3468	155.8	53.11	2.37	**1.07**	36.15
CryptoMinisat-INDU	0.94	0.97	0.9	0.9	0.91	**0.72**	**0.71**	237.1	5372	526.6	45.82	5.03	**4.14**	104.5
CryptoMinisat-SWV-IBM	0.75	0.67	0.69	0.83	0.62	**0.48**	**0.48**	265.5	1.307e+04	778.6	48.99	4.75	**2.78**	76.68
CryptoMinisat-IBM	1.04	0.67	0.55	0.56	0.53	**0.41**	**0.42**	215	8902	710.5	44.9	2.41	**1.49**	46.78
CryptoMinisat-SWV	0.89	0.79	0.68	0.66	0.63	**0.51**	**0.53**	181.1	3718	156.4	53.85	2.32	**1.03**	34.7
SPEAR-INDU	0.96	29.52	0.89	0.87	0.8	**0.58**	**0.6**	211.7	6402	1069	45.47	5.52	**4.25**	139.1
SPEAR-SWV-IBM	0.63	0.63	0.5	0.78	0.49	**0.38**	**0.36**	267.5	1.515e+04	337.8	48.48	4.9	**2.82**	81.06
SPEAR-IBM	0.6	0.55	0.48	0.66	0.5	**0.38**	**0.36**	243.9	9550	220.1	45.72	2.5	**1.56**	49.14
SPEAR-SWV	0.58	0.57	0.46	0.44	**0.47**	**0.34**	**0.34**	183.1	2618	113.5	56.09	2.38	**1.13**	43.74
tnm-RANDSAT	0.96	0.95	0.94	0.93	1.22	**0.88**	**0.86**	268.6	1.079e+04	527.2	46.21	7.64	**5.42**	137.8
SAPS-RANDSAT	0.86	0.81	0.71	0.78	0.86	**0.66**	**0.65**	306.5	1.554e+04	370.2	49.33	6.59	**5.04**	135.8
CPLEX-BIGMIX	0.91	0.93	0.91	1	0.85	**0.64**	**0.64**	139.9	1257	212.6	41.25	5.33	**3.54**	110.6
Gurobi-BIGMIX	1.21	1.22	1.23	1.26	1.43	**1.17**	**1.15**	129.6	1127	210	40.72	5.45	**3.69**	89.39
SCIP-BIGMIX	0.82	0.81	0.74	0.91	0.72	**0.57**	**0.57**	147.8	1722	203.6	39.51	5.08	**3.75**	99.29
lp_solve-BIGMIX	1.74	0.88	**0.6**	1.07	0.63	**0.5**	**0.47**	131.4	1342	204.7	43.27	**2.76**	4.92	120.5
CPLEX-CORLAT	0.48	**0.46**	0.5	**0.46**	0.62	**0.47**	**0.47**	274	8185	458.5	27.54	4.77	**3.4**	108
Gurobi-CORLAT	**0.38**	**0.37**	0.4	**0.37**	0.51	**0.38**	**0.37**	253.7	1.014e+04	407.7	28.58	4.71	**3.31**	100.7
SCIP-CORLAT	0.38	**0.38**	0.4	**0.37**	0.5	**0.38**	**0.37**	268.2	9769	430.8	26.89	5.12	**3.52**	108.4
lp_solve-CORLAT	0.42	**0.4**	**0.44**	**0.45**	0.54	**0.41**	**0.42**	280.7	4812	389.8	31.5	**2.63**	4.42	119.8
CPLEX-RCW	0.19	0.14	0.11	0.03	0.05	**0.02**	**0.02**	285.7	8474	495.1	25.84	4.81	**2.66**	91.36
CPLEX-REG	**0.38**	**0.38**	**0.38**	**0.38**	0.54	0.42	0.42	157.4	5586	458.8	24.95	4.56	**3.65**	111.6
CPLEX-CR	0.45	**0.43**	0.47	**0.43**	0.58	0.45	**0.44**	329.6	2.009e+04	705.6	29.92	11.44	**8.35**	244.9
CPLEX-CRR	0.41	0.39	0.42	**0.37**	0.47	**0.36**	**0.36**	482	4.007e+04	2130	35.3	20.36	**13.19**	396.3
LK-H-RUE	**0.61**	**0.61**	**0.61**	**0.61**	0.89	0.67	0.64	170.9	3128	627.5	22.95	11.49	**11.14**	269.7
LK-H-RCE	**0.71**	**0.7**	**0.71**	**0.71**	1.02	0.76	0.75	199.4	6775	1089	24.78	11.54	**10.79**	268.7
LK-H-TSPLIB	**1.09**	**0.93**	1.67	1.3	**1.21**	1.06	**0.88**	50	406.3	56.99	4.3	0.17	**0.11**	4.98
Concorde-RUE	**0.41**	**0.42**	**0.41**	**0.42**	0.59	0.45	0.44	243.4	7362	574.2	22.28	10.79	**9.9**	282.7
Concorde-RCE	0.33	**0.32**	0.33	0.34	0.46	0.35	0.35	220.8	1.038e+04	575.8	24.8	11.16	**10.18**	249.3
Concorde-TSPLIB	**0.95**	**0.57**	**0.71**	**0.87**	0.64	**0.52**	**0.52**	52.4	374.5	32.23	4.26	0.22	**0.12**	4.97

Table E.9: Quantitative comparison of models for runtime predictions on unseen instances. We report 10-fold cross-validation performance. Lower RMSE values are better (0 is optimal). Boldface indicates results not statistically significantly from the best.

Appendix E.5. Equivalent Results with Hyperparameter Optimization

In this section, we provide equivalent results to most of those presented in Sections 6 to 8 of the main article, but based on models using hyperparameter optimization.

Concerning experiments in the instance space, Table E.9 is the equivalent of Table 2 in the main article, Figure E.49 is the equivalent of Figure 4 in the main article, and Figure E.50 is the equivalent of Figure 5 in the main article.

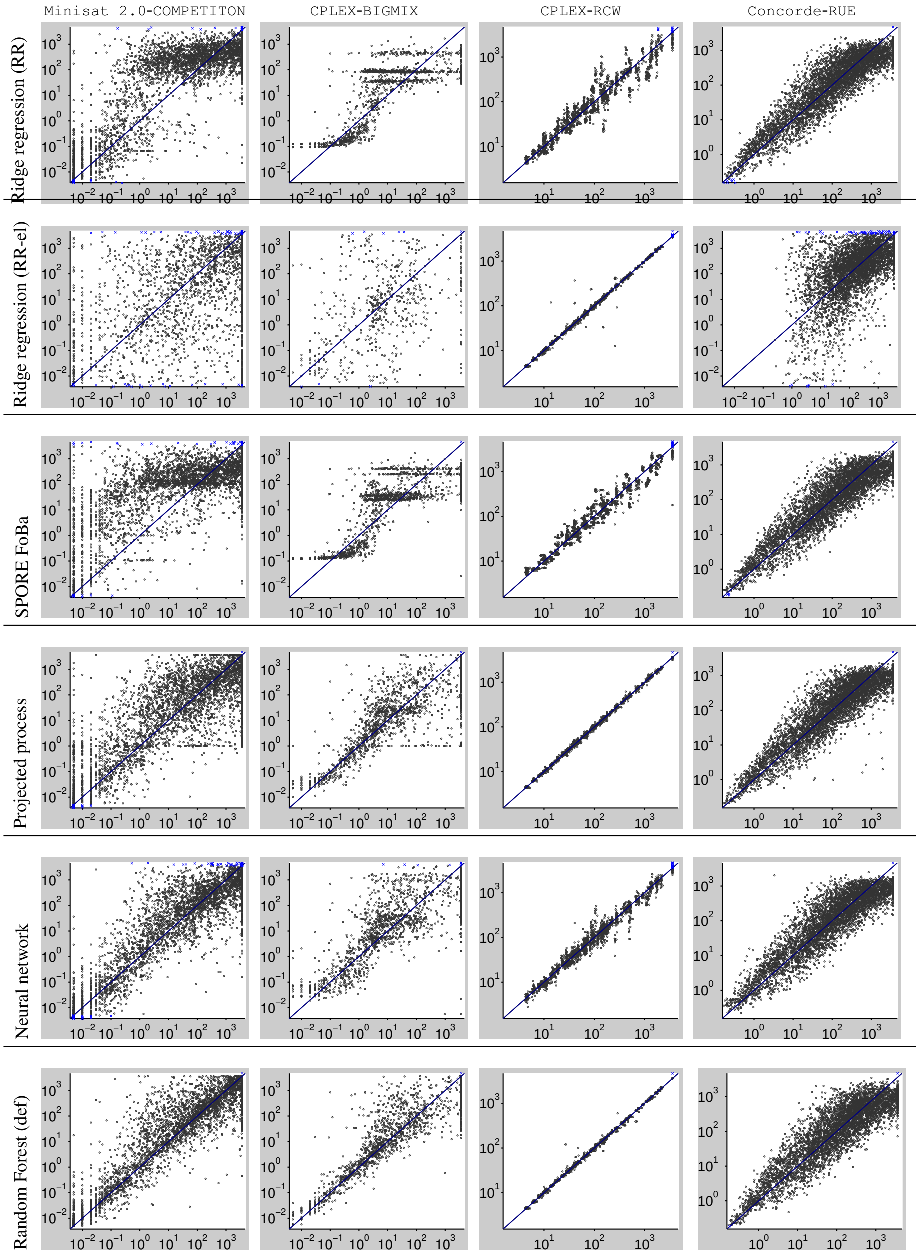
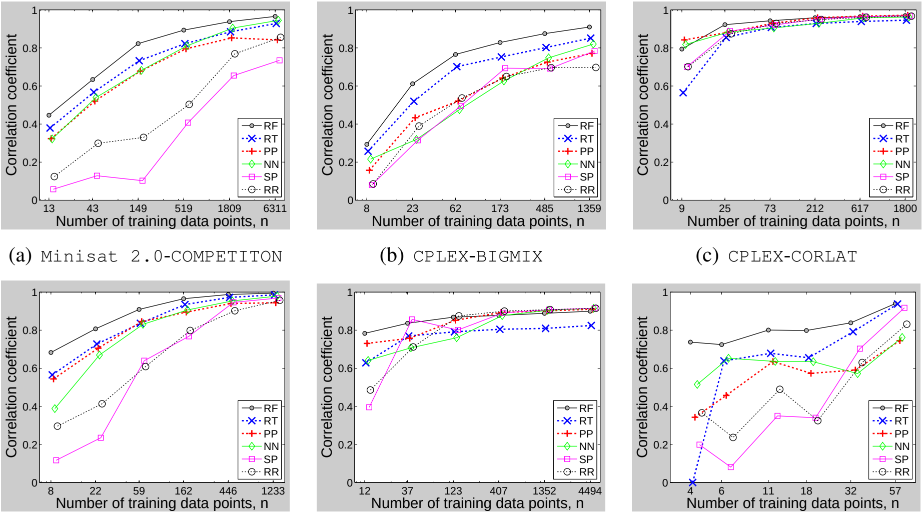


Figure E.49: Visual comparison of models for runtime predictions on unseen instances. The left-most column specifies the data set for the results in each row. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one instance. Predictions above $3\,000$ or below 0.001 are denoted by a blue cross rather than a black dot.



(a) `Minisat 2.0-COMPETITON`

(b) `CPLEX-BIGMIX`

(c) `CPLEX-CORLAT`

(d) `Minisat 2.0-SWV-IBM`

(e) `Concorde-RUE`

(f) `LK-H-TSPLIB`

Figure E.50: Prediction quality with varying numbers of training instances. For each model and number of training instances, we plot mean ± standard deviation of the correlation coefficient (CC) between true and predicted runtimes for new test instances; larger CC is better, 1 is perfect.

	RMSE							Time to learn model (s)						
Domain	RR	SP	NN	PP	RT	RF-def	RF	RR	SP	NN	PP	RT	RF-def	RF
CPLEX-BIGMIX	**0.26**	0.34	0.35	**0.24**	0.33	**0.25**	0.26	337.6	799.1	182.3	34.26	4.24	**2.98**	113.1
CPLEX-CORLAT	0.59	0.65	0.72	**0.53**	0.75	**0.55**	**0.56**	349.6	794.9	255.8	32.53	4.19	**3**	111.5
CPLEX-REG	0.43	0.47	0.57	0.42	0.49	**0.38**	**0.38**	363	3386	274.4	29.28	4	**2.86**	107
CPLEX-RCW	**0.21**	0.25	0.3	**0.21**	0.28	**0.21**	**0.21**	324.9	493.1	192.2	33.6	2.25	**1.93**	73.94
SPEAR-IBM	**0.25**	0.75	0.7	**0.25**	0.31	0.28	0.27	215.7	143.1	135.1	11.3	1.62	**1.51**	59.16
SPEAR-SWV	**0.36**	0.52	0.51	**0.35**	0.41	**0.36**	**0.37**	210.8	142.3	133.1	12.49	1.68	**1.52**	61.31

Table E.10: Quantitative comparison of models for runtime predictions on unseen configurations. We report 10-fold cross-validation performance. Lower RMSE is better (0 is optimal). Boldface indicates results not statistically significantly from the best.

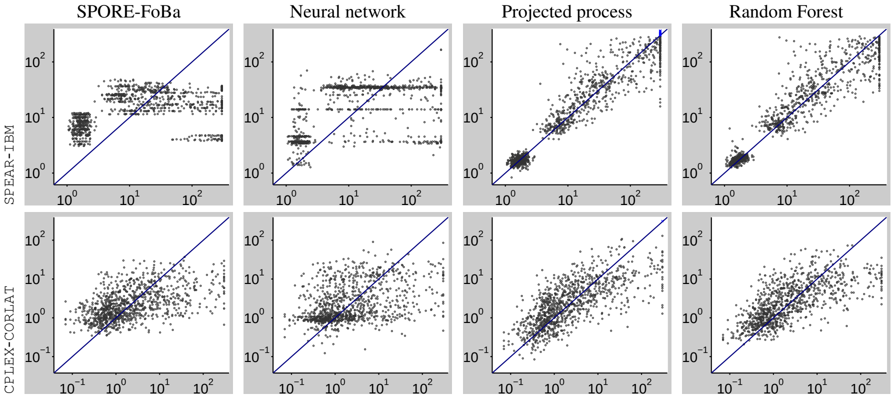
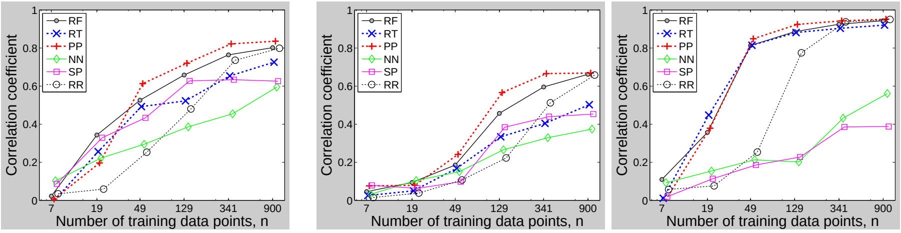


Figure E.51: Visual comparison of models for runtime predictions on previously unseen parameter configurations. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one parameter configuration.

Concerning experiments in the configuration space, Table E.10 is the equivalent of Table 5 in the main article, Figure E.51 is the equivalent of Figure 6 in the main article, and Figure E.52 is the equivalent of Figure 7 in the main article.



(a) `CPLEX-BIGMIX` (b) `CPLEX-CORLAT` (c) `SPEAR-IBM`

Figure E.52: Quality of predictions in the configuration space, as dependent on the number of training configurations. For each model and number of training instances, we plot mean ± standard deviation of the correlation coefficient (CC) between true and predicted runtimes for new test configurations.

	RMSE							**Time to learn model (s)**						
Domain	RR	SP	NN	PP	RT	RF-def	RF	RR	SP	NN	PP	RT	RF-def	RF
CPLEX-BIGMIX	4.1	2.9	0.7	0.78	0.74	0.55	**0.54**	1.1E3	6.8E4	2.9E3	1E2	70	**49**	1.7E3
CPLEX-CORLAT	0.51	0.51	0.57	0.53	0.67	**0.46**	0.49	1.4E3	5.8E4	3.2E3	75	47	**41**	1.2E3
CPLEX-REG	**0.17**	0.18	0.19	0.19	0.24	**0.17**	**0.17**	1.4E3	1.9E4	2.7E3	73	33	**31**	1E3
CPLEX-RCW	0.1	0.1	0.12	0.12	0.12	**0.09**	0.1	1.6E3	4.2E4	2.4E3	72	25	**23**	7.7E2
CPLEX-CR	0.39	0.4	0.43	0.42	0.52	**0.38**	**0.38**	1.2E3	7.5E4	5.1E3	87	64	**48**	2.5E3
CPLEX-CRR	0.34	0.34	0.37	0.39	0.43	**0.32**	**0.32**	1.4E3	9.1E4	4.6E3	1E2	66	**56**	1.8E3
SPEAR-IBM	0.61	0.75	0.53	0.52	0.57	**0.41**	0.42	9.4E2	8.3E4	3.7E3	74	54	**30**	1.5E3
SPEAR-SWV	0.57	0.51	0.61	0.54	0.55	**0.44**	0.45	1E3	6E4	2.1E3	64	43	**28**	8.7E2
SPEAR-SWV-IBM	0.62	0.59	0.64	0.65	0.59	**0.42**	0.45	9.6E2	1.1E5	4.3E3	1E2	54	**32**	2E3

Table E.11: Quantitative comparison of models for runtime predictions on unseen instances and configurations. Models were based on 10 000 data points.

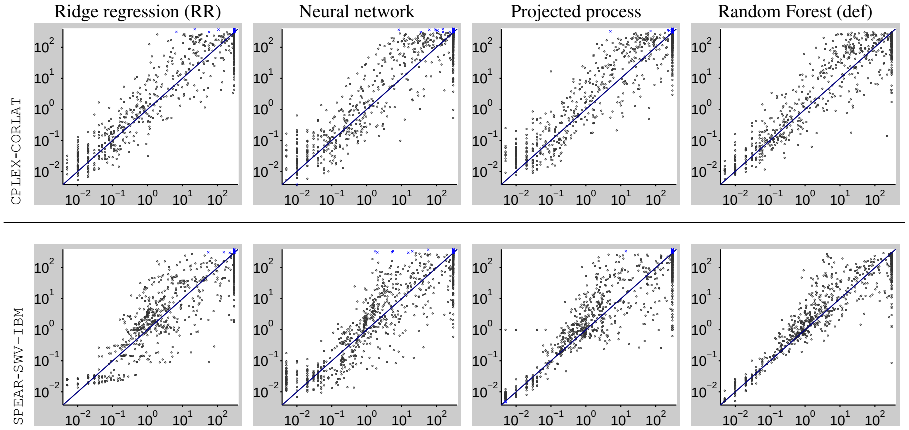


Figure E.53: Visual comparison of models for runtime predictions on pairs of previously unseen test configurations and instances. In each subfigure, the x-axis denotes true runtime and the y-axis cross-validated runtime as predicted by the respective model. Each dot represents one combination of an unseen instance and parameter configuration.

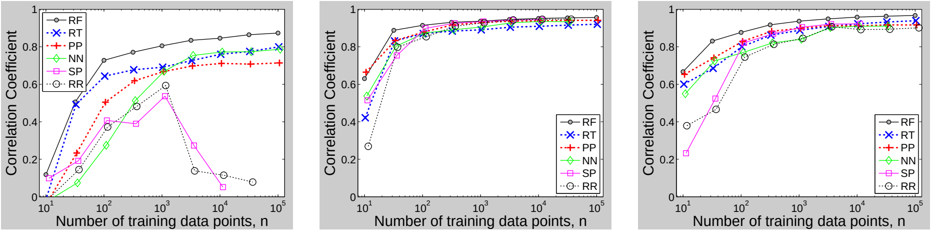
Concerning experiments in the combined space, Table E.11 is the equivalent of Table 6 in the main article, Figure E.53 is the equivalent of Figure 8 in the main article, Figure E.54 is the equivalent of Figure 9 in the main article, Table E.12 is the equivalent of Table 7 in the main article, Figure E.55 is the equivalent of Figure 10 in the main article, and Figure E.56 is the equivalent of Figure 11 in the main article.

References

[1] Leyton-Brown, K., Nudelman, E., & Shoham, Y. (2009). Empirical hardness models: methodology and a case study on combinatorial auctions. *Journal of the ACM*, *56*(4), 1–52.

[2] Schmee, J. & Hahn, G. J. (1979). A simple method for regression analysis with censored data. *Technometrics*, *21*(4), 417–432.

[3] Shawe-Taylor, J. & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.



(a) CPLEX-BIGMIX (b) CPLEX-CORLAT (c) SPEAR-SWV-IBM

Figure E.54: Quality of predictions in the joint instance/configuration space, varying the number of training data points. For each model and number of training data points, we plot mean correlation coefficients between true and predicted runtimes for new test instances and configurations. We omit standard deviations to avoid clutter, but they are very high for the two ridge regression variants.

		Training configurations							**Test configurations**						
Domain	**Instances**	RR	SP	NN	PP	RT	RF-def	RF	RR	SP	NN	PP	RT	RF-def	RF
CPLEX-BIGMIX	Training	0.57	0.52	0.54	0.65	0.59	**0.43**	**0.43**	0.57	0.52	0.55	0.65	0.62	**0.45**	**0.45**
	Test	4.1	2.9	0.69	0.78	0.71	0.54	**0.53**	4.1	2.9	0.7	0.78	0.74	0.55	**0.54**
CPLEX-CORLAT	Training	0.47	0.48	0.45	0.49	0.54	0.4	**0.38**	0.49	0.5	0.55	0.51	0.64	0.47	**0.46**
	Test	0.49	0.49	0.48	0.51	0.58	**0.4**	0.42	0.51	0.51	0.57	0.53	0.67	**0.46**	0.49
CPLEX-REG	Training	0.15	0.16	0.15	0.16	0.17	**0.12**	**0.12**	**0.16**	0.17	0.18	0.17	0.22	**0.16**	**0.16**
	Test	0.16	0.18	0.16	0.18	0.19	0.14	**0.13**	**0.17**	0.18	0.19	0.19	0.24	**0.17**	**0.17**
CPLEX-RCW	Training	0.09	0.09	0.11	0.1	0.08	**0.06**	**0.06**	0.1	0.1	0.12	0.11	0.12	**0.09**	0.1
	Test	0.09	0.09	0.11	0.11	0.08	**0.06**	**0.06**	0.1	0.1	0.12	0.12	0.12	**0.09**	0.1
CPLEX-CR	Training	0.37	0.37	0.37	0.4	0.45	**0.32**	**0.32**	0.38	0.38	0.42	0.41	0.49	**0.36**	**0.36**
	Test	0.38	0.38	0.38	0.41	0.47	**0.34**	**0.34**	0.39	0.4	0.43	0.42	0.52	**0.38**	**0.38**
CPLEX-CRR	Training	0.32	0.31	0.33	0.36	0.38	0.28	**0.27**	0.33	0.32	0.36	0.37	0.41	0.31	**0.3**
	Test	0.33	0.33	0.34	0.38	0.4	**0.29**	**0.29**	0.34	0.34	0.37	0.39	0.43	**0.32**	**0.32**
SPEAR-IBM	Training	0.53	0.48	0.49	0.48	0.43	0.36	**0.33**	0.53	0.48	0.5	0.48	0.45	0.37	**0.35**
	Test	0.61	0.75	0.52	0.52	0.57	0.41	**0.4**	0.61	0.75	0.53	0.52	0.57	**0.41**	0.42
SPEAR-SWV	Training	0.47	0.45	0.54	0.46	0.37	**0.3**	**0.3**	0.47	0.45	0.55	0.47	0.43	**0.34**	**0.34**
	Test	0.57	0.51	0.6	0.53	0.51	**0.4**	0.42	0.57	0.51	0.61	0.54	0.55	**0.44**	0.45
SPEAR-SWV-IBM	Training	0.58	0.55	0.59	0.62	0.48	0.38	**0.36**	0.58	0.55	0.6	0.62	0.5	0.39	**0.38**
	Test	0.62	0.59	0.63	0.65	0.58	**0.42**	0.44	0.62	0.59	0.64	0.65	0.59	**0.42**	0.45

Table E.12: Root mean squared error (RMSE) obtained by various empirical performance models for predicting the runtime of ⟨configuration, instance⟩ combinations. We trained on 10 000 randomly-sampled combinations of training configurations and instances, and report performance for the four combinations of training/test instances and training/test configurations.

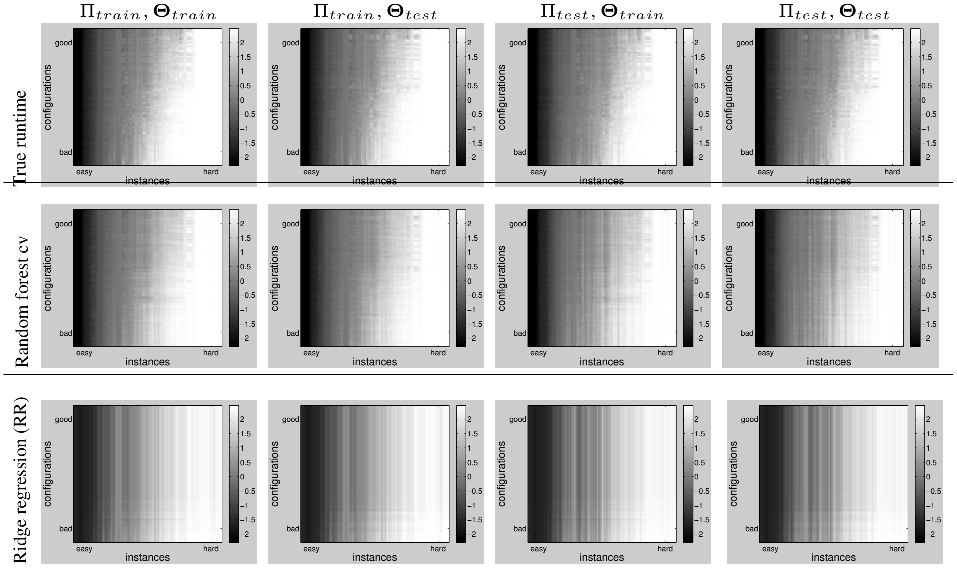


Figure E.55: True and predicted runtime matrices for scenario SPEAR-SWV-IBM, for all combinations of training/test instances (Π_{train} and Π_{test}, respectively) and training test configurations ($\mathbf{\Theta}_{train}$ and $\mathbf{\Theta}_{test}$, respectively). The matrices produced using regression trees are visually indistinguishable from those produced using random forests; those produced using all other methods not shown closely resemble those produced using ridge regression.)

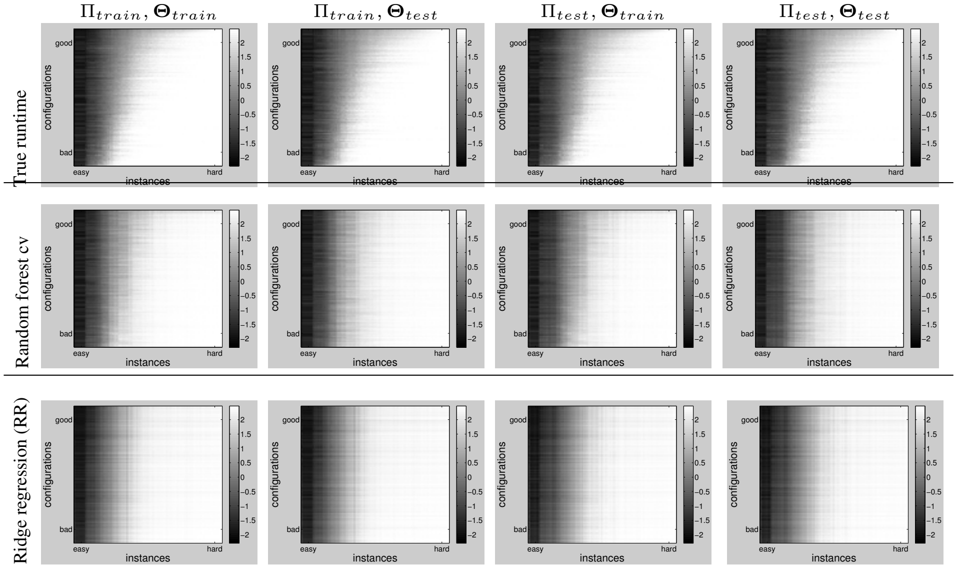


Figure E.56: True and predicted runtime matrices for scenario CPLEX-CORLAT, for all combinations of training/test instances ($\Pi_{train} and \Pi_{test}$, respectively) and training test configurations ($\mathbf{\Theta}_{train}$ and $\mathbf{\Theta}_{test}$, respectively).