1 Additional numerical results including XG-Boost algorithm

This section provides supplementary numerical results of exponential kernel-based consensual aggregation method. We include in this experiment the `XGboost` (Chen and Guestrin Chen and Guestrin (2016)) predictor, denoted by XGB, which is an outstanding method according to many applications and its performances in many Kaggle's challenges. In this simulation, the method is implemented using `xgboost` library of `R` software (Chen Chen et al. (2021)). We are interested in the behavior of the combining method when a strong predictive method is presented. The experiment is carried out on the same set of simulated and real datasets.

1.1 Simulated datasets

The results reported in this part are computed from 100 independent runs of the proposed combining estimation method implemented using the 10 models of simulated data in Section 4. The performances of uncorrelated and uncorrelated cases are presented in Table 1 and Table 2 respectively. Only Gaussian kernel is considered in this simulation as it stood out from the rest in the previous numerical experiments. Let Gauss Grid and Gauss GD stand for Gaussian kernel-based method obtained by grid search and gradient descent algorithm respectively. Note that each method is implemented on a computer with the following characteristics:

- System type: 64-bit operating system, x64-based processor.
- Processor: Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz.
- RAM: 16.0 GB.

As expected, XGB stands out from the rest of other basic regressors. Moreover, the performances of the aggregation methods are quite close to the best individual machine and sometimes even outperform the best one. We can also see that the performances of Gaussian kernel are quite similar indicating the right performance of gradient descent algorithm. Visually, Figure 1 and Figure 2 contain the boxplots of the results reported in Table 1 and Table 2 respectively. Moreover, The boxplots of running times of all the methods are given in Figture 3 and Figture 4 below.

Table 1: Average MSEs in the uncorrelated case.

Model	**Las**	**Rid**	**kNN**	**Tr**	**RF**	**XGB**	**COBRA**	**Gauss Grid**	**Gauss GD**
1	0.152 (0.016)	0.131 (0.013)	0.14 (0.015)	0.027 (0.004)	0.031 (0.004)	**0.005** (0.001)	0.011 (0.005)	**0.006** (0.001)	**0.006** (0.001)
2	1.306 (0.186)	0.755 (0.067)	0.849 (0.084)	1.077 (0.143)	**0.678** (0.059)	0.712 (0.074)	0.707 (0.061)	0.694 (0.063)	**0.693** (0.062)
3	0.653 (0.087)	0.658 (0.235)	1.463 (0.173)	0.779 (0.125)	0.610 (0.079)	**0.526** (0.064)	0.479 (0.045)	**0.453** (0.045)	**0.453** (0.044)
4	7.563 (1.083)	6.566 (1.411)	9.616 (1.358)	3.463 (0.718)	3.581 (0.449)	**2.509** (0.328)	2.819 (0.416)	**2.565** (0.341)	2.566 (0.338)
5	0.480 (0.045)	0.487 (0.065)	0.669 (0.085)	0.554 (0.067)	**0.413** (0.040)	0.442 (0.046)	0.411 (0.037)	0.399 (0.038)	**0.398** (0.038)
6	2.638 (0.514)	1.878 (0.286)	2.600 (0.292)	2.995 (0.362)	1.743 (0.225)	**1.529** (0.203)	1.370 (0.178)	**1.351** (0.192)	1.353 (0.191)
7	1.878 (0.380)	0.756 (0.105)	1.036 (0.120)	0.711 (0.099)	0.495 (0.058)	**0.475** (0.055)	0.473 (0.046)	**0.462** (0.051)	**0.462** (0.051)
8	0.124 (0.015)	0.122 (0.018)	0.199 (0.020)	0.158 (0.032)	**0.119** (0.012)	0.120 (0.022)	0.096 (0.013)	0.094 (0.014)	**0.093** (0.014)
9	1.544 (0.203)	2.899 (0.397)	3.504 (0.476)	1.767 (0.360)	1.429 (0.179)	**0.949** (0.161)	0.955 (0.122)	**0.868** (0.143)	0.869 (0.143)
10	1927.677 (267.480)	**1392.562** (172.288)	1668.111 (224.656)	2951.823 (431.816)	1511.842 (178.790)	1688.174 (219.798)	1496.756 (166.315)	**1491.847** (177.779)	1493.466 (173.211)

Table 2: Average MSEs in the correlated case.

Model	**Las**	**Rid**	**kNN**	**Tr**	**RF**	**XGB**	**COBRA**	**Gauss Grid**	**Gauss GD**
1	2.184 (0.468)	1.831 (0.416)	1.841 (0.401)	0.286 (0.123)	0.485 (0.193)	**0.064** (0.048)	0.193 (0.137)	0.064 (0.046)	**0.062** (0.047)
2	13.366 (2.277)	7.635 (1.291)	7.661 (1.155)	6.280 (1.230)	4.643 (0.782)	**4.308** (0.808)	4.450 (0.761)	3.992 (0.729)	**3.986** (0.736)
3	6.995 (4.080)	4.979 (1.362)	7.163 (1.605)	3.030 (1.029)	2.590 (0.951)	**1.562** (0.540)	2.485 (0.663)	1.431 (0.515)	**1.430** (0.544)
4	56.900 (11.211)	39.319 (9.450)	43.676 (10.033)	7.937 (2.076)	12.398 (4.434)	**4.994** (1.142)	8.217 (2.340)	5.361 (1.366)	**5.357** (1.431)
5	5.434 (1.994)	6.783 (3.726)	8.750 (3.391)	2.550 (1.217)	3.466 (2.060)	**1.253** (1.558)	2.473 (1.127)	0.500 (0.635)	**0.465** (0.621)
6	4.231 (0.916)	2.059 (0.394)	4.522 (0.615)	3.168 (0.519)	1.713 (0.247)	**1.324** (0.219)	**1.062** (0.132)	1.120 (0.131)	1.120 (0.132)
7	18.240 (5.532)	4.321 (0.823)	5.148 (0.996)	3.622 (0.844)	2.662 (0.582)	**2.139** (0.626)	2.430 (0.548)	2.368 (0.590)	**2.352** (0.583)
8	0.134 (0.017)	0.129 (0.020)	0.197 (0.021)	0.153 (0.029)	0.118 (0.011)	**0.111** (0.020)	0.092 (0.012)	**0.062** (0.013)	**0.062** (0.013)
9	40.629 (10.965)	30.688 (7.199)	37.252 (8.787)	13.083 (5.382)	13.040 (4.358)	**6.323** (2.705)	9.833 (3.443)	7.036 (3.208)	**6.845** (2.600)
10	6931.342 (949.032)	**5007.011** (968.808)	7360.055 (1237.711)	12529.912 (1933.860)	6754.950 (970.711)	8261.759 (1219.494)	5508.267 (729.912)	**5344.097** (879.113)	5453.242 (985.878)

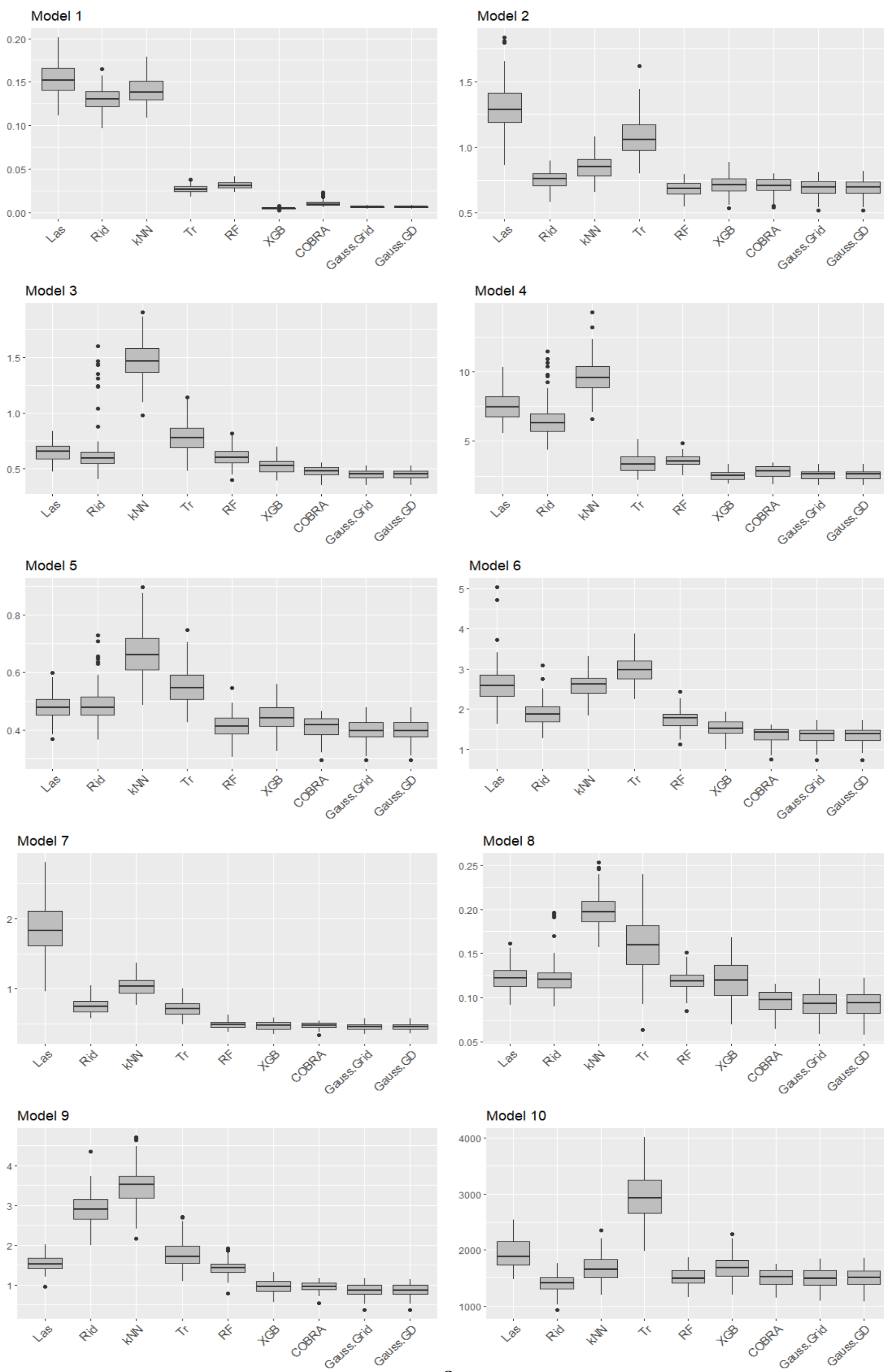

Figure 1: Boxplots of RMSEs of uncorrelated case.

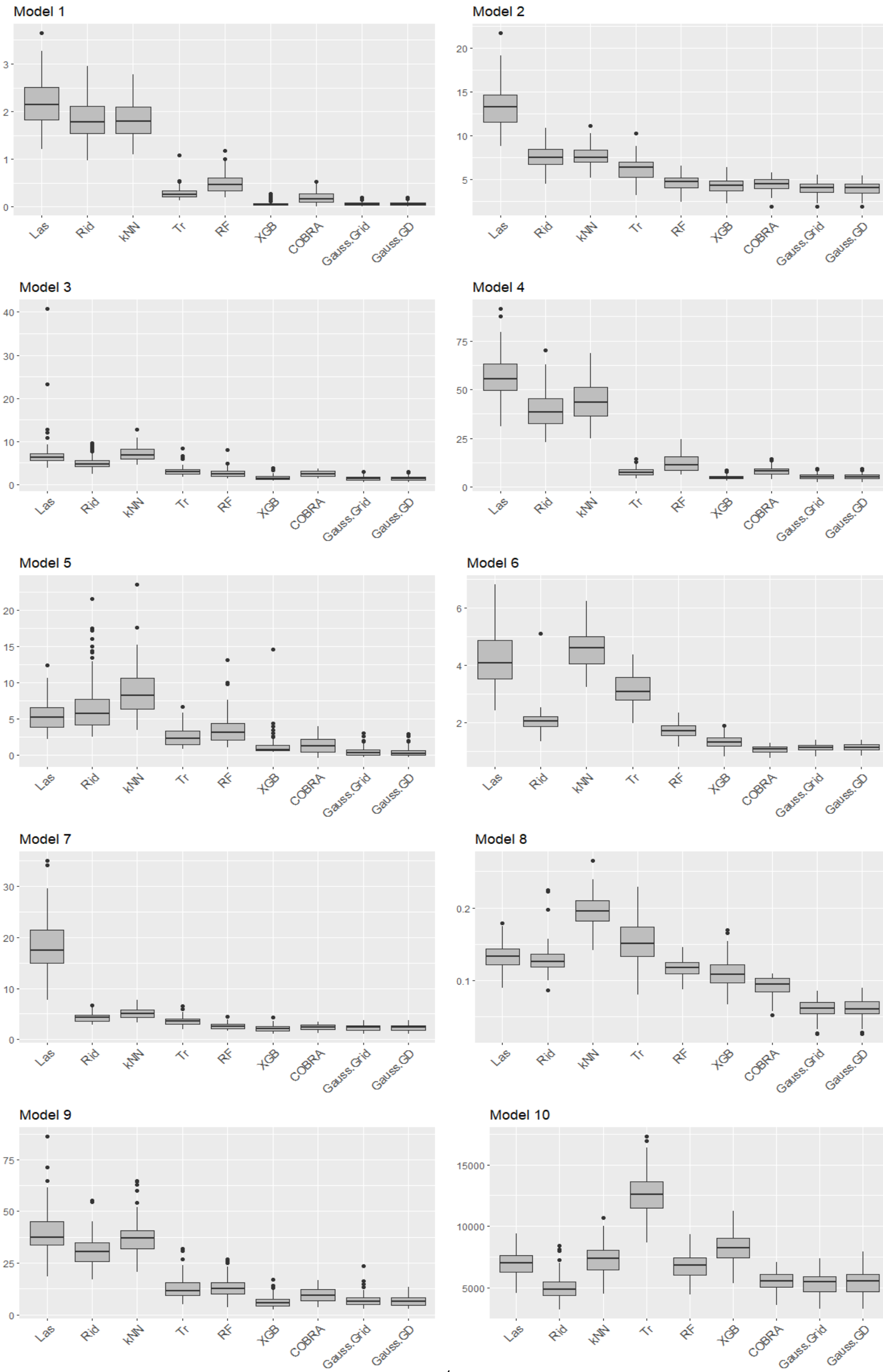

Figure 2: Boxplots of RMSEs of correlated case.

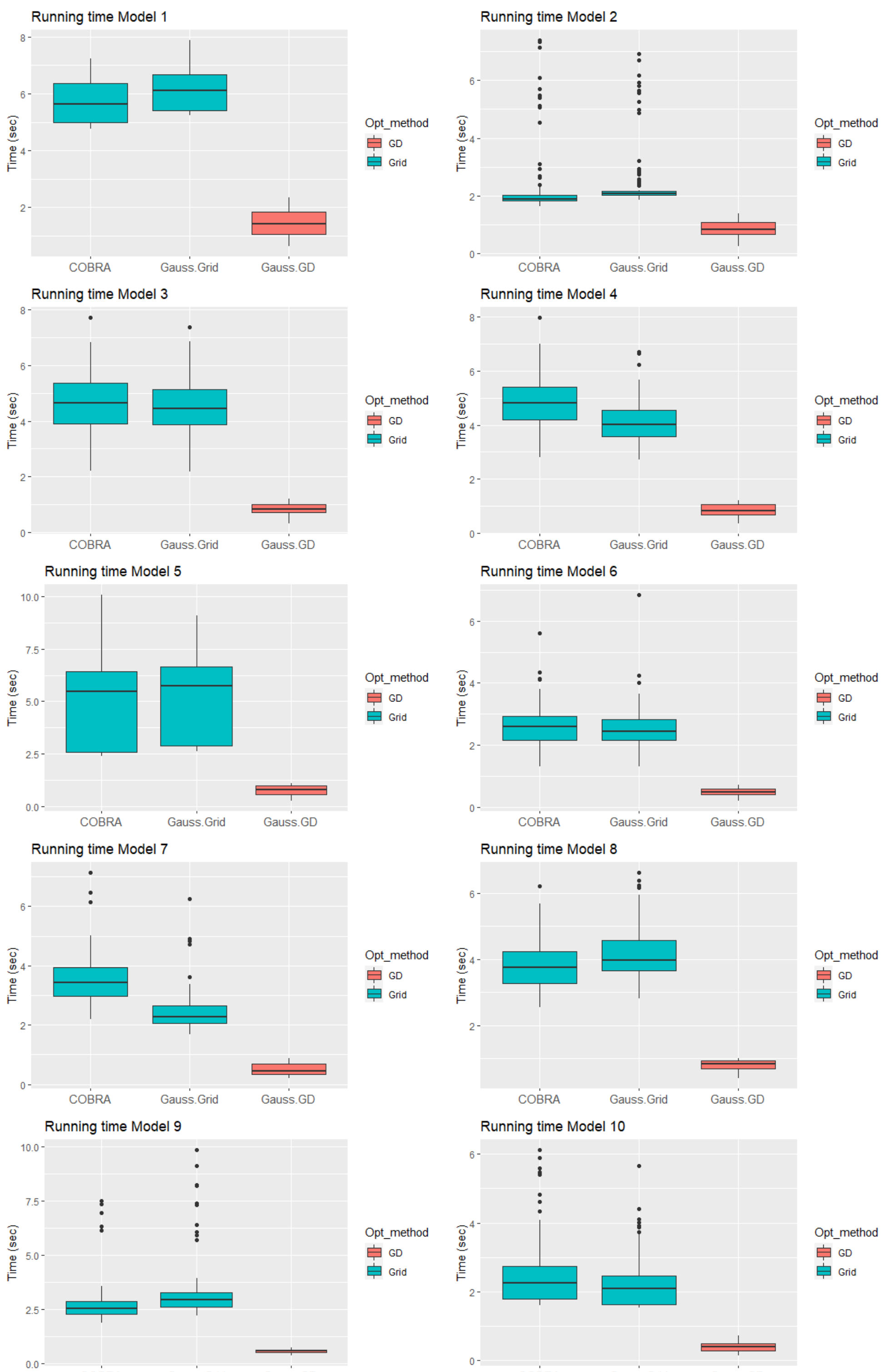

Figure 3: Boxplots of running times of uncorrelated case.

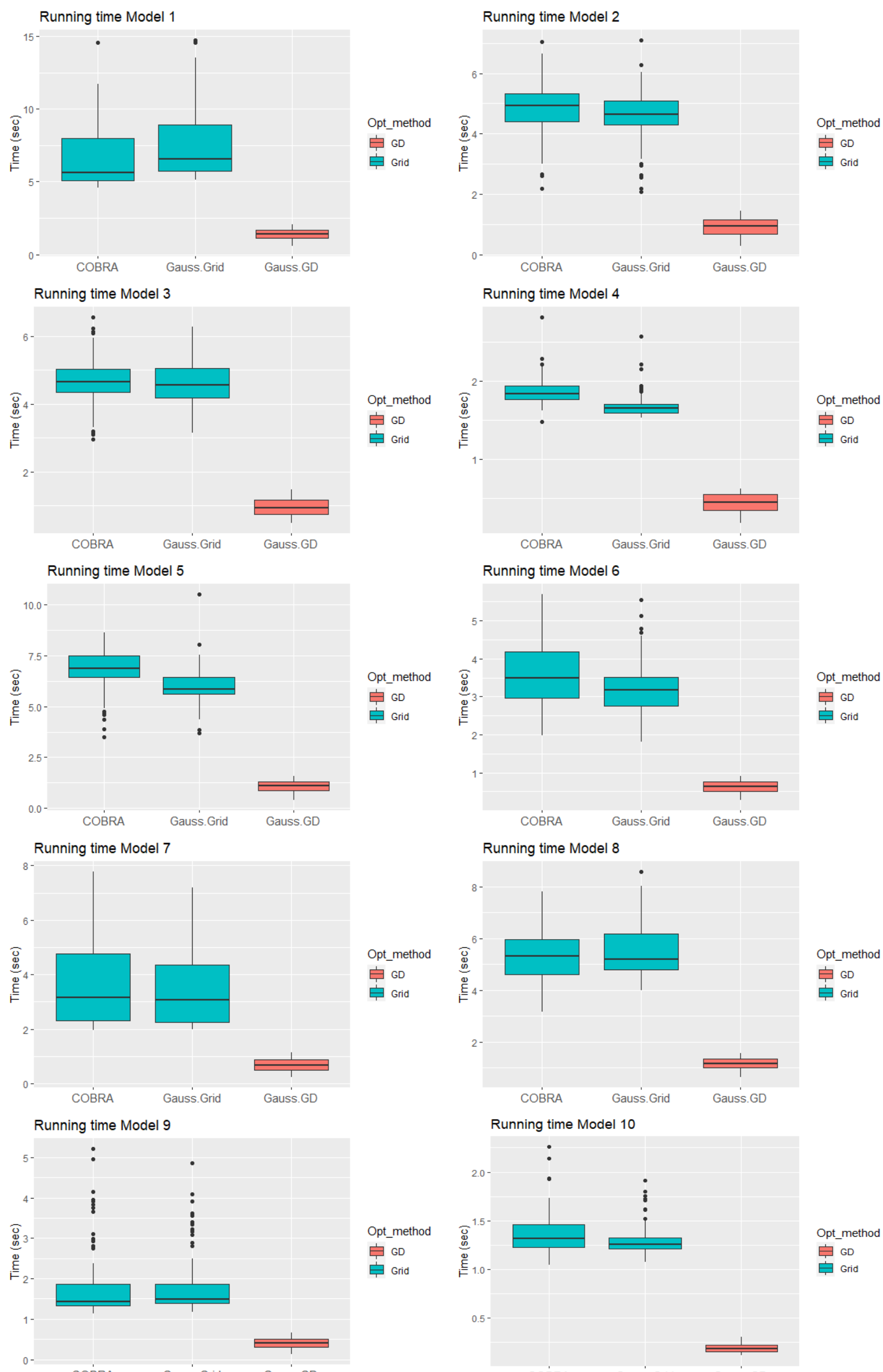

Figure 4: Boxplots of running times of correlated case.

1.2 Real datasets

With the same setting as in the previous part, this section reports the performances of all the methods evaluated on the five real-life datasets: Abalone, Air, Boston (`MASS` library of `R` software, see, Brian et al. Brian et al. (2021)), Turbine, and Wine. Moreover, the corresponding boxplots are given in Figure 5 below.

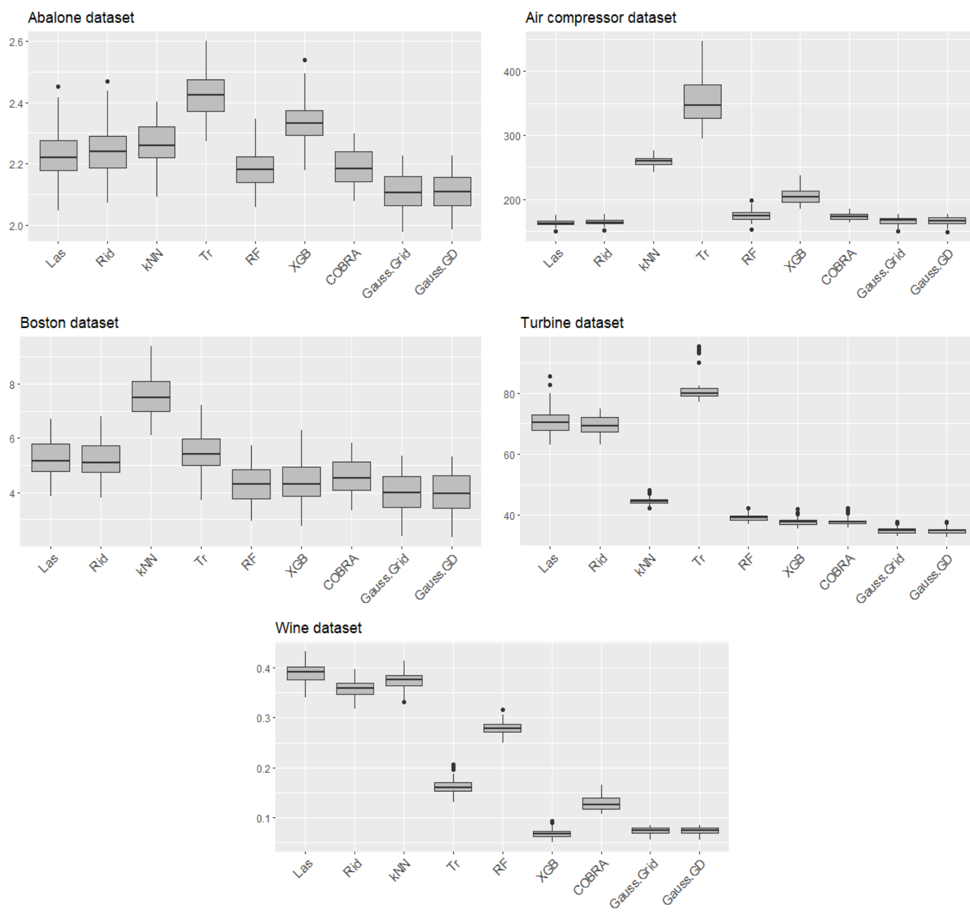

Figure 5: Boxplots of RMSEs of real datasets.

The associated RMSEs and standard errors are reported in Table 3 below.

Finally, the running times of all the methods are given in the Figure 6 below.

Table 3: Average RMSEs of real datasets.

Data	Las	Rid	kNN	Tr	RF	XGB	COBRA	Gauss Grid	Gauss GD
Abalone	2.233 (0.079)	2.247 (0.082)	2.264 (0.070)	2.424 (0.074)	**2.184** (0.061)	2.334 (0.068)	2.189 (0.062)	**2.110** (0.059)	**2.110** (0.058)
Air	**163.298** (4.635)	164.644 (4.685)	259.401 (6.892)	354.961 (34.906)	174.766 (7.617)	204.349 (11.804)	172.781 (4.952)	165.898 (6.216)	**165.872** (5.994)
Boston	5.247 (0.709)	5.218 (0.726)	7.558 (0.725)	5.467 (0.760)	**4.306** (0.684)	4.354 (0.780)	4.582 (0.659)	3.982 (0.775)	**3.963** (0.789)
Turbine	70.266 (3.671)	69.659 (2.795)	44.735 (1.155)	81.238 (4.393)	39.304 (1.153)	**37.938** (1.203)	37.974 (1.176)	34.968 (1.052)	**34.939** (1.047)
Wine	0.388 (0.019)	0.358 (0.016)	0.374 (0.018)	0.162 (0.013)	0.279 (0.013)	**0.068** (0.009)	0.129 (0.015)	**0.074** (0.007)	**0.074** (0.007)

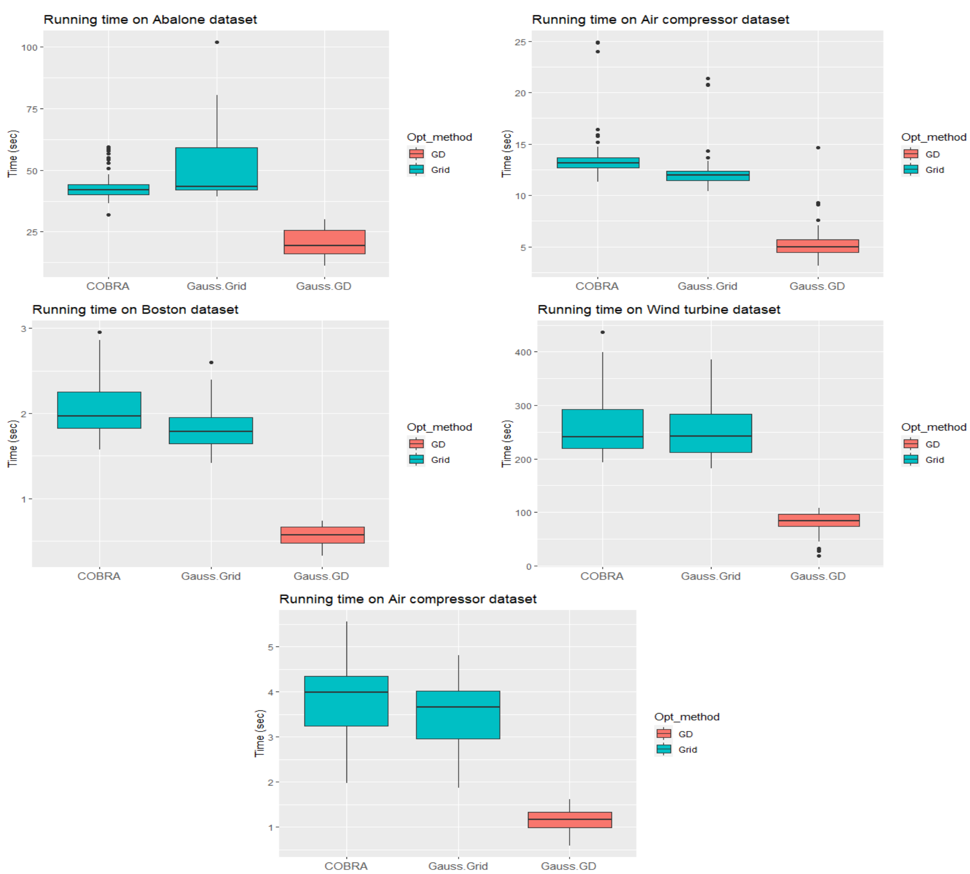

Figure 6: Boxplots of running times of real datasets.

References

Brian, R., Bill, V., Douglas, M.B., Kurt, H., Albrecht, G., David, F., 2021. Mass: Support functions and datasets for venables and ripley's mass. URL: https://CRAN.R-project.org/package=MASS.

Chen, T., Guestrin, C., 2016. Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, New York, NY, USA. p. 785–794. URL: https://doi.org/10.1145/2939672.2939785, doi:10.1145/2939672.2939785.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y., contributors, X., 2021. xgboost: Extreme gradient boosting. URL: https://CRAN.R-project.org/package=xgboost.