Received: 11 November 2022 | Revised: 10 September 2023 | Accepted: 21 September 2023

DOI: 10.1002/sim.9932

RESEARCH ARTICLE

Statistics in Medicine WILEY

A comparison of hyperparameter tuning procedures for clinical prediction models: A simulation study

Zoë S. Dunias[1] | **Ben Van Calster**[2,3] | **Dirk Timmerman**[2,4] | **Anne-Laure Boulesteix**[5,6] | **Maarten van Smeden**[1]

[1]Julius Center for Health Sciences and Primary Care, University Medical Center Utrecht, Utrecht, The Netherlands

[2]Department of Development and Regeneration, KU Leuven, Leuven, Belgium

[3]Department of Biomedical Data Sciences, Leiden University Medical Center, Leiden, The Netherlands

[4]Department of Obstetrics and Gynecology, University Hospitals Leuven, Leuven, Belgium

[5]Institute for Medical Information Processing, Biometry and Epidemiology, University of Munich, Munich, Germany

[6]Munich Center for Machine Learning (MCML), LMU Munich, Munich, Germany

Correspondence
Zoë S. Dunias, Julius Center for Health Sciences and Primary Care, University Medical Center Utrecht, Utrecht, The Netherlands.
Email: z.s.dunias-2@umcutrecht.nl

Tuning hyperparameters, such as the regularization parameter in Ridge or Lasso regression, is often aimed at improving the predictive performance of risk prediction models. In this study, various hyperparameter tuning procedures for clinical prediction models were systematically compared and evaluated in low-dimensional data. The focus was on out-of-sample predictive performance (discrimination, calibration, and overall prediction error) of risk prediction models developed using Ridge, Lasso, Elastic Net, or Random Forest. The influence of sample size, number of predictors and events fraction on performance of the hyperparameter tuning procedures was studied using extensive simulations. The results indicate important differences between tuning procedures in calibration performance, while generally showing similar discriminative performance. The one-standard-error rule for tuning applied to cross-validation (1SE CV) often resulted in severe miscalibration. Standard non-repeated and repeated cross-validation (both 5-fold and 10-fold) performed similarly well and outperformed the other tuning procedures. Bootstrap showed a slight tendency to more severe miscalibration than standard cross-validation-based tuning procedures. Differences between tuning procedures were larger for smaller sample sizes, lower events fractions and fewer predictors. These results imply that the choice of tuning procedure can have a profound influence on the predictive performance of prediction models. The results support the application of standard 5-fold or 10-fold cross-validation that minimizes out-of-sample prediction error. Despite an increased computational burden, we found no clear benefit of repeated over non-repeated cross-validation for hyperparameter tuning. We warn against the potentially detrimental effects on model calibration of the popular 1SE CV rule for tuning prediction models in low-dimensional settings.

KEYWORDS

cross-validation, hyperparameter tuning, penalized regression, prediction models, Random Forest

1 | INTRODUCTION

Clinical prediction models are used to estimate the probability of the presence of current disease (diagnoses) and future health status (prognosis) for individual patients.[1] To develop a prediction model, developers often rely on statistical learning methods such as penalized regression, tree-based methods, and neural networks. These methods involve one or more hyperparameters which generally determine the complexity of the model (eg, regularization parameter in Ridge regression).[2] Hyperparameters can be set to default values, which might not be generalizable across different datasets and research settings, or tuned to find their optimal values for a specific prediction problem at hand. Whereas first-level model parameters (eg, regression coefficients) are directly estimated from the data during model training, these second-level hyperparameters are often tuned using resampling methods.[3,4] The hyperparameters to be tuned and the procedure for tuning have to be chosen beforehand by the model-developer.

There are various procedures for hyperparameter tuning that can be used, many of which are based on cross-validation (CV). Besides regular K-fold CV, earlier studies proposed to perform repeated K-fold CV (eg, 10 × 10-fold CV)[5] aiming to obtain more stable estimates of predictive performance on which hyperparameter tuning is based. Moreover, the model-developer needs to decide on additional different configuration parameters, including the number of folds and the (predictive performance) optimization selection criterion. In standard CV for tuning, a certain out-of-sample predictive performance outcome is optimized over a hyperparameter space (eg, by minimizing deviance). Alternatively, the one-standard-error rule for tuning applied to CV (1SE CV) has become increasingly widespread,[6,7] where the hyperparameter value that results in the simplest model still within one standard error of the optimal model is selected. Although less commonly used for tuning, another popular resampling approach, namely bootstrapping, can also be used for hyperparameter tuning.[8]

Earlier studies on hyperparameter tuning have focused mostly on high-dimensional data,[3,9] where a large reduction in model complexity (ie, strong shrinkage) can be expected.[3] However, for clinical prediction models where most applications focus on low-dimensional data, a small reduction in model complexity (ie, weak shrinkage) can be expected. As recently argued by Ellenbach and colleagues,[3] guidance for hyperparameter tuning in low-dimensional settings is still lacking. The aim of this study is therefore to systematically evaluate and compare various hyperparameter tuning procedures for clinical prediction models in low-dimensional data. A simulation study is performed to evaluate out-of-sample predictive performance (discrimination, calibration, and overall prediction error) of clinical prediction models that are developed using several methods for dichotomous risk prediction (Ridge[10] and Lasso [Least absolute shrinkage and selection operator][11] logistic regression, Elastic Net[12] and Random Forest[13]). The effects of sample size, number of predictors, and events fraction on the performance of hyperparameter tuning procedures are examined.

This article is structured as follows. The models and estimation methods as well as the tuning procedures are described in Section 2. In Section 3 the simulation study design is described and in Section 4 the results are presented. An applied example is presented in Section 5. A discussion of the findings and its implications is provided in Section 6.

2 | MODELS

The interest is in estimating the probability of an event occurring ($Y = 1$) given values of p predictor variables, $\boldsymbol{X} = \{1, X_1, \dots, X_p\}$. For a patient i ($i = 1, \dots, N$), this probability is defined as $\pi_i = P(Y = 1|\boldsymbol{x}_i)$. The logistic regression function from which the probability of an event is predicted, is given by

$$\pi_i = \frac{1}{1 + exp\{-\boldsymbol{\beta}'\boldsymbol{x}_i\}}, \tag{1}$$

where the vector $\boldsymbol{\beta}$ contains an intercept and p regression coefficients (β_j). In unpenalized (ie, maximum likelihood) logistic regression, the coefficients $\boldsymbol{\beta}$ are estimated by maximizing the log-likelihood function[14]

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{N} \{y_i \log \pi_i + (1 - y_i) \log(1 - \pi_i)\}. \tag{2}$$

2.1 | Penalized regression

Ridge logistic regression is a penalized regression technique that shrinks the regression coefficients by implementing a penalty on their size.[15] Ridge regression estimates the regression coefficients $\boldsymbol{\beta}$ by subtracting the L2-penalty to the likelihood criterion to be maximized:

$$L(\boldsymbol{\beta}) - \lambda_R \sum_{j=1}^{p} \beta_j^2. \tag{3}$$

The log-likelihood is penalized proportionally to the sum of squared predictor coefficients (L2-penalty). In this equation, λ_R is a hyperparameter which determines the amount of shrinkage of coefficients; higher values of λ_R result in more shrinkage. The estimated coefficients $\boldsymbol{\beta}$ are shrunk towards zero but due to the squared penalty will not reach exactly zero.

Lasso logistic regression is a penalization technique similar to Ridge, which penalizes the log-likelihood proportionally to the sum of the absolute value of the regression coefficients (L1-penalty):[11]

$$L(\boldsymbol{\beta}) - \lambda_L \sum_{j=1}^{p} |\beta_j|. \tag{4}$$

The hyperparameter λ_L determines the amount of shrinkage of coefficients. By making use of the L1-penalty, which is non-differentiable at zero, regression coefficients can attain exactly zero. Hence, Lasso performs regression shrinkage as well as variable selection.

Elastic Net regression combines the Ridge and Lasso penalties, maximizing the following function:[12]

$$L(\boldsymbol{\beta}) - \lambda_{EN} \sum_{j=1}^{p} \left\{ \frac{1}{2}(1-\alpha)\beta_j^2 + \alpha|\beta_j| \right\}. \tag{5}$$

Elastic Net includes α as an additional hyperparameter, which functions as a penalty weight taking values between 0 and 1. Ridge and Lasso regression are special cases of Elastic Net, with $\alpha = 0$ and $\alpha = 1$, respectively.

2.2 | Random Forest

Random Forest, as introduced by Breiman,[13] is an ensemble technique that aggregates multiple decision trees to obtain a prediction. The individual decision trees are trained on different parts of the training set using the technique of bagging, which aims to decrease the variance of the aggregated model. Additional randomization of the decision trees is determined by the hyperparameters of the Random Forest algorithm.[16] Selecting a random subset of candidate variables at each split, the number of which is determined by the *mtry* hyperparameter, adds to the randomization of the trees. The criterium for splitting nodes, defined by the *splitrule* hyperparameter, can be set to selecting the cut point of the randomly selected candidate variables that minimizes the Gini impurity index[13] (*gini*) or selecting the cut point at random[17] (*extratrees*). Other hyperparameters include the minimum node size (ie, minimum number of observations in a terminal node), for which smaller values result in trees with more depth, and total number of trees. Hyperparameters are tuned to yield an optimal tradeoff in terms of achieving less correlated trees whilst maintaining reasonable strength of the trees. To obtain a prediction for a binary outcome, aggregation of a number T of decision trees is performed by a majority vote of the predictions based on the individual trees:[18]

$$\hat{p}_i = \frac{1}{T} \sum_{t=1}^{T} I(\hat{y}_{it} = 1), \tag{6}$$

where $\hat{y}_{it}$ denotes the predicted value based on tree t and $\hat{p}_i$ stands for the estimated probability of an event occurring for a patient i.

2.3 | Hyperparameter tuning procedures

The procedures to tune hyperparameters are often based on resampling techniques such as CV or bootstrap. In a K-fold CV approach, the training set is randomly partitioned into k parts, where $k-1$ parts are used for model training and the remaining kth part for validation. Training and validation is iterated until all k parts have been used for validation once, and the average of the performance estimates over k iterations is taken. An alternative CV-based approach used for the purpose of tuning that we also consider in this article is repeated K-fold CV. In repeated K-fold CV, the random partitioning into k parts as described above is repeated R times. *Tuning is performed by iterating the complete (non-repeated or repeated) K-fold CV process for different values of the hyperparameter(s) as defined in a prespecified hyperparameter*

TABLE 1 Full factorial simulation design: simulation factors.

Simulation factor	Factor levels
Sample size	Minimum required sample size for an expected shrinkage of 0.9:[21] n_m Half the minimum required sample size: $\frac{1}{2} \times n_m$ Twice the minimum required sample size: $2 \times n_m$
Events fraction	0.1, 0.3, 0.5

TABLE 2 Sample sizes of the development datasets for each scenario.

			Sensitivity analysis
Sample size criterium[a]	**Events fraction**	**Sample size values for 8 predictors**	**Sample size values for 16 predictors**
Minimum required sample size for an expected shrinkage of 0.9: n_m	0.1	917	1834
	0.3	408	815
	0.5	385	696
Half the minimum required sample size: $\frac{1}{2} \times n_m$	0.1	459	917
	0.3	204	408
	0.5	193	348
Twice the minimum required sample size: $2 \times n_m$	0.1	1834	3668
	0.3	816	1630
	0.5	770	1392

[a] Sample size values n_m were calculated based on the number of predictors and events fraction using a method proposed by Riley et al[21]

grid. Tuning is usually based on a grid search of a pre-defined set of possible values, selecting the value that optimizes a predictive performance criterion (eg, minimum deviance) in the validation sets. We additionally consider the popular one-standard-error rule for tuning applied to non-repeated and repeated CV, where the hyperparameter is selected that results in the simplest model within one standard error from the aforementioned model that optimizes the predictive performance criterion.[6] This standard error is calculated over the CV iterations.[19] Finally, we consider the default bootstrap as implemented in the commonly used *caret* package for model tuning.[20]

3 | SIMULATION METHODS

3.1 | Simulation setup

A factorial simulation design was carried out, varying the sample size and events fraction. The factor levels are provided in Table 1. The sample sizes were calculated based on Riley's[21] criteria, where the number of predictors and events fraction were used to obtain an expected shrinkage factor of 0.9 in an unpenalized logistic regression. Sample size values of the development datasets for each scenario are detailed in Table 2. The number of predictors was set to 8 predictors and predictor effects were distributed as follows: $\frac{1}{4}$ noise (predictor effects equal to zero), $\frac{1}{2}$ relatively weak (predictor effects equal to each other) and $\frac{1}{4}$ relatively strong (predictor effects set to three times the weaker effect). A total of 9 unique simulation scenarios (3×3) were executed in the main analysis, with 500 simulation runs per scenario.

For each simulation run, a development dataset was generated according to the simulation conditions (see Table 1). For each (hypothetical) individual i, the predictor values ($\boldsymbol{x}_i$) were generated by sampling from a standard multivariate normal distribution with equal pairwise correlations of 0.2. Subsequently, the binary outcome (y_i) was generated by sampling from a Bernoulli distribution conditional on the individuals predictor values and true predictors' effects. The intercept and predictor effects were set to ensure an (approximate) targeted discriminative performance (ie, AUC) of 0.75

TABLE 3 Prediction methods and corresponding hyperparameter(s) that were tuned.

Method	Hyperparameter(s)	Tuning range
Ridge logistic regression	λ_R	0–2 (201 equidistant values)
Lasso logistic regression	λ_L	0–2 (201 equidistant values)
Elastic Net	α	0–1 (11 equidistant values)
	λ_{EN}	0–2 (201 equidistant values)
Random Forest	min.node.size	1–10 (10 equidistant values)
	mtry	1–#predictors (#predictors equidistant values)
	splitrule	extratrees (fixed)

Note: min.node.size = minimum number of observations in a terminal node, mtry = the number of randomly drawn candidate variables at each split.

and the targeted events fraction (see Table 1) using large sample approximations (details in Appendix A). Four prediction models were estimated on the generated development data. The tuning procedures described in Section 3.2 were used to tune the prediction models considered in this study. For each simulation scenario, one independent validation dataset with a sample size of $N = 100\,000$ was generated using the same data-generating approach. The performance of the prediction models was evaluated on the corresponding validation dataset as described in Section 3.3.

In a sensitivity analysis, restricted to the penalized regression models due to computation time, the effect of the number of predictors was studied. The number of predictors was varied (factor levels: 8 and 16 predictors) and fully crossed with the simulation factors of sample size and events fraction, yielding an additional 9 unique simulation scenarios for the penalized regression models.

3.2 | Hyperparameter tuning settings

The following procedures for hyperparameter tuning were examined: 1) 5-fold CV, 2) 10-fold CV, 3) repeated 5-fold CV, 4) repeated 10-fold CV and 5) bootstrap. With regard to the repeated K-fold CV procedures, 5-fold CV was repeated 20 times (ie, 20 × 5-fold CV) and 10-fold CV was repeated 10 times (ie, 10 × 10-fold CV) to obtain the same number of splits. The bootstrap technique was applied using 500 bootstrap samples. For all tuning procedures (including bootstrap), selection of the hyperparameter value(s) was based on minimization of the deviance of the model. For the four CV-based tuning procedures, we also performed the analysis applying the one-standard-error rule for tuning as a selection criterion.[6] CV-based tuning in which the deviance is minimized is referred to hereafter as "standard" CV, whereas alternatively the indication "1SE" CV is used when the one-standard-error rule for tuning is applied.

For all model developments, (simple) grid search was used for tuning, where each combination of hyperparameter values restricted to a prespecified range was considered. The hyperparameters that were tuned for the different prediction methods and their tuning range are summarized in Table 3. For Random Forest, based on work by Probst, Boulesteix, and Bischl,[4] in this article the focus was on tuning *mtry* (ie, the number of randomly drawn candidate variables at each split) and minimum node size. The criterium for splitting nodes, *splitrule*, was set to splitting nodes by selecting, from all cut points of the randomly selected candidate variables, the cut point at random (*extratrees*).[17] The number of trees hyperparameter was kept fixed at 500, since the potential gain in predictive performance achieved by using more trees is usually small.[18]

3.3 | Predictive performance measures

To evaluate the out-of-sample predictive performance of the models, several performance measures were considered. Discriminative ability was assessed using the c-statistic. The c-statistic represents the probability that a randomly selected case with the event got assigned a higher risk score than a case without the event.[22] Higher values of the c-statistic indicate better discriminative performance.

The calibration slope (CS) and calibration in the large (CIL) were used to assess model calibration performance. The CS reflects the spread of the risk predictions and has a target value of 1; a slope < 1 indicates that risk predictions are too extreme, while a slope > 1 indicates that risk predictions are too narrowly distributed.[23] CIL is calculated as the average of the differences between the predicted risk and the overall event rate and has a target value of 0; CIL < 0 indicates systematic risk overestimation, while CIL > 0 indicates underestimation.[23] In addition, the square root of the mean squared distance from the target value of the log(slope) across all simulation runs was evaluated (RMSD):[24]

$$RMSD = \sqrt{\frac{\sum_{i=1}^{N_s}(\log(1) - \log(slope_i))^2}{N_s}}, \tag{7}$$

where N_s represents the number of simulation runs. Previous studies have shown that the performance of shrinkage methods is often variable over samples.[24] The RMSD measure takes into account both bias (ie, the slope's deviation from its target value of 1 on average) and variability.

Overall predictive performance was assessed using the square root of the mean squared prediction error (rMSPE) and mean absolute prediction error (MAPE). For the rMSPE and MAPE, prediction error was defined as the difference between the estimated probability based on the models under consideration and the true probability based on the data-generating model. Lower values indicate better overall predictive performance (lower prediction error).

3.4 | Software and estimation error handling

All simulations and analyses were performed using R (version 4.0.3)[25] on a high-performance computing facility. The following packages were used: the `caret`[20] package (version 6.0-86) for hyperparameter tuning and model fitting, implementing the `glmnet`[26] package (version 4.0-2) for estimating the Elastic Net, Ridge and Lasso regression models and the `ranger`[27] package (version 0.12.1) for estimating the Random Forest model. The (handling of) estimation errors are summarized in the supplementary materials (Table S1).

4 | SIMULATION RESULTS

In this section we will focus on simulation scenarios with an events fraction of 0.3. Similar result patterns were obtained for scenarios with events fractions of 0.1 and 0.5, which are described in detail in the supplementary materials. Result patterns for rMSPE were similar to MAPE and can therefore be found in the supplements. There were relatively few cases where the selected λ_R, λ_L and λ_{EN} were equal to the upper bound value of the prespecified hyperparameter grid (0.16%, 0.12%, and 0.14% for Ridge, Lasso and Elastic Net, respectively), which indicates sufficient coverage of the hyperparameter grid range.

Results for the various predictive performance outcomes are summarized in Table 4. The differences between tuning procedures in the predictive performance outcomes were generally consistent across the various models. Differences between tuning procedures were generally small in terms of the average c-statistic, except for 1SE CV resulting in slightly lower c-statistic values on average in case of Lasso regression and Random Forest (Figure 1, Table 4). In addition, CIL outcomes differed little between tuning procedures; all values approached the target value of zero, suggesting no systematic over- or underestimation of the predicted risks (Table 4).

We did observe substantial differences between tuning procedures in median calibration slopes (Figure 2, Table 4). The 1SE CV tuning procedures generally resulted in median calibration slopes well above unity for both non-repeated and repeated CV (both 5-fold and 10-fold), indicating too narrowly dispersed risk predictions (underfitting). The impact of the 1SE rule for tuning was particularly apparent in case of non-repeated K-fold CV for the penalized regression models, showing median calibration slopes far above unity. The 1SE CV tuning procedure also yielded median calibration slopes relatively far above unity for the Random Forest model (with events fractions of 0.3 and 0.5), although this effect was considerably smaller than for the penalized regression models and showed slightly less variance. Median calibration slopes differed little between standard CV and 1SE CV tuning for Random Forest in scenarios with a very low events fraction (value of 0.1) (Figure S7). Although to a lesser extent than 1SE CV tuning, bootstrap for tuning also resulted in median calibration slopes above unity (underfitting) across all models (Figure 2, Table 4). For all models, standard non-repeated and repeated CV for tuning (both 5-fold and 10-fold) performed similarly well considering the small difference between median calibration slopes. These tuning procedures appeared to outperform the other tuning procedures, showing median calibration slopes closest to unity (slightly underfitting).

TABLE 4 Results predictive performance outcomes of tuning procedures per model (restricted to scenario with number of predictors = 8, events fraction = 0.3 and sample size = n_m).

		c-statistic		CIL		CS		rMSPE		MAPE	
Model	**Tuning procedure**	**Mean**	**[SE]**	**Mean**	**[SE]**	**Median**	**[IQR]**	**Mean**	**[SE]**	**Mean**	**[SE]**
Ridge LR	5-fold CV	0.742	[0.000]	−0.003	[0.001]	1.112	[0.226]	0.061	[0.001]	0.047	[0.001]
	1SE 5-fold CV	0.742	[0.000]	−0.003	[0.001]	2.047	[0.821]	0.100	[0.001]	0.081	[0.001]
	10-fold CV	0.742	[0.000]	−0.003	[0.001]	1.099	[0.229]	0.060	[0.001]	0.047	[0.000]
	1SE 10-fold CV	0.742	[0.000]	−0.003	[0.001]	2.054	[0.644]	0.101	[0.001]	0.082	[0.001]
	repeated 5-fold CV	0.742	[0.000]	−0.003	[0.001]	1.120	[0.235]	0.060	[0.001]	0.047	[0.000]
	1SE repeated 5-fold CV	0.743	[0.000]	−0.003	[0.001]	1.410	[0.314]	0.071	[0.001]	0.057	[0.001]
	repeated 10-fold CV	0.742	[0.000]	−0.003	[0.001]	1.104	[0.239]	0.060	[0.001]	0.047	[0.000]
	1SE repeated 10-fold CV	0.743	[0.000]	−0.003	[0.001]	1.485	[0.333]	0.075	[0.001]	0.060	[0.001]
	Bootstrap	0.742	[0.000]	−0.003	[0.001]	1.203	[0.259]	0.063	[0.001]	0.049	[0.001]
Lasso LR	5-fold CV	0.741	[0.000]	−0.003	[0.001]	1.047	[0.225]	0.061	[0.001]	0.047	[0.001]
	1SE 5-fold CV	0.734	[0.000]	−0.003	[0.001]	1.659	[0.552]	0.096	[0.001]	0.077	[0.001]
	10-fold CV	0.741	[0.000]	−0.003	[0.001]	1.041	[0.218]	0.061	[0.001]	0.047	[0.000]
	1SE 10-fold CV	0.734	[0.000]	−0.003	[0.001]	1.723	[0.497]	0.097	[0.001]	0.078	[0.001]
	repeated 5-fold CV	0.741	[0.000]	−0.003	[0.001]	1.044	[0.223]	0.061	[0.001]	0.047	[0.000]
	1SE repeated 5-fold CV	0.740	[0.000]	−0.003	[0.001]	1.243	[0.286]	0.069	[0.001]	0.054	[0.001]
	repeated 10-fold CV	0.741	[0.000]	−0.003	[0.001]	1.034	[0.208]	0.061	[0.001]	0.047	[0.000]
	1SE repeated 10-fold CV	0.739	[0.000]	−0.003	[0.001]	1.297	[0.280]	0.072	[0.001]	0.057	[0.001]
	Bootstrap	0.740	[0.000]	−0.003	[0.001]	1.145	[0.226]	0.064	[0.001]	0.050	[0.000]

TABLE 4 (Continued)

		c-statistic		CIL		CS		rMSPE		MAPE	
Model	**Tuning procedure**	**Mean**	**[SE]**	**Mean**	**[SE]**	**Median**	**[IQR]**	**Mean**	**[SE]**	**Mean**	**[SE]**
Elastic Net	5-fold CV	0.741	[0.000]	-0.003	[0.001]	1.105	[0.232]	0.062	[0.001]	0.048	[0.001]
	1SE 5-fold CV	0.742	[0.000]	−0.003	[0.001]	1.940	[0.701]	0.098	[0.001]	0.079	[0.001]
	10-fold CV	0.741	[0.000]	−0.003	[0.001]	1.099	[0.228]	0.061	[0.001]	0.047	[0.000]
	1SE 10-fold CV	0.742	[0.000]	−0.003	[0.001]	2.012	[0.611]	0.100	[0.001]	0.081	[0.001]
	repeated 5-fold CV	0.741	[0.000]	−0.003	[0.001]	1.115	[0.241]	0.062	[0.001]	0.048	[0.000]
	1SE repeated 5-fold CV	0.742	[0.000]	−0.003	[0.001]	1.381	[0.338]	0.070	[0.001]	0.056	[0.001]
	repeated 10-fold CV	0.741	[0.000]	−0.003	[0.001]	1.102	[0.230]	0.061	[0.001]	0.047	[0.000]
	1SE repeated 10-fold CV	0.742	[0.000]	−0.003	[0.001]	1.451	[0.341]	0.074	[0.001]	0.059	[0.001]
	Bootstrap	0.741	[0.000]	−0.003	[0.001]	1.195	[0.264]	0.064	[0.001]	0.050	[0.001]
Random Forest	5-fold CV	0.725	[0.000]	0.003	[0.001]	1.141	[0.248]	0.092	[0.001]	0.072	[0.000]
	1SE 5-fold CV	0.722	[0.000]	−0.004	[0.001]	1.285	[0.161]	0.099	[0.000]	0.077	[0.000]
	10-fold CV	0.725	[0.000]	0.003	[0.001]	1.148	[0.252]	0.092	[0.001]	0.072	[0.000]
	1SE 10-fold CV	0.722	[0.000]	−0.005	[0.001]	1.289	[0.146]	0.099	[0.000]	0.077	[0.000]
	repeated 5-fold CV	0.727	[0.000]	0.003	[0.001]	1.180	[0.244]	0.090	[0.001]	0.070	[0.000]
	1SE repeated 5-fold CV	0.724	[0.000]	0.000	[0.001]	1.191	[0.261]	0.095	[0.001]	0.074	[0.000]
	repeated 10-fold CV	0.726	[0.000]	0.003	[0.001]	1.176	[0.241]	0.090	[0.001]	0.070	[0.000]
	1SE repeated 10-fold CV	0.723	[0.000]	−0.002	[0.001]	1.233	[0.279]	0.097	[0.001]	0.075	[0.000]
	Bootstrap	0.728	[0.000]	0.002	[0.001]	1.250	[0.243]	0.090	[0.001]	0.070	[0.000]

Abbreviations: CIL, calibration in the large; CS, calibration slope; CV, cross-validation; IQR, interquartile range; LR, logistic regression; MAPE, mean absolute prediction error; rMSPE, square root of the mean squared prediction error; SE, empirical standard error; 1SE, one-standard-error.

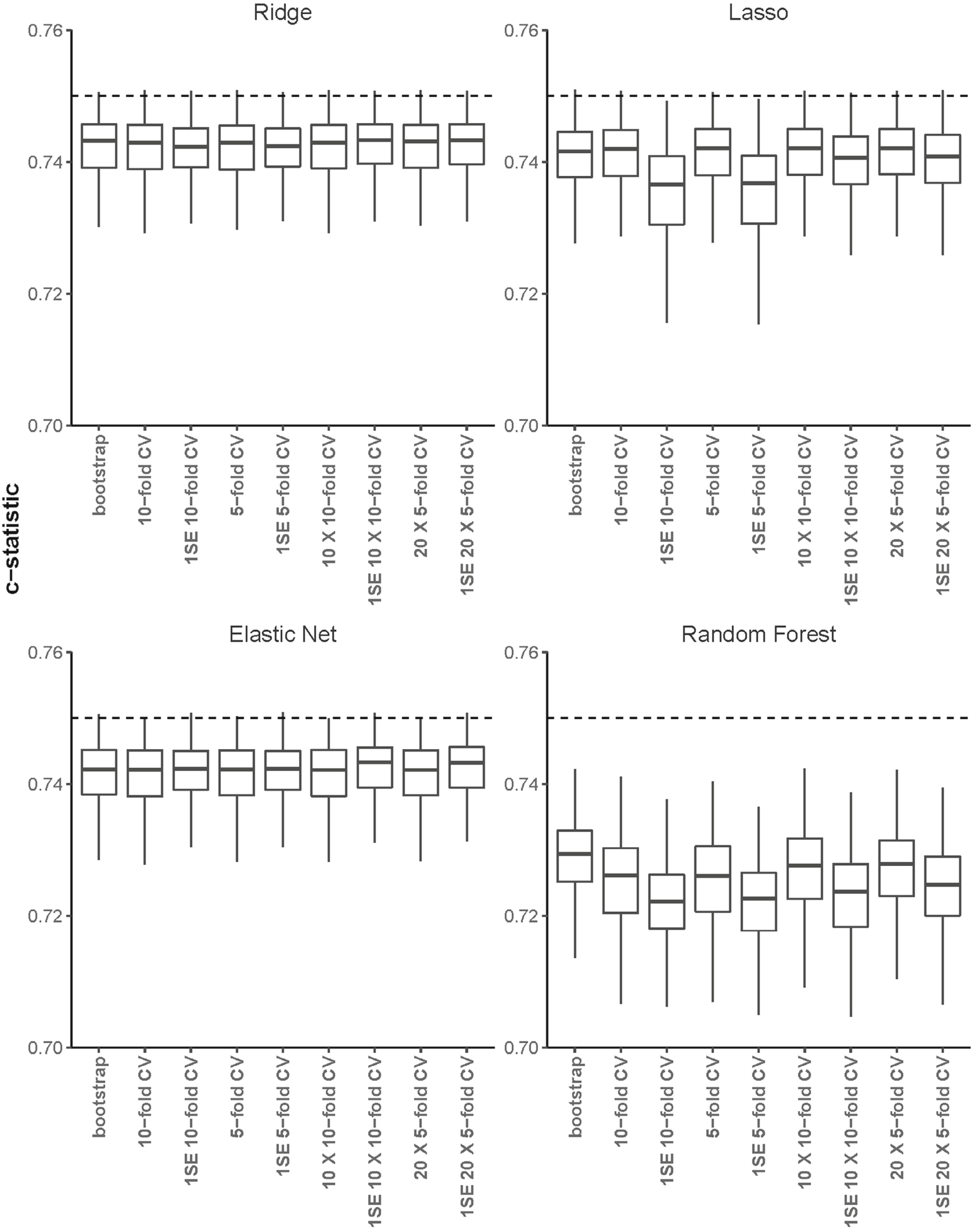

FIGURE 1 Box plots of the c-statistic (c-statistic of the data-generating model indicated by the dotted line) over the 500 simulation runs corresponding to various tuning procedures for each model (Ridge, Lasso, Elastic Net and Random Forest). Restricted to scenario with number of predictors = 8, events fraction = 0.3 and sample size = n_m. The whiskers have a maximum value of 1.5 times the interquartile range. CV, cross-validation; 1SE, one-standard-error.

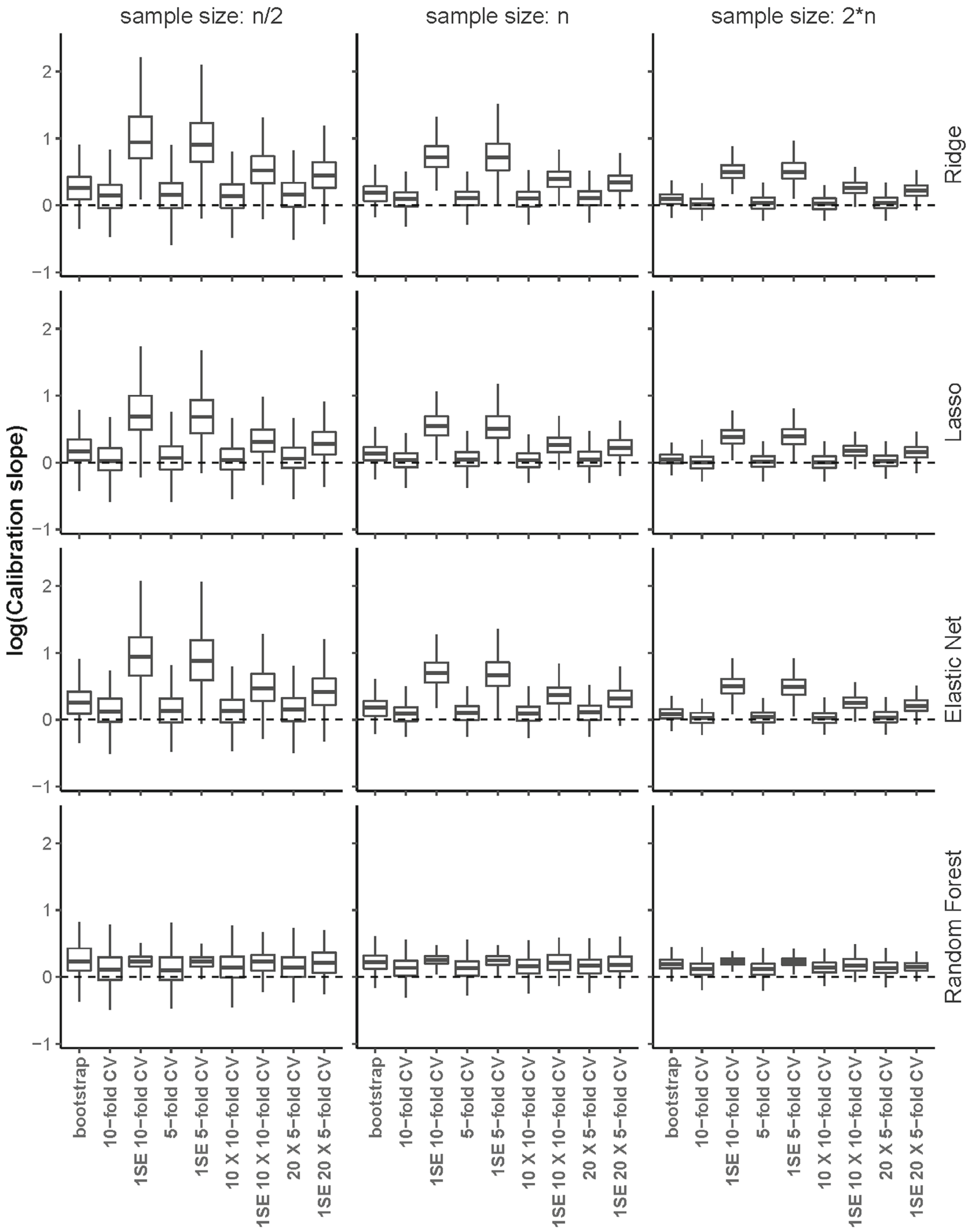

FIGURE 2 Box plots of the log(calibration slopes) (on a logarithmic scale a target value of 0 indicated by the dotted line) over the 500 simulation runs corresponding to various tuning procedures for each model (Ridge, Lasso, Elastic Net and Random Forest). Restricted to scenario with number of predictors = 8 and events fraction = 0.3. The whiskers have a maximum value of 1.5 times the interquartile range. CV, cross-validation; 1SE, one-standard-error.

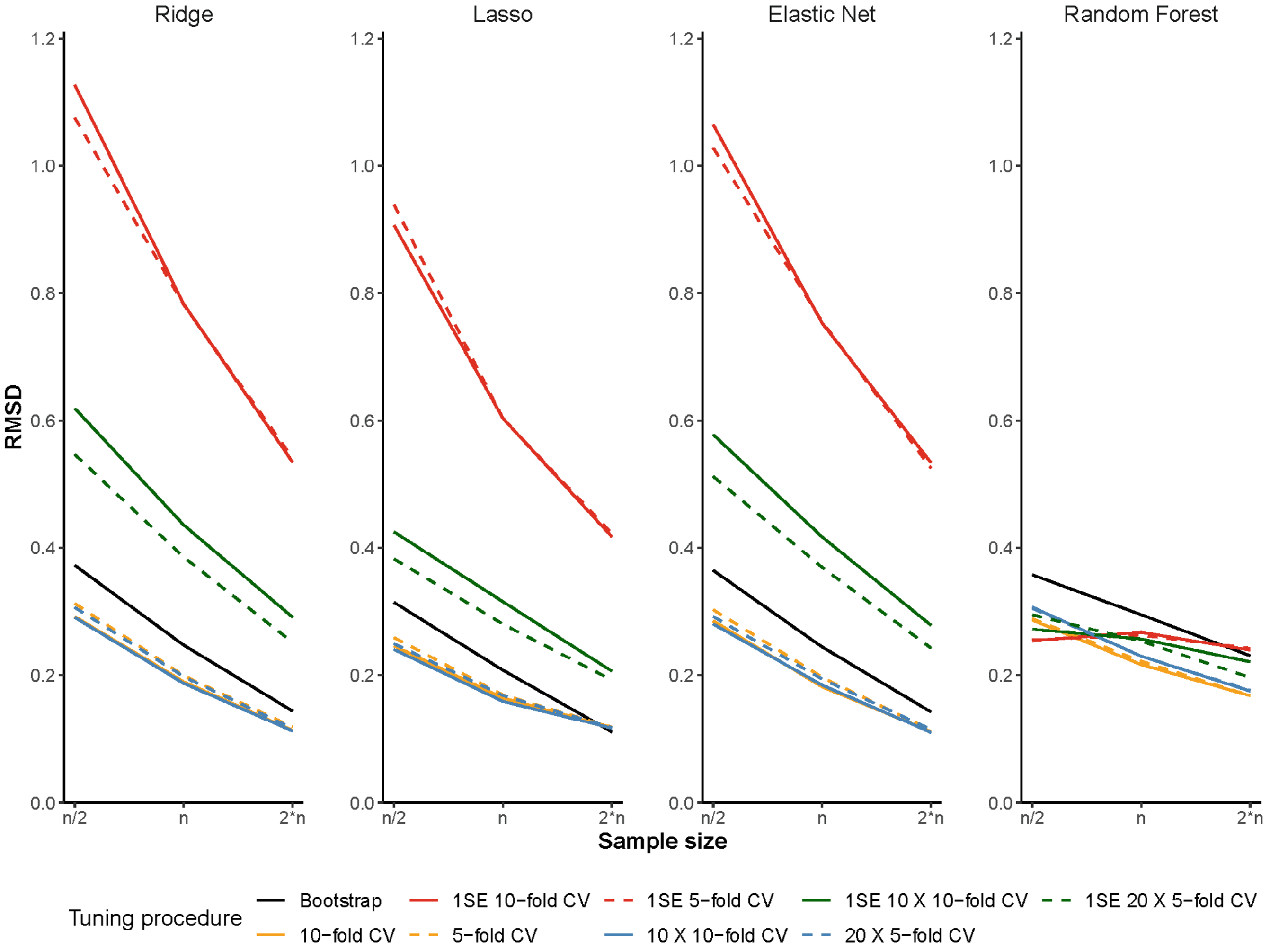

FIGURE 3 Root-mean-squared distance (RMSD) of the calibration slope over the 500 simulation runs corresponding to various tuning procedures for each model (Ridge, Lasso, Elastic Net and Random Forest). Restricted to scenario with number of predictors = 8 and events fraction = 0.3. CV, cross-validation; 1SE, one-standard-error.

RMSD was lowest for standard non-repeated and repeated CV for tuning (both 5-fold and 10-fold) (Figure 3), indicating better calibration performance. Bootstrap for tuning performed slightly worse compared to these tuning procedures, showing slightly higher RMSD. RMSD values were far higher for 1SE CV tuning as compared to other tuning procedures in case of the penalized regression models, but not for Random Forest. This was observed for repeated CV and even more so for non-repeated CV.

MAPE and rMSPE outcomes also differed between tuning procedures, mostly showing a similar performance pattern as RMSD. An exception was bootstrap-based tuning for Random Forest, which performed worst in terms of RMSD (highest) but best in terms of MAPE (lowest), although differences between tuning procedures for Random Forest were modest overall (Figures 3 and 4). MAPE and rMSPE did not differ substantially between standard non-repeated and repeated CV (for both 5-fold and 10-fold), for which MAPE and rMSPE were lowest in case of penalized regression (Figure 4, Figure S26). Bootstrap showed slightly higher MAPE and rMSPE than the standard CV-based tuning procedures for the penalized regression models. 1SE CV tuning consistently showed higher MAPE and rMSPE values than other tuning procedures, for repeated CV and even more so for non-repeated CV. This effect of the 1SE CV tuning procedure was observed for all models, although particularly apparent for the penalized regression models and to a lesser extent for Random Forest.

The result patterns for all predictive performance outcomes were generally found to be similar across simulation scenarios (ie, for situations with varying data characteristics), although differences between tuning procedures decreased with an increasing sample size and with events fractions increasing from 0.1 to 0.3 (differences between events fractions

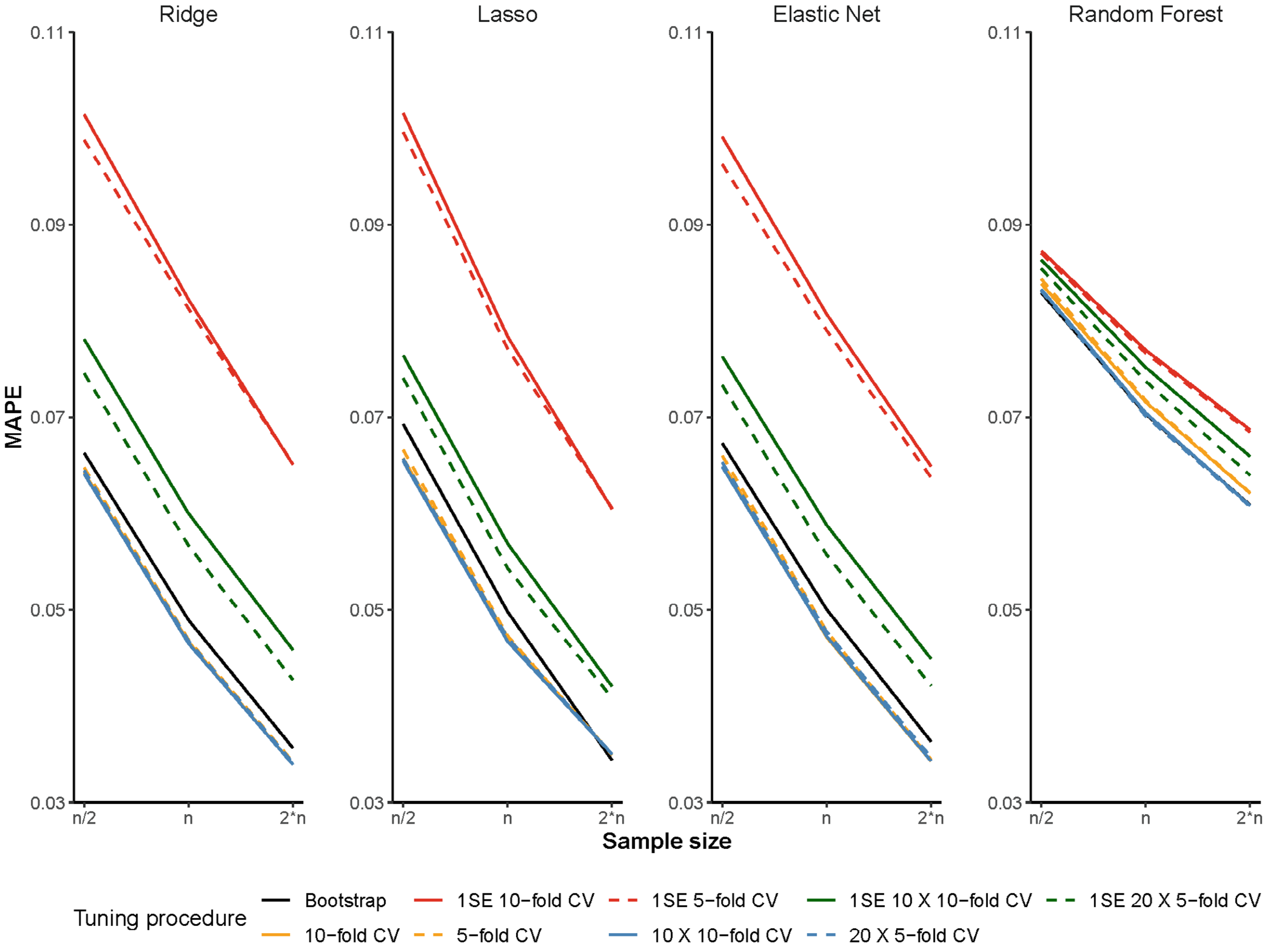

FIGURE 4 Mean absolute prediction error (MAPE) over the 500 simulation runs corresponding to various tuning procedures for each model (Ridge, Lasso, Elastic Net and Random Forest). Restricted to scenario with number of predictors = 8 and events fraction = 0.3. CV, cross-validation; 1SE, one-standard-error.

of 0.3 and 0.5 were generally small) (Figures 2–4, supplementary materials). This means that the impact of the tuning procedure used lessened with larger sample sizes and, to a certain level, a higher events fraction.

In a sensitivity analysis, the effect of number of predictors on the predictive performance of the tuning procedures was examined for the penalized regression models. Similar performance patterns were observed for all outcomes for scenarios with 16 predictors as compared to 8 predictors, although differences between tuning procedures decreased with a higher number of predictors (see supplementary materials).

5 | CASE STUDY: OVARIAN CANCER DATASET

A case study was performed in which hyperparameter tuning procedures and prediction methods were applied to clinical data collected by the International Ovarian Tumor Analysis (IOTA) group between 1999 and 2012.[28,29] The data included 5,914 women with at least one adnexal mass considered to require surgery as judged by a clinician and the outcome was whether an ovarian mass was benign (n = 3983; 67,3%) or malignant (n = 1931; 32.7%). The objective was to develop a clinical prediction model that estimated the probability of the presence of ovarian mass malignancy based on the following preoperatively measured predictors: age (continuous, in years), maximum diameter of the lesion (continuous, in mm) and number of papillary structures (ordinal, zero/one/two/three/four or more papillary structures).

Hyperparameter tuning and model estimation was based on a development set of n = 338 (minimum required sample size for an expected shrinkage of 0.9[21]) generated by drawing a random sample from the IOTA dataset. The predictive

TABLE 5 Case study results based on an ovarian cancer dataset.

		c-statistic	CIL	CS
Model	**Tuning procedure**	**Estimate [CI][a]**	**Estimate [CI]**	**Estimate [CI]**
Ridge LR	5-fold CV	0.779 [0.766–0.791]	0.000 [−0.012–0.011]	1.257 [1.170–1.343]
	1SE 5-fold CV	0.779 [0.767–0.792]	−0.005 [−0.017–0.006]	2.333 [2.173–2.494]
	10-fold CV	0.779 [0.766–0.792]	−0.002 [−0.013–0.009]	1.445 [1.346–1.544]
	1SE 10-fold CV	0.779 [0.766–0.792]	−0.005 [−0.016–0.006]	2.211 [2.060–2.363]
	Bootstrap	0.779 [0.766–0.791]	−0.001 [−0.012–0.010]	1.305 [1.215– 1.394]
Lasso LR	5-fold CV	0.778 [0.765–0.791]	−0.002 [−0.014–0.009]	1.549 [1.444–1.654]
	1SE 5-fold CV	0.778 [0.765–0.791]	−0.002 [−0.014–0.009]	1.549 [1.444–1.654]
	10-fold CV	0.779 [0.766–0.792]	−0.001 [−0.012–0.010]	1.364 [1.271–1.457]
	1SE 10-fold CV	0.778 [0.765–0.791]	−0.002 [−0.014–0.009]	1.549 [1.444–1.654]
	Bootstrap	0.781 [0.768–0.794]	0.001 [−0.010–0.012]	1.069 [0.996–1.141]
Elastic Net	5-fold CV	0.779 [0.767–0.792]	−0.002 [−0.013–0.009]	1.461 [1.361–1.560]
	1SE 5-fold CV	0.779 [0.766–0.792]	−0.005 [−0.016–0.007]	2.088 [1.945–2.232]
	10-fold CV	0.781 [0.769–0.794]	0.000 [−0.011–0.011]	1.224 [1.141– 1.307]
	1SE 10-fold CV	0.779 [0.767–0.792]	−0.006 [−0.017–0.006]	2.454 [2.286–2.623]
	Bootstrap	0.780 [0.767–0.792]	0.000 [−0.011–0.011]	1.180 [1.100–1.261]
Random Forest	5-fold CV	0.768 [0.756–0.781]	−0.013 [−0.024–0.002]	0.773 [0.719–0.827]
	1SE 5-fold CV	0.777 [0.765–0.790]	−0.011 [−0.022–0.000]	1.317 [1.229–1.406]
	10-fold CV	0.766 [0.753–0.778]	−0.011 [−0.022–0.001]	0.700 [0.650–0.749]
	1SE 10-fold CV	0.780 [0.767–0.792]	−0.011 [−0.022–0.000]	1.346 [1.256–1.436]
	Bootstrap	0.784 [0.772–0.796]	−0.009 [−0.020–0.002]	1.402 [1.311– 1.492]

Abbreviations: CI, 95% confidence interval; CS, calibration slope; CV, cross-validation; LR, logistic regression; 1SE, one-standard-error.

[a] CI based on 500 bootstrap replicates.

performance of the hyperparameter tuning procedures were subsequently evaluated using a validation set which comprised of the IOTA data not selected for the development set. The out-of-sample predictive performance of the models was evaluated based on the c-statistic, calibration slope and calibration in the large. The hyperparameter tuning procedures that were examined and their settings were the same in this case study as for the simulation studies (see Section 3.2). The hyperparameters that were tuned for the different prediction methods and their tuning range are summarized in Table 3.

Results generally showed little variation between tuning procedures in terms of the c-statistic, indicating similar discriminative performance (Table 5). CIL outcomes also differed little between tuning procedures; all values approached the target value of zero. The 1SE CV tuning procedure consistently resulted in calibration slopes relatively far above unity in the IOTA dataset, particularly in case of Ridge regression and Elastic Net (up to 2.333 and 2.454, respectively). While standard 5-fold and 10-fold CV showed calibration slopes above unity for the penalized regression models (although to a lesser extent than the 1SE CV tuning procedure), these resulted in calibration slopes below unity for Random Forest (minimum = 0.700). Calibration slopes were closest to unity for bootstrap for tuning in case of Lasso regression and Elastic Net in this dataset. This case study is in accordance with our simulation study results, showing that different tuning procedures can lead to important differences in the predictive performance of clinical prediction models, in particular with regard to their calibration.

6 | DISCUSSION

We studied the impact of various hyperparameter tuning procedures for penalized regression and Random Forest on the predictive performance of clinical prediction models. The results show that calibration performance and prediction errors can differ substantially between tuning procedures, while the effect on discriminative performance is generally much smaller. Notably, the 1SE CV tuning procedure generally resulted in poor calibration performance of the prediction model, showing signs of severe underfitting. Standard non-repeated and repeated CV outperformed the other tuning procedures in terms of calibration and prediction error, and were found to perform similarly (both 5-fold and 10-fold). Using bootstrap for tuning resulted in slightly worse calibration performance on average compared to the standard CV-based tuning procedures. These results were consistent across the penalized regression and tree-based models, where the impact of the 1SE CV rule for tuning was particularly apparent in case of penalized regression.

Differences between tuning procedures for Random Forest were generally smaller than for Ridge, Lasso and Elastic Net regression. This observation is consistent with earlier studies indicating that Random Forest is relatively easy to tune.[13] However, it should be noted that the differences between tuning procedures for Random Forest can still be substantial. In some simulation scenarios, especially those with a small sample size, the use of the 1SE rule for tuning can lead to severe miscalibration.

The 1SE CV tuning procedure showed both poor average performance and high variability in terms of calibration (median CS above unity and relatively high RMSD) for the penalized regression models. This was particularly the case for non-repeated CV, possibly because of less stable (more extreme) estimates relative to repeated CV. In addition, overall prediction error was highest for this tuning procedure (highest average MAPE and rMSPE). Although the 1SE rule for tuning can be expected to result in some degree of underfitting of the prediction model as this approach targets a more parsimonious model, the magnitude of the observed underfitting by the 1SE rule and the detrimental consequences for model calibration were often substantial. Noticeable is that while Breiman et al[6] and Hastie et al[7] have advised the use of 1SE CV for tuning, our findings showed this tuning procedure resulted in too narrowly dispersed risk predictions (underfitting). This is possibly related to the focus on low-dimensional data in our study and shows that high-dimensional data might be a precondition for the effective use of 1SE CV for tuning. Another explanation is that calibration of prediction models is often not evaluated, although considered important for clinical risk prediction,[23] and hence the effects of tuning procedures may so far have been largely unnoticed.

Overall, calibration performance was best for standard non-repeated and repeated CV as compared to the other tuning procedures, both in terms of average performance and variability (median CS close to unity and lowest RMSD). Moreover, for these tuning procedures, the overall prediction error was lowest (lowest average MAPE and rMSPE). Bootstrap showed slightly higher prediction error (higher average MAPE and rMSPE) and poorer calibration performance (median calibration slope above unity and higher RMSD) as compared to the standard CV-based tuning procedures. This is in agreement with earlier studies related to support vector machine models,[19,30] demonstrating optimal hyperparameter tuning for standard K-fold CV (performing slightly better than bootstrap). The impact of the tuning procedure used became less apparent with larger sample sizes, a higher number of predictors and, to a certain level, a higher events fraction.

This study's results have the following implications. Although different tuning procedures perform similarly in terms of discriminative performance, calibration performance can differ substantially between tuning procedures. For development of (low-dimensional) clinical prediction models, we generally advise against the use of the 1SE rule for tuning. In general, the standard CV-based approaches perform better than bootstrap, while little difference was observed between non-repeated and repeated CV.

The results of our study can be considered in the broader context of studies on resampling methods for prediction model development and validation. While our study and other work[19,30] indicate that CV may be the preferred approach over bootstrap for hyperparameter tuning, earlier studies[31,32] have suggested the bootstrap as a preferred approach for internal validation in the context of low-dimensional prediction modeling. Other studies have shown that in the context of resampling methods for tuning regularization (ie, shrinkage) parameters in small, low-dimensional data samples, even the preferred methods can perform poorly.[24,33,34] Furthermore, our study shows limited benefit of using repeated over non-repeated CV for hyperparameter tuning, despite an increased computational burden. For internal validation, however, earlier work reports on the gain in performance of repeated CV as compared to non-repeated CV.[31]

This study has some limitations. Firstly, data were generated under a logistic regression model. The simulation results were designed to evaluate tuning procedures within models and do not allow for a fair comparison between modeling procedures. Secondly, our study focused on tuning by grid search, which is the default in most software. Besides the decision of which tuning procedure to use, the model-developer has to define the range of the candidate hyperparameter values for tuning (ie, the grid configurations). In earlier studies the default grid values was found often to be insufficient in low-dimensional settings.[24,35] Hence, in this study we used an extensive grid search, resulting in high computational time. Given the substantial computational efforts required for the extensive grid search, especially in simulation studies, future research may focus on more computationally efficient approaches to hyperparameter tuning searches. Thirdly, this study addressed some of the most commonly used approaches for hyperparameter tuning and focused on resampling procedures for this purpose. There are other approaches for hyperparameter tuning, for instance those that optimize other measures of out-of-sample predictive performance or that avoid resampling. For example, information criteria such as Akaike's Information Criteria[36] can also be used for tuning penalized regression models. Future research into these tuning procedures in low-dimensional data settings could complement the current findings. Another interesting direction for further research is the evaluation of the 1SE CV rule relative to dimensionality of the predictor space focusing on the calibration performance.

In conclusion, the choice of tuning procedure can have a profound influence on the predictive performance of clinical prediction models. The increasingly popular 1SE CV tuning procedure should be used with caution, taking into account the specific data characteristics under consideration. Knowledge of the tuning procedure differences discussed contribute to the optimalization of the usage of clinical prediction models.

DATA AVAILABILITY STATEMENT

The ovarian cancer dataset used in the case study cannot be shared due to privacy and ethical restrictions.

ORCID

Zoë S. Dunias https://orcid.org/0000-0001-9412-0123
Ben Van Calster https://orcid.org/0000-0003-1613-7450
Dirk Timmerman https://orcid.org/0000-0002-3707-6645
Anne-Laure Boulesteix https://orcid.org/0000-0002-2729-0947
Maarten van Smeden https://orcid.org/0000-0002-5529-1541

REFERENCES

1. Van Smeden M, Reitsma JB, Riley RD, Collins GS, Moons KGM. Clinical prediction models: diagnosis versus prognosis. *J Clin Epidemiol*. 2021;132:142-145.
2. Christodoulou E, Ma J, Collins GS, Steyerberg EW, Verbakel JY, Van Calster B. A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *J Clin Epidemiol*. 2019;110:12-22.
3. Ellenbach N, Boulesteix AL, Bischl B, Unger K, Hornung R. Improved outcome prediction across data sources through robust parameter tuning. *J Classif*. 2020;38:212-231.
4. Probst P, Boulesteix AL, Bischl B. Tunability: importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*. 2019;20:1-32.
5. Witten IH, Frank E, Hall MA. *Data Mining: Practical Machine Learning Tools and Techniques*. Massachusetts, USA: Morgan Kaufmann; 2011.
6. Breiman L, Friedman J, Stone CJ, Olshen RA. *Classification and Regression Trees*. Boca Raton: CRC press; 1984.
7. Hastie T, Tibshirani R, Friedman J. *In the Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer; 2009.
8. Léger C, Romano JP. Bootstrap choice of tuning parameters. *Ann Instit Stat Math*. 1990;42:709-735.
9. Fan Y, Tang CY. Tuning parameter selection in high-dimensional penalized likelihood. *J R Stat Soc Ser B*. 2013;75:531-552.
10. Cessie SL, Houwelingen JC. Ridge estimators in logistic regression. *Appl Stat*. 1992;41:191-201.
11. Tibshirani R. Regression shrinkage and selection via the Lasso. *J R Stat Soc B*. 1996;58:267-288.
12. Zou H, Hastie T. Regularization and variable selection via the elastic net. *J R Stat Soc B*. 2005;67:301-320.
13. Breiman L. Random forests. *Mach Learn*. 2001;45:5-32.
14. Agresti A. *Categorical Data Analysis. 2*. New Jersey: John Wiley & Sons, Inc.; 2002.
15. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*. 2000;42:80-86.
16. Probst P, Wright MN, Boulesteix AL. Hyperparameters and tuning strategies for random forest. *WIREs Data Min Knowl Discov*. 2019;9(e1301):1-15.
17. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Mach Learn*. 2006;63:3-42.
18. Probst P, Boulesteix AL. To tune or not to tune the number of trees in random forest. *J Mach Learn Res*. 2018;18:1-18.
19. Anguita D, Boni A, Ridella S, Rivieccio F, Sterpi D. *Theoretical and Practical Model Selection Methods for Support Vector Classifiers*. Berlin: Springer; 2005:177.
20. Kuhn M. Building predictive models in R using the caret package. *J Stat Softw*. 2008;28:1-26.
21. Riley RD, Ensor J, Snell KIE, et al. Calculating the sample size required for developing a clinical prediction model. *BMJ*. 2020;368:1-12.
22. Harrell FE Jr. *Regression Modeling Strategies*. New York: Springer; 2001.
23. Van Calster B, McLernon DJ, Van Smeden M, Wynants L, Steyerberg EW. Calibration: the Achilles heel of predictive analytics. *BMC Med*. 2019;17(230):1-7.
24. Van Calster B, Van Smeden M, De Cock B, Steyerberg EW. Regression shrinkage methods for clinical prediction models do not guarantee improved performance: simulation study. *Stat Methods Med Res*. 2020;29:1-13.

25. Team RC. *A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing; 2020.
26. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*. 2010;33:1-22.
27. Wright MN, Wager S, Probst P. A fast implementation of random forests. 2019: 123-136.
28. Kaijser J, Bourne T, Valentin L, et al. Improving strategies for diagnosing ovarian cancer: a summary of the International Ovarian Tumor Analysis (IOTA) studies. *Ultrasound Obstet Gynecol*. 2013;41:9-20.
29. Van Calster B, Van Hoorde K, Valentin L, et al. Evaluating the risk of ovarian cancer before surgery using the ADNEX model to differentiate between benign, borderline, early and advanced stage invasive, and secondary metastatic tumours: prospective multicentre diagnostic study. *BMJ*. 2014;349:g5920.
30. Duarte E, Wainer J. Empirical comparison of cross-validation and internal metrics for tuning SVM hyperparameters. *Pattern Recognit Lett*. 2017;88:6-11.
31. Steyerberg EW, Harrell FE Jr, Borsboom GJ, Eijkemans MJ, Vergouwe Y, Habbema JD. Internal validation of predictive models: efficiency of some procedures for logistic regression analysis. *J Clin Epidemiol*. 2001;54:774-781.
32. Austin PC, Steyerberg EW. Events per variable (EPV) and the relative performance of different strategies for estimating the out-of-sample validity of logistic regression models. *Stat Methods Med Res*. 2017;26:796-808.
33. Riley RD, Snell KIE, Martin GP, et al. Penalization and shrinkage methods produced unreliable clinical prediction models especially when sample size was small. *J Clin Epidemiol*. 2021;132:88-96.
34. Šinkovec H, Heinze G, Blagus R, Geroldinger A. To tune or not to tune, a case study of ridge logistic regression in small or sparse datasets. *BMC Med Res Methodol*. 2021;21:199.
35. Van Smeden M, Moons KGM, Groot dJAH, et al. Sample size for binary logistic prediction models: Beyond events per variable criteria. *Stat Methods Med Res*. 2019;28:2455-2474.
36. Akaike H. Information theory and an extension of the maximum likelihood principle. In Second International Symposium on Information Theory; 1973; Budapest.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Dunias ZS, Van Calster B, Timmerman D, Boulesteix AL, van Smeden M. A comparison of hyperparameter tuning procedures for clinical prediction models: A simulation study. *Statistics in Medicine*. 2024;43(6):1119-1134. doi: 10.1002/sim.9932

APPENDIX A. PARAMETERS DATA-GENERATING MODEL

Parameters of the data-generating model for simulation of the data were obtained as follows: the intercept and predictor effects were set to ensure an (approximate) targeted discriminative performance (ie, AUC) of 0.75 and the targeted events fraction (see Table 1) using large sample approximations. A vector $\boldsymbol{\beta}$ of regression coefficients and an intercept was estimated for a generated predictor variable dataset ($N = 100\,000$) through an optimization algorithm (using the *optim* function in R (version 4.0.3)[25]) that uses a loss function which minimizes: the sum of (1) the squared difference between the targeted AUC and the observed AUC of the generated data set, and (2) the squared difference between the targeted events fraction and the average predicted probability for an event in the generated dataset. This process was repeated 20 times. In order to obtain reliable estimates, the average over these 20 repetitions was taken. The resulting estimates were validated on an independent validation dataset ($N = 100\,000$).