On Cross-Validation for MLP Model Evaluation

Tommi Kärkkäinen

University of Jyväskylä,
Department of Mathematical Information Technology
tommi.karkkainen@jyu.fi

Abstract. Cross-validation is a popular technique for model selection and evaluation. The purpose is to provide an estimate of generalization error using mean error over test folds. Typical recommendation is to use ten-fold stratified cross-validation in classification problems. In this paper, we perform a set of experiments to explore the characteristics of cross-validation, when dealing with model evaluation of Multilayer Perceptron neural network. We test two variants of stratification, where the nonstandard one takes into account classwise data density in addition to pure class frequency. Based on computational experiments, many common beliefs are challenged and some interesting conclusions drawn.

Keywords: Cross-validation, Multilayer Perceptron, Model Selection.

1 Introduction

Cross-validation (CV) is a popular technique for model selection and evaluation, whose roots go back, at least, to 1960's [1]. The purpose of dividing data into independent training and test sets is to enable estimation of the generalization error of a trained model. This is especially important for the so-called universal approximators, of which Multilayer Perceptron neural network (MLP) is an example [2]. More precisely, with a real data set of input-output samples, a model that can represent unknown functions very accurately is prone to overlearning. Hence, its complexity should be determined using generalization as primary focus instead of training accuracy.

Kohavi [3] is probably the most well-known reference of CV in machine learning. Using six classification benchmarks with over 500 instances from UCI repository, two classification algorithms, C4.5 and naive Bayes, and amount of misclassifications in percentages as error indicator, stratified 10-CV (i.e., with ten folds) was concluded as the recommended approach. This approach has then established itself as a kind of community practice. For example, in [4] it is stated at page 153 (in relation to the amount of folds in CV): "Why 10? Extensive tests on numerous different datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error, and there is also some theoretical evidence that backs this up." No references are given. At page 2980 in [5], which deals with nonlinear regression, it is stated that "Many simulation and empirical studies have verified that a reliable

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 291–300, 2014.

estimate of [generalization] Err can be obtained with $k = 10$ for $N > 100$ as recommended by Davison and Hinkley (1997)." The precise argument by Davison and Hinkley [6] at page 294, in the context of linear regression, is to take $k = \min\{N^{1/2}, 10\}$ due to *practical experience*: "taking $k > 10$ may be too computationally intensive... while taking groups of size at least $N^{\frac{1}{2}}$ should perturb the data sufficiently to give small variance of the estimate." No computational experiments are performed to support the claim.

Actually it has been observed in many articles that use of CV is not straightforward. In [7] it is shown, in the least-squares-estimation context, that for an unstable procedure the predictive loss, i.e., difference between "crystal ball" and cross-validation based selection, is large. The difficulties of using (10-)CV for the MLP model selection were already addressed in [8]: With fourteen UCI benchmark data sets the experiments showed that CV is only slightly better than random selection of MLP architecture and that the smallest size of the hidden layer tested provided almost the same generalization performance than the repeated folding. Based on experiments with the same UCI data sets as Kohavi [3], for ID3 and Info-Fuzzy Network classifiers with 2-CV, [9] ended up with "CV uncertainty principle": "the more accurate is a model induced from a small amount of real-world data, the less reliable are the values of simultaneously measured cross-validation estimates." Finally, in [10] large and general review of CV is given. Among the overall conclusions it is stated that i) usually CV overestimates generalization error compared to training error, ii) CV method with minimal variance [of generalization error estimate] seems strongly framework-dependent, and iii) the issue of "optimal" amount of folds in CV is not straightforward.

Hence, the purpose of this paper is to perform a set of experiments to explore the characteristics of cross-validation, when dealing with model evaluation of MLP. We test two variants of stratification, where the new approach takes into account classwise data densities [11] in addition to pure class frequency. To simplify the analysis, we restrict to ten folds. The contents are as follows: in Section 2 we summarize the model, the learning problem, and the actual algorithms. Then, in Section 3, a sequence of computational experiments with observations and subconclusions is presented. Finally, general conclusions are summarized in Section 4.

2 Methods and Algorithms

2.1 MLP

Action of MLP in a layerwise form, with given input vector $\mathbf{x} \in R^{n_0}$, can be formalized as follows [12]: $\mathbf{o}^0 = \mathbf{x}, \quad \mathbf{o}^l = \mathcal{F}(\mathbf{W}^l \tilde{\mathbf{o}}^{(l-1)})$ for $l = 1, \ldots, L$. Here the layer number has been placed as an upper index and by $\tilde{}$ we indicate the vector enlargement to include bias. This places these nodes in a layer as first column of the layer's weight matrix which then has the factorization $\mathbf{W}^l = \left[\mathbf{W}_0^l \ \mathbf{W}_1^l\right]$. $\mathcal{F}(\cdot)$ denotes the application of activation functions. We restrict to networks with one hidden layer so that the two unknown weight matrices are $\mathbf{W}^1 \in \mathbb{R}^{n_1 \times (n_0+1)}$

Algorithm 1. Determination of neural network model using cross-validation

Input: Data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$.
Output: Feedforward neural network.
Define $\boldsymbol{\beta}$, $n1max$, $nfolds$, and $nits$
for $n_1 \leftarrow 2$ **to** $n1max$ **do**
  for $regs \leftarrow 1$ **to** $|\boldsymbol{\beta}|$ **do**
    Create $nfolds$ using cross-validation
    for $k \leftarrow 1$ **to** $nfolds$ **do**
      for $i \leftarrow 1$ **to** $nits$ **do**
        Initialize $(\mathbf{W}^1, \mathbf{W}^2)$ from $\mathcal{U}([-1, 1])$
        Minimize (1) with current n_1 and $\boldsymbol{\beta}(regs)$ over kth training set
        Store network for smallest training set error
      Compute test_error over kth test set for the stored network
    Store network for the smallest mean{test_error}

and $\mathbf{W}^2 \in \mathbb{R}^{n_2 \times (n_1+1)}$. Using the given learning data $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^{n_0}$ and $\mathbf{y}_i \in \mathbb{R}^{n_2}$, determination of weights is realized by minimizing the cost functional

$$\mathcal{J}(\mathbf{W}^1, \mathbf{W}^2) = \frac{1}{2N} \sum_{i=1}^{N} \|\mathbf{e}_i\|^2 + \frac{\beta}{2n_1} \sum_{i,j} \left(|\mathbf{W}^1_{i,j}|^2 + |(\mathbf{W}^2_1)_{i,j}|^2 \right) \tag{1}$$

for $\mathbf{e}_i = \mathbf{W}^2 \tilde{\mathcal{F}}(\mathbf{W}^1 \tilde{\mathbf{x}}_i) - \mathbf{y}_i$ and $\beta \geq 0$. The linear second layer and the special form of regularization omitting the bias-column $\mathbf{W}^2_0$ are due to Corollary 1 in [12]: For every locally optimal MLP-network with the cost functional (1), satisfying $\nabla_{\mathbf{W}^2} \mathcal{J} = \mathbf{0}$, the average error $\frac{1}{N} \sum_{i=1}^N \mathbf{e}_i$ is zero. Hence, every locally $\mathbf{W}^2$-optimal network provides un unbiased nonlinear estimator for the learning data, independently on the regularization coefficient β.

The actual determination of MLP is documented in Algorithm 1. The main point is to realize systematic grid search over the complexity landscape, determined by n_1 (size of network) and β (size of weights; the larger β the closer to zero). Hence, $n1max$ determines largest size of the hidden layer and predefined values in vector $\boldsymbol{\beta} = \{\beta_r\}$ define possible regularization coefficients. Due to preliminary testing, we use here $\beta_r = 10^{-r}, r = 2, \ldots, 6$. Moreover, with the fixed parameters we always create new folds to sample the CV approaches below. The parameter $nits$ determines the amount of local restarts with random generation/initialization of weights from the uniform distribution. This is the simplest globalization strategy when minimization of (1) will be done locally.

2.2 Two Folding Approaches for CV

We apply two stratification strategies for folding: the standard random creation where class frequencies of the whole training data are approximated in folds (SCV). As the second approach, DOB-SCV (Distribution Optimally Balanced Standard CV) as proposed in [11] was implemented, see Algorithm 2. In this

Algorithm 2. Distribution Optimally Balanced Standard CV (DOB-SCV)

Input: Data $(\mathbf{X}, C) = \{\mathbf{x}_i, c_i\}_{i=1}^{N}$ of inputs and class labels and amount of folds k.
Output: k non-disjoint folds $F_l, l = 1, \ldots, k$, such that $\mathbf{X} = \cup_{l=1}^{k} F_l$.

```
for each class j and input data X_j = {x_i | c_i = j} do
    while |X_j| ≥ k do
        Let x_1 be random observation from X_j
        Let x_2, ..., x_k be k − 1 closest neighbors of x_1 from X_j
        Let F_l = F_l ∪ {x_l} and X_j = X_j \ {x_l}, l = 1, ..., k
    Place the remaining observations from X_j into different folds F_l, l = 1, ..., |X_j|
```

approach, using the division of a random observation from class j and its $k-1$ nearest class neighbors to different folds, classwise densities in addition to frequencies are approximated in all the folds. We remind that in [11,13] the extensive experimentation on various data sets and classifiers did not include MLP as classifier, not to mention the particular optimization problem (1) that we solve here.

3 Computational Experiments

All methods described in the previous section were implemented and tested on MATLAB (R2013b running on 64-bit Windows 7). For SCV, *cvpartition* routine is used. Minimization of (1) is based on MATLAB's unconstrained minimization routine *fminunc*, using layerwise sensitivity calculus from [12] for computing gradients. Standard sigmoid $s(x) = 1/(1 + \exp(-x))$ is used as the activation function. All input variables are preprocessed into the range $[0, 1]$ of $s(x)$ to balance the overall scaling of unknowns [12]. Class encoding is realized in the well-known manner by using standard basis in $\mathbf{R}^{n_2}$: the lth unit vector is used as target output for an input $\mathbf{x}_i$ from class C_l.

As benchmark data we use "Segmentation" from UCI repository, which is multiclass ($n_2 = 7$ classes) and many-input ($n_0 = 17$ input variables when two nearly constant ones are omitted) data set with small training set "Sgm (Train)" and large, separate validation set "Sgm (Test)". These sets are documented in Table 1. In what follows, we use the term Training error, TrE, for the mean error which is computed over the training sets, i.e. the subsets of "Sgm(Train)" without the test folds. Similarly, Test error TsE refers to mean error over test folds. With Generalization error GeE, we refer to the error which is computed using the validation set "Sgm(Test)".

Table 1. UCI classification data sets for CV experiments

Name	N	Class frequencies	Comments
Sgm (Train)	210	[30 30 30 30 30 30 30]	Features 3–4 removed
Sgm (Test)	2100	[300 300 300 300 300 300 300]	Features 3–4 removed

Table 2. 10-CV results for misclassification rate in percentages as error measure

10-SCV	10-SCV	3x10-SCV
7/1e-5/3.5/12.4(6.4)	12/1e-3/6.0/11.9(7.9)	11/1e-3/6.0/13.3(8.3)
10-DOB-SCV	10-DOB-SCV	3x10-DOB-SCV
7/1e-5/4.2/13.8(4.2)	9/1e-3/6.7/12.4(4.0)	6/1e-6/4.7/12.9(5.8)

3.1 Misclassification Rate in Percentages as Error Measure

In Table 2 first set of results using SCV and DOB-SCV with Algorithm 1 are given. We use $nits = 2$ and $n1max = 12$. In the results, n_1^* and β^* for the smallest TsE, its standard deviation Std, and the corresponding TrE are given. The actual result format is thus n_1^*/β^*/TrE/TsE(Std). For both folding approaches the algorithm is first tested two times separately, and then three times repeated folding for fixed n_1 and β is performed so that the errors are then computed over 30 training and test set errors. As error measure the misclassification rate in percentages is used.

From Table 2 one notices very high unstability of the results. Training and Test errors are very different, best parameters between tests vary a lot, standard deviations are large (round 30%–65% of means) and they do not decrease when folding is reiterated. For DOB-SCV, Stds are typically smaller compared to SCV, but there is no real difference in TsEs. Altogether one ends up with high uncertainty with these results, especially when the relationship between Training and Test errors, visualized using scatter plots in Fig. 1, is taken into account. The quantized form of the discrete error measure does not allow accurate evaluation of MLP models with different complexity, which is reflected as high variability in parameter choices.

3.2 Testing Predictive Error Measures

Next we test whether the discrete approximation of classification error could be one reason for difficulties with CV. Instead of misclassification rate in percentages, we test two error measures which are typical for estimating the actual prediction error:

$$e_{\mathrm{MR}} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{n_2}(\mathcal{N}(\mathbf{x}_i)-\mathbf{y}_i)_j^2} \quad \text{(Mean-Root-Squared-Error)},$$

$$e_{\mathrm{RM}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n_2}(\mathcal{N}(\mathbf{x}_i)-\mathbf{y}_i)_j^2} \quad \text{(Root-Mean-Squared-Error)}.$$

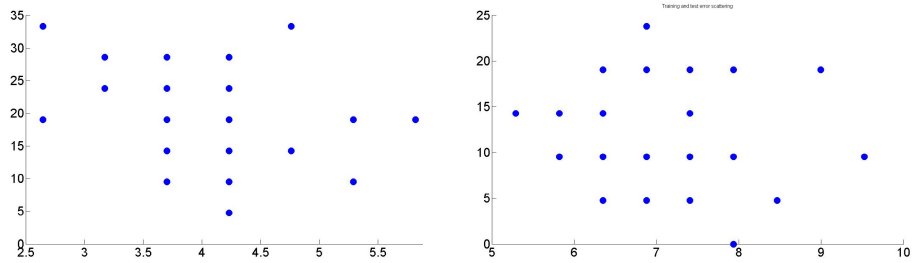

Fig. 1. Training/Test error scatterings for 3xSCV with $n_1 = 7$ and $\beta = 10^{-5}$ (left) and for 3xDOB-SCV with $n_1 = 9$ and $\beta = 10^{-3}$ (right)

Even if the definitions are very close to each other, the amount of observations weights the rest of the error measure differently ($1/N$ in e_{MR} compared to $1/\sqrt{N}$ in e_{RM}) and we want to find out how this affects comparisons of Training, Test, and Generalization errors which are computed with data sets of different sizes.

In Fig. 2 scatter plots of training and test set errors for SCV are given for all locally optimal MLPs obtained with Algorithm 1 for $n_1 = 5, \ldots, 8$ and $nits = 2$. It is visually clear that e_{MR} reflects the positive correlation between the two errors in a better way than e_{RM}. The visual appearance and the conclusion are precisely the same for DOB-SCV.

Using e_{MR} as CV error measure in MLP model evaluation in Algorithm 1, we obtain the following choices of parameters using the same grid search as in Table 2: for SCV, $n_1 = 5$ and $\beta = 10^{-5}$ and for DOB-SCV, $n_1 = 7$ and $\beta = 10^{-5}$. We then fix these and reiterate the two folding approaches three times with $nits = 5$. The individual results and their grand mean over different foldings are documented in Table 3.

From Table 3 we conclude that Training error underestimates and Test error overestimates Generalization error. For different foldings, SCV results are this time more stable than those of DOB-SCV. However, there is one remarkable difference in the characteristics of the results.

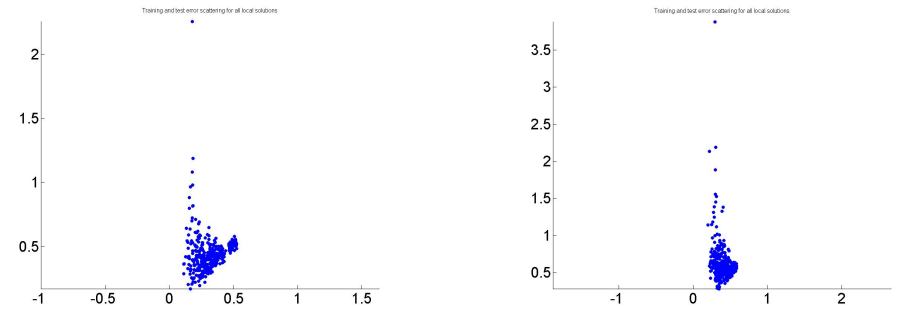

Fig. 2. Training/test set error scatterings with SCV: e_{MR} (left) and e_{RM} (right)

Table 3. Repeated CV with e_{MR} as error measure

	3xSCV			3xDOB-SCV		
Fold	TrE	TsE	GeE	TrE	TeE	GeE
1st	0.3065	0.3532	0.3443	0.1908	0.3421	0.3439
2nd	0.3063	0.3618	0.3480	0.1976	0.4478	0.3497
3rd	0.3080	0.3552	0.3497	0.1906	0.4852	0.3816
Grand	0.3069	0.3567	0.3473	0.1930	0.4250	0.3584

Namely, for one particular folding, one observation from the original training data belongs to exactly one test set due to disjoint division. Hence, for one observation the maximum amount of false test classifications over the three foldings is precisely three. Next, for 3xSCV with $n_1 = 5$ and $\beta = 10^{-5}$ and for 3xDOB-SCV with $n_1 = 7$ and $\beta = 10^{-5}$ we checked their classwise behavior in this respect, i.e. report the amount of indices per class where this maximum of three false test classifications per one observation is reached:

SCV: [1 0 4 29 3 1 0] = 38 cases,
DOB-SCV: [1 0 4 7 3 3 0] = 18 cases.

Hence, for SCV the pure inside-class randomness can produce high variability between classwise test accuracies (because the test folds can be very different from each other) whereas DOB-SCV compensates this dramatically better, through and due to distributional balancing. Notice that in the mean accuracy estimates without separating the classes, such behavior is completely hidden, as witnessed in Tables 2 and 3. We also conclude that class 4 is the most difficult one, so that to improve the classification performance more observations from that class should be contained in the training data.

3.3 Predictive CV with Modified Data Sets

To this end, we remove 30 random observations of class 4 from the original "Sgm(Test)" and add them to "Sgm(Train)". The previous stepwise experimentation is repeated as follows: i) With three repetitions apply the grid search for n_1 and β with $nits = 2$ using e_{MR} as error measure (cf. Table 2), ii) Compute mean errors with the chosen parameters with three repetitions of foldings (cf. Table 3) and visually assess an error scattering plot, and iii) check the classwise error rates from the test folds.

Result of Step i) is given in Table 4. We have obtained much higher stability in the parameter choice for both folding approaches. Also standard deviations are smaller compared to mean errors (around 20%–25% of means). Still, Training errors deviate from Test errors.

Table 4. Best parameters and errors for 10-CV with modified data sets

10-SCV	10-SCV	3x10-SCV
6/1e-6/0.21/0.36(0.07)	6/1e-6/0.25/0.37(0.10)	6/1e-5/0.28/0.37(0.07)
10-DOB-SCV	10-DOB-SCV	3x10-DOB-SCV
7/1e-6/0.19/0.34(0.09)	6/1e-5/0.28/0.37(0.08)	6/1e-5/0.27/0.36(0.08)

As for Step ii), we fix the parameters according to Table 4 as $n_1^* = 6$ and $\beta^* = 5 \cdot 10^{-6}$ for SCV and $n_1^* = 6$ and $\beta^* = 10^{-5}$ for DOB-SCV. The result with these choices for repeated foldings are given in Table 5.

We conclude from Table 5, especially compared to Table 3, that the increase of the size of training set from 210 to 240, that yielded to increase of the size of the hidden layer by one for SCV, then increased the differences between Training and Test errors. The common trend of Generalization error underestimation by Training error and overestimation by Test error remains. With the choices of parameters, slightly smaller Generalization errors are this time obtained with DOB-SCV compared to SCV. Now, for TrE's and TsE's the behavior of two folding approaches is similar.

Scatter plots for training and validation set errors with the two approaches are depicted in Figure 3. We see that both folding approaches yield to positive correlation between these errors, with DOB-SCV capturing such a desired behavior slightly better.

To this end, for the three repetitions in Table 5 and taking into account only those cases where an observation was always wrongly classified in a test set, we obtained the following amount of misclassifications per class:
SCV: [1 0 3 8 5 6 0] = 23 cases,
DOB-SCV: [1 0 4 6 7 8 0] = = 26 cases.
We conclude that the modifications of training and validation sets paid off, especially for SCV, by means of improved classwise balance of the classification accuracy and significantly smaller overall misclassification rate. For DOB-SCV, the amount of complete misclassifications increased significantly, from 18 into 26, because the emphasis on class 4 had negative effect on accuracies in classes 5 and 6. The smaller amount obtained by SCV does not imply superiority over DOB-SCV but just the fact that the result was obtained with more flexible model, i.e. with slightly smaller β^*.

Table 5. Repeated CV with modified sets

	3xSCV			3xDOB-SCV		
Fold	TrE	TsE	GeE	TrE	TeE	GeE
1st	0.2724	0.3754	0.3570	0.2728	0.4008	0.3597
2nd	0.2688	0.4080	0.3723	0.2795	0.3804	0.3570
3rd	0.2712	0.3840	0.3658	0.2743	0.3728	0.3402
Grand	0.2708	0.3891	0.3650	0.2755	0.3847	0.3532

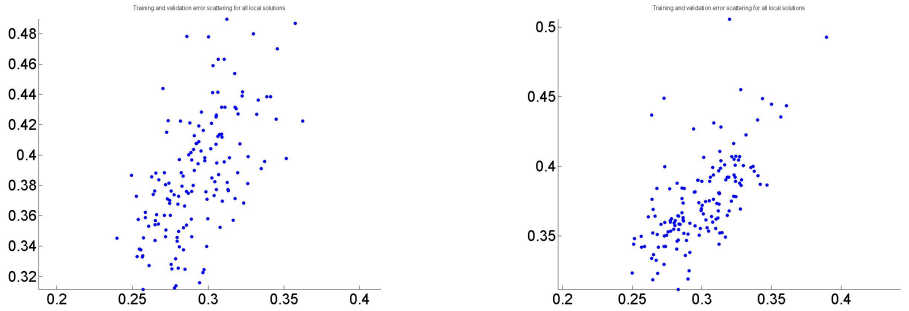

Fig. 3. Scatter plots of training/validation set errors for 3xSCV (left) and 3xDOB-SCV (right) with modified data sets

4 Conclusions

The performed set of experiments illustrate some difficulties related to model assessment with cross-validation. The general statistical assumptions on fixed input and input-output conditional distributions are not necessarily valid with real data sets. The amount of folds and the actual folding strategy have an effect on the behavior of CV. The error measure used with different data sets (training, test, validation) affects error computations and, hence, the form of obtained relationships underlying model selection. Especially when a universal prediction model, like MLP, is used in classification with typical output encoding, a discrete and quantized error measure suppress the precious information reflecting the quality of the model. In any case, estimation of the generalization error through test folds is only an approximation, and with all the experiments and techniques used here, we always ended up to overestimate the true generalization error using mean over ten test folds. Similarly, the standard deviation of generalization error estimate can remain large and does not necessarily decrease with repeated folding. We conclude, by comparing the Std estimates obtained with different parameters (typically with simpler network - with smaller size of hidden layer or larger regularization coefficient - we end up with smaller variance), that this estimate reflects more the variability of the model itself instead of one model's actual classification performance. Such an observation might be valid for universal approximators in general. We also illustrated that the quality of data has an effect on cross-validation results, especially when using the standard, stratified CV.

Through all the computational experiments performed we found that DOB-SCV folding approach could be better suited for real data sets, because it potentially provides better differentiation of a classifier's true performance through more homogenous test folds. This conclusion coincides with the results in [11] that were obtained with other classifiers and for larger set of folding approaches. Moreover, if classwise deviations in accuracy are revealed, one can, with sample data sets, augment the training data set accordingly or, in real applications,

launch a new data collection campaign to improve the overall classification performance. The findings here are obtained with only one benchmark data set with $k = 10$ folds, so further experiments with larger sample of real data sets and different amount of folds should be carried out in the future.

References

1. Elisseeff, A., Pontil, M.: Leave-one-out error and stability of learning algorithms with applications. NATO Science Series, Sub Series III: Computer and Systems Sciences 190, 111–130 (2003)
2. Pinkus, A.: Approximation theory of the MLP model in neural networks. Acta Numerica, 143–195 (1999)
3. Kohavi, R.: Study of cross-validation and bootstrap for accuracy estimation and model selection. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI 1995), pp. 1137–1143 (1995)
4. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, Burlington (2011)
5. Borra, S., Ciaccio, A.D.: Measuring the prediction error. a comparison of cross-validation, bootstrap and covariance penalty methods. Computational Statistics and Data Analysis 54, 2976–2989 (2010)
6. Davison, A.C., Hinkley, D.V.: Bootstrap Methods and their Applications. Cambridge Series on Statistical and Probabilistic Mathematics. Cambridge University Press (1997)
7. Breiman, L.: Heuristics of instability and stabilization in model selection. The Annals of Statistics 24(6), 2350–2383 (1996)
8. Andersen, T., Martinez, T.: Cross validation and MLP architecture selection. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 1999), pp. 1614–1619 (1999)
9. Last, M.: The uncertainty principle of cross-validation. In: Proceedings of the IEEE International Conference on Granular Computing (GrC 2006), pp. 275–280 (2006)
10. Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Statistics Surveys 4, 40–79 (2010)
11. Moreno-Torres, J.G., Sáez, J.A., Herrera, F.: Study on the impact of partition-induced dataset shift on k-fold cross-validation. IEEE Transactions on Neural Networks and Learning Systems 23(8), 1304–1312 (2012)
12. Kärkkäinen, T.: MLP in layer-wise form with applications to weight decay. Neural Computation 14, 1451–1480 (2002)
13. López, V., Fernández, A., Herrera, F.: On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. Information Sciences 257, 1–13 (2014)