Les Cahiers du GERAD ISSN: 0711–2440

On removing diverse data for training machine learning models

K.T. Ton, D. Aloise, C. Contardo

G–2022–38

August 2022

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2022 – Bibliothèque et Archives Canada, 2022

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2022 – Library and Archives Canada, 2022

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

On removing diverse data for training machine learning models

Kim T. Ton [a, b]

Daniel Aloise [a, b]

Claudio Contardo [a, c]

[a] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[b] *Département de génie informatique et génie logiciel, Polytechnique Montréal, Montréal (Qc), Canada H3C 3A7*

[c] *École de génie et d'informatique Gina Cody, Université Concordia, Montréal (Qc), Canada, H3G 1M8*

kim-thuyen.ton@polymtl.ca
daniel.aloise@polymtl.ca
claudio.contardo@concordia.ca

August 2022
Les Cahiers du GERAD
G–2022–38

Abstract : Providing the right data to a machine learning model is an important step to insure the performance of the model. Non-compliant training data instances may lead to wrong predictions yielding models that cannot be used in production. Instance or prototype selection methods are often used to curate training sets thus leading to more reliable and efficient models. In this work, we investigate if diversity is helpful as a criterion for choosing which instances to remove from a given training set. We test our hypothesis against a random selection method and Mahalanobis outlier selection, using benchmark data sets with different data characteristics. Our computational experiments demonstrate that selection by diversity achieves better classification performance than random selection, and can hence be considered as an alternative data selection criterion for effective model training.

Acknowledgements: The authors would like to thank professors Eduardo Pardo and Abraham Duarte for providing us the OBMA code and executable. This research was partially funded by Natural Sciences and Engineering Research Council of Canada (NSERC) under grants DG–2017–05617 and DG–2020–06311 for its financial support.

1 Introduction

Machine learning (ML) has often been perceived as requiring the largest possible amount of data to gain accuracy in predicting a behavior. Typically, there are three stages for a ML model: preprocessing, training and decision/prediction [1]. During preprocessing, the provided training set might be transformed before being fed to the ML model. In the sequel, during the training stage, the model processes the training set to generalize rules and formulas for prediction with a minimum amount of classification errors. Finally, at the prediction stage, new unlabelled data instances are given to the ML model which must predict a class or a value for them.

Nowadays, several preprocessing methods exist to ensure that only the right data is given to an ML model – a concern that accompanies the ML field since its origin [2, 3]. For example, a model that overfits or underfits the training data will result in poor predicting capabilities as they lose their ability to generalize over unseen data [4]. In addition, being able to reduce the amount of data needed to correctly train a ML model is crucial to speed up its training process and save memory resources.

Preprocessing techniques can be mainly categorized into feature selection, instance selection, and outlier detection methods [2, 3]. All of them seek to decrease the amount of data fed to a classification model. Feature selection methods reduce the training dataset by decreasing the number of used features, therefore the dimension of the data. Those methods weight the features in order of relevance and remove the least important ones [5]. Both outlier detection methods and instance selection methods work by reducing the amount of data instances. A method can either focus on removing noisy instances, superfluous data instances or both. Noisy or outlier data instances deteriorate the performance of classifiers when added to the training set while superfluous instances do not impact the performance when removed [6]. Outlier detection methods, as the name indicates, focus on removing outliers from the dataset [7].

The goal of an instance selection method is to speedup the model training by reducing the size of the training set without impacting the model's performance [6, 8, 9]. An instance selection method can either start with an empty training set and add data instances, or start with all the data then remove instances. The selection criterion is usually based on a performance metric or a selection formula. With a metric performance, the methods reduce the training set as long as the classification performance stays above a predefined threshold [9]. With a selection formula, the stopping condition is typically defined by the user, e.g. number of needed instances, logical tests, etc. Multiple criteria can be combined together to achieve a more complex method [10].

In this paper, we investigate if *diversity* can be used as an effective criterion for removing instances within selection methods. Our concept of diversity is related to variety among the data instances, which is quantified by the observed dissimilarities among them [11]. Our research hypothesis is that the removed data instances are diverse, representing instances less likely to belong together to the same class. Thus, given that one decides to reduce the training size of a ML model, these data instances are rather selected to be suppressed.

We test our approach on classifying eight different benchmark datasets, comparing it with two baseline methods. The first one selects data instances for suppression completely at random whereas the second consists of the classical Mahalanobis outlier detection method [6, 12].

The paper is organized as follows. In Section 2, we present the *maximum diversity problem* which is optimized to decide the data instances to be removed from the available training set. In Section 3, we explain our instance selection method based on maximum diversity. Section 4 describes the experimental methodology used to test our research hypothesis. In Section 5, we present and discuss the performed computational results. Finally, in Section 6, we present our concluding remarks.

2 The maximum diversity problem

Given a set of n data instances $U = \{u_1, \ldots, u_n\}$ for which a symmetric dissimilarity matrix $D = \{d_{ij} : 1 \leq i, j \leq n\}$ is defined such that $d_{ii} = 0$ and $d_{ij} \geq 0$ for every $1 \leq i < j \leq n$, the *maximum diversity problem* (MDP) consists of selecting a subset $P \subset U$ of size $p < n$ such as the sum of dissimilarities between the elements of P is maximum. The problem is formalized as:

$$\begin{aligned} \max & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} x_i x_j \\ \text{subject} & \sum_{i=1}^{n} x_i = p \\ & x_i \in \{0, 1\} \qquad \forall i = 1, \ldots, n. \end{aligned} \tag{1}$$

The MDP arises in many real-life applications. For example, in facility location, one may be interested in locating competing stores in a city as far as possible, or to place trash/pollutant storage as to not concentrate exposure in one area of the town [13, 14]. The MDP is also applied in biology for deciding about ecosystems re-population or for genetic engineering to produce more resilient plants [15, 16, 17, 18, 19], or for product design where companies want to have products that are different from their competitor [20]. The problem was shown to be NP-hard by Kuo et al [11].

Several exact and heuristic methods have been proposed in the literature to solve the MDP [14, 21, 22]. The state-of-the-art exact method for the MDP is due to Martí et al. [23] who proposed a branch-and-bound able to optimally solve medium-size instances with $n = 100$ in 1 hour of CPU time. Regarding heuristics, Martí et al. [24] have very recently performed an exhaustive comparison of state-of-the-art heuristics on the MDPLIB 2.0 - Maximum Diversity Problem Library available at https://www.uv.es/ rmarti/paper/mdp.html. Among the compared methods, the OBMA method of Zhou et al. [25] emerges as the best heuristic.

3 Instance selection by maximum diversity

Using the MDP as underlying optimization model, we propose a new instance selection method which removes p data instances from the training set. The so-called *Max Diversity Instance Selection Method* (`MaxDivSelec`) is described in Algorithm 1. `MaxDivSelec` proceeds by removing a total of p data instances from the classes of the training dataset. For that, it solves a MDP in each class. The algorithm starts by initializing the training set with all labelled data instances (line 1). After that, the algorithm iterates (lines 2-8) over each class label $c = 1, \ldots, k$ of the training dataset. In line 3, the data instances of class c are isolated in $X_c \subseteq X$, and then the covariance matrix X_c is computed in line 4. Then, in line 5, a matrix D_c of distances is computed for each pair of instances in class c. In our case, $D = (d_{ij})$ are computed as Mahalanobis distances, i.e.,

$$d_{ij} = \sqrt{(x_i - x_j)^T \Sigma^{-1} (x_i - x_j)}. \tag{2}$$

We note that Σ is approximated by singular value decomposition (SVD) factorization if it is singular [26]. In the sequel, a MDP solver – in our case OBMA – is called to solve an MDP problem for D_c, selecting the $\overline{p}_c$ points of maximum diversity in class c. More details on how $\overline{p}_c$ is computed are given in Section 4.3. The algorithm then removes the selected data instances from the training set in line 7. Finally, the reduced training set T is returned in line 9.

Let $X_c \subseteq X$ be the matrix of dimension $n_c \times s$ composed by the data instances of class c in X Compute the covariance matrix Σ_c of X_c Compute a distance matrix D_c of dimension $n_c \times n_c$ $R \leftarrow$ `SolveMDP`$(D_c, \overline{p}_c)$ $T \leftarrow T \setminus R$

Algorithm 1 `MaxDivSelec`

Input: X: labelled dataset of dimension $n \times s$,
$\overline{p}$: array of dimension $1 \times k$ with the number of instances to be removed per class

$T \leftarrow X$
for $c = 1, \ldots, k$ **do**
  Let $X_c \subseteq X$ be the matrix of dimension $n_c \times s$ composed by the data instances of class c in X
  Compute the covariance matrix Σ_c of X_c
  Compute a distance matrix D_c of dimension $n_c \times n_c$
  $R \leftarrow$ `SolveMDP`$(D_c, \overline{p}_c)$
  $T \leftarrow T \setminus R$
end for
return T

Figure 1 illustrates the use of `MaxDivSelec` on a 2D synthetic dataset consisting of two gaussians with 40 data instances each. The first gaussian is generated with μ=0 and σ=0.5, while the second has a μ=-3 and σ=1. In the example five data instances are removed from each gaussian.

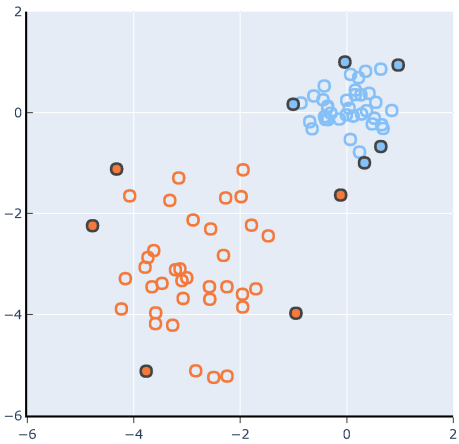


Figure 1: Illustration of the data instances selected by `MaxDivSelec` for $\overline{p}_1 = 5$ and $\overline{p}_2 = 5$, which corresponds to 12.5% of the whole dataset

4 Experimental methodology

4.1 K-nearest neighbors

In order to prove our hypothesis about effectiveness of `MaxDivSelec` as a data instance selection method for classification models, we had to choose one representative ML model from which our conclusions could be better generalized.

The K-nearest neighbor (KNN) model is a simple yet effective supervised classifier [27]. It predicts the class of an unseen instance by finding its K closest data instances from the training set. The unlabelled instance is then assigned to the majority class among them. The KNN classification model was a natural choice for our experiments for three reasons:

(i) it relies on a distance metric – as well as the MDP.
(iii) it is quite tolerant to outliers and noisy data.
(iii) its classification performance, memory usage and computing times are tightly linked to the number of data instances used for training. [1]

[1] KNN makes use of the so-called *lazy training* or *instance-based learning*. It simply queries over the data to make a prediction [28].

4.2 Baseline methods

We compared MaxDivSelec against two other selection methods. The first method, called `Random`, corresponds to our null hypothesis. It simply chooses p data instances to remove at random, with equal probability.

The second method, denoted here `Mahalanobis`, is well-known in the literature [29]. It removes outliers by computing the Mahalanobis distance (2) from each data instance to the centroid of the class it belongs. Larger values of the Mahalanobis distance indicate a greater outlier likelihood. The method aims to improve model classification by removing from the training dataset the instances which are too different to be statistically part of a class.

Algorithm 2 presents the pseudo-code of method `Mahalanobis` which returns a set T of data instances for model training. The method computes for each available labelled data instance its Mahalanobis distance to the centroid of the class to which it belongs. In the sequel, the algorithm removes, from each class c, $\overline{p}_c$ data instances whose Mahalanobis distances are the largest computed.

Algorithm 2 `Mahalanobis method`

Input: X: labelled dataset of dimension $n \times s$,
$\overline{p}$: array of dimension $1 \times k$ with the number of instances to be removed per class

$T \leftarrow X$
for $c = 1, \ldots, k$ **do**
  Let $X_c \subseteq X$ be the matrix of dimension $n_c \times s$ composed by the data instances of class c in X
  Compute the covariance matrix Σ_c of X_c
  for each data instance x_ℓ of class c **do**
    Compute the Mahalanobis distance $d_\ell = \sqrt{(x_\ell - \mu_c)^{\mathrm{T}} \Sigma_c^{-1} (x_\ell - \mu_c)}$ between x_ℓ and the centroid μ_c of class c
  end for
  $R \leftarrow$ the $\overline{p}_c$ instances of class c with largest d
  $T \leftarrow T \setminus R$
end for
return T

Figure 2 illustrates the application of the `Mahalanobis` method over the same synthetic example of the two the last section. Here, again, five data instances are removed from each gaussian. We remark that the selections performed by `MaxDivSelec` and `Mahalanobis` differ of two data instances only.

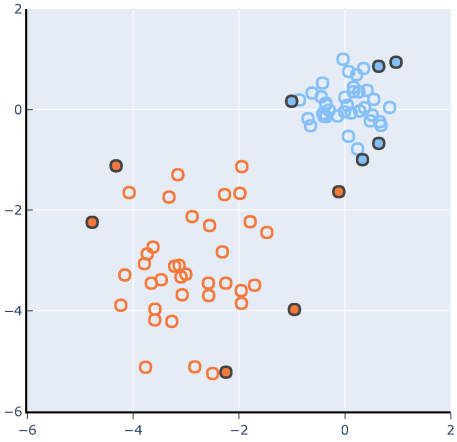


Figure 2: Illustration of the data instances selected by `Mahalanobis` for $\overline{p}_1 = 5$ and $\overline{p}_2 = 5$, which corresponds to 12.5% of the whole dataset

4.3 The α parameter

The α parameter controls the percentage of data instances to be removed from the training set. For example, for $\alpha = 50\%$ and $n = 100$, 50 data instances are suppressed from the training dataset ($p = 50\% \times 100$). This amount is split proportionally across the provided classes. Thus, the method removes $\overline{p}_c = \alpha \times n_c$ data instances from each class $c = 1, \ldots, k$, rounding it to the closest integer. At the end, some adjustments must be performed so that $\sum_{c=1}^{k} \overline{p}_c = p$. Considering $p' = p - \sum_{c=1}^{k} \overline{p}_c$ as the number of adjustments, we either remove or add a data instance to the final training set depending whether p' is positive or negative. The adjustments performed in each class are limited to one, and are performed from the class with the largest amount of data instances to the least populated class. To illustrate, for a training set with 5 classes such that $n_1 = 10$, $n_2 = 10$, $n_3 = 10$, $n_4 = 30$, $n_5 = 40$, for $\alpha = 50\%$ (that is, $p = 50$), we obtain $\overline{p}_1 = 5$, $\overline{p}_2 = 5$,$\overline{p}_3 = 5$, $\overline{p}_4 = 15$, $\overline{p}_5 = 20$. Because $p' = 0$, there is no need to adjust the values. For the same training set, by taking $\alpha = 12.5\%$ (that is, $p = 13$), we have $\overline{p}_1 = 1$, $\overline{p}_2 = 1$, $\overline{p}_3 = 1$, $\overline{p}_4 = 4$, $\overline{p}_5 = 5$. Since, in this case, $p' = 13 - 12 = 1$, $\overline{p}_5$ is adjusted to 6, making a total of $p = 13$ data instances.

5 Computational experiments

5.1 Datasets

We compare the presented methods over different real-world benchmark datasets. The used datasets are shown in Table 1. The different number of classes and attributes across them are aimed to test how well the compared methods handle complex classification problems. All datasets were numerically normalized in each attribute dimension before use.

Table 1: Table with datasets' characteristics.

Dataset	n	#classes	#attributes
Iris [30]	150	3	4
Seeds [31]	210	3	7
Dermatology	358	6	34
Ionosphere [32]	351	2	34
Breast cancer Wisc. [33]	683	2	9
Mammographic [34]	830	2	5
Contraceptive[35]	1473	3	9
Abalone [36]	4177	29	8

5.2 Evaluation

We used different classification performance metrics depending on whether the classification problem was: (i) binary or multiclass, and (ii) balanced or unbalanced. A binary classification problem is one in which prediction is done for two classes only, while a multiclass problem involves more than two classes. A balanced problem supposes that the number of data instances of each class is approximately the same, while an unbalanced classification task has the majority of the data instances belonging to a subset of the provided classes. To accommodate those different categories of problems, three performances metrics were used, namely accuracy, RMSE and the F1-score.

The accuracy score is a classification performance metric often used for supervised classification problems [37]. It compares the prediction to the ground-truth class thus computing the ratio of right predictions. In a binary classification problem, there exist four possible cases for a given prediction: a True Positive (TP), a False Positive (FP), a True Negative (TN) and a False Negative (FN) [38]. The two *true* cases happen when the model predict the correct class (positive or negative). Conversely, the *false* cases happen when the model predicts the opposite class. For example, a FP occurs when the

model predicts a positive class for an actual negative data instance. The accuracy score is is given by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

An accuracy value of 1 means that the model predicted 100% of the classes correctly while a score of 0 means that none of the predictions was correct. The accuracy metric can also be generalized for k classes as

$$Accuracy = \frac{\sum_{c=1}^{k} T_c}{\sum_{c=1}^{k} T_c + \sum_{c=1}^{k} F_c} \tag{4}$$

where T_c is the number of TP for class c, and F_i the number of FP for that same class.

The Root Mean Squared Error (RMSE) score is a performance metric commonly used for multiclass models [39] for which the classes are ordered somehow. Thus, for an expected value of 0, predicting 1 is less "wrong" than predicting 10, for instance. The RMSE is equal to the squared root of the mean of squared errors between the predictions and the ground-truth values. The formula with y' and y as the predicted and ground-truth values, respectively, and n as the number of predictions is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \tag{5}$$

The score represent by how much the model is off on average from the expected values. A score of 0 means that no error was made and the classifier is perfect.

The last and more complex performance metric is the F1-score also called the F-measure. It is used for unbalanced binary classification problems. It is a suitable score for when the model has to predict well one class in particular amongst others. The *recall* refers to the model's capacity to detect the positive class of interest among the total amount of positive samples, while the *precision* is the model's capacity to well classify TP data instances over the total amount of instances predicted as members of the positive class [38].

The recall and the precision of a model are computed as:

$$recall = \frac{TP}{TP + FN} \qquad precision = \frac{TP}{TP + FP}. \tag{6}$$

The F1-score is finally calculated as the geometric mean of both measures:

$$\text{F1-score} = 2 \cdot \frac{recall \cdot precision}{recall + precision} \tag{7}$$

The Iris, Seeds, Dermatology and Contraceptive datasets are assessed according to the accuracy score since they are balanced. The Abalone dataset is an unbalanced dataset with more than two ordered classes. Consequently, KNN's classification performance is evaluated according to the RMSE score for that dataset. Finally, datasets Ionosphere, Breast cancer Wisconsin and Mammographic datasets are evaluated according to F1-score since they consist of binary labelled data, with one majority class.

5.3 Cross-validation

The three methods `MaxDivSelec`, `Random` and `Mahalanobis` are tested with the KNN classifier for $K \in \{3, 5, 10\}$ and $\alpha \in \{0.125, 0.25, 0.5\}$, which yields a total of 9 combinations of parameters to be tested. The KNN classifier uses the Euclidean distance. To generalize our results, a 5-fold cross-validation is used to produce multiple test sets. A 5-fold cross-validation separates the data into 5 sets where each set is used as the test set while the rest is used as the training set [40]. That means that

80% of the dataset is used for training while the other 20% is used for the testing. For this experiment, ten 5-fold cross-validation processes are made to produce a total of 50 pairs of training/test sets. The instance selection methods are employed on the training set of each fold. Since `Random` is a stochastic method, it is executed 20 times with a different seed (0 to 19) for each fold.

Table 2 reports the benchmark performance scores of the KNN classifier for each dataset. They correspond to the classifier's mean performance when using the whole set of labelled data instances for training, i.e., without instance selection. The datasets are grouped in the table according to the used performance metric.

Table 2: Mean benchmark performance of KNN [Accuracy, F1-score, RMSE] for each tested dataset

Dataset	k=3	k=5	k=10
Iris	0.961	0.961	0.964
Seed	0.920	0.930	0.922
Dermatology	0.954	0.956	0.959
Contraceptive	0.458	0.483	0.502
Ionosphere	0.716	0.722	0.707
Breast cancer Wisconsin	0.946	0.953	0.953
Mammographic	0.771	0.789	0.792
Abalone	2.856	2.801	2.692

6 Results and discussion

Our computational results for methods `Random`, `Mahalanobis` and `MaxDivSec` are displayed as box plots to focus on the classification performance distributions of the methods over the tested folds. Besides, we present line charts of the mean performance obtained by each method. We present here a subset of the results, but all box plots can be checked at https://ktton.github.io/master-research/. Results are grouped by the number of used neighbors K and by the resulting training size after instance selection. By grouping the results, we can better evaluate how the parameters K and α affect classification performance.

The methods are compared regarding their general behavior, but also on their worst-case result. The worst-case result is the lowest performance result achieved by the method for a given data fold. Regarding accuracy and F1-score that corresponds to the lowest obtained score for a tested fold, whereas for the RMSE tha corresponds to the highest obtained score. The results are further analysed by means of a Wilcoxon statistical test with a confidence level of 5 %. That test tell us if the results achieved by our method are statistically different from those obtained by the baseline methods `Random` and `Mahalanobis`.

6.1 Classification performance results

First, we checked the general performance of the instance selection methods for each dataset. For the smallest datasets (regarding n), the three methods presented similar performance. To illustrate that, Figures 3a and 4a show the results for the dataset Seeds. Moreover, the obtained means are not far from the benchmark KNN performance, which means that using instance selection methods for small datasets does not incur significant losses of classification performance.

Regarding the largest datasets Breast cancer, Mammographic masses, Ionosphere, Contraceptive and Abalone, we observe a major difference between the classification metrics obtained by the different methods. Performing instance selection with `Random` appear to incur more varied classification performance than by using `Mahalanobis` and `MaxDivSec`, as shown in the box plots of Figures 3b, 3c and 3d. Figures 4b, 4c and 4d show the mean performance of each method for the same three datasets.

The Mahalanobis and MaxDivSec methods outperform the Random method particularly for Ionosphere (Figure 4c) and Abalone (Figure 4d). Besides, we note across the plots that the Mahalanobis and MaxDivSelec methods obtain better mean classification scores than those obtained by Random. These score differences become larger as more instances are removed from the training sample. In fact, for these instances, the Mahalanobis and MaxDivSelec methods perform better than the benchmark performance obtained by the KNN classifier using the whole data for training. We can hence conclude that for these datasets restricting the training data to relevant instances is important for increasing the generalization capability of the model.

Regarding the worst-case performance, MaxDivSelec always obtains better or equal worst-case classification results than Random. Having a better worst-case scenario means that our instance selection method is more robust regarding the posterior classification performance of the classifier when predicting the labels of unseen data. When compared to Mahalanobis, our method appears to have similar worst-case performance.

Finally, we also analyse the classification performance for varied values of α, and number of neighbors K used by the KNN classifier. Our conclusions are as follows:

- For the smallest datasets and breast cancer Wisconsin, the classification performance is lightly affected by K and α.
- For both the Abalone and Contraceptive datasets, the classification performance improves as K increases for methods MaxDivSelec, Mahalanobis and Random.
- For the Ionosphere dataset, the classification performance decreases as K increases for methods MaxDivSelec, Mahalanobis and Random, especially with α=50%.
- For the Abalone , Contraceptive, Mammographic and Ionosphere datasets, the classification performance of MaxDivSelec and Mahalanobis increases as α gets larger, i.e., as more data instances are removed from the training set. They are actually better than the benchmark performance presented in Table 2 except for the Mammographic dataset.

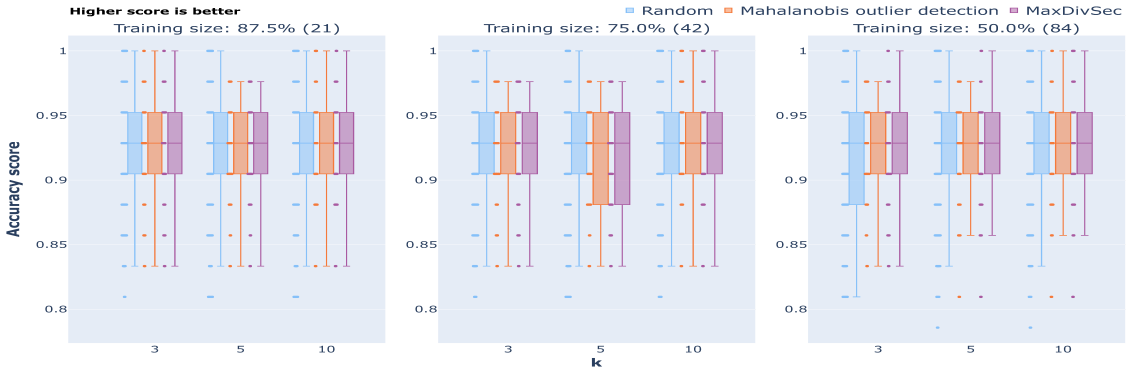
6.2 Wilcoxon tests

This section presents Wilcoxon signed-ranks tests [41] in order to compare the obtained results in terms of statistical significance.

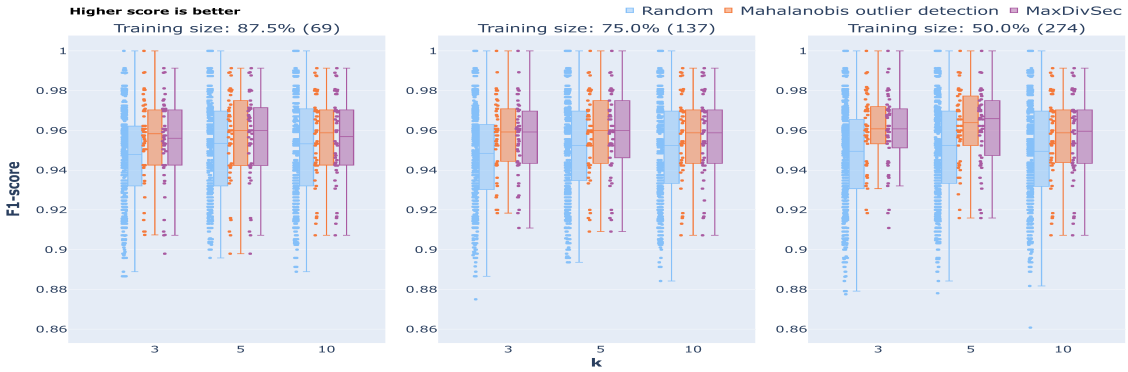
For each dataset, we compared the different methods Random, Mahalanobis and MaxDivSelec on each combination of $K = 3, 5$ and 10, and $\alpha = 0.25, 0.50$ and 0.75, totalizing nine Wilcoxon tests per dataset. Two hypothesis are tested. First, we check if the two result distributions are similar (i.e., the median of differences = 0). If that first hypothesis is rejected, this means that the second hypothesis is true, i.e., that the methods obtain statistically different results (the median of differences <0) We used a confidence level of 5% meaning that the p-value must be smaller than 0.05 to reject an hypothesis. The Wilcoxon test results are reported in Tables 3 and 4 for each dataset.

Table 3 shows the results of the comparisons of the methods MaxDivSec and Mahalanobis with the method Random. We observe that our instance selection method MaxDivSelec is statistically different from the Random method for most of K and α combinations for all the datasets. We can also verify the same behaviour with the Mahalanobis method except for the Seed dataset. The Mahalanobis method is only different for two combinations. Moreover, when our method is different from the Random method, it is most of the time better.

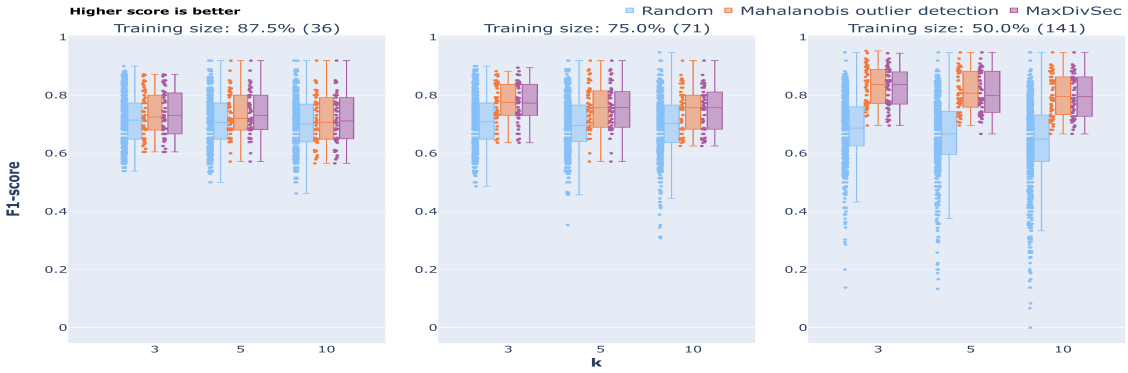
In Table 4, we show the Wilcoxon results obtained when comparing our method MaxDivSec with the Mahalanobis method. We notice that MaxDivSec is seldom different or better than Mahalanobis except for two datasets: Contraceptive and Mammographic. However, such difference does not mean that the first is necessarily better than the later. In most of the cases, MaxDivSelec is not statistically different from the Mahalanobis method.



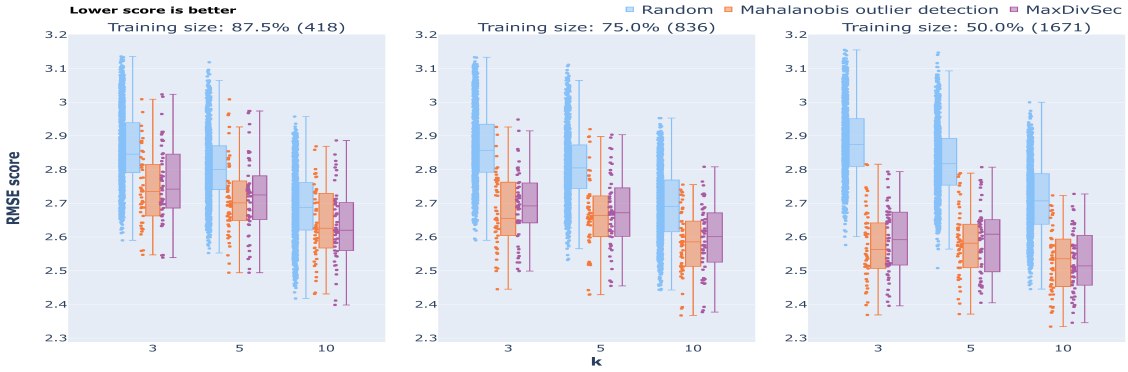
(a) Seeds dataset



(b) Cancer dataset

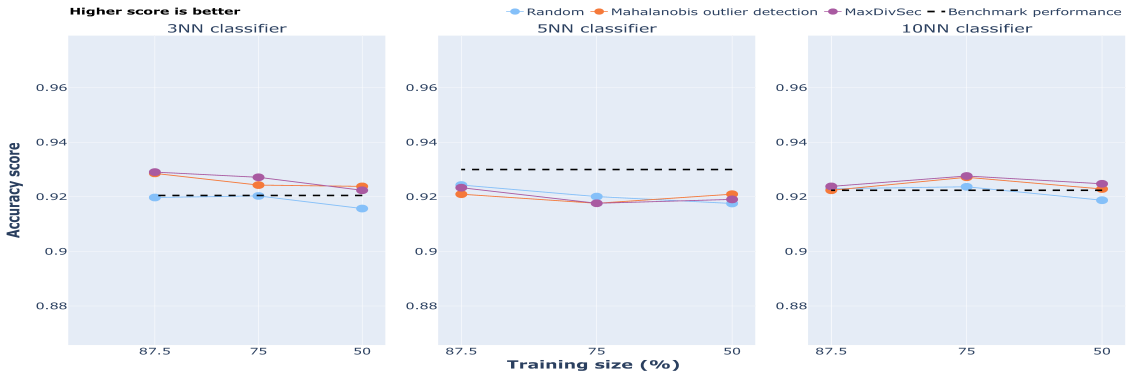


(c) Ionosphere dataset

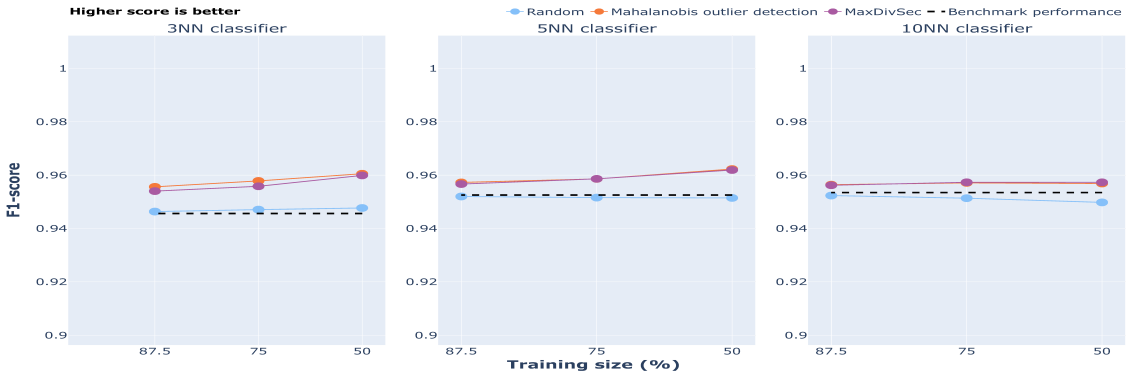


(d) Abalone dataset

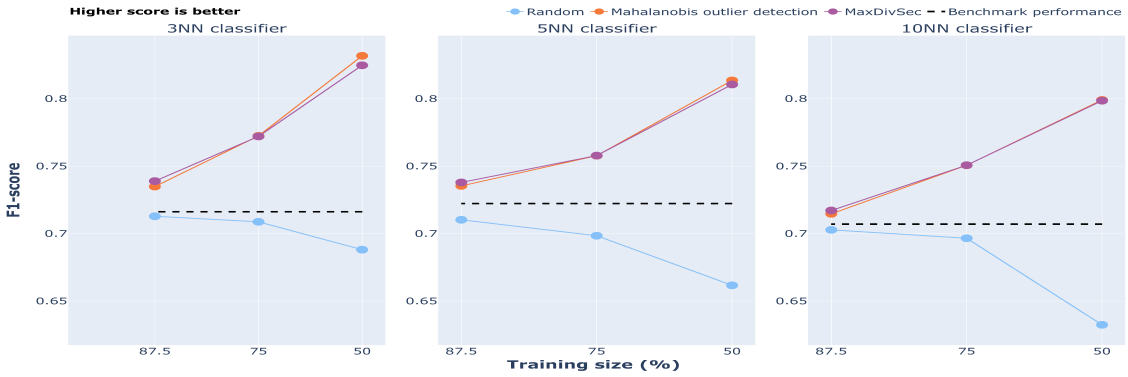
Figure 3: Boxplot results grouped by training size



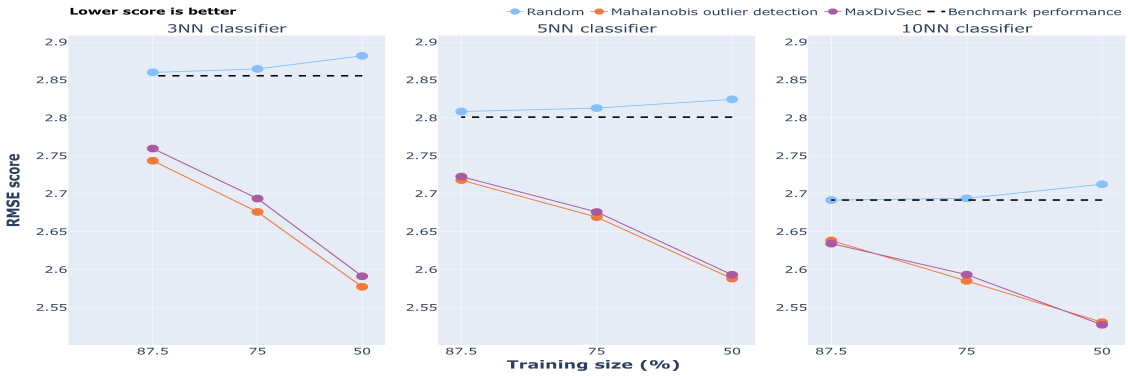
(a) Seeds dataset



(b) Cancer dataset



(c) Ionosphere dataset



(d) Abalone dataset

Figure 4: Mean classification results grouped by K

Table 3: Wilcoxon test results with a confidence level of 5% for comparing `MaxDivSec` and `Mahalanobis` with `Random`

Dataset	`MaxDivSec`		`Mahalanobis`	
	Different	Better	Different	Better
Iris	5/9	4/5	7/9	5/7
Seed	4/9	4/4	2/9	2/2
Dermatology	5/9	5/5	6/9	6/6
Ionosphere	9/9	9/9	9/9	9/9
Cancer	8/9	8/8	8/9	8/8
Mammographic	8/9	4/8	6/9	4/6
Contraceptive	9/9	9/9	8/9	8/8
Abalone	9/9	9/9	9/9	9/9

Table 4: Wilcoxon test results with a confidence level of 5% for comparing `MaxDivSec` with `Mahalanobis`

Dataset	Different	Better
Iris	1/9	1/1
Seed	0/9	0/0
Dermatology	0/9	0/0
Ionosphere	0/9	0/0
Cancer	1/9	0/1
Mammographic	7/9	3/7
Contraceptive	8/9	8/8
Abalone	3/9	0/3

7 Concluding remarks

This paper proposed to investigate the use of diversity for selecting data instances for model training. With that purpose, we proposed `MaxDivSec`, an algorithm that proceeds by removing from the training set of a machine learning model the subset of data instances for which its associated diversity is maximum. We compared `MaxDivSec`, regarding the classification performance of a target classifier, with two other baseline instance selection methods, one that random selects data instances for suppression and another based on the removal of data outliers. Our results demonstrated that diversity is actually a good criterion for data instance selection as the obtained results by `MaxDivSec` led to superior classification performance in the vast majority of the tested scenarios when compared to the random approach. However, the proposed method was not shown to be significantly different from the method based on the suppression of outliers. Finally, although we demonstrated by our experiments that maximum diversity is effective on selecting data instances for model training, its computation still requires the solution of a NP-hard problem either exactly or heuristically.

References

[1] C. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), chapter Introduction, pp. 1–66. Springer-Verlag, Berlin, Heidelberg, (2006).

[2] D. Pyle, Data Preparation for Data Mining. (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999), 1st edition.

[3] S. Zhang, C. Zhang, and Q. Yang, Data preparation for data mining, Applied Artificial Intelligence. 17 (5–6), 375–381, (2003). doi: 10.1080/713827180. URL https://doi.org/10.1080/713827180.

[4] G. I. Webb, Overfitting. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning, In Encyclopedia of Machine Learning, chapter O, pp. 744–744. Springer US, Boston, MA, (2010).

[5] L. C. Molina, L. Belanche, and A. Nebot. Feature selection algorithms: a survey and experimental evaluation. In 2002 IEEE International Conference on Data Mining, 2002. Proceedings., pp. 306–313, (2002). doi: 10.1109/ICDM.2002.1183917.

[6] S. Garcia, J. Derrac, J. Cano, and F. Herrera, Prototype selection for nearest neighbor classification: Taxonomy and empirical study, IEEE Transactions on Pattern Analysis and Machine Intelligence. 34(3), 417–435 (March, 2012). doi: 10.1109/TPAMI.2011.142.

[7] T. Dasu and T. Johnson, Exploratory Data Mining and Data Cleaning. (John Wiley & Sons, Inc., USA, 2003), 1 edition.

[8] J. Nalepa and M. Kawulok, Selecting training sets for support vector machines: a review, Artificial Intelligence Review. 52(2), 857–900 (Aug, 2019). doi: 10.1007/s10462-017-9611-1. URL https://doi.org/10.1007/s10462-017-9611-1.

[9] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler, A review of instance selection methods, Artificial Intelligence Review. 34(2), 133–143 (may, 2010). doi: 10.1007/s10462-010-9165-y.

[10] M. Blachnik, Ensembles of instance selection methods: a comparative study, International Journal of Applied Mathematics and Computer Science. 29(1), 151–68 (Mar., 2019). URL http://dx.doi.org/10.2478/amcs-2019-0012.

[11] C.-C. Kuo, F. Glover, and K. S. Dhir, Analyzing and modeling the maximum diversity problem by zero-one programming*, Decision Sciences. **24**(6), 1171–1185, (1993). doi: https://doi.org/10.1111/j.1540-5915.1993.tb00509.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-5915.1993.tb00509.x.

[12] A. Zimek and P. Filzmoser, There and back again: Outlier detection between statistical reasoning and data mining algorithms, WIREs Data Mining and Knowledge Discovery. 8(6), e1280, (2018). doi: https://doi.org/10.1002/widm.1280. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1280.

[13] M. J. Kuby, Programming models for facility dispersion: The p-dispersion and maxisum dispersion problems, Geographical Analysis. 19(4), 315–329, (1987). doi: 10.1111/j.1538-4632.1987.tb00133.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1538-4632.1987.tb00133.x.

[14] F. Glover, C.-C. Kuo, and K. S. Dhir, Heuristic algorithms for the maximum diversity problem, Journal of Information and Optimization Sciences. 19(1), 109–132, (1998). doi: 10.1080/02522667.1998.10699366. URL https://doi.org/10.1080/02522667.1998.10699366.

[15] F. Glover, K. Ching-Chung, and K. S. Dhir, A discrete optimization model for preserving biological diversity, Applied Mathematical Modelling. **19**(11), 696–701, (1995). doi: https://doi.org/10.1016/0307-904X(95)00083-V. URL https://www.sciencedirect.com/science/article/pii/0307904X9500083V.

[16] A. Duarte and R. Martí, Tabu search and grasp for the maximum diversity problem, European Journal of Operational Research. 178(1), 71–84, (2007). doi: https://doi.org/10.1016/j.ejor.2006.01.021. URL https://www.sciencedirect.com/science/article/pii/S0377221706000634.

[17] K. Ralls, P. Sunnucks, R. C. Lacy, and R. Frankham, Genetic rescue: A critique of the evidence supports maximizing genetic diversity rather than minimizing the introduction of putatively harmful genetic variation, Biological Conservation. 251, 108784, (2020). doi: https://doi.org/10.1016/j.biocon.2020.108784. URL https://www.sciencedirect.com/science/article/pii/S0006320720308429.

[18] A. A. Hoffmann, A. D. Miller, and A. R. Weeks, Genetic mixing for population management: From genetic rescue to provenancing, Evolutionary Applications. 14(3), 634–652, (2021). doi: https://doi.org/10.1111/eva.13154. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/eva.13154.

[19] T. Leinster and M. W. Meckes, Maximizing diversity in biology and beyond, Entropy. 18(3), (2016). doi: 10.3390/e18030088. URL https://www.mdpi.com/1099-4300/18/3/88.

[20] T. von Ghyczy, Product diversity and proliferation as a new mode of competing in the motor car, Int. J. of Vehicle Design. 6(4/5), 423–425, (1985).

[21] M. Lozano, D. Molina, and C. García-Martínez, Iterated greedy for the maximum diversity problem, European Journal of Operational Research. **214**(1), 31–38, (2011). doi: https://doi.org/10.1016/j.ejor.2011.04.018. URL https://www.sciencedirect.com/science/article/pii/S0377221711003626.

[22] R. Martí, M. Gallego, and A. Duarte. Optsicom project, (2010). URL http://grafo.etsii.urjc.es/optsicom/mdp/.

[23] R. Martí, M. Gallego, and A. Duarte, A branch and bound algorithm for the maximum diversity problem, European Journal of Operational Research. 200(1), 36–44, (2010).

[24] R. Martí, A. Martínez-Gavaraa, S. Pérez-Pelób, and J. Sánchez-Orob, Discrete diversity and dispersion maximization: A review and an empirical analysis from an OR perspective, submitted March 2021. (2021).

[25] Y. Zhou, J.-K. Hao, and B. Duval, Opposition-based memetic search for the maximum diversity problem, IEEE Transactions on Evolutionary Computation. 21(5), 731–745, (2017).

[26] G. Strang, The Singular Value Decomposition (SVD). In: Introduction to Linear Algebra, In Introduction to Linear Algebra, chapter 7, pp. 364–400. Wellesley-Cambridge Press, Wellesley, MA, fifth edition, (2009).

[27] T. Cover and P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory. 13(1), 21–27, (1967). doi: 10.1109/TIT.1967.1053964.

[28] E. Keogh, Instance-Based Learning, In Encyclopedia of Machine Learning, chapter I, pp. 549–550. Springer US, Boston, MA, (2010). doi: 10.1007/978-0-387-30164-8_409. URL https://doi.org/10.1007/978-0-387-30164-8_409.

[29] J. Fernández Pierna, F. Wahl, O. de Noord, and D. Massart, Methods for outlier detection in prediction, Chemometrics and Intelligent Laboratory Systems. 63(1), 27 – 39, (2002). doi: https://doi.org/10.1016/S0169-7439(02)00034-5. URL http://www.sciencedirect.com/science/article/pii/S0169743902000345. Chemometrics 2002 S.I.

[30] R. Fisher. UCI machine learning repository: Iris data set, (1936). URL https://archive.ics.uci.edu/ml/datasets/iris.

[31] M. Charytanowicz, J. Niewczas, P. Kulczycki, P. A. Kowalski, S. Łukasik, and S. Żak. UCI machine learning repository: Seeds data set, (2012). URL https://archive.ics.uci.edu/ml/datasets/seeds.

[32] V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker. UCI machine learning repository: Ionosphere data set, (1989). URL https://archive.ics.uci.edu/ml/datasets/ionosphere.

[33] W. H. Wolberg and O. L. Mangasarian, Multisurface method of pattern separation for medical diagnosis applied to breast cytology, Proc Natl Acad Sci U S A. 87(23), 9193–9196 (Dec, 1990).

[34] M. Elter, R. Schulz-Wendtland, and T. Wittenberg. UCI machine learning repository: Mammographic mass data set, (2007). URL http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass.

[35] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. UCI machine learning repository: Contraceptive method data set, (1999). URL http://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice.

[36] W. J. Nash, T. L. Sellers, S. R. Talbot, A. J. Cawthorn, and W. B. Ford. UCI machine learning repository: Abalone data set, (1994). URL https://archive.ics.uci.edu/ml/datasets/Abalone.

[37] C. Sammut and G. I. Webb, Eds., Accuracy. In: Encyclopedia of Machine Learning, In eds. C. Sammut and G. I. Webb, Encyclopedia of Machine Learning, chapter A, pp. 9–10. Springer US, Boston, MA, (2010).

[38] K. M. Ting, Precision and Recall. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning, In Encyclopedia of Machine Learning, chapter P, pp. 781–781. Springer US, Boston, MA, (2010).

[39] C. Sammut and G. I. Webb, Eds., Mean Squared Error. In: Encyclopedia of Machine Learningr, In eds. C. Sammut and G. I. Webb, Encyclopedia of Machine Learning, chapter M, pp. 653–653. Springer US, Boston, MA, (2010).

[40] C. Sammut and G. I. Webb, Eds., Cross-Validation. In: Encyclopedia of Machine Learning, In eds. C. Sammut and G. I. Webb, Encyclopedia of Machine Learning, chapter C, pp. 249–249. Springer US, Boston, MA, (2010).

[41] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin. 1(6), 80–83, (1945). URL http://www.jstor.org/stable/3001968.