The Multi-Ranked Classifiers Comparison

Norbert Jankowski

Abstract Is it true that everybody knows how to compare classifiers in terms of reliability? Probably not, since it is so common that just after reading a paper we feel that the classifiers' performance analysis is not exhaustive and we would like to see more information or more trustworthy information. The goal of this paper is to propose a method of multi-classifier comparison on several benchmark data sets. The proposed method is trustworthy, deeper, and more informative (multi-aspect). Thanks to this method, we can see much more than overall performance. Today, we need methods which not only answer the question whether a given method is the best, because it almost never is. Apart from the general strength assessment of a learning machine we need to know when (and whether) its performance is outstanding or whether its performance is unique.

1 Introduction

The proposed method of classifiers comparison is based on known statistical elements like accuracy, statistical tests, and rankings in general. For clarity, let us define *accuracy* as the fraction of correctly classified instances to the whole number of instances m:

$$acc(m, D) = 1 - err(m, D) = \frac{1}{m} \sum_{\langle \mathbf{x}, y \rangle \in D, y = m(\mathbf{x})} 1 \tag{1}$$

In highly unbalanced cases (when the numbers of class instances differ significantly), it is recommended to use a balanced version of accuracy:

N. Jankowski (✉)
Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
e-mail: norbert@is.umk.pl

R. Burduk et al. (eds.), *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, Advances in Intelligent Systems and Computing 403, DOI 10.1007/978-3-319-26227-7_11

$$bacc(m, D) = \frac{1}{K} \sum_{k=1}^{K} \frac{\sum_{x \in D^k, m(\mathbf{x}_i)=k} 1}{|D^k|}, \tag{2}$$

where K is the number of classes, and $D^k = \{\langle \mathbf{x}_i, y_i \rangle : \langle \mathbf{x}_i, y_i \rangle \in D \wedge y_i = k\}$ is a set of pairs belonging to the kth class. As it can be seen that an error in classification of an instance of a smaller class is more strongly weighted, accordingly to class counts' proportions. The most common testing tool is the *cross-validation* test which divides randomly a given data set D into p equally counted subsets D_i. In consequence, we obtain p training–testing data set pairs $[D'_i, D_i]$, where $D'_i = D \backslash D_i$. Next, we have p phases of classifier learning and testing. The average accuracy over the testing part defines estimated accuracy ($eacc = 1/p \sum_{i=1}^{p} acc(m_i, D_i)$), where m_i is a classifier learnt on the D'_i data. However, such estimation should not be considered trustworthy, and it is recommended to repeat the cross-validation process q times (usually 10 times). For more about parametrization of cross-validation and their statistical relations see [2]. Now, the accuracy estimation is based on much more tests. Assume that

$$Eacc_m^D = [acc_1, \ldots, acc_{pq}] \tag{3}$$

is a vector of accuracies for all p test parts and for all q repetitions of cross-validation ($p * q$ single tests). It is highly recommended to use a *stratified* version of cross-validation. This test additionally keeps the proportions of classes in subsequent D_i sets to be close to the proportions of class counts in D. To keep the process of classifier comparison as trustworthy as possible it is also recommended to control the seed in the drawing process of training and testing parts. This means that each classifier should be trained and tested on the same training data and testing data. This is even more important when we use statistical tests like paired tests (e.g., paired t-test). In case of paired tests, it is obligatory to train and test all classifiers on the same data. Except the accuracy and the error, the reader should in some cases consider usage of other factors like recall, precision, specificity or confusion matrix, for more see [3, 6]. Statistical tests can serve as an important tool in classifier comparison. The reason for that is quite simple. Let us consider an example of two tests where average accuracies were equal to 0.87 and 0.879. In such case, we cannot directly claim that one of those classifiers is significantly better than the other one, however they differ in accuracy values. It is because the variances of classification of test data for both classifiers have a crucial role. Thanks to the statistical tests, the significance of results can be calculated. For detailed description on how to calculate statistical test, see [2, 8]. The most oftenly used statistical test in the context of machine learning is the t-test, which goal is to check whether the accuracy mean of the first classifier is significantly greater than the accuracy mean of the second classifier (in such case we use the *one tail* test version). The null hypothesis is that the first classifier is not better than the second one. Another possibility is to check whether the accuracy means of two populations are significantly different or not (this is two tail test version). The one tail is slightly more

advised, as we usually have to choose the better one. And to calculate this test for two classifiers m and m' the accuracy differences $Eacc_m^D - Eacc_{m'}^D$ are used (the paired version of test). For classifiers' comparison the t-test can be used in one of two versions: paired or unpaired. The unpaired version has to be used only if the classifiers learned on different draws from data set. Practically, for a machine learning task it is not difficult to use the same distribution for learning and testing. If we used the same data distribution in cross-validation (or Monte Carlo as well), then the paired version is a more reliable test and we should avoid using the unpaired version if possible. The necessary condition to use one of the t-tests is that the test samples are approximately normally distributed. In a case where test samples are not normally distributed, there are two other interesting options. For the paired case, the Mann–Whitney test can be used and for unpaired, the Wilcoxon test. The last two test are ranked tests (accuracies are first transformed to ranks and are then further analyzed). Another resourceful test for computational intelligence which is rarely used is the McNemmar test. This test is designed to analyze whether the correctnesses of instance vectors for two classifiers are not statistically different. In that case, the test is not based on accuracies (or equivalently on errors) but on the correctness of each data instance. Even if two classifiers are characterized by the same means and similar variations they may differ in classification of appropriate instances. To compare more than two classifiers, the Anova test can be used, but its usability is somewhat limited. The base goal of this test is to calculate whether any two classifiers in a group are significantly unequal. The limitation of this test stems from the fact that in a case of comparing several classifiers, we are usually sure that same classifiers will differ, but we are interested in how they differ, not just whether they differ.

1.1 Common Traps in Learning Machine Comparison

The description below mostly concerns classification testing *traps*, but indeed most of the *traps* are of universal behavior. The ultimate solution or a trap? In so many cases a seemingly trustworthy comparison can be easily misleading. There are some types of commonly repeated errors in numerous articles. To avoid those problems in the future we can enumerate some of them.

1. The overall average accuracy as a measure of classifier performance. In some papers average accuracies are averaged over several data sets for given learning machines. Of course this information can be useful, however if we try to compare such averages obtained from two (or more) learning machines, such comparison is not trustworthy. It happens that in case of one or few data sets in a tested group the average accuracies differ strongly between classifiers and then, even if one classifier has a better overall accuracy, realistically, in case of most data sets, its performance may be significantly worst.
2. Another commonly observed scheme of classifier comparison is calculating how many of the given classifiers were not worse (the average accuracy was not smaller significantly) than others. The results of such calculation is the number of wins for all classifiers over several data sets. Such information is really interesting because

winners are certainly positive. However, it is somewhat risky in the following case: assume the first classifier wins a few times more than the second classifier and each win was significantly better, but just by a bit. A problem arises if for the second classifier the wins are much more than just *a bit* significant.

3. In some cases, benchmark data set was originally divided into two parts: the training and testing. If for a given classifier's configuration we train the classifier just once using the training part and then test use the testing part, then the test result is trustworthy. A problem arises when we repeatedly: learn a classifier, test the classifier, then basing on the test results we tune the configuration of the classifier. In such scenarios, researchers do not test different configurations but learn with validation using the testing data as validation data. Presentation of such results means a presentation of unreliable data.
4. One of the very typical errors in comparing classifiers is when the cross-validation testing is prepared after supervised[1] data transformation/preprocessing. Any supervised preprocessing must be embedded inside each cross-validation fold—before every classifier learning, first the data must be preprocessed for each cv-fold. Without following this scenario, the results can differ too strongly and are unreliable.
5. In some articles authors propose a new method but the conclusions are sometimes based just on a few data set benchmarks. However, the authors claim that the method works always and is universal (really?). Such scenario should also raise suspicions—just a few data sets should mean 'sometimes', not almost 'always'. A close problem to the above one is when authors claim that a method is *scalable* while the results are presented only on small data sets and the computational complexity is not investigated and is probably far from linear. Although it happens that a new method is proposed just in context of one problem (one data set) and this scenario can be correct.
6. Another unfair type of construction of comparisons is based on consciously misleading testing procedures. One of the most common examples of such problem is the usage of atypical parametrization of a test. For example, the usage of monte carlo randomization in place of commonly used cross-validation for given benchmarks. Generally, monte carlo randomization is correct, but if in context of the results for benchmark data sets all previous authors used cross-validation, then if we see monte carlo randomization, we can be sure of one thing: we cannot compare those results with the previous ones. They should be considered negligible. Another way of erring in the test procedure is to select different error measures, even though the author knows which measure was selected in previous articles about the considered problem (benchmark). Again, new results will not be trustworthily compared. Of course some of traps are entered unconsciously while others, consciously. I hope the above examples will help to avoid some mistakes in the future. The goal of the following part is to present how to plan a classifier comparison to be clear, trustworthy, informative and deep.

[1] Supervised process (learning of data transformation) means to use the class labels.

2 The Multi-ranked Classifier Comparison

The main goal of this section is to present the multi-ranked classifiers comparison which will analyze a series of classifiers over a sequence of benchmark data sets. Except the standard mean accuracies and the number of wins, we plan to present additional supporting information which significantly simplify the estimation of the role of a given classifier compared to others. Let us assume we have a sequence of benchmark data sets D_i ($i \in \{1, \ldots d\}$) and a series of classifiers m_j ($j \in \{1, \ldots, T\}$). First, as usually, we need the accuracy vectors from cross-validation tests for every classifier and for every benchmark data set. This gives us a matrix of $Eacc_{m_j}^{D_i}$ (remember that $Eacc_{m_j}^{D_i}$ is a vector, not a scalar). The matrix of mean test accuracies $\bar{a}_j^i$ for a machine m_j and benchmark D_i is the base of further presentation. Strictly, the $\bar{a}_j^i$ is the mean of $Eacc_{m_j}^{D_i}$ vector accuracies. Additionally, we define the σ_j^i to be the standard deviation of $Eacc_{m_j}^{D_i}$.

2.1 Machine Ranks and Significance Groups

For given data D_i, we can group machines in accordance with their mean accuracies using the paired t-test.[2] Such groups will be assigned to a rank. We can define the rank assignment for machines as follows:

- the machine with highest accuracy mean is ranked 1,
- all machines whose accuracy means are not significantly smaller (measured with t-test) are also ranked with 1,
- rank 2 is assigned to the machine of highest accuracy amongst those whose accuracy is significantly smaller than the machine's first ranked with 1,
- rank 2 is assigned to all machines which have not been ranked yet, whose accuracies are insignificantly smaller than the first machine's ranked with 2,
- the following ranks are assigned in the same way.

All machines with the same rank compose a *significance group*. Let's define r_j^i as the rank of machine m_j obtained for benchmark data D_i. Such ranks forms rank groups, each group is composed of machines which are characterized by the same (insignificantly different) performance. This feature is so important because for different benchmark data sets the spread between accuracies varies. Additionally, ranks are independent of the differences between mean accuracies of different benchmark data sets. This feature is important for comparing the results for two (or more) benchmarks, basing on ranks, instead of comparing mean accuracies for different

[2] To compute paired t-test for machine s and t and data D_i use the vector of differences: $Eacc_{m_s}^{D_i} - Eacc_{m_t}^{D_i}$.

benchmarks. Additionally, let us define $\bar{r}_j$ to be a mean rank for machine m_j across benchmarks and σ_j^r as the standard deviation of ranks for a given machine m_j. The mean ranking is the best estimate of overall performance. If it is close to 1, it means that the machine is usually the winning one. And standard deviation informs us of the changes across benchmark data sets. Winners versus ranking: Typically, authors of classifier comparisons use the division into two parts for a given benchmark: winners (machines with best accuracies-insignificantly different) and losers (machines which perform significantly worse than the best one). Such binary spread is sometimes not adequate—in some cases the mean accuracies naturally form more than two groups of performance and division into two groups in fact hides some information. The reader will able to observe in the example below that in case of some benchmarks, the ranks form several groups of performance which reflect several levels of performance degradation. And across several benchmarks we can simply observe how frequently the performance of a given classifier degraded and how deeply.

2.2 Winners and Unique Winners

Observation of machines which win for given data sets is important, but apart from the observation of winners we should also observe machines which are unique winners. A unique winner is a machine which is the best for a given data set and no other machine is insignificantly worse. Such machines are not redundant in contrary to nonunique winners, which can be substituted by another machine(-s). Define the w_i to be the number of wins for machine m_i (win means that machine is the best one for given data or insignificantly worse). Define the u_i to be a count of unique wins of machine m_i, which is the number of wins while no other machine has the same rank (unique win means that only one machine has rank 1 for given benchmark).

2.3 Multi-ranked Classifiers Comparison

The above part of this section has presented all necessary definitions to present the proposed classifier comparison. This comparison will consist of

- mean accuracy and standard deviation for each machine and each benchmark with its rank,
- overall mean accuracy per machine with its standard deviation,
- overall mean rank per machine with its standard deviation,
- wins count and unique wins count.

All this information is nested in the matrix below:

	m_1	m_2	$\cdots$	m_p
D_1	$\bar{a}_1^1 \pm \sigma_1^1(r_1^1)$	$\bar{a}_2^1 \pm \sigma_2^1(r_2^1)$	$\cdots$	$\bar{a}_p^1 \pm \sigma_p^1(r_p^1)$
D_2	$\bar{a}_1^2 \pm \sigma_1^2(r_1^2)$	$\bar{a}_2^2 \pm \sigma_2^2(r_2^2)$	$\cdots$	$\bar{a}_p^2 \pm \sigma_p^2(r_p^2)$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
D_q	$\bar{a}_1^q \pm \sigma_1^q(r_1^q)$	$\bar{a}_2^q \pm \sigma_2^q(r_2^q)$	$\cdots$	$\bar{a}_p^q \pm \sigma_p^q(r_p^q)$
Mean Accuracy	$\bar{a}_1^* \pm \sigma_1^*$	$\bar{a}_2^* \pm \sigma_2^*$	$\cdots$	$\bar{a}_p^* \pm \sigma_p^*$
Mean Rank	$\bar{r}_1 \pm \sigma_1^r$	$\bar{r}_2 \pm \sigma_2^r$	$\cdots$	$\bar{r}_p \pm \sigma_p^r$
Wins[unique wins]	$w_1[u_1]$	$w_2[u_2]$	$\cdots$	$w_p[u_p]$

(4)

2.4 An Example of Multi-ranked Comparison

Probably, the best way to see the attractiveness of the presented comparison method is to analyze a real world example. 40 benchmark data sets from the UCI machine learning repository [7] were selected to present the comparison below. Two neural networks, k Nearest neighbor [1], and two types of Support Vector Machines (linear and gaussian) [9] were selected to compare with the proposed method. The first neural network is a simple linear model (no hidden layer) learned by pseudo-inverse matrix (via singular values decomposition). The linear model $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ is learned by:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{X}^\dagger\mathbf{y} \tag{5}$$

where $\mathbf{X}$ is a matrix of input data, $\mathbf{y}$ label (class) vector and $\mathbf{X}^\dagger$ is pseudo-inverse matrix. The above equation is a solution for the goal:

$$J_s(\mathbf{w}) = ||\mathbf{X}\mathbf{w} - \mathbf{y}||^2 = \sum_{i=1}^{m}(\mathbf{w}^T\mathbf{x}_i - y_i)^2 \tag{6}$$

obtained by zeroing the gradient. The next neural network is a nonlinear model generated by a set of gaussian kernels $(k_1, \ldots, k_l)$, and learned in a similar way as above networks after transforming the original space into the space obtained by kernels. It means that instead of $\mathbf{X}$ in Eq. 5 the matrix F is used:

$$F_{ij} = k_j(\mathbf{x}_{z_j}; \mathbf{x}_i), \tag{7}$$

where $\mathbf{x}_{z_j}$ are randomly selected between all data vectors. Such construction of neural networks is equivalent to Extreme learning machines [4, 5]. Note that the parameters of all learning machines were not optimized because the goal of this paper is not to achieve optimal performance of given machines, but to present the attractiveness of the comparison method.

Table 1 The multi-ranked classifiers comparison

	NN-linear	NN-Gauss	kNN	L-SVM	SVM
Autos	71.06±9.11(2)	**77.37±9.37(1)**	64.75±10.8(3)	53.31±10.4(5)	59.2±9.61(4)
Balance-scale	49.44±5.06(5)	**90.8±1.56(1)**	88.81±2.45(3)	84.48±2.84(4)	89.5±1.89(2)
Breast-cancer-diagnostic	43.01±2.91(5)	90.65±3.29(4)	96.8±2.26(2)	**97.4±1.93(1)**	95.9±2.47(3)
Breast-cancer-original	85.18±3.52(3)	95.74±2.34(2)	**96.82±2.01(1)**	**96.65±2.21(1)**	**96.97±1.98(1)**
Breast-cancer-prognostic	76.59±4.36(2)	75.87±5.79(2)	76.39±8.11(2)	**80.14±8.54(1)**	76.38±4.04(2)
Breast-tissue	**64.43±13.3(1)**	52.45±15.7(2)	**66.77±14.3(1)**	43.42±7.92(3)	42.45±8.79(3)
Car-evaluation	75.01±2.11(5)	89.02±1.63(2)	**93.79±1.31(1)**	82.9±2.31(4)	88.49±1.53(3)
Cardiotocography-1	73.31±3(3)	74.02±2.62(2)	**76.09±3.08(1)**	58.79±2.53(5)	71.65±2.3(4)
Cardiotocography-2	40.84±15.4(5)	88.93±1.69(3)	**91.15±1.5(1)**	87.6±1.52(4)	90.57±1.51(2)
Chess-rook-versus-pawn	66.45±9.28(5)	70.4±1.96(4)	94.86±1.17(3)	96.92±0.855(2)	**98.42±0.726(1)**
CMC	49±3.8(2)	**55.35±3.44(1)**	49.45±4.21(2)	19.24±2.6(4)	30.67±3.07(3)
Vongressional-voting	70.25±7.49(4)	94.53±4.37(2)	91.91±5.07(3)	95.04±3.97(2)	**96.3±3.22(1)**
Connectionist-bench-sonar	52.43±5.05(5)	57.59±4.91(4)	**82.65±6.79(1)**	75.86±7.94(3)	78.55±6.79(2)
Connectionist-bench-vowel	53.66±5(4)	87.48±4.25(2)	**94.72±2.65(1)**	26.66±3.77(5)	64.32±4.69(3)
Cylinder-bands	73.42±8.3(2)	36.98±2.22(5)	64.93±8.44(4)	**76.43±7.71(1)**	67.14±2.55(3)
Dermatology	**93.43±4.1(1)**	**94.07±3.99(1)**	92.82±3.87(2)	**93.4±3.99(1)**	86.69±5.23(3)
Ecoli	**86.13±5.05(1)**	**85.46±5.17(1)**	**85.75±5.29(1)**	76.06±6.62(3)	83.25±5.23(2)
Glass	54.46±8.39(4)	**67.83±9.13(1)**	65.68±7.65(2)	35.67±6.52(5)	57.22±8.07(3)
Habermans-survival	**73.78±2.82(1)**	71.65±6.4(2)	70.99±6.44(3)	72.46±2.77(2)	**73.77±3.93(1)**
Hepatitis	83.75±10.9(3)	**91.25±9.32(1)**	87.38±12.1(2)	81.63±10.3(3)	88.25±9.03(2)
Ionosphere	73.3±4.54(4)	66.55±3.4(5)	84.67±4.56(3)	88.15±4.75(2)	**94.87±3.65(1)**

(continued)

Table 1 (continued)

	NN-linear	NN-Gauss	kNN	L-SVM	SVM
Iris	51.73±8.79(4)	94.33±5.22(2)	**95±5.79(1)**	77.8±7.58(3)	**96.13±5.03(1)**
Libras-movement	65.81±6.7(3)	67.75±7.37(2)	**77.11±5.99(1)**	50.33±6.39(4)	47.97±5.97(5)
Liver-disorders	43.34±2.37(4)	67.55±8.25(2)	61.02±7.74(3)	69.22±5.68(2)	**71.13±6.41(1)**
Lymph	80.1±9.6(2)	**85.65±10(1)**	80.47±10(2)	80.17±9.75(2)	79.97±10.1(2)
Monks-problems-1	50±0.81(5)	96.42±2.7(3)	99.66±0.838(2)	74.64±3.5(4)	**100±0(1)**
Monks-problems-2	**65.73±0.877(1)**	64.01±4.55(2)	56.18±4.47(4)	**65.73±0.877(1)**	60.88±4.22(3)
Monks-problems-3	60.3±3.69(3)	98.09±1.93(2)	**98.77±1.2(1)**	**98.92±1.12(1)**	**98.92±1.12(1)**
Parkinsons	82.67±4.86(4)	**92.41±5.48(1)**	**91.82±6.37(1)**	87.79±6.29(3)	89.81±5.19(2)
Pima-Indians-diabetes	36.13±1.66(5)	74.95±4.78(3)	73.97±4.52(4)	**77.24±4.76(1)**	76.47±4.65(2)
Sonar	52.43±5.05(5)	57.59±4.91(4)	**82.65±6.79(1)**	75.86±7.94(3)	78.55±6.79(2)
Spambase	43.2±0.807(5)	68.12±1.92(4)	90.92±1.29(3)	**92.79±0.962(1)**	91.65±1.2(2)
Spect-heart	79.42±2.05(2)	**82.13±6.66(1)**	**81.84±7.21(1)**	**80.86±7.43(1)**	**82.58±6.89(1)**
Spectf-heart	79.27±2.17(2)	**80.61±5.93(1)**	72.73±7.47(3)	79±6.96(2)	78.22±5.25(2)
Statlog-Australian-credit	62.04±3.76(5)	71.64±4.4(4)	79.61±4.44(3)	**84.74±4.42(1)**	83±4.32(2)
Statlog-German-credit	70.59±1.04(4)	70.74±2.77(4)	72.4±3.7(3)	**76.55±3.91(1)**	75.21±2.82(2)
Statlog-heart	78.3±7.16(2)	79.15±7.73(2)	**82.15±7.9(1)**	**83.67±6.7(1)**	**83.44±7.53(1)**
Statlog-vehicle	59.65±3.99(5)	**75.87±4.47(1)**	72.95±4.06(2)	68.4±4.89(3)	66.05±3.84(4)
Teaching-assistant	**54.99±11.7(1)**	**56.82±10.3(1)**	42.4±11.5(3)	54.09±12(2)	40.44±12.4(3)
Thyroid-disease	14.76±26.3(5)	95.25±0.584(2)	94.94±0.499(3)	93.7±0.35(4)	**95.41±0.526(1)**
Mean accuracy	63.48±5.92	77.33±5.06	80.59±5.4	74.84±5.09	78.16±4.61
Mean rank	3.35±0.24	2.25±0.2	2.1±0.16	2.525±0.22	2.175±0.17
Wins [unique]	6[0]	13[8]	15[7]	13[7]	12[6]

Table 2 The multi-ranked classifiers comparison—version II

	NN-linear	NN-Gauss	kNN	L-SVM	SVM
Autos	71.06±9.11(2)	**83.92±9.1(1)**	64.75±10.8(3)	53.31±10.4(5)	59.2±9.61(4)
Balance-scale	49.44±5.06(5)	**90.21±1.93(1)**	88.81±2.45(3)	84.48±2.84(4)	89.5±1.89(2)
Breast-cancer-diagnostic	43.01±2.91(5)	92.13±3.01(4)	96.8±2.26(2)	**97.4±1.93(1)**	95.9±2.47(3)
Breast-cancer-original	85.18±3.52(3)	95.48±2.49(2)	**96.82±2.01(1)**	**96.65±2.21(1)**	**96.97±1.98(1)**
Breast-cancer-prognostic	76.59±4.36(2)	76.55±6.55(2)	76.39±8.11(2)	**80.14±8.54(1)**	76.38±4.04(2)
Breast-tissue	**64.43±13.3(1)**	48.94±16(2)	**66.77±14.3(1)**	43.42±7.92(3)	42.45±8.79(3)
Car-evaluation	75.01±2.11(5)	92.56±1.61(2)	**93.79±1.31(1)**	82.9±2.31(4)	88.49±1.53(3)
Cardiotocography-1	73.31±3(3)	**78.21±2.37(1)**	76.09±3.08(2)	58.79±2.53(5)	71.65±2.3(4)
Cardiotocography-2	40.84±15.4(4)	**90.93±1.57(1)**	**91.15±1.5(1)**	87.6±1.52(3)	90.57±1.51(2)
Chess-rook-versus-pawn	66.45±9.28(5)	77.16±1.94(4)	94.86±1.17(3)	96.92±0.855(2)	**98.42±0.726(1)**
CMC	49±3.8(2)	**54.26±3.59(1)**	49.45±4.21(2)	19.24±2.6(4)	30.67±3.07(3)
Congressional-voting	70.25±7.49(4)	**95.66±3.58(1)**	91.91±5.07(3)	95.04±3.97(2)	**96.3±3.22(1)**
Connectionist-bench-sonar	52.43±5.05(5)	60.96±5.04(4)	**82.65±6.79(1)**	75.86±7.94(3)	78.55±6.79(2)
Connectionist-bench-vowel	53.66±5(4)	**96.82±2.31(1)**	94.72±2.65(2)	26.66±3.77(5)	64.32±4.69(3)
Cylinder-bands	73.42±8.3(2)	38.49±2.39(5)	64.93±8.44(4)	**76.43±7.71(1)**	67.14±2.55(3)
Dermatology	93.43±4.1(2)	**95.13±3.43(1)**	92.82±3.87(2)	93.4±3.99(2)	86.69±5.23(3)
Ecoli	**86.13±5.05(1)**	82.55±5.32(2)	**85.75±5.29(1)**	76.06±6.62(3)	83.25±5.23(2)
Glass	54.46±8.39(3)	**64.51±7.74(1)**	**65.68±7.65(1)**	35.67±6.52(4)	57.22±8.07(2)
Habermans-survival	**73.78±2.82(1)**	70.57±6.32(3)	70.99±6.44(3)	72.46±2.77(2)	**73.77±3.93(1)**
Hepatitis	83.75±10.9(3)	**91.25±9.32(1)**	87.38±12.1(2)	81.63±10.3(3)	88.25±9.03(2)
Ionosphere	73.3±4.54(4)	67.27±4.3(5)	84.67±4.56(3)	88.15±4.75(2)	**94.87±3.65(1)**
Iris	51.73±8.79(4)	94.47±5.36(2)	**95±5.79(1)**	77.8±7.58(3)	**96.13±5.03(1)**
Libras-movement	65.81±6.7(3)	**80.92±6.12(1)**	77.11±5.99(2)	50.33±6.39(4)	47.97±5.97(5)
Liver-disorders	43.34±2.37(5)	64.4±7.5(3)	61.02±7.74(4)	69.22±5.68(2)	**71.13±6.41(1)**

(continued)

Table 2 (continued)

	NN-linear	NN-Gauss	kNN	L-SVM	SVM
Lymph	80.1±9.6(2)	**84.84±9.33(1)**	80.47±10(2)	80.17±9.75(2)	79.97±10.1(2)
Monks-problems-1	50±0.81(5)	99.87±0.526(2)	99.66±0.838(3)	74.64±3.5(4)	**100±0(1)**
Monks-problems-2	**65.73±0.877(1)**	64.02±5.9(2)	56.18±4.47(4)	**65.73±0.877(1)**	60.88±4.22(3)
Monks-problems-3	60.3±3.69(3)	98.61±1.31(2)	**98.77±1.2(1)**	**98.92±1.12(1)**	**98.92±1.12(1)**
Parkinsons	82.67±4.86(5)	**94.84±4.47(1)**	91.82±6.37(2)	87.79±6.29(4)	89.81±5.19(3)
Pima-Indians-diabetes	36.13±1.66(4)	74.17±4.65(3)	73.97±4.52(3)	**77.24±4.76(1)**	76.47±4.65(2)
Sonar	52.43±5.05(5)	60.96±5.04(4)	**82.65±6.79(1)**	75.86±7.94(3)	78.55±6.79(2)
Spambase	43.2±0.807(5)	71.94±1.94(4)	90.92±1.29(3)	**92.79±0.962(1)**	91.65±1.2(2)
Spect-heart	79.42±2.05(2)	**82.17±6.19(1)**	**81.84±7.21(1)**	**80.86±7.43(1)**	**82.58±6.89(1)**
Spectf-heart	**79.27±2.17(1)**	**79.89±6.18(1)**	72.73±7.47(3)	**79±6.96(1)**	78.22±5.25(2)
Statlog-Australian-credit	62.04±3.76(5)	75.29±4.47(4)	79.61±4.44(3)	**84.74±4.42(1)**	83±4.32(2)
Statlog-German-credit	70.59±1.04(4)	71.72±3.3(3)	72.4±3.7(3)	**76.55±3.91(1)**	75.21±2.82(2)
Statlog-heart	78.3±7.16(2)	78.37±7.44(2)	**82.15±7.9(1)**	**83.67±6.7(1)**	**83.44±7.53(1)**
Statlog-vehicle	59.65±3.99(5)	**79.22±3.69(1)**	72.95±4.06(2)	68.4±4.89(3)	66.05±3.84(4)
Teaching-assistant	54.99±11.7(2)	**63.65±10.3(1)**	42.4±11.5(3)	54.09±12(2)	40.44±12.4(3)
Thyroid-disease	14.76±26.3(5)	**95.77±0.595(1)**	94.94±0.499(3)	93.7±0.35(4)	95.41±0.526(2)
Mean accuracy	63.48±5.92	78.97±4.86	80.59±5.4	74.84±5.09	78.16±4.61
Mean rank	3.35±0.23	2.1±0.2	2.2±0.15	2.5±0.21	2.2±0.16
Wins [unique]	5[0]	18[13]	12[3]	13[7]	11[4]

All results in accordance with the above definitions was presented in Tables 1 and 2 which have the same form as matrix in the Eq. 4. The difference between tables lies in the number of kernels used to learn the NN-Gauss neural network—the numbers of kernels were equal to 80 and 160, respectively. Starting from the top of Table 1 we can analyze significance groups for selected benchmark data. In contrary to a presentation based only on wins and defeats here we can observe that in case of several benchmarks data the numbers of significance groups spread from 2 even up to 5 (5 is the maximum of course). The number of significance groups is very often relatively huge—it is close to the maximum—and is directly related to the diversity of model's performance. Divergent performance of quality is directly correlated with the numbers of significance groups. In case of a presentation based on wins and defeats, this feature is invisible. After the rows which present accuracies statistics and significance groups we come to the sum-up information. The first row informs about commonly used average accuracies over all benchmark data sets. Next row presents the information about the average rank for each classifier. The best ranking informs us about the best classifier over all benchmarks. Note that the best average accuracy over all benchmarks may not be as good an estimation of the best classifier as the machine with the best average rank. It is because the magnitudes of average accuracy for machines are independent, which can significantly bias the rank, and this can be seen in Table 2. The last row informs us about the number of wins and the number of unique wins. The best number of wins is quite closely related to the best rank. But more special information is captured by the unique wins. This informs us about the uniqueness of a given machine. Larger number of unique wins means a more unique and more significant machine. If the number of unique wins is really small, it means that such machine can be simply substituted by another machine. It shows that machine redundancy can be very easily analyzed. Compare the two tables to see how the redundancy can change. Additionally, all non-small unique win counters are connected with nonredundant winning machines.

3 Summary

Model selection is one of most important tasks in machine learning and computational intelligence. The comparison of classifiers should as informative as possible, and should not hide any important information. Typically, we observe that classifiers comparisons are oversimplified and in consequence, to select a model, we need another results which comment the behavior of learning and the obtained results. The proposed scheme of multi-ranked classifiers comparison bases on the same statistical tools but calculates more and different features for the prepared test. Thanks to the proposed scheme, we can easily analyze information like

- significance groups which describe difference in performance for a given benchmark without the bias of variance,
- the overall best classifier information is based mostly on the averaged ranks, which may be additionally compared with the win counts,
- machine uniqueness and machine redundancy,
- the best winner machine (the machine with the most wins),
- detailed information about performance for a given machine and a given benchmark.

Such classifier comparison significantly simplifies the process of results analysis and the model(-s) selection is simplified.

References

1. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. Inst. Electr. Electron. Eng. Trans. Inf. Theory **13**(1), 21–27 (1967)
2. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Comput. **10**(7), 1895–1923 (1998)
3. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2001)
4. Huang, G.-B., Zhu, Q.-Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: International Joint Conference on Neural Networks, pp. 985–990. IEEE Press (2004)
5. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing **70**, 489–501 (2006)
6. Larose, D.: Discovering Knowledge in Data. An Introduction to Data Mining. Wiley, New York (2005)
7. Merz, C.J., Murphy, P.M.: UCI repository of machine learning databases (1998). http://www.ics.uci.edu/~mlearn/MLRepository.html
8. Montgomery, D.C., Runger, G.C.: Applied Statistics and Probability for Engineers. Wiley, New York (2002)
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)