2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

New Untrained Aggregation Methods for Classifier Combination

Bartosz Krawczyk, Michał Woźniak

***Abstract*— The combined classification is a promising direction in pattern recognition and there are numerous methods that deal with forming classifier ensembles. The most popular approaches employ voting, where the final decision of compound classifier is a combination of individual classifiers' outputs, i.e., class labels or support functions. This paper concentrates on the problem how to design an effective combination rule, which takes into consideration the values of support functions returned by the individual classifiers. Because in many practical tasks we do not have a training set at our disposal, then we express our interest in aggregation methods which do not require learning. A special attention is paid to weighted aggregation, especially when the different weights depend on particular support function of a given individual classifier. We propose a novel approach for untrained combination of support functions using the Gaussian function to assign mentioned above weights. The computer experiments carried out on the set of benchmark data sets confirm the advantages of the proposed approach for particular cases, especially when the number of class labels is high.**

I. INTRODUCTION

There are numerous propositions on how to automate the classification process. Nevertheless, according to Wolpert's *no free lunch theorem* there is not a single classifier that is appropriate for all the tasks we are facing, since each classifier has its own domain of competence [1]. For a given classification task we often have a pool of classifiers at our disposal. What is interesting, the set of incorrectly classified objects by all individual classifiers is typically small [2]. This observation means that even if individual classifiers do not have high qualities, they could produce a considerably good compound classifier, e.g., by nominating the most competent individual classifier for a given object. This approach is called classifier selection and was firstly described in [3]. Notwithstanding, the problem is how to find the most competent classifier.

The considered approach is called a multiple classifier system (MCS), combined classifier or classifier ensemble [4] and this idea is highlighted as a hot topic in machine learning [5], [6], [7].

We face three main groups of problems connected with the MCS design process:

- How the individual classifiers should be interconnected in the ensemble. Most of the applications use the parallel topology, but we should notice that it is possible to connect the individual classifiers using a serial or a hybrid architecture [8]. In the case of serial topology, we can order the individual classifiers according to their cost and ask them for the decision sequentially. We start asking from the cheapest classifier and if the decision is not robust enough then the next classifier is asked to make a decision.
- The next issue focuses on how to select a valuable pool of individual classifiers. Usually, we can collect a huge number of classifiers. However, we should notice that such a pool should consist of diverse and high quality classifiers, i.e., mutually complementary models, because apart from increasing the computational complexity, combining similar classifiers will not contribute much to the MCS under construction. We can use so-called diversity measures [9], [10] to select classifiers to ensemble. This measure should assure that a pool of the selected classifiers has desired properties. Here, we have to notice that there is not an universal diversity measure for all classification tasks and formulating more appropriate diversity measures is an ongoing challenge for the pattern recognition community. Additionally, process of classifier selection (known as ensemble pruning) still waits for a proper attention as well. Due to the high computational complexity of full-search over all of the possible classifier subgroups, several heuristic groups of methods were proposed [11]:
  - Ranking-based methods use an evaluation measure for classifier ranking and choose only the first best ones.
  - Clustering-based methods cluster a pool of classifiers according to the criterion based on a chosen diversity measure and then prune each of the cluster.
  - Optimization-based methods consider ensemble pruning as an optimization problem and apply heuristic techniques to solve it.
- The last concern is related to designing a combination rule, aimed at creating a mechanism that can exploit the strengths of the selected classifiers and combine them optimally, i.e, the greatest effort is concentrated on combining the outputs of elementary classifiers.

In this work we focus on the last issue, i.e., the collective decision making method, which is called the combination rule, fuser or combiner. Nevertheless, we have to emphasize that each of the mentioned above concern is important and has a significant impact in the quality of MSC. Let us systematize methods of classifier combination, taking into consideration what kind of the information is used to

Bartosz Krawczyk and Michał Woźniak are with the Department of Systems and Computer Networks, Wrocław University of Technology, Poland (email: {bartosz.krawczyk,michal.wozniak}@pwr.wroc.pl).

This work was supported by the Polish National Science Center under the grant no. DEC-2013/09/B/ST6/02264. The work was supported by EC under FP7, Coordination and Support Action, Grant Agreement Number 316097, ENGINE European Research Centre of Network Intelligence for Innovation Enhancement (http:// engine.pwr.wroc.pl/)

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

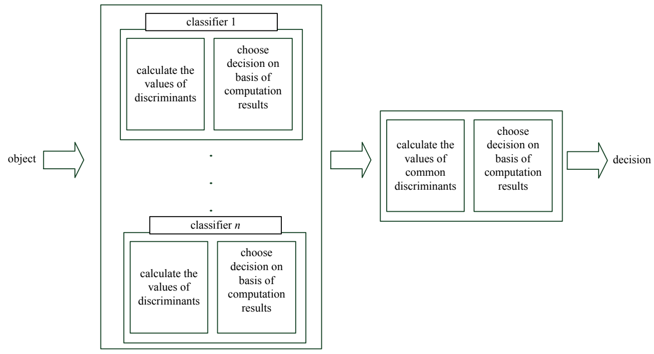

Fig. 1. Idea of a combined classifier based on support functions [7].

establish the decision of a combination rule.

- Methods that make decisions on the basis of class labels returned by the individual classifiers.
- Methods that propose constructing new support functions based on supports returned by the individual classifiers.

The former group includes mainly voting algorithms [12], [13]. Many works have been devoted to combined classifier quality, but most of them formulated the conclusions under the strong and convenient assumptions, such as particular cases of the majority vote [14]. Unfortunately, such assumptions and restrictions are useless to solve typical practical problems. We should also mention those works that suggest to train the weights [15], usually assigned to individual classifiers, which seems to be an attractive alternative method [16], [17]. Initially only voting schemes were implemented, but in later works more advanced methods were proposed as *Multistage Organization with Majority Voting* [18] or the *Behavior Knowledge Space* [19].

The second group of combination methods is based on support functions. Firstly, the Borda count [20] should be mentioned, which makes a decision on the basis of class ranks returned by each classifiers. In general the support function is a measure of support given in favor of a distinguished class, as neural network output, *posterior* probability [21] or fuzzy membership function. There are many approaches dealing with this problem as [22], in which the optimal projective fuser was presented. Opitz and Shavlik [23] combined neural networks outputs according to their accuracy, while Rokach and Maimon [24] employ probabilistic approach. Several analytical properties of aggregating methods were discussed e.g. in [25], [26]. Basically, the aggregating methods, which do not require a learning procedure, use simple operators as the maximum, minimum, sum, product, or average value. Also, the framework of the combined classifier based on support functions of individual classifiers is depicted in Fig. I. It is worth mentioning that an important approach called '*mixture of experts*' [27] combines classifiers' outputs using so-called *gating function* depending on classifiers' input. Moreover, Tresp and Taniguchi [28] proposed a linear weighted function for this model. On the basis of the mentioned model, Cheeseman [29] proposed the *mixture of Gaussian*. We should also mention the *Decision Templates* [30] which estimates the most typical profiles for each support function values (returned by individual classifiers) for each of the class and to make a decision the most similar profile (to the set of supports returned by individual classifiers for a given observation) should be find. Of course, the crucial component of this approach is so-called similarity measure used to finding the most similar decision profile.

In this work we concentrate on an alternative approach, called the weighted aggregating. The main contribution of this work is the proposition of the novel weighted aggregation operators which do not require learning and their evaluation on the basis of computer experiments carried out on the pool of benchmark data sets which confirm the usefulness of proposed approach for a particular classification tasks.

The next section presents a short introduction to weighted aggregating, discusses the motivations of the research, and includes introduction into proposed combination rule. Then, the results of the experimental comparative study of proposed combiners and well known untrained aggregation methods are presented. The last section concludes the paper.

II. UNTRAINED AGGREGATION METHODS

In this section we focus on the weighted aggregating, therefore let's assume that each individual classifier makes a decision on the basis of the values of support functions.

A. Weighted aggregation

Let $\Pi = \left\{\Psi^{(1)}, \Psi^{(2)}, ..., \Psi^{(n)}\right\}$ be the pool of n individual classifiers and $F_{i,k}(x)$ stands for a support function that is assigned to class i $(i \in \mathcal{M} = \{1, ..., M\})$ for a given observation x $\left(x \in \mathcal{X} = \left[x^{(1),...,x^{(d)}}\right]^d\right)$, and which is used by the k-th classifier $\Psi^{(k)}$ from the pool Π.

The combined classifier $\Psi(x)$ uses the following decision rule

$$\Psi(x) = \arg\max_{k \in M} F_k(x), \tag{1}$$

where $F_k(x)$ is the weighted combination of the support functions of the individual classifiers from Π for the class k . Let us consider the possibilities of weight assigning [17]:

1) **Weights dependent on classifier** - this is a traditional approach where weights are connected with classifier and each support function of the k-th classifier is weighted by the same value w_k.
2) **Weights dependent on classifier and feature vector** - weight $w_k(x))$ is assigned to the k-th classifier and for a given x which has the same value for each support function used by it.
3) **Weights dependent on classifier and class number** - weight $w_{i,k}$ is assigned to the k-th classifier and the i-th class. For a given classifier, weights assigned for different classes could be different.

4) **Weights dependent on classifier, class number, and feature vector** - weight $w_{i,k}(x)$ is assigned to the k-th classifier, but for given x its value could be diverse for different support functions assigned to each class.

It is worth noting that Wozniak and Zmyslony [31] argued that the most promising direction is that the weights depend on the classifier number and the class label, because cases where weights are dependent on the feature vector are *de facto* function estimation problems that require additional assumptions about them and which usually lead to a parametric case of function estimation.

$$F_i(x) = \sum_{k=1}^{n} w_{i,k} F_{i,k}(x) \text{ and } \forall i \in \mathcal{M} \sum_{k=1}^{n} w_{i,k} = 1. \quad (2)$$

The considered case does not require any additional assumptions and the formulation of the optimization task is quite simple. They proposed two simple frameworks of combination rule learning, the fist one employs neural approach, while the second one exploit strength of evolutionary approach. The set of computer experiments carried out on the wide range of benchmark data set confirmed the usefulness and quality of proposed methods. Nevertheless, such approach requires additional learning information, usually in the form of learning set to train the combination rule. For many practical cases such a set could be unavailable, while we can have a pool of already trained classifiers at our disposal. Although, we do not have additional learning examples the aggregating operators are *de facto* weighted aggregating taking into consideration weights dependent on classifier, class label, and feature values.

$$F_i(x) = \sum_{k=1}^{n} w_{i,k}(x) F_{i,k}(x) \text{ and } \forall i \in \mathcal{M} \sum_{k=1}^{n} w_{i,k}(x) = 1. \quad (3)$$

Here, we have to stress that such operators assign the weights arbitrary, i.e., weight values are not results of the learning task. Therefore untrained aggregation methods can be only applied in such a case. On the other hand, simple aggregation are subject to very restrictive conditions what severely limits they practical use [32].

B. Proposed combination rules

In this work we employ the forth proposition of weight assigning. Therefore, we should look for new untrained aggregation operators which could exploit the competencies of the individual classifiers. The simple operators as maximum or average usually behave reasonably well but their work could be spoil by very unprecise estimators of the support functions used by only a few classifiers from a pool. Therefore we propose the modifications of the mentioned above operators which take into consideration all available support functions returned by the individual classifiers from the pool, but the functions which have the similar values to maximum or average have the strongest impact in the final value of the common support function calculated by using eq. 3. The proposed operators are called N-AVG and N-MAX and can be calculated according to the Alg. 1. The only difference is the calculation of the $\overline{F}_i(x)$.
For N-AVG it is calculated according to

$$\overline{F}_i(x) = \frac{\sum_{k=1}^{N} F_{i,k}(x)}{N}, \quad (4)$$

and for N-MAX using the following formulae

$$\overline{F}_i(x) = \max_{k \in \mathcal{M}} F_{i,k}(x). \quad (5)$$

Algorithm 1 General framework for the weight calculation

Require: Π - pool of elementary classifiers
$F_{i,k}(x)$ - support function value for each class i returned by each individual classifier k from Π
Ensure: $w_{i,k}(x)$ - weights assigned to each support function $F_{i,k}(x)$ which could be used in eq.3

for $i := 1$ **to** M **do**
  $w := 0$
  Calculate $\overline{F}_i(x)$ according to eq. 5 for N-MAX or according to eq. 4 for N-AVG
  for $k := 1$ **to** n **do**
    $w_{i,k}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(F_{i,k}(x) - \overline{F}_i(x))}{2\sigma^2}\right)$
    $w := w + w_{i,k}(x)$
  end for
  for $k := 1$ **to** n **do**
    $w_{i,k} := \frac{w_{i,k}(x)}{w}$
  end for
end for

The only parameter of the proposed operators is σ which equivalent of standard deviation in normal distribution. For our research we fix $\sigma = 0.5$.

III. Experimental Investigations

The aims of the experiment were to check the performance of the two proposed aggregation operators N-AVG and N-MAX and to compare them with several popular methods for aggregating classifiers.

A. Datasets

In total we chose 10 datasets from the UCI Repository. For datasets with missing values (autos, cleveland and dermatology), instances without full set of features available were removed. Details of the chosen datasets are given in Table I.

B. Set-up

As a base classifier, we have decided to use neural network (NN) - realized as a multi-layer perceptron, trained with back-propagation algorithm, with number of neurons depending on the considered dataset: in the input layer equal to the number of features, in the output layer equal to the number of classes and in the hidden layer equal to half of the sum of neurons in previously mentioned layers. Each model was initialized with random starting values and their

TABLE I

DETAILS OF DATASETS USED IN THE EXPERIMENTS.

No.	Name	Objects	Features	Classes
1.	Autos	159	25	6
2.	Car	1728	6	4
3.	Cleveland	297	13	5
4.	Dermatology	366	33	6
5.	Ecoli	336	7	8
6.	Flare	1389	10	6
7.	Lymphography	148	18	4
8.	Segment	2310	19	7
9.	Vehicle	846	18	4
10.	Yeast	1484	8	10

training process was stopped prematurely after 200 iterations, in order to assure the initial diversity of the pool and that we are working on weak classifiers.

The pool of classifiers used for experiments was homogeneous and consisted of 10 neural networks.

As a reference methods we decided to use popular classifier combination algorithms: majority voting (MV), maximum of support (MAX), average of supports (AVG), median of supports (MED), product (PRO) and weights calculated according to the individual accuracy of each classifier (IND) [4].

For a pairwise comparison, we use a 5x2 combined CV F-test [33]. It uses a five times two fold cross validation as its basis. With this, we obtain training and testing sets of an equal size. It is an all versus all comparison test. Test score is expressed with the probability that two classifiers have similar error rates (null hypothesis). An alternative hypothesis assumes, that these two classifiers have different error rates. If the difference in error rates is small, it can mean that two models are constructed with similar error. This implies that the hypothesis should not be rejected. If the difference is large, we can assume that two classifiers exhibit different errors and we should reject the given hypothesis.

For assessing the ranks of classifiers over all examined benchmarks, we use a Friedman ranking test [34]. It checks, if the assigned ranks are significantly different from assigning to each classifier an average rank.

We use the Shaffer post-hoc test [35] to find out which of the tested methods are distinctive among an n x n comparison. The post-hoc procedure is based on a specific value of the significance level α. Additionally, the obtained p-values should be examined in order to check how different given two algorithms are.

C. Results

Results of the experiments, presented according to the accuracy and reduction rate of the examined methods, are given in Table II. Outputs of Shaffer post-hoc test over accuracy are given in Table III. We can derive the following conclusions from the experiments:

- The proposed operators behaved reasonably well and outperformed, with statistical significance, all of the traditional methods for 5 out 10 data sets.
- Modifications of the average operator N-AVG was significantly better than the original one in 3 out 10 experiments, while N-MAX (and N-AVG as well) was not significantly better than the original maximum operator.
- The Shaffer test confirmed that the combination rule which takes into consideration additional information (coming e.g. individual classifier accuracy) can outperform untrained operators. This confirmed our intuition, because the trained combination rule usually behave better than untrained one, what was confirmed in the literature.
- This test also showed that N-MAX is a slightly better than N-AVG, and what is interesting it can outperform most of the traditional untrained approaches except maximum operator.
- Analyzing characteristics of the used data benchmark sets we can suppose that proposed operators work well especially for the classification task where the number of possible classes is a quite high, but additional computer experiments should be carried out to confirm this dependency.
- Each of the proposed operators outperform majority voting for almost all data sets. We can conclude, that in the case of an absence of additional learning examples (which can be used to train the combination rule) the untrained aggregation is a better choice than voting methods. This observation is also known and confirmed by other researches as [36].
- Our proposed methods allow to establish efficient weighted combination rules with a low computational complexity. Trained fusers require an additional processing time, which increases the complexity of the ensemble. Our methods, due to their low complexity, seem as an attractive proposition for real-life problems with limitations on processing time, e.g., ensembles for data streams.

IV. CONCLUSIONS

The paper presented two novel untrained aggregation operators which could be used in the case of the absence of additional learning material to train the combination rule. Otherwise the trained combination rule should be advised. The proposed methods could be valuable alternatives for the traditional aggregating operators which do not required learning and could be used in the mentioned above case instead of voting methods, of course in the case that we can access to the support function values of individual classifiers. The computer experiments confirmed that performances of the proposed methods are satisfactory compared to the traditionally untrained operators, especially for tasks when the number of possible classes is high. Therefore, we are going to continue the work on the proposed models, especially we would like to carried out the wider range of computer experiments which would define precisely the type of the classification tasks when the N-AVG and N-MAX could be used.

References

[1] D. H. Wolpert, "The supervised learning no-free-lunch theorems," in *In Proc. 6th Online World Conference on Soft Computing in Industrial Applications*, 2001, pp. 25–42.

[2] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[3] L. Rastrigin and R. H. Erenstein, *Method of Collective Recognition*. Energoizdat, Moscow, 1981.

[4] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[5] L. Rokach, *Pattern classification using ensemble methods*, ser. Series in machine perception and artificial intelligence. World Scientific, 2010.

[6] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*, 1st ed. Chapman & Hall/CRC, 2012.

[7] M. Wozniak, *Hybrid Classifiers - Methods of Data, Knowledge, and Classifier Combination*, ser. Studies in Computational Intelligence. Springer, 2014, vol. 519.

[8] B. Cyganek, "One-class support vector ensembles for image segmentation and classification," *Journal of Mathematical Imaging and Vision*, vol. 42, no. 2-3, pp. 103–117, 2012.

[9] D. Partridge and W. Krzanowski, "Software diversity: practical statistics for its measurement and exploitation," *Information and Software Technology*, vol. 39, no. 10, pp. 707 – 717, 1997.

[10] G. Brown and L. I. Kuncheva, ""good" and "bad" diversity in majority vote ensembles," in *MCS*, 2010, pp. 124–133.

[11] G. Tsoumakas, I. Partalas, and I. Vlahavas, "An ensemble pruning primer," in *Applications of supervised and unsupervised ensemble methods*. Springer, 2009, pp. 1–13.

[12] B. Biggio, G. Fumera, and F. Roli, "Bayesian analysis of linear combiners," in *Proceedings of the 7th international conference on Multiple classifier systems*, ser. MCS'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 292–301.

[13] L. Xu, A. Krzyzak, and C. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 3, pp. 418 –435, may/jun 1992.

[14] L. Hansen and P. Salamon, "Neural network ensembles," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 10, pp. 993 –1001, oct 1990.

[15] R. Burduk, "Classifier fusion with interval-valued weights," *Pattern Recognition Letters*, vol. 34, no. 14, pp. 1623–1629, 2013.

[16] K. Woods, J. Kegelmeyer, W.P., and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 4, pp. 405 –410, 1997.

[17] M. Woźniak and K. Jackowski, "Some remarks on chosen methods of classifier fusion based on weighted voting," in *Hybrid Artificial Intelligence Systems*, ser. Lecture Notes in Computer Science, E. Corchado, X. Wu, E. Oja, A. Herrero, and B. Baruque, Eds. Springer Berlin / Heidelberg, 2009, vol. 5572, pp. 541–548.

[18] D. Ruta and B. Gabrys, "A theoretical analysis of the limits of majority voting errors for multiple classifier systems," *Pattern Anal. Appl.*, vol. 5, no. 4, pp. 333–350, 2002.

[19] Y. Huang and C. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 1, pp. 90 –94, jan 1995.

[20] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 66–75, Jan. 1994.

[21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[22] N. S. V. Rao, "A generic sensor fusion problem: Classification and function estimation." in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, F. Roli, J. Kittler, and T. Windeatt, Eds., vol. 3077. Springer, 2004, pp. 16–30.

[23] D. W. Opitz and J. W. Shavlik, "Generating accurate and diverse members of a neural-network ensemble," in *NIPS*, 1995, pp. 535–541.

[24] L. Rokach and O. Maimon, "Feature set decomposition for decision trees," *Intell. Data Anal.*, vol. 9, no. 2, pp. 131–158, Mar. 2005.

[25] G. Fumera and F. Roli, "A theoretical and experimental analysis of linear combiners for multiple classifier systems," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 6, pp. 942 –956, june 2005.

[26] M. Wozniak, "Experiments on linear combiners," in *Information Technologies in Biomedicine*, ser. Advances in Soft Computing, E. Pietka and J. Kawa, Eds. Springer Berlin / Heidelberg, 2008, vol. 47, pp. 445–452.

[27] R. A. Jacobs, "Methods for combining experts' probability assessments," *Neural Computation*, vol. 7, no. 5, pp. 867–888, 1995.

[28] V. Tresp and M. Taniguchi, "Combining estimators using non-constant weighting functions," in *Advances in Neural Information Processing Systems 7*. MIT Press, 1995, pp. 419–426.

[29] P. Cheeseman, M. Self, J. Kelly, J. Stutz, W. Taylor, and D. Freeman, "AutoClass: a Bayesian classification system," in *Machine Learning: Proceedings of the Fifth International Workshop*. Morgan Kaufmann, 1988.

[30] L. Kuncheva, J. C. Bezdek, and R. P. W. Duin, "Decision templates for multiple classifier fusion: an experimental comparison," *Pattern Recognition*, vol. 34, no. 2, pp. 299–314, 2001.

[31] M. Wozniak and M. Zmyslony, "Combining classifiers using trained fuser - analytical and experimental results," *Neural Network World*, vol. 13, no. 7, pp. 925–934, 2010.

[32] R. Duin, "The combining classifier: to train or not to train?" in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2, 2002, pp. 765 – 770 vol.2.

[33] E. Alpaydin, "Combined 5 x 2 cv f test for comparing supervised classification learning algorithms," *Neural Computation*, vol. 11, no. 8, pp. 1885–1892, 1999.

[34] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[35] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044–2064, 2010.

[36] M. Wozniak and M. Zmyslony, "Designing combining classifier with trained fuser - analytical and experimental evaluation," *Neural Network World*, vol. 20, no. 7, pp. 925–934, 2010.

TABLE II

COMPARISON OF THE CLASSIFIER COMBINATION METHODS, WITH RESPECT TO THEIR ACCURACY [%]. SMALL NUMBERS UNDER ACCURACIES STAND FOR INDEXES OF METHODS, FROM WHICH THE CONSIDERED ONE IS STATISTICALLY SUPERIOR. LAST ROW STANDS FOR THE AVG. RANK AFTER THE FRIEDMAN TEST.

Dataset	MV[1]	MAX[2]	AVG[3]	MED[4]	PRO[5]	IND[6]	N-AVG[7]	N-MAX[8]
1.	62.34 –	65.84 1,3,4,5	64.23 1,5	65.01 1,3,5	63.05 –	65.43 1,3,5	67.54 ALL	66.32 1,3,4,5,6
2.	89.12 3,4,5,6,7	89.23 3,4,5,6,7	88.43 4,5,6,7	87.34 4	85.31 4	87.51 4	87.26 3,4,5,6,7	89.04
3.	52.38 –	57.23 1,5,7	57.43 1,5,7	58.12 1,2,5,7	55.64 1	58.71 ALL	55.02 1	57.14 1,5,7
4.	93.23 –	95.75 1,5,7	95.05 1,5	95.48 1,5,7	92.87 –	95.12 1,5	94.67 1	95.83 1,5,7
5.	71.02 –	77.43 1,3,4,5	75.36 1,4,5	72.18 1	71.61 –	78.79 1,2,3,4,5,7	79.62 ALL	77.60 13,4,5
6.	74.31 2,4,5	72.69 –	75.72 1,2,4,5,6,7	73.42 2	73.12 2	74.28 2,4,5	73.90 2,4,5	77.12 ALL
7.	82.27 ALL	80.32 5	80.87 5	81.12 2,5	79.32 –	80.32 5	81.12 2,5	80.32 5
8.	86.23 4,5	86.74 4,5	87.54 1,2,4,5,7	84.32 –	85.62 4	87.22 1,2,4,5,7	86.89 4,5	91.21 ALL
9.	66.43 –	74.03 1,3,4,5,7	72.63 1,4,5,7	69.03 1,5	67.90 1	75.15 ALL	70.12 1,4,5	73.87 1,3,4,5,7
10.	43.41 –	52.36 1,3,4,5,7	49.78 1,4,5	46.03 1,5	45.02 1	54.39 1,2,3,4,5,7	50.11 1,4,5	57.98 ALL
Avg. rank	4.51	3.21	5.72	6.48	7.62	2.78	3.02	2.66

TABLE III

SHAFFER TEST FOR COMPARISON BETWEEN THE PROPOSED COMBINATION METHODS AND REFERENCE FUSERS. SYMBOL '=' STANDS FOR CLASSIFIERS WITHOUT SIGNIFICANT DIFFERENCES, '+' FOR SITUATION IN WHICH THE METHOD ON THE LEFT IS SUPERIOR AND '-' VICE VERSA.

hypothesis	p-value
N-AVG vs MV	+ (0.0423)
N-AVG vs MAX	= (0.3895)
N-AVG vs AVG	= (0.4263)
N-AVG vs MED	= (0.2371)
N-AVG vs PRO	+ (0.0136)
N-AVG vs IND	- (0.398)
N-MAX vs MV	+ (0.0262)
N-MAX vs MAX	= (0.4211)
N-MAX vs AVG	+ (0.0249)
N-MAX vs MED	+ (0.0106)
N-MAX vs PRO	+ (0.0097)
N-MAX vs IND	= (0.2768)
N-AVG vs N-MAX	- (0.0314)