https://doi.org/10.1007/s10044-017-0677-9

Improving optimum-path forest learning using bag-of-classifiers and confidence measures

Silas Evandro Nachif Fernandes[1] · João Paulo Papa[2]

Received: 21 June 2017 / Accepted: 11 December 2017 / Published online: 18 December 2017
© Springer-Verlag London Ltd., part of Springer Nature 2017

Abstract

Machine learning techniques have been actively pursued in the last years, mainly due to the great number of applications that make use of some sort of intelligent mechanism for decision-making processes. In this work, we presented an ensemble of optimum-path forest (OPF) classifiers, which consists into combining different instances that compute a score-based confidence level for each training sample in order to turn the classification process "smarter", i.e., more reliable. Such confidence level encodes the level of effectiveness of each training sample, and it can be used to avoid ties during the OPF competition process. Experimental results over fifteen benchmarking datasets have shown the effectiveness and efficiency of the proposed approach for classification problems, with more accurate results in more than 67% of the datasets considered in this work. Additionally, we also considered a bagging strategy for comparison purposes, and we showed the proposed approach can lead to considerably better results.

Keywords Optimum-path forest · Supervised learning · Classifier ensemble

1 Introduction

The study of systems that make use of multiple classifiers has become an area of great interest in pattern recognition. A large number of methods to combine classifiers have been proposed recently, thus allowing the fusion of different strategies aiming at improving the effectiveness of the whole system [8, 12, 13]. Researchers have suggested that a combination of decisions provided by several classifiers can result in a better recognition rate than using a sole classifier, or even using the best classifier from a collection [16]. In fact, it is expected that each classifier of this collection may learn different aspects of the data. Thus, the deficiencies of each technique can be offset by the improvements of others. Among the different methods proposed in the literature [19], bagging [6], Boosting [18] and Random Subspaces [14] as the most widely used methods. The Random Subspaces technique creates multiple classifiers using different spaces of features, while bagging generates different learners by randomly selecting subsets of samples to train base classifiers. Although Boosting also uses part of the data to train the classifiers, the most difficult samples to be classified have a higher probability of being selected to compose the final training set.

The output of the classification algorithms can be roughly categorized into three levels [1]: abstract, rankings and confidence. In the first level, the classifiers associate a single label to each dataset sample, while in ranking-based approaches the possible labels for a sample are stored in a priority queue according to some criterion. In confidence-oriented techniques, the classifier computes some metric that will reflect the probability of each label being assigned to a particular sample.

Papa et al. [23, 24] introduced the optimum-path forest (OPF) classifier which is a graph-based supervised pattern recognition technique with interesting results in terms of efficiency and effectiveness, comparable to the ones obtained by Support Vector Machines (SVMs) [7, 32] but faster for training. The idea of OPF is to model

✉ João Paulo Papa
papa@fc.unesp.br

Silas Evandro Nachif Fernandes
silas.fernandes@dc.ufscar.br

[1] Department of Computing, Federal University of São Carlos - UFSCar, Rodovia Washington Luís, Km 235 - SP 310, São Carlos, São Paulo 13565-905, Brazil

[2] Department of Computing, São Paulo State University - UNESP, Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, Bauru, São Paulo 17033-360, Brazil

the pattern recognition task as a graph partition problem where a set of key samples (prototypes) acts as rulers of this competition process. Such samples try to conquer the remaining ones offering to them optimum-path costs, and when a sample is conquered, it receives the label of its conqueror. A new variant of the OPF classifier that makes use of a k-nearest neighborhood (k-nn) graph named OPF$_{knn}$ was proposed by Papa and Falcão [21, 22, 25], and its semi-supervised version has been presented by Amorim et al. [4]. An interesting property stated by Souza et al. [29] concerns that OPF is equivalent to 1-NN when all training samples are used as prototypes. In addition, the same authors presented the k-OPF as a natural extension of the OPF classifier and showed that k-OPF and the well-known k-nearest neighbors technique are similar to each other under some situations. This is interesting in light of a recent work by Amancio et al. [3], which showed that k-nearest neighbors may perform so as well as Support Vector Machines. Ponti and Papa [27] showed the training step of OPF classifier can be more efficient and effective when subsets of the original training set are used rather than the whole set. In the same year, Ponti et al. [28] proposed the combination of OPF classifiers using Markov Random Fields model as a decision graph and Game Theory to compute the final decision, i.e., each classifier is seen as player and each classifier decision (class label) is modeled as a strategy. Fernandes et al. [10] showed an improved version of the naïve OPF classifier that computes a score-based confidence level for each training sample in order to turn the classification process "smarter", i.e., more reliable. Ponti and Rossi [26] investigated different data undersampling approaches and their influence in ensembles of OPF-based classifiers. Fernandes et al. [11] introduced meta-heuristic optimization techniques for pruning OPF-based classifiers in the context of land cover classification. However, there are very few studies on combining OPF classifiers to improve effectiveness in the classification process.

In this paper, an ensemble of OPF score-based confidence classifiers is proposed, which consider not only the optimum-path value from a given sample in the classification process, but also its confidence value measured by means of a score index computed over a validating set. In a nutshell, the idea is to exploit the combination of OPF using bag-of-classifiers by using optimum-path costs that consider confidence values coming from different classifiers. It is shown this approach can overcome traditional OPF in several datasets, i.e., providing a refinement of OPF classification process, even when learn on smaller training sets, as well as it can perform training faster than its standard version when using the same amount of data. The proposed approach also improves the recent results presented by Fernandes et al. [10] and also extends such approach in the context of OPF$_{knn}$ classifier.

The remainder of the paper is organized as follows. Sections 2 and 3 present the OPF background theory and the proposed approach for ensemble-oriented classification with score-based confidence computation, respectively. Section 4 describes the methodology and the experimental results. Finally, conclusions and future works are stated in Sect. 5.

2 Optimum-path forest

In this section, the theoretical foundation of the naïve OPF is discussed. Given some key nodes (prototypes), they will compete among themselves aiming at conquering the remaining nodes. Thus, the algorithm outputs an optimum-path forest, which is a collection of optimum-path trees (OPTs) rooted at each prototype. This work employs the OPF classifier proposed by Papa et al. [23, 24], which is explained in more details as follows.

Let $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ be a labeled dataset, such that $\mathcal{D}_1$ and $\mathcal{D}_2$ stand for the training and test sets, respectively. Let $\mathcal{S} \subset \mathcal{D}_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes). Let $(\mathcal{D}_1, \mathcal{A})$ be a complete graph whose nodes are the samples in $\mathcal{D}_1$, and any pair of samples defines an edges in $\mathcal{A} = \mathcal{D}_1 \times \mathcal{D}_1$. Additionally, let π_s be a path in $(\mathcal{D}_1, \mathcal{A})$ with terminus at sample $\mathbf{s} \in \mathcal{D}_1$.

The OPF algorithm proposed by Papa et al. [23, 24] employs the path-cost function $f_{\max}$ due to its theoretical properties for estimating prototypes (Sect. 2.1 gives further details about this procedure):

$$\begin{aligned} f_{\max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in \mathcal{S} \\ +\infty & \text{otherwise}, \end{cases} \\ f_{\max}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{\max}(\pi_s), d(s, t)\}, \end{aligned} \quad (1)$$

where $d(s, t)$ stands for a distance between nodes s and t, such that $s, t \in \mathcal{D}_1$. Therefore, $f_{\max}(\pi_s)$ computes the maximum distance between adjacent samples in π_s, when π_s is not a trivial path. In short, the OPF algorithm tries to minimize $f_{\max}(\pi_t), \forall t \in \mathcal{D}_1$.

2.1 Training phase

Say that $\mathcal{S}^*$ is an optimum set of prototypes when OPF algorithm minimizes the classification errors for every $\mathbf{s} \in \mathcal{D}_1$. Given that $\mathcal{S}^*$ can be found by exploiting the theoretical relation between the minimum-spanning tree and the optimum-path tree for $f_{\max}$ [2], the training essentially consists in finding $\mathcal{S}^*$ and an OPF classifier rooted at $\mathcal{S}^*$. By computing a minimum-spanning tree (MST) in the complete graph $(\mathcal{D}_1, \mathcal{A})$ obtains a connected acyclic graph whose nodes are all samples of $\mathcal{D}_1$ and the edges are undirected and weighted by the distances d between adjacent samples. In the MST, every pair of samples is connected by a single path, which is optimum according to $f_{\max}$. Hence, the minimum-spanning tree contains one optimum-path tree for any selected root node.

The optimum prototypes are the closest elements of the MST with different labels in $\mathcal{D}_1$ (i.e., elements that fall in the frontier of the classes). By removing the edges between different classes, their adjacent samples become prototypes in $\mathcal{S}^*$, and OPF algorithm can define an optimum-path forest with minimum classification errors in $\mathcal{D}_1$.

2.2 Classification phase

For any sample $\mathbf{t} \in \mathcal{D}_2$, we consider all edges connecting $\mathbf{t}$ with samples $\mathbf{s} \in \mathcal{D}_1$, as though $\mathbf{t}$ were part of the training graph. Considering all possible paths from $\mathcal{S}^*$ to $\mathbf{t}$, we find the optimum-path $\mathcal{P}^*(\mathbf{t})$ from $\mathcal{S}^*$ and label $\mathbf{t}$ with the class $\lambda(\mathcal{R}(\mathbf{t}))$ of its most strongly connected prototype $\mathcal{R}(\mathbf{t}) \in \mathcal{S}^*$. This path can be identified incrementally, by evaluating the optimum cost $\mathcal{C}(\mathbf{t})$ as follows:

$$\mathcal{C}(\mathbf{t}) = \min_{\forall \mathbf{s} \in \mathcal{D}_1} \{\max\{\mathcal{C}(\mathbf{s}), d(\mathbf{s}, \mathbf{t})\}\}. \tag{2}$$

Let the node $s^* \in \mathcal{D}_1$ be the one that satisfies Eq. 2 (i.e., the predecessor $\mathcal{P}(\mathbf{t})$ in the optimum-path $\mathcal{P}^*(\mathbf{t})$). Given that $L(s^*) = \lambda(\mathcal{R}(\mathbf{t}))$, the classification simply assigns $L(s^*)$ as the class of $\mathbf{t}$. An error occurs when $L(\mathbf{s}^*) \neq \lambda(\mathbf{t})$. An interesting point to be considered concerns with the relation between OPF and the nearest neighbor classifier (NN). Although OPF uses the distance between samples to compose the cost to be offered to them, the path-cost function encodes the power of connectivity of the samples that fall in the same path, being much more powerful than the sole distance. Therefore, this means OPF is not a distance-based classifier. Additionally, Papa et al. [24] showed that OPF is quite different than NN, being those techniques exactly the same only when all training samples become prototypes.

3 Ensemble of classifiers with score-based confidence levels

In this section, the confidence level proposed by Fernandes et al. [10] is first introduced in Sect. 3.1, followed by the proposed approach based on ensemble of classifiers to improve the OPF learning process using that confidence levels in Sect. 3.2.

3.1 Score-based confidence levels

In order to extract the confidence level, the dataset $\mathcal{D}$ is partitioned into three subsets, say that $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_v \cup \mathcal{D}_2$ where $\mathcal{D}_1$, $\mathcal{D}_v$ and $\mathcal{D}_2$ stand for the training, validating and testing sets, respectively. It is worth nothing to say all subsets have their respective graph representation as being $(\mathcal{D}_1, \mathcal{A})$, $(\mathcal{D}_v, \mathcal{A})$ and $(\mathcal{D}_2, \mathcal{A})$, as defined in Sect. 2. Therefore, the same definition applied for $\mathcal{D}_1$ and $\mathcal{D}_2$ can also be adopted for $\mathcal{D}_v$.

The approach proposed by Fernandes et al. [10] to calculate scores aims at training OPF classifier over $\mathcal{D}_1$ for further classification of $\mathcal{D}_v$ using the same methodology described in Sect. 2. The main difference is that each training sample receives a reliability level $\phi(\cdot)$, which is computed by means of its individual performance (recognition rate) over the validating set. The training samples $\mathbf{s} \in \mathcal{D}_1$ start with $\phi(\mathbf{s}) = 0$, and if $\mathbf{s}$ classifies some validating sample, then $\phi(\mathbf{s}) = \phi(\mathbf{s}) + 1$; if misclassification occurs, then $\phi(\mathbf{s}) = \phi(\mathbf{s}) - 1$. Later on, the final $\phi(\mathbf{s})$ is computed based on the average of hits and errors for each sample $\mathbf{t} \in \mathcal{D}_v$ conquered by $\mathbf{s}$. Also, considering the aforementioned approach, a sample $\mathbf{s} \in \mathcal{D}_1$ that did not participate from any classification process would be scored as $\phi(\mathbf{s}) = 0$, and thus may be penalized, since the higher the score the most reliable that sample is. Therefore, for such samples are assigned $\phi(\mathbf{s}) \rightarrow +1$ to give them a chance to perform a classification process over the unseen (test) data without any disadvantage. Thus, at the end of the classification process over the validating set $\mathcal{D}_v$ have a score measure $\phi(\mathbf{s}) \in [0, 1], \forall \mathbf{s} \in \mathcal{D}_1$, which can be used as a confidence level of that sample. In short, there are three possible confidence levels:

- $\phi(\mathbf{s}) = 0$: it means sample $\mathbf{s}$ did not perform a good work on classifying samples, since it has misclassified all samples. Therefore, samples with score equal to 0 *may not be reliable*;
- $0 < \phi(\mathbf{s}) < 1$: it means sample $\mathbf{s}$ has misclassified some samples, as well as it has also assigned correct labels to some of them. Notice the larger the errors, the lower is the sample's reliability. Samples with scores that fall in this range *may be reliable*; and
- $\phi(\mathbf{s}) = 1$: it means either sample $\mathbf{s}$ did not participate in any classification process or $\mathbf{s}$ assigned the correct label to all its conquered samples, which means $\mathbf{s}$ is a *reliable sample* according to our definition.

Algorithm 1 implements the procedure described above. Lines 1–4 initialize the score of each training sample, and Line 5 performs the OPF training step over $(\mathcal{D}_1, \mathcal{A})$. The core of the algorithm is performed in Lines 6–15: the classification process of a validation sample $\mathbf{t}$ is performed in Line 7 using traditional OPF classification procedure. Let

$\mathbf{s}^* \in \mathcal{D}_1$ be the sample that has conquered $\mathbf{t}$: in this case, the counter $n(\cdot)$ of samples classified by $\mathbf{s}^*$ is then increased in Line 8. Additionally, if $\mathbf{t}$ is misclassified, the counter $e(\cdot)$ is decreased for that training sample $\mathbf{s}^*$ in Line 10. The loop in Lines 11–15 is responsible for computing the final score for each training sample. Lines 12–13 set the score of a sample that did not participate in any classification process to 1, as mentioned above.

Algorithm 1: Confidence levels algorithm

```
Input: A λ-labeled graph-based representations for both
       training and validation sets, i.e., $(\mathcal{D}_1, \mathcal{A})$ and $(\mathcal{D}_v, \mathcal{A})$,
       respectively.
Output: Confidence level $\phi(\mathbf{s})$, $\forall \mathbf{s} \in \mathcal{D}_1$.
Auxiliary: Arrays $n(\cdot)$ and $e(\cdot)$.
1  for each $s \in \mathcal{D}_1$ do
2      $n(\mathbf{s})$=0;
3      $e(\mathbf{s})$=0;
4      $\phi(\mathbf{s})$=0;
5  Train OPF algorithm over $(\mathcal{D}_1, \mathcal{A})$ according to Section 2.1;
6  for each $\mathbf{t} \in \mathcal{D}_v$ do
7      Let $\mathbf{s}^* \in \mathcal{D}_1$ be the sample that classified $\mathbf{t}$ with label $L(\mathbf{t})$
       according to Equation 2;
8      $n(\mathbf{s}^*) \leftarrow n(\mathbf{s}^*) + 1$;
9      if $\lambda(t) \neq L(t)$ then
10         $e(\mathbf{s}^*) \leftarrow e(\mathbf{s}^*) - 1$;
11 for each $s \in \mathcal{D}_1$ do
12     if $n(s) = 0$ then
13         $\phi(\mathbf{s}) \leftarrow 1$;
14     else
15         $\phi(\mathbf{s}) \leftarrow \frac{n(\mathbf{s})+e(\mathbf{s})}{n(\mathbf{s})}$;
```

After calculating the confidence levels for each training sample, one needs to modify the naïve OPF classification procedure in order to consider such information during the label assignment. In order to fulfill this purpose, Fernandes et al. [10] proposed a modification in the OPF classification procedure (Eq. 2) as follows:

$$\mathscr{C}'(\mathbf{t}) = \min_{\forall \mathbf{s} \in \mathcal{D}_1} \left\{ \left(\frac{1}{\phi(\mathbf{s}) + \epsilon} \right) * \max\{\mathscr{C}(\mathbf{s}), d(\mathbf{s}, \mathbf{t})\} \right\}, \tag{3}$$

where $\epsilon = 10^{-4}$ is employed to avoid numerical instabilities. Therefore, the idea of the first term in the Eq. 3 is to penalize samples with *low confidence* values by increasing their costs. In short, the amount of penalty is inversely proportional to the sample's confidence level. For the sake of explanation, we provided a graphic illustration of the working mechanism considering the confidence level-based optimum-path forest classifier proposed by Fernandes et al. [10]. Let OPF* be the classifier trained on $\mathcal{D}_1 \cup \mathcal{D}_v$, and OPF$_c$ the confidence-based approach proposed by Fernandes et al. [10]. The idea is to show the situations in which the approach that uses confidence levels may overcome standard OPF by making use of the reliability of a given training sample when classifying others. Figure 1 depicts the training (hexagon) and validating (remaining samples) sets with respect to "Synthetic1" dataset (Table 1), which comprises two classes (squares and circles) with a high amount of data overlapping. Now, let us consider the highlighted zone displayed in Fig. 1, which is zoomed and represented in Figs. 2a–c, corresponding to the same set of samples for OPF, OPF* and OPF$_c$, respectively. Samples 'A', 'B' and 'C' are part of the training set, while sample 'D' belongs to the validating set; and the "circle" is a test sample that can be classified by either 'A', 'B' or 'C' (we showed the competition process between 'A' and 'B' only). Considering standard OPF (Fig. 2a), we can observe sample 'B' (solid edge) has provided a better path-cost than sample 'A' (dashed edge), thus conquering the test sample and also misclassifying it, since its true label is "circle", i.e., the same label as 'A'. The same situation can be observed for OPF* in Fig. 2b, meaning that larger training sets may not be helpful for learning patterns in highly overlapped regions (as aforementioned, OPF* is trained over $\mathcal{D}_1 \cup \mathcal{D}_v$). However, if we consider the confidence values in OPF$_c$ (Fig. 2c), we can notice that sample 'B' has been penalized with a lower confidence level than sample 'A', thus reflecting in the cost provided to the test sample, which is more suitable considering now sample 'A' (solid edge), since it has a better confidence level. Therefore, the classification based on the training samples' reliability allows OPF$_c$ to be more accurate in some situations, mainly in highly overlapped datasets. Finally, Fig. 2d depicts the regions of the training space according to the domain of confidence value through the natural neighbor interpolation [15]. We can observe the "darkest regions" (confidence value close to zero) stand for the ones with high levels of overlapping

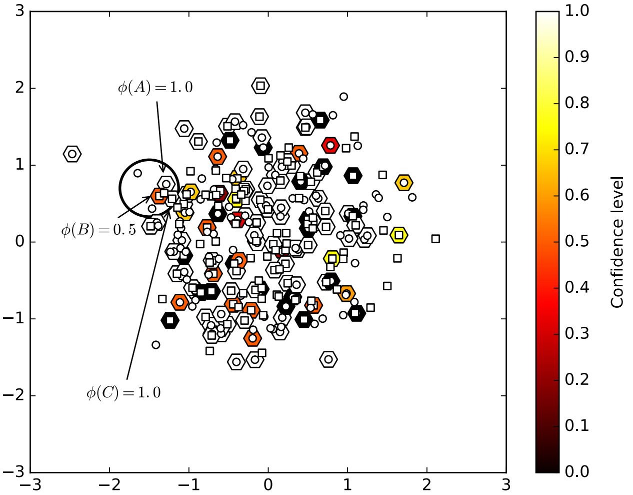
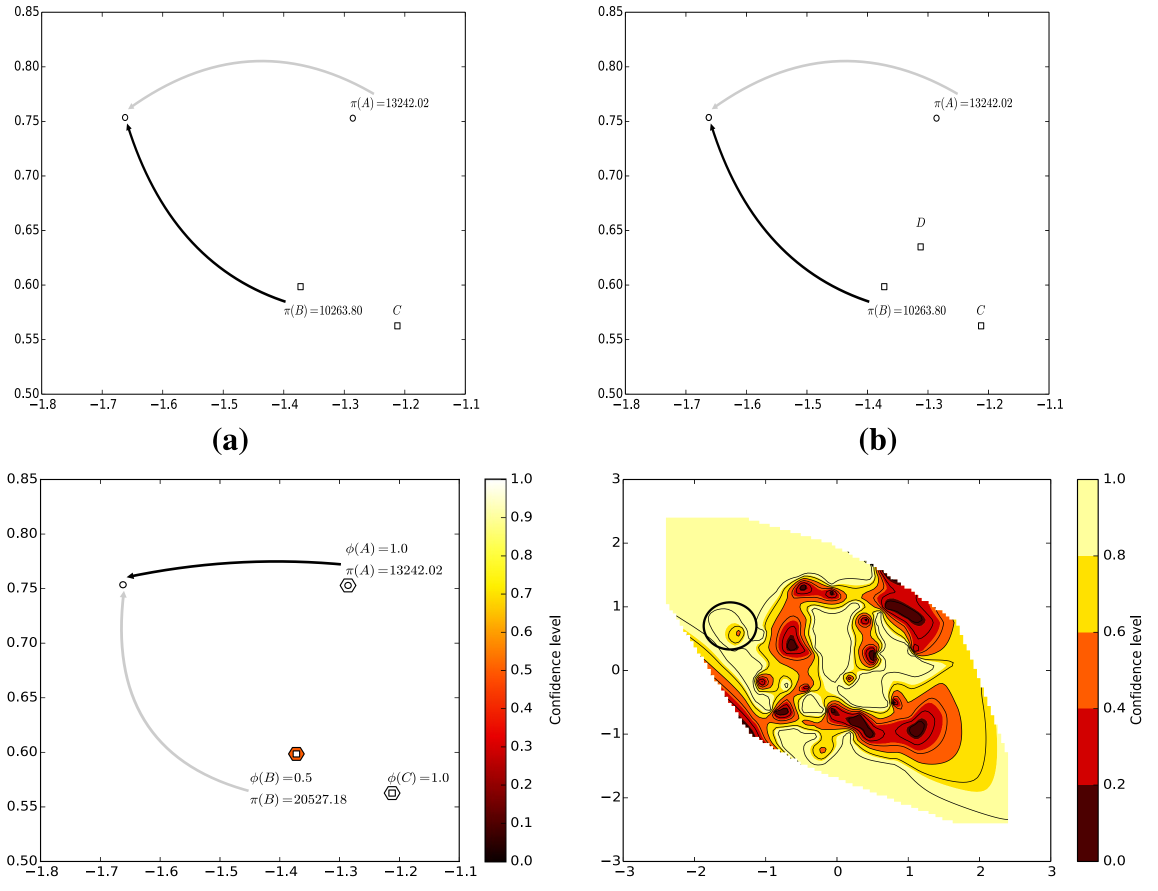


Fig. 1 Graphic representation of each training sample of "Synthetic1" (Table 1) dataset according to its confidence level



(a) **(b)** **(c)** **(d)**

Fig. 2 Example of a classification process in the test set for **a** OPF, **b** OPF* and **c** OPFc, and the **d** graphic representation for the dispersion zones of each training sample of "Synthetic1" (Table 1) dataset

Table 1 Description of the datasets

Dataset	# samples	# features	# classes
aflw	8193	4096	2
Pima-Indians-Diabetes	768	8	2
Statlog-Australian	690	14	2
Statlog-dna	5186	180	3
Statlog-Heart	270	13	2
Synthetic1	500	2	2
Synthetic2	1000	2	2
Synthetic3	200	2	2
Synthetic4	100,000	4	4
UCI-a1a	32,561	123	2
UCI-Ionosphere	351	34	2
UCI-Liver-disorders	345	6	2
usps	9298	256	10
w1a	49,749	300	4
yahoo-web-directory-topics	1106	10,629	4

in Fig. 1, supporting the idea that training samples that fall in such regions may not be reliable enough for classifying others, as well as training samples that are located nearby to outliers, which have a high probability to be misclassified in traditional pattern recognition techniques. Therefore, if a training sample misclassifies an outlier from the validating set in the OPF$_c$, its confidence level will drop, thus raising its classification cost over the test samples.

3.2 Ensemble-based confidence levels

In this section, a new approach is presented based on bag-of-classifiers and confidence measures to improve OPF effectiveness. Since OPF classifier uses the abstract output method only, i.e., the output of the classifier is a single label, the OPF based on confidence levels also returns the very same output. Xu et al. [31] defined an interesting approach to combine the outputs of L classifiers in an ensemble depending on the information obtained from the individual members. Such approach considers that each classifier assigns a class label to every sample in the dataset. Therefore, the ensemble of classifiers generates a collection of L possible outputs to each sample.

Let $\mathscr{M} = \{\mathscr{M}_1, \mathscr{M}_2, \ldots, \mathscr{M}_L\}$ be a set of L classifiers, and $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ be a set of K class labels. Roughly speaking, each classifier takes an n-dimensional input vector and associates it to a class label, i.e., $M_i : \mathfrak{R}^n \rightarrow \Omega$, $i = 1, 2, \ldots, L$. Therefore, for any sample $\mathbf{z}$ to be classified, the ensemble of classifiers generates a collection $\Psi_{\mathbf{z}} = [\psi_{\mathbf{z}}(\mathscr{M}_1), \ldots, \psi_{\mathbf{z}}(\mathscr{M}_L)]$ of possible outputs, where $\psi_{\mathbf{z}}(\mathscr{M}_i)$ stands for the output of classifier $\mathscr{M}_i$ considering sample $\mathbf{z}$.

The idea is to partition the training set $\mathscr{D}_1$ into L subsets $\mathscr{D}_1^j$, i.e., $\mathscr{D}_1 = \mathscr{D}_1^1 \cup \mathscr{D}_1^2 \cup \cdots \cup \mathscr{D}_1^L$, such that each classifier $\mathscr{M}_i$ will be trained over $\mathscr{D}_1^i$, $i = 1, 2, \ldots L$. The proposed approach employs the confidence-based procedure presented in Sect. 3.1 for each trained classifier $\mathscr{M}_i$ using the validating set $\mathscr{D}_v$, i.e., this means we shall associate a score level for each sample from the different training folds. After calculating the score levels in the validating phase, the classification takes place using Eq. 3, and the possible outputs are assigned to each sample $\mathbf{z} \in \mathscr{D}_2$. The classification of that sample is performed through the majority vote. We also evaluated the proposed approach using ensembles composed of two distinct OPF versions: OPF with complete graph [24] and with a k-NN graph, i.e., OPF$_{knn}$ [21]. Additionally, we apply the same idea of confidence levels in OPF$_{knn}$ classification step as well, since the work by Fernandes et al. [10] used OPF with complete graph only.

4 Methodology and experimental results

The proposed ensemble confidence-based OPF classifier is compared with standard OPF using fifteen real and synthetic different benchmark classification problems.[1,2] The datasets were normalized as follows:

$$\mathbf{t}' = \frac{\mathbf{t} - \mu}{\rho} \tag{4}$$

where μ denotes the mean, and ρ stands for its standard deviation. Also, $\mathbf{t}$ and $\mathbf{t}'$ correspond to the original and normalized features, respectively. Table 1 presents the main characteristics of each dataset.

In regard to the methodology, each dataset was partitioned into three subsets: training (40%), validating (20%) and testing sets (40%), hereinafter denoted as 40:20:40. For each range, training, validating and testing sets were selected randomly and the process was repeated twenty times (Stratified k-fold cross-validation).[3] It is worth noting the standard OPF was trained over $\mathscr{D}_1 \cup \mathscr{D}_v$ considering the aforementioned subsets. In order to provide a consistent experimental evaluation, the following classifiers were compared: (a) standard OPF; (b) the baseline classifier which uses the confidence-based OPF proposed by Fernandes et al. [10], (OPF$_c$); and (c) the proposed work using three base OPF$_c$ classifiers and a combination of decisions provided by majority voting, defined as *ensemble* OPFc. Furthermore, to evaluate the impact with other OPF variants, we conducted two more experiments: (d) one that combines OPF$_{knn}$ and OPF$_c$, i.e., an ensemble with two base OPF$_c$ classifiers and one OPF$_{knn}$ (hereinafter called OPF$_c$ + OPF$_{knn}$); and (e) one last approach composed of OPF$_{knn}$ using the very same confidence-based idea of OPF$_c$, but now adapted to this variant that uses a k-neighborhood graph (defined as OPF$_{knnC}$). In this case, the ensemble also contains three base classifiers, one OPF$_{knnC}$ and two OPF$_c$. The pipeline of experimental evaluation using the bag-of-classifiers ensemble is illustrated in Fig. 3.

We used three base classifiers only, since we observed no significant gains using more classifiers. The rationale behind that is related to the numbers of samples available for the learning process of each base classifier, since the more classifiers we use, the smaller the training sets. The idea is to look for effectiveness by using the analysis of confidence levels in conjunction with efficiency by using disjoint sets to accelerate and improve the final decision-making process by combining decisions. In addition, we compared the proposed pipeline with the bagging strategy using an ensemble of three classifiers aggregated by using different bootstrapped samples of the original training data.

Table 2 presents the mean accuracies and standard deviation over all datasets, being the recognition rates computed according to Papa et al. [24], and Table 3 presents the bagging strategy concerning the very same group of datasets. In addition, the F-measure metric was calculated for the very same group of datasets concerning the proposed approach and bagging in Tables 4, 5, respectively. The most accurate techniques considering the Wilcoxon test [30] (with significance of 0.05) are highlighted in bold.

We can observe the proposed ensemble-based OPF has obtained the best results in 10 out 15 datasets according to Table 2. It is worth noting that the bagging strategy allowed the best results in only 2 out the 15 datasets concerning the accuracy results (Table 3) of ensemble-based approaches and provided better results for only “UCI-Ionosphere”, “UCI-Liver-disorders”, “usps” and “yahoo-web-directory-topics” datasets. In regard to Tables 4 and 5, the F-measure values showed a similar behavior to that observed in the accuracy. The main idea in computing confidence levels for

[1] http://archive.ics.uci.edu/ml.

[2] http://lrs.icg.tugraz.at/research/aflw.

[3] Notice the percentages have been empirically chosen, being more intuitive to provide a larger validating set for calculating the confidence levels.

Fig. 3 Pipeline of the experimental evaluation: **a** standard and **b** baseline approaches, **c** the proposed approach using three base OPFc classifiers, **d** using two base OPF$_c$ and one OPF$_{knn}$ classifier, and **e** one that combines OPF$_{knnC}$ (OPF$_{knn}$ with confidence levels) and two OPF$_c$

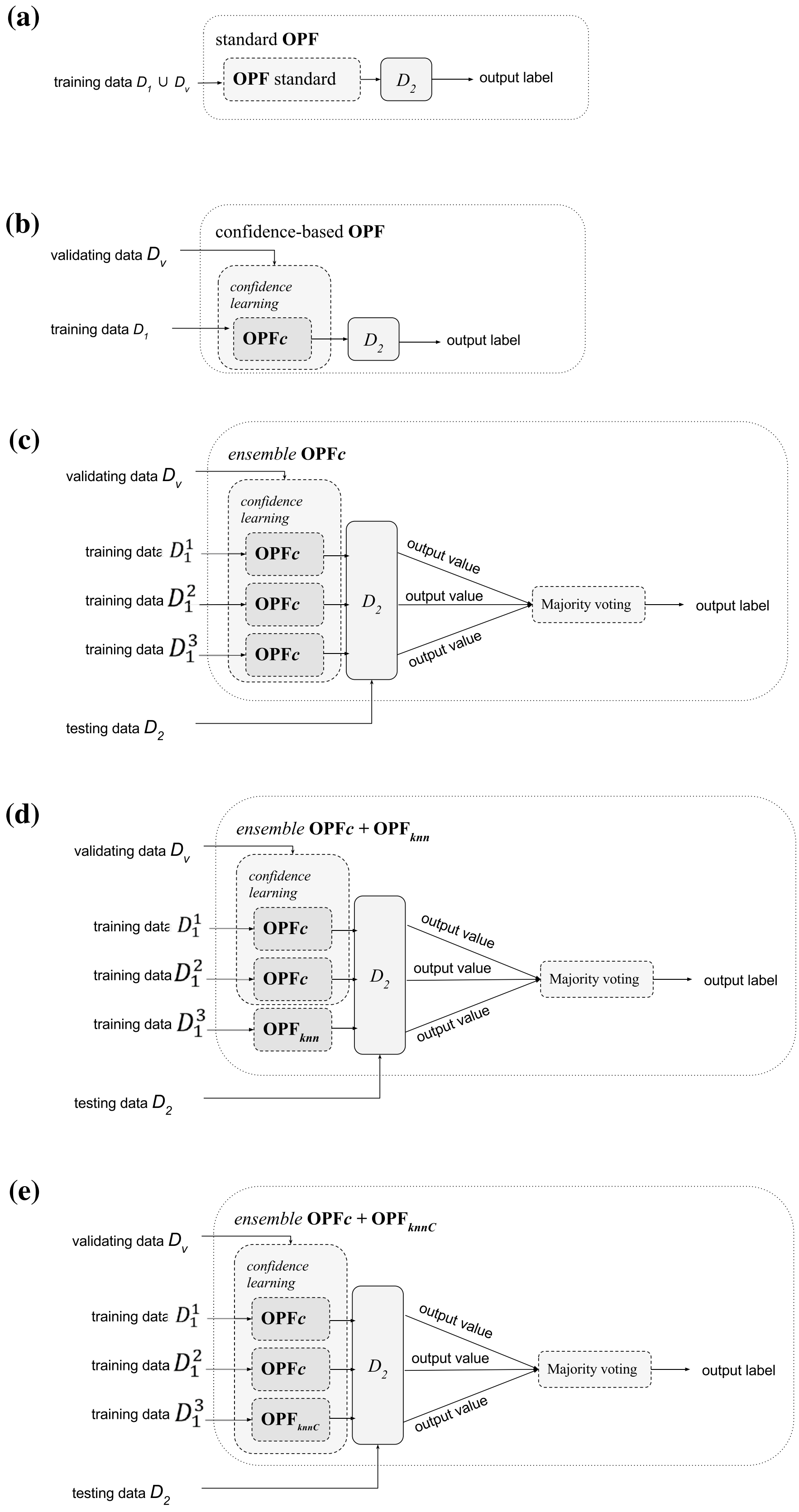


Table 2 Mean accuracy results (%) and standard deviation over all datasets for standard OPF, OPF*c*, and ensemble under different configuration

Dataset	OPF	OPF*c*	*ensemble*	*ensemble*	*ensemble*
			OPF*c*	OPF*c* +	OPF*c* +
				OPF_{knn}	OPF_{knnC}
aflw	89.9047 ± 0.5281	89.5206 ± 0.3943	**90.3768** ± 0.4024	**90.4056** ± 0.4628	**90.3581** ± 0.4106
Pima-Indians-Diabetes	65.9111 ± 3.0189	66.2384 ± 2.0851	**69.1421** ± 2.6485	**68.9981** ± 2.7670	**68.9981** ± 2.7670
Statlog-Australian	77.8525 ± 2.0016	79.4537 ± 2.4155	**83.3438** ± 1.2958	82.8686 ± 1.5526	82.8686 ± 1.5526
Statlog-dna	**89.7546** ± 0.5335	87.3751 ± 0.6745	86.0323 ± 0.7358	85.9107 ± 0.6781	85.9793 ± 0.9693
Statlog-Heart	76.5104 ± 1.9464	76.2604 ± 6.4816	**79.1979** ± 3.8658	**78.5312** ± 4.3359	**78.5312** ± 4.3359
Synthetic1	51.8000 ± 1.3910	52.8750 ± 1.3863	**55.4750** ± 2.2106	**55.3500** ± 2.3932	**55.3500** ± 2.3932
Synthetic2	70.4500 ± 1.4374	74.2250 ± 1.7605	**78.9875** ± 2.3907	**78.3625** ± 1.7686	**78.4000** ± 1.8104
Synthetic3	92.7790 ± 3.8120	93.4329 ± 1.5429	**94.3872** ± 2.4503	**94.3857** ± 1.5597	**94.3857** ± 1.5597
Synthetic4	85.5993 ± 0.1442	86.7918 ± 0.1146	**88.7461** ± 0.0970	88.5103 ± 0.0747	88.5173 ± 0.0789
UCI-a1a	67.5772 ± 0.8289	69.6517 ± 1.0403	72.2513 ± 0.4794	**72.6258** ± 0.4591	**72.4733** ± 0.6518
UCI-Ionosphere	**80.0997** ± 2.6531	**80.0180** ± 3.0253	77.6144 ± 2.5317	75.3513 ± 4.3467	75.3513 ± 4.3467
UCI-Liver-disorders	**59.5765** ± 2.0933	**60.3276** ± 2.1215	57.2144 ± 2.1227	57.1379 ± 3.8046	57.1379 ± 3.8046
usps	**97.4527** ± 0.1750	97.0480 ± 0.1811	96.4015 ± 0.1930	96.4647 ± 0.1698	96.4936 ± 0.1723
w1a	76.7155 ± 1.3150	78.3214 ± 1.2836	81.4666 ± 0.6995	**83.0393** ± 0.7084	**82.6786** ± 0.7421
yahoo-web-directory-topics	63.3961 ± 3.7948	**65.4839** ± 3.4701	60.7265 ± 5.5222	60.1739 ± 6.4093	61.3214 ± 5.5594

The most accurate techniques for the Wilcoxon test are highlighted in bold

Table 3 Mean accuracy results (%) and standard deviation over all datasets using bagging strategy for ensemble-based OPF under different configurations

Dataset	OPF	OPF*c*	*ensemble*	*ensemble*	*ensemble*
			OPF*c*	OPF*c* +	OPF*c* +
				OPF_{knn}	OPF_{knnC}
aflw	**89.9047** ± 0.5281	89.5206 ± 0.3943	89.9325 ± 0.3935	**90.1234** ± 0.3722	**90.1234** ± 0.3722
Pima-Indians-Diabetes	**65.9111** ± 3.0189	**66.2384** ± 2.0851	**65.3771** ± 2.1528	**65.5015** ± 2.3142	**65.5015** ± 2.3142
Statlog-Australian	77.8525 ± 2.0016	**79.4537** ± 2.4155	**80.6516** ± 1.8472	**80.5711** ± 1.6151	**80.5711** ± 1.6151
Statlog-dna	**89.7546** ± 0.5335	87.3751 ± 0.6745	89.1084 ± 0.5769	88.9461 ± 0.5885	88.9461 ± 0.5885
Statlog-Heart	**76.5104** ± 1.9464	**76.2604** ± 6.4816	**78.2604** ± 3.8860	**77.8750** ± 4.0239	**77.8750** ± 4.0239
Synthetic1	51.8000 ± 1.3910	**52.8750** ± 1.3863	**54.0250** ± 2.4160	**54.0000** ± 2.3292	**54.0000** ± 2.3292
Synthetic2	70.4500 ± 1.4374	**74.2250** ± 1.7605	**74.9500** ± 1.6155	**74.7625** ± 1.6573	**74.7625** ± 1.6573
Synthetic3	**92.7790** ± 3.8120	**93.4329** ± 1.5429	**93.8750** ± 2.0879	**93.8750** ± 2.1973	**93.8750** ± 2.1973
Synthetic4	85.5993 ± 0.1442	**86.7918** ± 0.1146	**86.8453** ± 0.1460	86.8229 ± 0.1435	86.8229 ± 0.1435
UCI-a1a	67.5772 ± 0.8289	69.6517 ± 1.0403	70.8360 ± 0.7138	**71.3890** ± 0.4872	**71.3890** ± 0.4872
UCI-Ionosphere	80.0997 ± 2.6531	80.0180 ± 3.0253	**82.7206** ± 3.3906	**82.2369** ± 3.0783	**82.2369** ± 3.0783
UCI-Liver-disorders	**59.5765** ± 2.0933	**60.3276** ± 2.1215	**59.0884** ± 4.0186	**59.6562** ± 4.3026	**59.6562** ± 4.3026
usps	**97.4527** ± 0.1750	97.0480 ± 0.1811	97.2822 ± 0.1625	**97.3464** ± 0.1700	**97.3464** ± 0.1700
w1a	76.7155 ± 1.3150	78.3214 ± 1.2836	81.4666 ± 0.8151	**83.0951** ± 1.1398	**83.0951** ± 1.1398
yahoo-web-directory-topics	63.3961 ± 3.7948	**65.4839** ± 3.4701	**62.3424** ± 6.4529	**62.4055** ± 6.1363	**62.4055** ± 6.1363

The most accurate techniques for the Wilcoxon test are highlighted in bold

each training sample and further applying Eq. 3 as the path-cost function is to avoid ties during the competition process. Roughly speaking, a tie means we have two (at least) different samples that offer the same optimum cost to another sample. The problem occurs when such samples belong to different classes, which may lead OPF to a misclassification. Therefore, by considering the confidence level in the path-cost function, we can rely on samples that are more "trustable" than others. However, by using bagging strategies, since the validating set is the same and we have ensembles composed of training samples that are sampled with reposition, we can also have training samples with the very same

Table 4 Mean F-measure values over all datasets for standard OPF, OPFc, and ensemble under different configuration

Dataset	OPF	OPFc	*ensemble*	*ensemble*	*ensemble*
			OPFc	OPFc +	OPFc +
				OPF$_{knn}$	OPF$_{knnC}$
aflw	0.8937 ± 0.0057	0.8909 ± 0.0041	**0.9000** ± 0.0042	**0.9000** ± 0.0049	**0.8995** ± 0.0043
Pima-Indians-Diabetes	0.6925 ± 0.0244	0.6979 ± 0.0152	**0.7396** ± 0.0231	**0.7354** ± 0.0217	**0.7354** ± 0.0217
Statlog-Australian	0.7825 ± 0.0199	0.7973 ± 0.0234	**0.8377** ± 0.0124	0.8329 ± 0.0147	0.8329 ± 0.0147
Statlog-dna	**0.8481** ± 0.0074	0.8150 ± 0.0097	0.8095 ± 0.0098	0.7993 ± 0.0094	0.8011 ± 0.0127
Statlog-Heart	0.7690 ± 0.0184	0.7662 ± 0.0647	**0.7968** ± 0.0383	**0.7912** ± 0.0423	**0.7912** ± 0.0423
Synthetic1	0.5180 ± 0.0139	0.5288 ± 0.0139	**0.5547** ± 0.0221	**0.5535** ± 0.0239	**0.5535** ± 0.0239
Synthetic2	0.7045 ± 0.0144	0.7423 ± 0.0176	**0.7899** ± 0.0239	**0.7836** ± 0.0177	**0.7840** ± 0.0181
Synthetic3	**0.9278** ± 0.0381	0.9340 ± 0.0157	**0.9432** ± 0.0248	**0.9432** ± 0.0158	**0.9432** ± 0.0158
Synthetic4	0.7840 ± 0.0022	0.8019 ± 0.0017	**0.8312** ± 0.0015	0.8277 ± 0.0011	0.8278 ± 0.0012
UCI-a1a	0.7312 ± 0.0134	0.7622 ± 0.0148	**0.8139** ± 0.0032	**0.8141** ± 0.0029	**0.8141** ± 0.0044
UCI-Ionosphere	**0.8504** ± 0.0200	**0.8482** ± 0.0214	0.8319 ± 0.0172	0.8163 ± 0.0300	0.8163 ± 0.0300
UCI-Liver-disorders	**0.6047** ± 0.0198	**0.6203** ± 0.0232	**0.6040** ± 0.0188	0.6014 ± 0.0369	0.6014 ± 0.0369
usps	**0.9592** ± 0.0028	0.9526 ± 0.0031	0.9429 ± 0.0030	0.9439 ± 0.0026	0.9444 ± 0.0028
w1a	0.6650 ± 0.0659	0.7519 ± 0.0699	0.8865 ± 0.0318	**0.9632** ± 0.0357	**0.9417** ± 0.0446
yahoo-web-directory-topics	**0.6275** ± 0.0166	**0.6396** ± 0.0243	**0.6534** ± 0.0644	**0.6479** ± 0.0617	**0.6283** ± 0.0667

The most accurate techniques for the Wilcoxon test are highlighted in bold

Table 5 Mean F-measure values over all datasets using bagging strategy for ensemble-based OPF under different configurations

Dataset	OPF	OPFc	*ensemble*	*ensemble*	*ensemble*
			OPFc	OPFc +	OPFc +
				OPF$_{knn}$	OPF$_{knnC}$
aflw	**0.8937** ± 0.0057	0.8909 ± 0.0041	0.8950 ± 0.0039	**0.8969** ± 0.0038	**0.8969** ± 0.0038
Pima-Indians-Diabetes	**0.6925** ± 0.0244	**0.6979** ± 0.0152	**0.6929** ± 0.0168	**0.6945** ± 0.0187	**0.6945** ± 0.0187
Statlog-Australian	0.7825 ± 0.0199	**0.7973** ± 0.0234	**0.8100** ± 0.0180	**0.8096** ± 0.0153	**0.8096** ± 0.0153
Statlog-dna	**0.8481** ± 0.0074	0.8150 ± 0.0097	0.8378 ± 0.0076	0.8362 ± 0.0087	0.8362 ± 0.0087
Statlog-Heart	**0.7690** ± 0.0184	**0.7662** ± 0.0647	**0.7880** ± 0.0377	**0.7838** ± 0.0389	**0.7838** ± 0.0389
Synthetic1	0.5180 ± 0.0139	**0.5288** ± 0.0139	**0.5403** ± 0.0242	**0.5400** ± 0.0233	**0.5400** ± 0.0233
Synthetic2	0.7045 ± 0.0144	**0.7423** ± 0.0176	**0.7495** ± 0.0162	**0.7476** ± 0.0166	**0.7476** ± 0.0166
Synthetic3	**0.9278** ± 0.0381	**0.9340** ± 0.0157	**0.9388** ± 0.0209	**0.9388** ± 0.0220	**0.9388** ± 0.0220
Synthetic4	0.7840 ± 0.0022	**0.8019** ± 0.0017	**0.8027** ± 0.0022	0.8024 ± 0.0022	0.8024 ± 0.0022
UCI-a1a	0.7312 ± 0.0134	0.7622 ± 0.0148	0.7789 ± 0.0081	**0.7868** ± 0.0050	**0.7868** ± 0.0050
UCI-Ionosphere	0.8504 ± 0.0200	0.8482 ± 0.0214	**0.8681** ± 0.0258	**0.8649** ± 0.0231	**0.8649** ± 0.0231
UCI-Liver-disorders	**0.6047** ± 0.0198	**0.6203** ± 0.0232	**0.6029** ± 0.0413	**0.6080** ± 0.0426	**0.6080** ± 0.0426
usps	**0.9592** ± 0.0028	0.9526 ± 0.0031	0.9565 ± 0.0026	**0.9575** ± 0.0028	**0.9575** ± 0.0028
w1a	0.6650 ± 0.0659	0.7519 ± 0.0699	0.8700 ± 0.0485	**0.9410** ± 0.0577	**0.9410** ± 0.0577
yahoo-web-directory-topics	**0.6275** ± 0.0166	**0.6396** ± 0.0243	**0.6402** ± 0.0299	**0.6335** ± 0.0308	**0.6335** ± 0.0308

The most accurate techniques for the Wilcoxon test are highlighted in bold

confidence level (such value is computed over the very same validating set for all training subsets). In this context, avoiding ties will no longer be possible, degenerating to the original OPF and thus affecting the results as we can observe in Tables 2 and 3. It was not possible to establish some specific situation considering the dataset configuration (e.g., number of classes and the number features) in which *ensemble* OPF$_c$ could be better than OPF and OPF$_c$, although it seems the proposed approach has obtained the top results in situations with highly overlapped regions. Taking a look at Fig. 4a, b ("Synthetic1" and "Synthetic2" dataset, respectively), we can observe a considerable amount of overlapping among samples of different classes, thus being more useful to learn patterns with OPF$_c$ and, consequently, we can obtain more

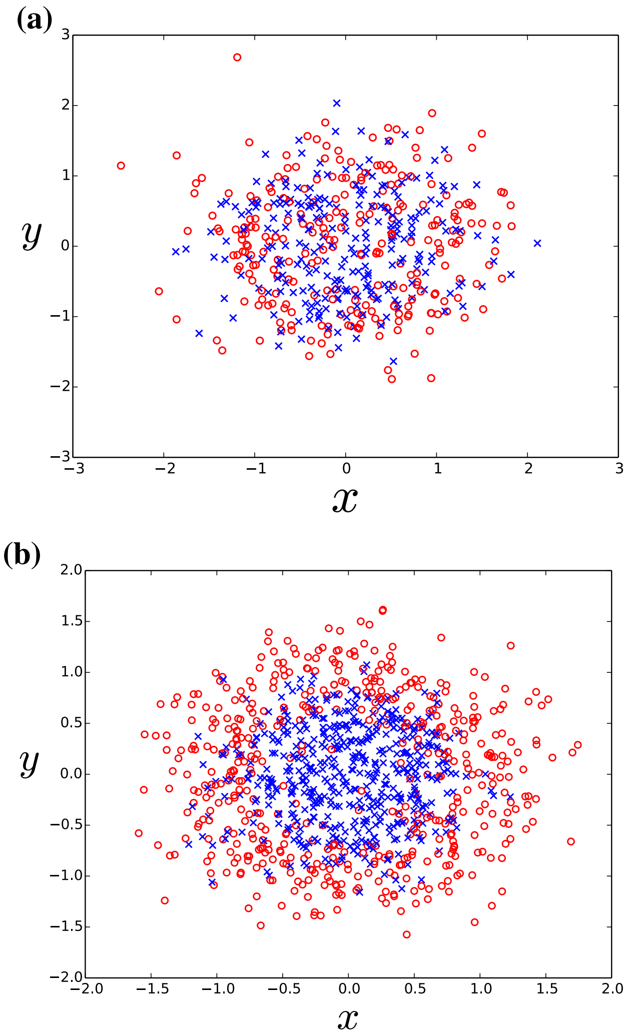


Fig. 4 Graphic representation containing all samples of **a** “Synthetic1”, **b** “Synthetic2” dataset

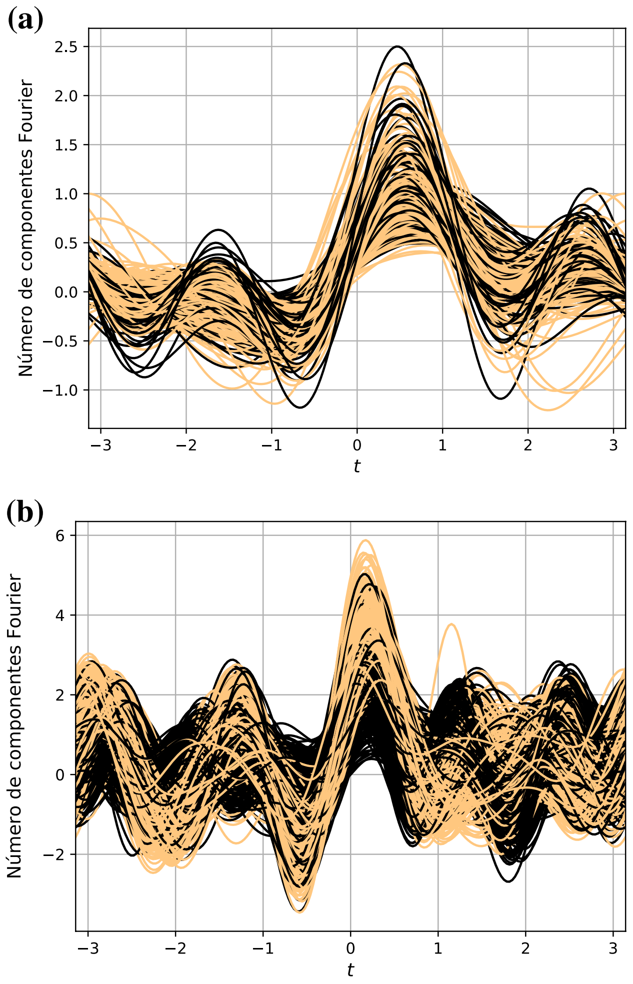


Fig. 5 Andrews plot considering **a** “UCI-Liver-disorders” and **b** “Statlog-Australian” dataset in the range of $-\pi < t < \pi$

effective results with *ensemble* OPF$_c$, since the feature space ends up being partitioned into different subregions.

However, the above situation usually does not occur in datasets that do not behave as “Synthetic1” or “Synthetic2” dataset, i.e., they do not have a considerable amount of overlapped regions. Well-behaved datasets seem to be better generalized by standard OPF approach. Therefore, for some situations it is more important to count with a larger dataset instead of an ensemble of classifiers.

Another aspect to be considered concerns the situations where the *ensembles* do not outperform OPF. If one takes a look at Table 2, the “UCI-Liver-disorders” and “Statlog-Australian” datasets can be included in the aforementioned situation. In order to have some insight about the amount of overlapping on that datasets, we employed the Andrews curve method [5], which represents high-dimensional feature spaces by means of finite Fourier series. The transformation has to maintain some inherent properties of the data, thus making possible to identify some behaviors of the data [17]. Each line in this plot stands for a sample, and the color corresponds to a given class. Figure 5a, b depicts the Andrews plot considering “UCI-Liver-disorders” and “Statlog-Australian” datasets, respectively.

Clearly, the datasets contain a considerable amount of overlapped regions, which is a strong indicator that *ensemble* OPF$_c$ is more robust to such situations than OPF. Errors during the classification process are highly associated to the so-called tie-regions, which stand for regions in the feature space where a testing sample can be conquered by more than one training sample.

As mentioned above, OPF elects the prototype nodes as being the nearest samples from different classes, which can be found out through a MST computation over the training graph. Actually, if one has a unique MST, which means all edge-weights are different to each other, the OPF classification error on that graph would be zero, since the

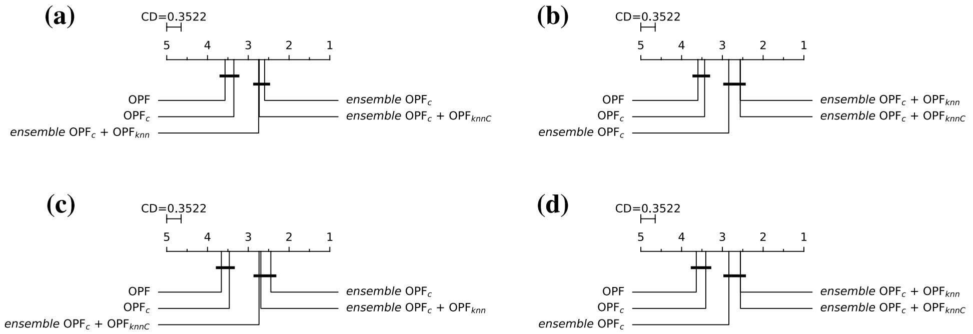


Fig. 6 Comparison of all approaches against to each other according to the average accuracies for **a** proposed approach and **b** bagging, and F-measure values for **c** proposed approach and **d** bagging concerning the Nemenyi test. Groups that are not significantly different (at $p = 0.05$) are connected to each other

Table 6 Computational load (in seconds) and standard deviation over all datasets concerning standard OPF, OPFc, and ensemble under different configuration with respect to the training time (training + calculating scores when using confidence levels)

Dataset	OPF	OPFc	*ensemble*	*ensemble*	*ensemble*
			OPFc	OPFc +	OPFc +
				OPF$_{knn}$	OPF$_{knnC}$
aflw	104.21 ± 0.9067	61.05 ± 0.6330	30.46 ± 0.4904	35.54 ± 0.5175	43.22 ± 0.5387
Pima-Indians-Diabetes	0.0111 ± 0.0001	0.0069 ± 0.0005	0.0035 ± 0.0000	0.0040 ± 0.0000	0.0048 ± 0.0000
Statlog-Australian	0.0105 ± 0.0011	0.0061 ± 0.0004	0.0030 ± 0.0002	0.0037 ± 0.0004	0.0042 ± 0.0000
Statlog-dna	2.1375 ± 0.1125	1.2724 ± 0.0960	0.6276 ± 0.0371	0.7118 ± 0.0065	0.8628 ± 0.0082
Statlog-Heart	0.0016 ± 0.0001	0.0010 ± 0.0001	0.0005 ± 0.0001	0.0006 ± 0.0001	0.0008 ± 0.0001
Synthetic1	0.0041 ± 0.0003	0.0026 ± 0.0003	0.0014 ± 0.0001	0.0017 ± 0.0002	0.0021 ± 0.0003
Synthetic2	0.0187 ± 0.0016	0.0105 ± 0.0001	0.0052 ± 0.0001	0.0060 ± 0.0001	0.0072 ± 0.0001
Synthetic3	0.0009 ± 0.0001	0.0005 ± 0.0000	0.0003 ± 0.0000	0.0003 ± 0.0001	0.0004 ± 0.0000
Synthetic4	225.15 ± 2.7878	186.18 ± 9.6521	73.11 ± 4.3084	72.50 ± 3.8332	85.44 ± 4.8247
UCI-a1a	80.54 ± 2.3472	60.24 ± 2.8357	25.92 ± 6.0696	27.19 ± 4.4609	32.02 ± 4.6964
UCI-Ionosphere	0.0036 ± 0.0002	0.0021 ± 0.0002	0.0011 ± 0.0001	0.0013 ± 0.0001	0.0016 ± 0.0001
UCI-Liver-disorders	0.0022 ± 0.0003	0.0015 ± 0.0002	0.0009 ± 0.0001	0.0009 ± 0.0001	0.0011 ± 0.0002
usps	11.46 ± 0.1212	7.6728 ± 0.1381	2.6886 ± 0.0414	3.0855 ± 0.0371	3.7447 ± 0.0413
w1a	382.12 ± 3.1877	260.08 ± 5.1479	141.95 ± 1.9616	141.64 ± 0.9371	164.72 ± 1.9294
yahoo-web-directory-topics	3.8479 ± 0.4005	2.4794 ± 0.2286	1.5434 ± 0.0626	1.7210 ± 0.0224	2.0890 ± 0.0244

optimum-paths from a prototype node to the remaining samples follow the shape of the MST. Therefore, as we are positioning the prototypes on the boundary of the classes, it is no longer possible for a sample from a given class to conquer a sample from another class. However, the above situation does not occur in practice, since there is a high probability of multiple MSTs in large datasets. In the standard OPF implementation, although the values of the possible optimum-paths that are going to be offered to a given graph node may be the same from samples from different classes, the one which reaches that node first will conquer it. In the *ensemble* OPF$_c$, when subsets of the original training set are used rather than the whole set, multiple MSTs provide distinct conquering processes, that together with the confidence level procedure improves the effectiveness of the classification phase.

In order to provide a robust statistical analysis, we performed the nonparametric Friedman test, which is used to rank the algorithms for each dataset separately. In case of Friedman test provides meaningful results to reject the null-hypothesis (i.e., all techniques are equivalent), we can perform a post hoc test further. For this purpose, we conducted the Nemenyi test [9, 20], which allows us to verify whether there is a critical difference (CD) among techniques or not. The results of the Nemenyi test can be represented in a simple diagram, in which the average ranks of the methods are plotted on the horizontal axis, where the lower the average rank is, the better the technique is. Moreover, the groups with no significant difference are connected with a horizontal line. Figure 6 depicts the statistical analysis considering the average accuracy over the test set. As one can observe, the proposed *ensemble* OPF$_c$ and *ensemble* OPF$_c$ + OPF$_{knnC}$ can be considered the most accurate techniques. Lastly, in the second group, we have the standard OPF and OPF$_c$ approaches. Such test reflects the fact *ensemble* OPF$_c$ and *ensemble* OPF$_c$ + OPF$_{knnC}$ achieved the best accuracy rates

Table 7 Computational load (in seconds) and standard deviation over all datasets using bagging strategy concerning ensemble-based OPF under different configurations with respect to the training time (training + validating)

Dataset	*ensemble* OPF*c*	*ensemble* OPF*c* + OPF_{knn}	*ensemble* OPF*c* + OPF_{knnC}
aflw	110.45 ± 0.7817	161.96 ± 1.2288	172.91 ± 0.9507
Pima-Indians-Diabetes	0.0122 ± 0.0001	0.0174 ± 0.0001	0.0185 ± 0.0001
Statlog-Australian	0.0112 ± 0.0007	0.0157 ± 0.0001	0.0167 ± 0.0001
Statlog-dna	2.1733 ± 0.0503	3.2010 ± 0.0440	3.4121 ± 0.0434
Statlog-Heart	0.0018 ± 0.0001	0.0025 ± 0.0000	0.0027 ± 0.0000
Synthetic1	0.0046 ± 0.0003	0.0068 ± 0.0005	0.0072 ± 0.0004
Synthetic2	0.0192 ± 0.0001	0.0271 ± 0.0001	0.0288 ± 0.0001
Synthetic3	0.0010 ± 0.0001	0.0014 ± 0.0002	0.0015 ± 0.0002
Synthetic4	213.13 ± 21.84	276.71 ± 9.7410	294.53 ± 10.05
UCI-a1a	101.67 ± 2.2934	123.66 ± 1.4978	131.40 ± 2.1094
UCI-Ionosphere	0.0038 ± 0.0000	0.0056 ± 0.0001	0.0060 ± 0.0000
UCI-Liver-disorders	0.0025 ± 0.0002	0.0036 ± 0.0003	0.0038 ± 0.0003
usps	9.4173 ± 0.0705	13.57 ± 0.0835	14.56 ± 0.3095
w1a	412.36 ± 1.5981	532.65 ± 2.2717	563.87 ± 3.3816
yahoo-web-directory-topics	3.9878 ± 0.2614	6.6294 ± 0.1824	7.1195 ± 0.1768

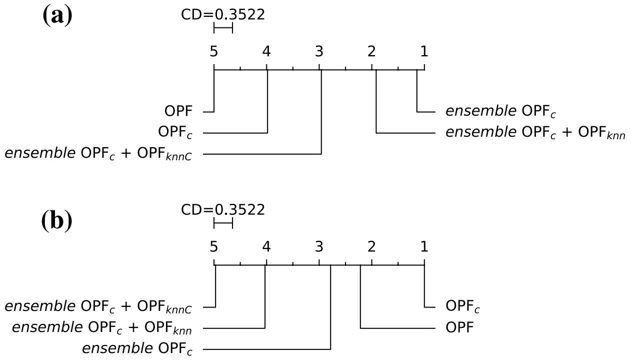


Fig. 7 Nemenyi statistical test regarding the computational load concerning to the training (training + calculating scores) phase for **a** proposed approach and **b** bagging strategy. Groups that are not significantly different (at $p = 0.05$) are connected to each other

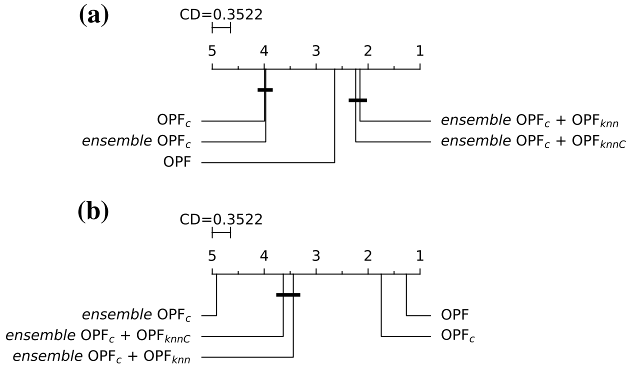


Fig. 8 Nemenyi statistical test regarding the computational load concerning to the testing phase for **a** proposed approach and **b** bagging strategy. Groups that are not significantly different (at $p = 0.05$) are connected to each other

in the majority of datasets. In fact, the statistical test did not point out a CD between each pair of ensemble-based OPF variants, which means they performed similarly in some problems.

In regard to the computational load, Tables 6 and 7 show the mean computational load considering the training (training + calculating scores when using confidence levels) phase concerning to the proposed approach and bagging approach, respectively. As expected, ensemble-based OPF is faster than standard OPF and OPF$_c$, since training into smaller subregions (disjoint training sets) is faster than training in all training data [27]. On average, i.e., considering all 15 datasets, *ensemble* OPF$_c$ has been about 2.929 times faster than standard OPF, and 2.095 times faster than OPF$_c$. Concerning the bagging strategy (Table 7), the ensemble approach was the slowest than standard OPF and OPF$_c$, since it is generated L training sets $\mathcal{D}_1^L$ by sampling from $\mathcal{D}_1$ uniformly and with replacement.

The statistical analysis for training (training + calculating scores) and testing phases is shown in Figs. 7 and 8, respectively. Figure 7a emphasizes the *ensemble* OPF$_c$ as the fastest approach in the training phase. Then, *ensemble* OPF$_c$ + OPF$_{knnC}$ showed intermediate performance, and lastly the standard OPF as the slowest one for training phase. In regard to the bagging training phase, as expected, the *ensemble* method was the slowest one, being OPF$_c$ the fastest approach, since it trains into smaller training sets (notice that the standard OPF was trained over $\mathcal{D}_1 \cup \mathcal{D}_v$, and OPF$_c$ was trained over $\mathcal{D}_1$).

In regard to the testing phase, we can stand out three groups in Fig. 8a: the first one composed of two ensemble-based: one with OPF$_c$ + OPF$_{knnC}$ and another with OPF$_{knn}$ (the fastest ones), wherewith there is no CD between them; then, the standard OPF showed intermediate performance; and the other group with *ensemble* OPF$_c$ and OPF$_c$ (the slowest ones). On average, the standard OPF has been about 1.190 times faster than *ensemble* OPF$_c$ in the testing phase, since there is more than one classifier in ensemble-based OPF. However, the ensemble-based with OPF$_{knn}$ and OPF$_{knnC}$ appeared as the fastest approaches for the testing phase, being OPF$_{knn}$ less expensive concerning the computational load for the testing phase. Regarding the testing phase using bagging strategy (Fig. 8b), its expected that the ensemble using bagging with replacement can result in a slower test phase. In short, we can drawn some conclusions:

- the proposed approach can improve standard OPF and OPF$_c$ classification results by ensemble-based OPF using a confidence levels for each training sample;
- the proposed approach provides a faster training phase; and
- bagging-based design of ensembles does not seem to help the proposed approaches, since it can lead to a number of samples with the very same confidence level.

5 Conclusions and future works

In this work, we introduced the idea of using OPF such as a bag-of-classifiers with a confidence measures to improve OPF recognition rate. The idea is to build an ensemble of classifiers using OPF with confidence-based approach proposed by Fernandes et al. [10], i.e., we want to exploit the combination of classifiers by majority votes while using confidence values and a modified formulation for OPF classification. We also validated the proposed approach in two different variants of the OPF classifier and with a bagging strategy for designing ensembles of classifiers.

Experiments over 15 datasets showed the robustness of the proposed approaches, which obtained the best results in 10 datasets and a less costly training phase when using disjoint sets compared to the bagging approach. The proposed approach also obtain better results in highly overlapped datasets, which may occur in practice. Additionally, the techniques introduced in this work are usually faster in the training phase when compared to traditional OPF (trained over $\mathcal{D}_1 \cup \mathcal{D}_v$) and OPF$_c$ (approximately 2.929 times faster than standard OPF). Future works will be guided to explore ensemble pruning strategies for the OPF classifier considering meta-and hyper-heuristics.

Acknowledgements The authors are grateful to FAPESP grants #2013/07375-0, #2014/16250-9, #2014/12236-1, and #2016/19403-6, Capes, and CNPq grants #470571/2016-6 and #306166/2014-3 for their financial support. The authors are also grateful to the reviewers for their insightful comments.

References

1. Al-Ani A, Deriche M (2002) A new technique for combining multiple classifiers using the dempster–shafer theory of evidence. J Artif Intell Res 17(1):333–361
2. Allène C, Audibert JY, Couprie M, Keriven R (2010) Some links between extremum spanning forests, watersheds and min-cuts. Image Vis Comput 28(10):1460–1471
3. Amancio DR, Comin CH, Casanova D, Travieso G, Bruno OM, Rodrigues FA, Costa LF (2014) A systematic comparison of supervised classifiers. PLoS ONE 9(4):e94,137
4. Amorim WP, Falcão AX, Papa JP, Carvalho MH (2016) Improving semi-supervised learning through optimum connectivity. Pattern Recogn 60:72–85
5. Andrews DF (1972) Plots of high-dimensional data. Biometrics 28(1):125–136
6. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
7. Castillo E, Peteiro-Barral D, Berdiñas BG, Fontenla-Romero O (2015) Distributed one-class support vector machine. Int J Neural Syst 25(07):1550,029
8. Dash JK, Mukhopadhyay S (2016) Similarity learning for texture image retrieval using multiple classifier system. Multimed Tools Appl 1–25. doi:10.1007/s11042-016-4228-y
9. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
10. Fernandes SEN, Scheirer W, Cox DD (2015) Papa JP progress in pattern recognition, image analysis, computer vision, and applications: 20th Iberoamerican congress, CIARP 2015, Montevideo, Uruguay, November 9–12, 2015, Proceedings, chap. improving optimum-path forest classification using confidence measures, pp 619–625. Springer International Publishing, Cham
11. Fernandes SEN, Souza AN, Gastaldello DS, Pereira DR, Papa JP (2017) Pruning optimum-path forest ensembles using metaheuristic optimization for land-cover classification. Int J Remote Sens 38:5736–5762
12. Folino G, Pisani FS (2015) Combining ensemble of classifiers by using genetic programming for cyber security applications. Springer International Publishing, Cham, pp 54–66
13. Giacinto G, Roli F, Fumera G (2000) Selection of classifiers based on multiple classifier behaviour. Springer, Berlin, pp 87–93
14. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844
15. Hunter JD (2007) Matplotlib: a 2d graphics environment. Comput Sci Eng 9(3):90–95
16. Kittler J, Hatef M, Duin RPW, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20(3):226–239
17. Koziol J, Hacke W (1991) A bivariate version of andrews plots. IEEE Trans Biomed Eng 38(12):1271–1274
18. Kuncheva L, Skurichina M, Duin RPW (2002) An experimental study on diversity for bagging and boosting with linear classifiers. Inf Fus 3(4):245–258
19. Kuncheva LI (2004) Combining pattern classifiers: methods and algorithms. Wiley-Interscience, New York
20. Nemenyi P (1963) Distribution-free multiple comparisons. Princeton University, Princeton
21. Papa JP, Falcão AX (2008) A new variant of the optimum-path forest classifier. In: Proceedings of the 4th international

symposium on advances in visual computing, Lecture Notes in Computer Science, Springer, Berlin, pp 935–944

22. Papa JP, Falcão AX (2009) A learning algorithm for the optimum-path forest classifier. In: Torsello A, Escolano F, Brun L (eds) Graph-based representations in pattern recognition, vol 5534. Lecture Notes in Computer Science. Springer, Berlin Heidelberg, pp 195–204
23. Papa JP, Falcão AX, Albuquerque VHC, Tavares JMRS (2012) Efficient supervised optimum-path forest classification for large datasets. Pattern Recogn 45(1):512–520
24. Papa JP, Falcão AX, Suzuki CTN (2009) Supervised pattern classification based on optimum-path forest. Int J Imaging Syst Technol 19(2):120–131
25. Papa JP, Fernandes SEN, Falcão AX (2017) Optimum-path forest based on k-connectivity: theory and applications. Pattern Recogn Lett 87:117–126
26. Ponti M, Rossi I (2013) Ensembles of optimum-path forest classifiers using input data manipulation and undersampling. Multiple Classif Syst 7872:236–246
27. Ponti MP, Papa JP (2011) Improving accuracy and speed of optimum-path forest classifier using combination of disjoint training subsets. In: Sansone C, Kittler J, Roli F (eds) Multiple classifier systems, vol 6713. Lecture Notes in Computer Science. Springer, Berlin, pp 237–248
28. Ponti MP, Papa JP, Levada ALM (2011) A Markov random field model for combining optimum-path forest classifiers using decision graphs and game strategy approach. In: San Martin C, Kim SW (eds) Progress in pattern recognition, image analysis, computer vision, and applications, Lecture Notes in Computer Science, vol 7042, pp 581–590. Springer, Berlin
29. Souza R, Rittner L, Lotufo RA (2014) A comparison between k-optimum path forest and k-nearest neighbors supervised classifiers. Pattern Recogn Lett 39:2–10
30. Wilcoxon F (1945) Individual comparisons by ranking methods. Biom Bull 1(6):80–83
31. Xu L, Krzyzak A, Suen C (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans Syst Man Cybern 22(3):418–435
32. Zhang Y, Zhou W, Yuan S (2015) Multifractal analysis and relevance vector machine-based automatic seizure detection in intracranial EEG. Int J Neural Syst 25(6):1550020