AUTHOR(S): NGUYEN, T.T., NGUYEN, M.P., PHAM, X.C., LIEW, A. W.-C. and PEDRYCZ, W.

TITLE: Combining heterogeneous classifiers via granular prototypes.

YEAR: 2018

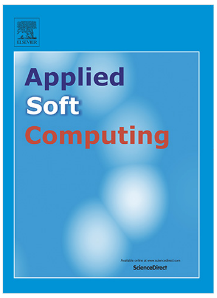
Publisher citation: NGUYEN, T.T., NGUYEN, M.P., PHAM, X.C., LIEW, A. W.-C. and PEDRYCZ, W. 2018. Combining heterogeneous classifiers via granular prototypes. Applied soft computing [online], 73, pages 795-815. Available from: https://doi.org/10.1016/j.asoc.2018.09.021

OpenAIR citation: NGUYEN, T.T., NGUYEN, M.P., PHAM, X.C., LIEW, A. W.-C. and PEDRYCZ, W. 2018. Combining heterogeneous classifiers via granular prototypes. Applied soft computing, 73, pages 795-815. Held on OpenAIR [online]. Available from: https://openair.rgu.ac.uk/

Publisher copyright statement:

OpenAIR takedown statement:

Accepted Manuscript



Combining heterogeneous classifiers via granular prototypes

Tien Thanh Nguyen, Mai Phuong Nguyen, Xuan Cuong Pham, Alan Wee-Chung Liew, Witold Pedrycz

PII: S1568-4946(18)30538-6
DOI: https://doi.org/10.1016/j.asoc.2018.09.021
Reference: ASOC 5100

To appear in: *Applied Soft Computing Journal*

Received date : 16 May 2017
Revised date : 6 September 2018
Accepted date : 13 September 2018

Please cite this article as: T.T. Nguyen, et al., Combining heterogeneous classifiers via granular prototypes, *Applied Soft Computing Journal* (2018), https://doi.org/10.1016/j.asoc.2018.09.021

Highlights:

- We modeled the base classifiers' output by using a granular prototype formalized as a vector of intervals.
- We defined a way to quantify the distance between the base classifiers' output on an observation and a granular prototype.
- We proposed a novel framework to combine multiple classifiers in an ensemble system
- The proposed method is highly competitive to several state-of-the-art ensemble methods.

Combining Heterogeneous Classifiers via Granular Prototypes

Tien Thanh Nguyen[1], Mai Phuong Nguyen[2], Xuan Cuong Pham[3], Alan Wee-Chung Liew[3], and Witold Pedrycz[4]

[1] School of Computer Science and Digital Media, Robert Gordon University, Aberdeen, Scotland, United Kingdom

[2] FPT Company, Hanoi, Vietnam

[3] School of Information and Communication Technology, Griffith University, Australia

[4] Department of Electrical & Computer Engineering
University of Alberta, Edmonton, AB, T6R 2V4 Canada

Abstract: In this study, a novel framework to combine multiple classifiers in an ensemble system is introduced. Here we exploit the concept of information granule to construct granular prototypes for each class on the outputs of an ensemble of base classifiers. In the proposed method, uncertainty in the outputs of the base classifiers on training observations is captured by an interval-based representation. To predict the class label for a new observation, we first determine the distances between the output of the base classifiers for this observation and the class prototypes, then the predicted class label is obtained by choosing the label associated with the shortest distance. In the experimental study, we combine several learning algorithms to build the ensemble system and conduct experiments on the UCI, colon cancer, and selected CLEF2009 datasets. The experimental results demonstrate that the proposed framework outperforms several benchmarked algorithms including two trainable combining methods, i.e., Decision Template and Two Stages Ensemble System, AdaBoost, Random Forest, L2-loss Linear Support Vector Machine, and Decision Tree.

Keywords: Ensemble method, multiple classifiers system, information granule, information uncertainty, supervised learning

1. Introduction

Supervised learning is an active research area in the machine learning community. Many algorithms resulting from different learning methodologies have been introduced to learn the relationship between feature vectors and class labels with the aim of generating discriminative decision model. Experiments have shown that there is no single learning algorithm that performs well on all datasets. A learner can achieve high accuracy on some data sets but high error rate on others. Ensemble learning, where multiple learning algorithms are combined into a single framework to obtain a better discriminative decision model, offers a viable solution [1].

Dietterich [2] showed the benefit of combining multiple classifiers from three aspects: statistical, computational, and representational. When a classifier is learned on a given training set, it gives a hypothesis about the relationship between the feature vectors and the class labels. With a small number of training data, different hypotheses (classifiers) can produce the same error rate on the training data. It might happen that a poor hypothesis is chosen to predict the label of an unseen sample. By combining several hypotheses, we can reduce the risk of choosing a wrong hypothesis. From the computational aspect, many algorithms perform local search to obtain locally optimum solution. In ensemble methods, by changing the starting point of algorithms, we can have a better approximation of the unknown relationship than that of a single learning algorithm. Finally, the unknown relationship in some cases cannot be modeled by a single hypothesis. By using a combination of multiple hypotheses, a better approximation for the relationship can be achieved.

In ensemble method, different "models" could refer to the different learning algorithms or to a set of generic classifiers generated by learning a unique learning algorithm on many different training sets [3]. Each learning algorithm learns a classifier on a given training set to describe the relationship between the feature vector and the class label of the training observations. The generated classifier returns the posterior probabilities, i.e., numerical class memberships that an observation belongs to different classes. A combination method is then used to aggregate the outputs of all classifiers to generate the discriminative model. As each classifier may output different results on each observation, uncertainty is introduced.

A combiner which can capture the facet of uncertainty when combining the base classifiers' outputs would be desirable. In the literature, several combiners have been introduced based on this consideration, such as fuzzy IF-THEN rule-based combiner [4] and Decision Template method [5]. In this study, we propose an ensemble framework based on modeling the uncertainty in the base classifiers' output using interval-based representations [6, 7]. Here interval-based representations are generated by the notion of information granularity. Starting from the pioneering work of Zadeh [8-10], the concept of information granules have been used to model human cognitive and decision-making activities [11-13], and have been applied to many real-world applications [14].

In homogeneous ensemble methods like AdaBoost [15], Bagging [16], and Random Forest [17], the focus is on the generation of-new training schemes from the original training set. Meanwhile, in the heterogeneous ensemble systems, a fixed set of different learning algorithms learns on the same training set to generate the different base classifiers. The outputs of these classifiers (called meta-data of Level1 data) are then combined to make the final prediction [3-5, 18]. In this type of ensembles, the approach is focused on designing algorithms that combine the meta-data to achieve higher accuracy than that using a single classifier. In this work, we use the principle of justifiable information granularity to generate granular prototypes resulting from the outputs, i.e. the meta-data, of a set of base classifiers of heterogeneous ensemble obtained from the training observations. By defining a distance function between a feature vector and a granular prototype, we propose a novel combining algorithm for the heterogeneous ensemble systems via a shortest distance-based mechanism.

The novelty of our work lies in the following:

(i) To the best of our knowledge, this is the first approach that models the uncertainty in the meta-data of training observations by using the granular prototype formalized as a vector of intervals.

(ii) We define a way to quantify the distance between the meta-data (a numerical vector) of an observation and a granular prototype (a vector of intervals).

(iii) We propose a novel combining algorithms for heterogeneous ensemble system via a shortest distance-based mechanism.

The paper is organized as follows. In Section 2, heterogeneous ensemble method and the concept of justifiable granularity in the design of information granules are introduced. In Section 3, the novel combining method based on the idea of justifiable granularity is proposed. Experimental results are presented in Section 4; here the results of the proposed method are compared with the results produced by a number of benchmark algorithms when using 26 datasets. Finally, the conclusions are presented in Section 5.

TABLE.1. SUMMARY OF MAIN NOTATION

Notation	Description
$\mathcal{D}$	Observed data or training set
$\mathbf{x}$	Observation
M	Number of classes
N	Number of training observations
N_m	Number of training observations belonging to m^{th} class
K	Number of learning algorithms
$\{y_m\}_{m=1,\ldots,M}$	Set of labels
$\{\mathcal{K}_k\}_{k=1,\ldots,K}$	K learning algorithms
$\{BC_k\}_{k=1,\ldots,K}$	K base classifiers associated with K learning algorithms
$\mathbf{L}$	Meta-data or Level1 data of $\mathcal{D}$
$\mathbf{L}(\mathbf{x})$	Meta-data or Level1 data of observation $\mathbf{x}$
$\mathbf{L}_m$	Meta-data or Level1 data related to the m^{th} class
$\mathbf{L}_{m,j}$	j^{th} column of $\mathbf{L}_m$
$\mathrm{C}\{\cdot\}$	Relative cardinality of a set
$\left[\underline{v_{mj}}, \overline{v_{mj}}\right]$	Interval computed from j^{th} attribute of $\mathbf{L}_m$ ($j = 1, \ldots, MK;\ m = 1, \ldots, M$)
$\mathbf{V}_m = \left\{\left[\underline{v_{mj}}, \overline{v_{mj}}\right]\right\}_{j=1,\ldots,MK}$	Granular prototype for the m^{th} class ($m = 1, \ldots, M$)
$\mathcal{V} = \{\mathbf{V}_m\}_{m=1,\ldots,M}$	Set of M prototypes
$d(x, [\cdot])$	Distance between scalar x and interval
$\mathbf{d}(\mathbf{t}, \mathbf{V})$	Distance between a vector $\mathbf{t}$ and an interval prototype $\mathbf{V}$

2. Related Work

2.1. Ensemble method

Over the past years, many approaches related to ensemble methods have been proposed, and there are different taxonomies of ensemble methods [1, 18-22]. We follow the taxonomy in [22] in which ensemble methods are divided into two types:

- Homogeneous ensemble: A set of classifiers are generated on different training sets obtained from an original one by using the same learning algorithm. The outputs of these classifiers are combined to give the final decision. Several state-of-the-art ensemble methods in the literature are AdaBoost [15], Bagging [16], and Random Forest [17].
- Heterogeneous ensemble: Several different learning algorithms are learned on the same training set to generate the different base classifiers. The heterogeneous ensemble focuses more on the combining strategies on the meta-data [3, 18, 23-26]) to achieve higher accuracy than a single classifier.

In the literature, besides the practical applications of ensemble methods in many areas, research on ensemble methods can be divided into three aspects:

- Design of new ensemble systems: Several recent research efforts have focused on designing new ensemble systems. Rodríguez et al. [27] proposed the Rotation Forest in which principal component analysis (PCA) is applied to each of the K subsets randomly selected from a feature set. The K axis rotations form the new features for a base classifier. Blaser and Fryzlewicz [28] designed a novel ensemble system by generating random rotation matrices to rotate the feature space before generating the base classifiers. Wu [29] proposed a new ensemble learning paradigm with the consideration of implicit supplementary information about the performance orderings for the trained base classifiers in previous literature. By measuring the similarity between the two learning tasks, the supplementary ordering information for the trained classifiers of a given learning task can be inferred so as to obtain the optimal combining weights of the trained classifiers. Moreover, several ensemble systems were developed for different learning paradigms such as incremental learning [30-32], semi-supervised learning [33], and multi-label learning [34, 35]. For instance, Pham et al. [31] combined random projections and Hoeffding tree to construct an incremental online ensemble learning system. Krawczyk and Cano [32] incrementally learnt a threshold for each arrived instance in the online heterogeneous ensemble system. Classifiers were selected for the prediction if their support on each instance exceeds the threshold. Wu et al. [35] proposed ML-FOREST algorithm to learn an ensemble of hierarchical multi-label classifier trees to reveal the intrinsic label dependencies. Finally, besides the two popular combiners i.e. Sum and Majority Vote [4, 36], novel combining algorithms were introduced to enhance the task of combining on classifiers' outputs. For example, Kuncheva et al. [18] used the Ordered Weighted Averaging (OWA) operators to aggregate the classifiers' outputs. Wang et al. [37] proposed a new fusion scheme based on the upper integrals. Costa et al. [38] used the generalized mixture functions as a combining algorithm in which the weight each classifier put on a class was set dynamically in the combination process.

- **Enhancing existing ensemble methods**: This approach focuses on techniques to enhance the performance of some popular ensemble methods such as Boosting [15], Bagging [16], Random Forest [17], and Random Subspace [39]. Several classifier selection or redundant classifier pruning methods were proposed for this purpose, e.g., dynamic classifiers selection [40, 41], instance-based pruning [42], clustering-and-selection approach [43], and double pruning scheme (static and dynamic pruning working together) [44]. There are also hybrid approaches to weigh base classifiers in Random Subspace [45], and weigh feature subspaces in Bagging [46]. Yu et al. [47] proposed the hybrid incremental ensemble learning which combines feature space-based learning and sample space-based learning in a single framework. Several methods have been introduced to improve the performance of AdaBoost, for example by maximizing the margin between training samples of different classes via linear programming in LPBoost [48], via quadratic programming in TotalBoost [49], and learning from skewed training data in RUSBoost [50] to handle imbalanced datasets.

- Study on properties of the ensemble: The research studies the properties of an ensemble system such as diversity, margin, and generalization error bound, and their relationships and uses them to enhance the ensemble's performance. For instance, Kuncheva et al. [51] studied ten diversity measures and examined the relationships between the accuracy and measures of diversity. Tang et al. [52] theoretically analyzed six diversity measures to understand the relations between them and the concept of margin. Gao and Zhou [53] obtained a tight generalization error bound by considering the empirical average margin and margin variance. Wang et al. [54] studied the relationship between the model's generalization ability and fuzziness of fuzzy classifiers. Kuncheva et al. [55] derived bounds with a kappa-error diagram which is used to analyze the performance of ensemble systems. Li et al. [56] extended the definition of margin based on the classification confidence of the base classifiers. The weights of the base classifiers then were computed by minimizing the margin induced classification loss. Gou et al. [57] studied margin and diversity of ensemble systems and applied them to the ensemble pruning process.

2.2. Heterogeneous ensemble method

In this paper, we are concerned with the heterogeneous ensemble method. For an observation $\mathbf{x}$, let $\mathrm{P}_k(y_m|\mathbf{x})$ be the probability that $\mathbf{x}$ belongs to the class with the label y_m given by the k^{th} classifier. Kuncheva et al. [7] summarized three types of output for $\mathbf{x}$ for each $k = 1, \dots, K$:

- Crisp Label: return only class label $\mathrm{P}_k(y_m|\mathbf{x}) \in \{0,1\}$ and $\sum_m \mathrm{P}_k(y_m|\mathbf{x}) = 1$.
- Fuzzy Label: return posterior probabilities that $\mathbf{x}$ belongs to classes, i.e. $\mathrm{P}_k(y_m|\mathbf{x}) \in [0,1]$ and $\sum_m \mathrm{P}_k(y_m|\mathbf{x}) = 1$.
- Possibilistic Label: the same as fuzzy label but does not require the sum of all posterior probabilities to equal one, i.e. $\mathrm{P}_k(y_m|\mathbf{x}) \in [0,1]$ and $\sum_m \mathrm{P}_k(y_m|\mathbf{x}) > 0$.

In this study, we consider the meta-data in the form of the fuzzy label. The meta-data of N training observations is a $N \times MK$ posterior probability matrix $\{\mathrm{P}_k(y_m|\,\mathbf{x}_n)\}m = 1, \dots, M;\ k = 1, \dots, K;\ n = 1, \dots, N$ defined by:

$$\mathbf{L} = \begin{bmatrix} \mathrm{P}_1(y_1|\mathbf{x}_1) & \cdots & \mathrm{P}_1(y_M|\mathbf{x}_1) & \cdots & \mathrm{P}_K(y_1|\mathbf{x}_1) & \cdots & \mathrm{P}_K(y_M|\mathbf{x}_1) \\ \mathrm{P}_1(y_1|\mathbf{x}_2) & \cdots & \mathrm{P}_1(y_M|\mathbf{x}_2) & \cdots & \mathrm{P}_K(y_1|\mathbf{x}_2) & \cdots & \mathrm{P}_K(y_M|\mathbf{x}_2) \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathrm{P}_1(y_1|\mathbf{x}_N) & \cdots & \mathrm{P}_1(y_M|\mathbf{x}_N) & \cdots & \mathrm{P}_K(y_1|\mathbf{x}_N) & \cdots & \mathrm{P}_K(y_M|\mathbf{x}_N) \end{bmatrix} \quad (1)$$

whereas the meta-data of an observation $\mathbf{x}$ is given by:

$$\mathbf{L}(\mathbf{x}) = [\mathrm{P}_1(y_1|\mathbf{x}) \quad \cdots \quad \mathrm{P}_1(y_M|\mathbf{x}) \quad \cdots \quad \mathrm{P}_K(y_1|\mathbf{x}) \quad \cdots \quad \mathrm{P}_K(y_M|\mathbf{x})] \quad (2)$$

There are two techniques to combine the meta-data, namely the fixed combining methods and the trainable combining methods [3, 22]. The advantage of applying fixed combining methods for an ensemble system is that no training based on the meta-data of training observations is needed; as a result, they have less time complexity than their counterparts. Several popular fixed combining methods are Sum Rule, Product Rule, Max Rule, Min Rule, Median Rule, and Majority Vote Rule [4, 36], in which Majority Vote Rule and Sum Rule are the most popular. Kittler et al. [36] showed that the Sum Rule is developed under two assumptions: "conditional independence of respective representations used by the classifiers and classes being highly ambiguous", and Sum Rule generally results in the most reliable predictions. Kuncheva [58] proved the theoretical probability of error related to different rules by making assumptions about normal and uniform distribution. The Ordered Weighted Averaging operator (OWA), one of the most well-known operators applied to Decision Making Systems, has also been applied to the combiners in ensemble systems [18, 26]. This operator is used to compute average value based on weight, but instead of focusing on the original meta-data like in the fixed rules, it is linked to the order of data. As a result, the predictions at specific locations can receive more attention than the others.

In contrast, trainable combining methods utilize the knowledge in the meta-data of the training set to obtain the prediction model. Although the computational cost would increase, they generally lead to higher classification accuracy [3]. The trainable combining methods are based on the stacked generalization paradigm (also called stacking algorithm) that was first proposed by Wolpert [59]. Stacking algorithm first trains several first-level learners on the original training set using different learning algorithms. Then another learning algorithm (also called the combining algorithm) is trained on the predictions of the first-level learners to obtain the second-level learner.

Trainable combining methods are constructed based on the meta-data of the training observations which can be obtained via the Cross Validation procedure [3, 60, 61]. First, the training set is divided into several disjoint parts of equal size. One part plays the role of testing in turn, and the rests assume the role of training during the training phase. The meta-data of the observations in testing part is obtained by classifiers learned on the training part. Several strategies have been proposed to exploit label information in $\mathbf{L}$ in the combining method in which two well-known approaches are weight-based classifiers methods and the meta-data modeling-based methods.

The first strategy is based on the assumption that each classifier is assigned a different weight for each class label, and a combining algorithm is then conducted based on the M linear combinations of posterior probabilities and the associated weights for the M classes. The predicted class label for an unseen observation is decided by selecting the maximum value among these combinations. Several methods have been proposed to weigh the base classifiers. Ting et al. [61] proposed the MLR method by solving M Linear Regression models corresponding to the M classes based on the meta-data and the training data labels in crisp form to find these combining weights. Zhang and Zhou [62] used linear programming to find the weights of the base classifiers. Sen et al. [63] introduced a method inspired by MLR which uses a hinge loss function in the combiner. By using this function with regularization, three different combinations were proposed, namely weighted sum, dependent weighted sum, and linear stacked generalization, based on different regularizations with group sparsity.

On the other hand, the second strategy aims to construct the M representations on the meta-data associated with the M class labels. The discriminative decision model is obtained based on the similarity between these representations and the meta-data of unseen observation. Kuncheva et al. [5] introduced Decision Template method in which the representations (called the Decision Template) are acquired by averaging values of the meta-data belonging to each class. The class label is assigned to unseen observation if the associated Decision Template is nearest to its meta-data. The advantage of Decision Template method is that it saves time in both training and classification due to its simple computation. However, this method could have high error rate if the classifiers do not have high enough accuracy due to the fact that the simple Decision Template may not provide a good representation for a particular class. Nguyen et al. [3] modeled the likelihood distribution of the meta-data associated with each class label by a Gaussian distribution computed using Variational Inference method. The combining algorithm is then obtained using Bayesian theorem where an unseen observation is assigned to the class label associated with the maximum posterior probability.

There are trainable combining methods that do not belong to the above strategies. Merz [60] proposed SCANN, an ensemble method compose of Stacking, Correspondence Analysis (CA) and kNN. In this method, CA is applied to an indicator matrix formed on the meta-data and the true labels of the training observations. After that, kNN is used to classify unseen observations in the new scaled space. The method is sometimes impractical due to the singularity characteristic of the indicator matrix which cannot be handled by CA. Moreover, the classification process of SCANN is more complicated than that of other combining classifier algorithms, and this increases the classification time. Nguyen et al. [24] learned a Decision Tree C4.5 on the meta-data of the training set to create the second-level classifier. This model is combined with Genetic Algorithm to select the subset of features on the meta-data. Another approach is Meta Decision Tree [64], a new Decision Tree on the meta-data where at each node, a classifier is chosen instead of selecting a value for splitting an attribute. The entropy and maximum posterior probability are also added to the meta-data to enhance the discrimination ability but no theoretical basis was provided about the effectiveness of that expansion. Zhang and Duin [22] compared the performance of several heterogeneous ensemble methods with fixed combining rules and several second-level learners such as Naïve Bayes classifier and Fisher classifier. The experiments on just one hand gesture dataset with 3 different sizes of the training set, however, do not present a convincing comparison. Recently, Nguyen et al. [4] proposed a hybrid combining classifier system in which fuzzy rules work on the meta-data to produce the classification model. Although that system outperforms other fuzzy rules-based methods and ensemble methods in the experiment since the uncertainty in the meta-data can be captured by the fuzzy rules, the training process has high time complexity than other training combining methods due to a large number of rules generated.

2.3. The principle of justifiable information granularity

Normally, point statistics such as mean, median and skewness are often used to describe the data in many real-world applications. However, in many scenarios, point-wise information is less useful for subsequent reasoning [12]. Instead, information granularity which explicitly models the inherent uncertainty present in the data is more preferred. In this study, we aim to design information granule to describe sample data $\mathbf{D}$ in the form of an interval $\Omega = [a, b]$ in which a and b are lower and upper bounds of the interval, respectively. There are two intuitively compelling requirements needed to be considered [65-67]. First, the information granule Ω should reflect the existing data in such a way that the interval set becomes more legitimate as more data are within the bounds of Ω. On the other hand, information granule should exhibit high specificity. This implies that the smaller (more compact) the information granule is, the better (higher specificity) it is.

We apply the principle of justifiable granularity [14, 66] to construct interval Ω to satisfy the two requirements above. As the distribution of $\mathbf{D}$ is generally not known in advance, the experimental evidence can be determined by the cardinality $\mathrm{C}\{\mathbf{D}\}$ of the set of elements in $\mathbf{D}$ falling within the bounds of Ω. More generally, an increasing function f_1 of $\mathrm{C}\{\mathbf{D}\}$ can be considered in the form of:

$$f_1(\mathrm{C}\{\mathbf{D}\}) = (\mathrm{C}\{\mathbf{D}\})^{\beta}, \beta > 0 \tag{3}$$

Meanwhile, the specificity of the interval can be specified based on its length since shorter interval results in better specificity. In the same way, we use a continuous non-increasing function of the length of the interval expressed in the form:

$$f_2(|a-b|) = \exp(-\alpha|a-b|),\ \alpha > 0 \tag{4}$$

in which $|a-b|$ is the length of interval $\Omega = [a, b]$.

The two requirements above lead to the following optimization problem:

$$\begin{cases} f_1\{\mathrm{C}\{\mathbf{D}\}\} \to max \\ f_2\{|b-a|\} \to max \end{cases} \tag{5}$$

It is noted that the two objective functions in (5) are in conflict since increasing $f_1\{\mathrm{C}\{\mathbf{D}\}\}$ would increase $|a-b|$, resulting in the decrease in $f_2(u)$. A compromise can be reached by using the product of these two functions and maximizing the expression with respect to the bounds of the interval:

$$f_1(\mathrm{C}\{\mathbf{D}\}) \times f_2(|a-b|) \tag{6}$$

We choose the median of data in $\mathbf{D}$ (denoted by $med(\mathbf{D})$) as the numerical representative of the set of data around which Ω is created. Here, we only discuss the procedure to construct b (a is determined similarly). Based on (3), (4), and (6), we compute the compromise associated with b:

$$V(b) = (\mathrm{C}\{x_k \in \mathbf{D} \mid med(\mathbf{D}) \le x_k \le b\})^{\beta} \times \exp\big(-\alpha(|med(\mathbf{D}) - b|)\big) \tag{7}$$

The optimal upper bound of the interval is determined by maximizing the values of $V(b)$ i.e.

$$b_{opt} = \arg\max \ \{V(b) | b \ge med(\mathbf{D}), b \in \mathbf{D}\} \tag{8}$$

The optimal lower bound is found in the same manner

$$a_{opt} = \arg\max \ \{V(a) | a \le med(\mathbf{D}), a \in \mathbf{D}\} \tag{9}$$

where

$$V(a) = (\mathrm{C}\{x_k \in \mathbf{D} \mid a \le x_k \le med(\mathbf{D})\})^{\beta} \times \exp\big(-\alpha(|med(\mathbf{D}) - a|)\big) \tag{10}$$

A special case is noted in proposition 1 when the principle of justifiable granularity is applied to the two-class classification problems.

Proposition 1: If $[a_{opt}, b_{opt}]$ is the interval built by justifiable granularity on the meta-data associated with the first class label of a two-class classification problem, the interval associated with the other class label is $[1 - b_{opt}, 1 - a_{opt}]$ (See Appendix for the detailed proof)

Therefore, for binary classification, interval construction is only needed for the first class label while the interval for the second class label can be derived directly from the first one.

3. Proposed framework

In this paper, we focus on developing a classification framework by applying justifiable granularity to the meta-data of training observations. Specifically, we model the uncertainty in the base classifier outputs by constructing class interval associated with each class label (called granular prototype) from the meta-data. The proposed framework is illustrated in Figures 1, 2, and 3.

$$\mathbf{L}_m = \begin{bmatrix} \mathrm{P}_1(y_1|\mathbf{x}_1^m) & \cdots & \mathrm{P}_1(y_M|\mathbf{x}_1^m) & \cdots & \mathrm{P}_K(y_1|\mathbf{x}_1^m) & \cdots & \mathrm{P}_K(y_M|\mathbf{x}_1^m) \\ \mathrm{P}_1(y_1|\mathbf{x}_2^m) & \cdots & \mathrm{P}_1(y_M|\mathbf{x}_2^m) & \cdots & \mathrm{P}_K(y_1|\mathbf{x}_2^m) & \cdots & \mathrm{P}_K(y_M|\mathbf{x}_2^m) \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \mathrm{P}_1(y_1|\mathbf{x}_{N_m}^m) & \cdots & \mathrm{P}_1(y_M|\mathbf{x}_{N_m}^m) & \cdots & \mathrm{P}_K(y_1|\mathbf{x}_{N_m}^m) & \cdots & \mathrm{P}_K(y_M|\mathbf{x}_{N_m}^m) \end{bmatrix} \quad (11)$$

$$\begin{array}{ccccc} \Downarrow & & \Downarrow & & \Downarrow \\ \left[\underline{\mathrm{P}_1(y_1|.)}, \overline{\mathrm{P}_1(y_1|.)}\right] & \ldots & \left[\underline{\mathrm{P}_1(y_M|.)}, \overline{\mathrm{P}_1(y_M|.)}\right] & \ldots & \left[\underline{\mathrm{P}_K(y_M|.)}, \overline{\mathrm{P}_K(y_M|.)}\right] \end{array}$$

We use a Cross Validation-based procedure to generate the meta-data from the training set (see Figure 1). Specifically, T-fold Cross Validation is applied to the training set $\mathcal{D}$ to obtain T disjoint parts $\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_T$, $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset \ (i \neq j)$, and $|\mathcal{D}_1| \approx \cdots \approx |\mathcal{D}_T|$. Meta-data of observations in $\mathcal{D}_i$ is then formed by the classifiers (denoted by BC_k^{-i}) generated by learning the K learning algorithms on $\widetilde{\mathcal{D}}_i = \mathcal{D} - \mathcal{D}_i$. The meta-data of all training observations belonging to $\mathcal{D}$ is finally obtained by concatenating all meta-data from each $\mathcal{D}_i$ into the form of matrix $\mathbf{L}$ given by (1). Since class labels of training observations are known in advance, $\mathbf{L}$ can be separated into M groups corresponding to the M class labels i.e. $\mathbf{L}_m = \{\mathbf{L}(\mathbf{x})|y(\mathbf{x}) = y_m\}(m = 1, \ldots, M)$. If the meta-data of the m^{th} class contains N_m observations, $\mathbf{L}_m$ is a $N_m \times MK$ matrix as shown in (11). On the j^{th} column of $\mathbf{L}_m (j = 1, \ldots, MK)$, the principle of justifiable granularity is applied to obtain the interval to represent all elements (posterior probabilities) in that column.

Let $V_{mj} = \left[\underline{v_{mj}}, \overline{v_{mj}}\right]$ denotes the interval obtained on the j^{th} column $(j = 1, \ldots, MK)$ of $\mathbf{L}_m$. After looping though all MK columns, we obtain MK intervals associated with MK columns of $\mathbf{L}_m$, denoted by $\mathbf{V}_m = \{V_{mj}\}_{j=1,\ldots,MK}$. Doing this for all M, we obtain a set of M granular prototypes i.e. $\mathcal{V} = \{\mathbf{V}_m\}(m = 1, \ldots, M)$ representing the M class labels. $\mathbf{V}_m$ is our novel information granules representation for the m^{th} class label. Note that $\mathbf{V}_m$ is a vector of interval values and is different to the representation used in Decision Template method [5] where mean value is used to describe the posterior probabilities in each column of $\mathbf{L}_m$. At the end, the base classifiers $\{BC_k\}$ $(k = 1, \ldots, K)$ are generated by learning the K learning algorithms on entire training set $\mathcal{D}$. The training process will output the M granular prototypes $\mathcal{V} = \{\mathbf{V}_m\}$ associated with M class labels and K base classifiers $\{BC_k\}$. These outputs will be used as the input for the classification process.

During classification, for each unlabeled observation $\mathbf{x}^u$, we compute its meta-data $\mathbf{L}(\mathbf{x}^u)$ in the form of vector (2) by classifying $\mathbf{x}^u$ with the K base classifiers BC_k. The class label for $\mathbf{x}^u$ is predicted by calculating the distance between $\mathbf{L}(\mathbf{x}^u)$ and prototype $\mathbf{V}_m (m = 1, \dots, M)$ and then selecting the smallest value among all distances. To do this, we need to define the distance between a numerical vector and a granular prototype.

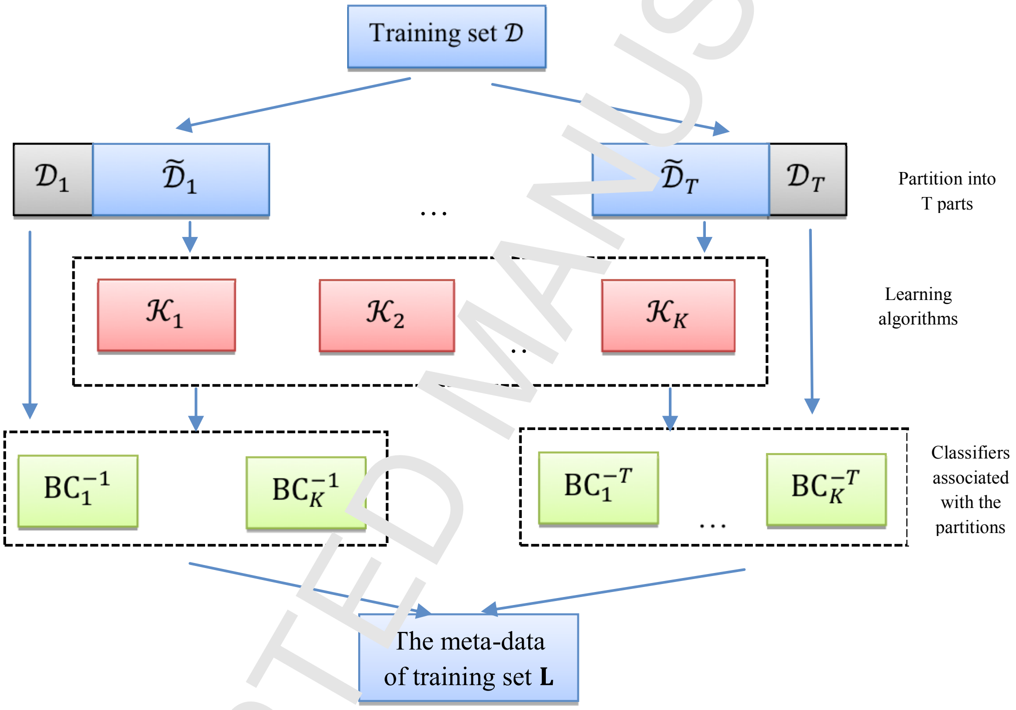


Fig.1. Meta-data generation

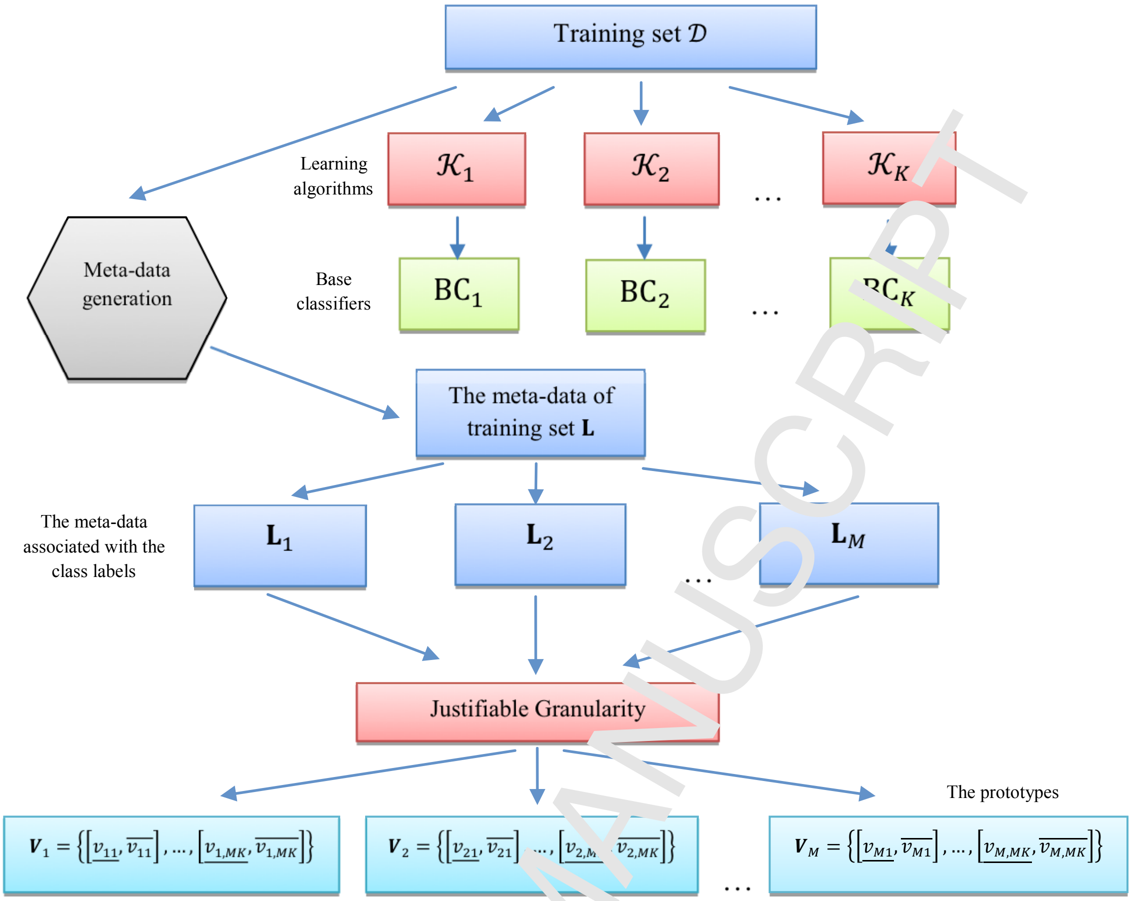

Training set $\mathcal{D}$
Learning algorithms
$\mathcal{K}_1$
$\mathcal{K}_2$
$\mathcal{K}_K$
Meta-data generation
Base classifiers
BC_1
BC_2
BC_K
The meta-data of training set **L**
The meta-data associated with the class labels
$\mathbf{L}_1$
$\mathbf{L}_2$
$\mathbf{L}_M$
Justifiable Granularity
The prototypes
$V_1=\{[\underline{v_{11}},\overline{v_{11}}],\dots,[\underline{v_{1,MK}},\overline{v_{1,MK}}]\}$
$V_2=\{[\underline{v_{21}},\overline{v_{21}}],\dots,[\underline{v_{2,MK}},\overline{v_{2,MK}}]\}$
$V_M=\{[\underline{v_{M1}},\overline{v_{M1}}],\dots,[\underline{v_{M,MK}},\overline{v_{M,MK}}]\}$

Fig.2. Training process of the proposed method

We start with the definition of a distance between a numerical value and an interval. The distance is inspired by the distance between two intervals $d([x_1,x_2],[a,b]) = \max\{|x_1-a|,|x_2-b|\}$ as defined in [68]. Since we can write $x=[x,x]$, we define the distance between a numerical value and an interval as:

Definition 1: The distance between a numerical value x and an interval $[a,b]$ is given by:

$$d(x,[a,b]) = \max\{|x-a|,|x-b|\} \tag{12}$$

Several interesting properties of the distance function in (12) are listed below. They can be regarded as a generalization of the "classical" vector distance. The proof of these properties is covered in the Appendix. These properties ensure that the distance function defined in (12) is a proper metric. For example, Property 3 ensures that any prediction that falls inside the interval is more reliable than those that fall outside the interval. Property 4 ensures that if x_1 is close to $[a,b]$ and x_2 is close to $[a,b]$, then x_1 and x_2 must be close to each other.

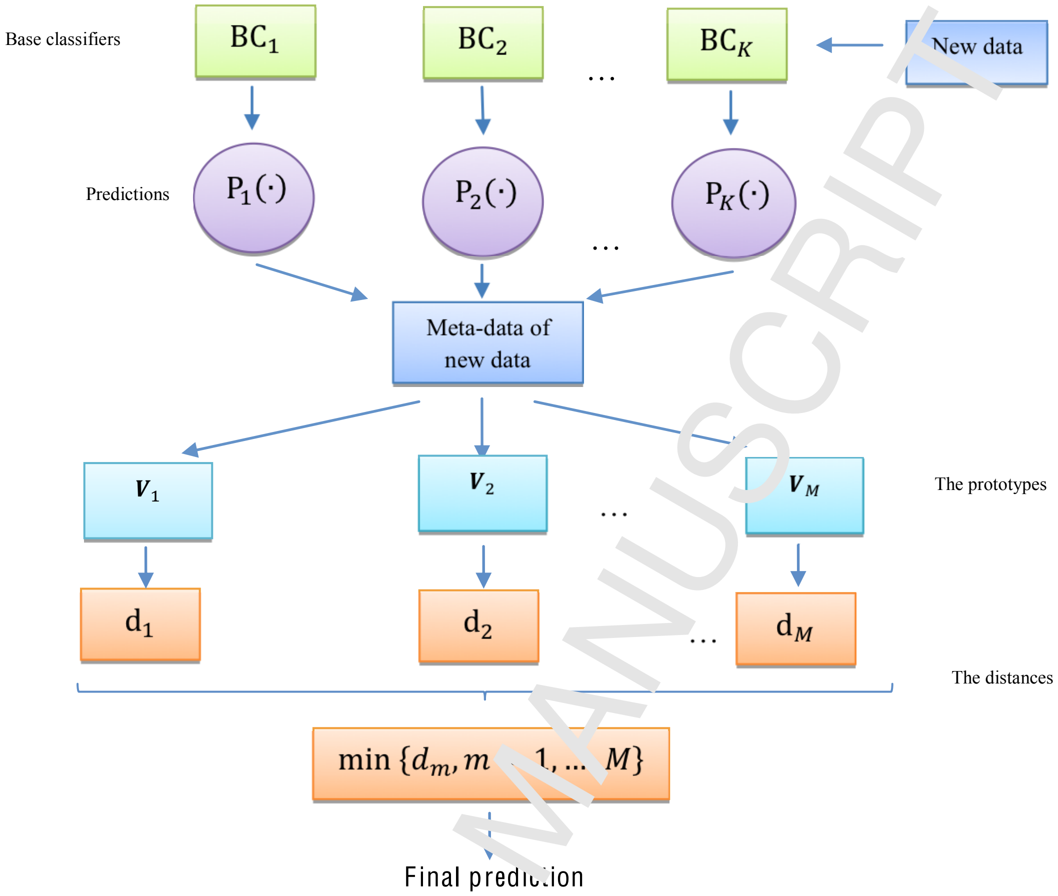


Fig.3.Classification process of the proposed method

Property 1 (Positive Definiteness): $d(x,[a,b]) \geq 0$ and $d(x,[a,b]) = 0 \leftrightarrow x = a = b$ (13)

Property 2 (Equality): $d(x_1,[a,b]) = d(x_2,[a,b]) \Leftrightarrow x_1 = x_2$ or $x_1 + x_2 = a + b$ (14)

Property 3 (Consistency):If $x_1 \in [a,b]$ and $x_2 \notin [a,b]$ then $d(x_1,[a,b]) < d(x_2,[a,b])$ (15)

Property 4 (Triangle Inequality): $d(x_1,[a,b]) \leq d(x_1,x_2) + d(x_2,[a,b])$ (16)

Property 5 (Symmetry):$d(x,[a,b]) = d([a,b],x)$ (17)

Property 6 (Scale Invariance):$d(\alpha x,[\alpha,\alpha][a,b]) = |\alpha| d(x,[a,b])$ (18)

Property 7 (Translation Invariance): $d(x,[a,b]) = d(x + \alpha,[a,b] + [\alpha,\alpha])$ (19)

Using Definition 1, we can define the distance between a numerical vector and a granular prototype as:

Definition 2: The distance between a vector $\mathbf{t}$ and a granular prototype $\mathbf{V} = \{V_j\}(j = 1, \dots, |\mathbf{V}|)$ is defined by:

$$\mathbf{d}(\mathbf{t}, \mathbf{V}) = \sum_{j=1}^{|\mathbf{V}|} d(t_j, V_j) \tag{20}$$

in which $d(t_j, V_j)$ is the distance between the j^{th} attribute of $\mathbf{t}$ and the interval V_j given by (12).

Two important properties of $\mathbf{d}(\mathbf{t}, \mathbf{V})$ are outlined as follows

Property 8 (Consistency): If $\mathbf{t}_1 = \{t_{1j}\}\, t_{1j} \in V_j$ and $\mathbf{t}_2 = \{t_{2j}\}\, t_{2j} \notin V_j\ \forall j = 1, \dots, |\mathbf{V}|$ then

$$\mathbf{d}(\mathbf{t_1}, \mathbf{V}) < \mathbf{d}(\mathbf{t_2}, \mathbf{V}) \tag{21}$$

Property 9 (Triangle Inequality): $\mathbf{d}(\mathbf{t}_1, \mathbf{V}) \le \mathbf{d}(\mathbf{t}_1, \mathbf{t}_2) + \mathbf{d}(\mathbf{t}_2, \mathbf{V})$ (22)

where $\mathbf{d}(\mathbf{t}_1, \mathbf{t}_2)$ is the distance between two vector $\mathbf{t}_1$ and $\mathbf{t}_2$. Their proof is presented in Appendix.

We can now compute the distance between the meta-data of unlabeled observation $\mathbf{x}^u$, i.e. $\mathbf{L}(\mathbf{x}^u)$, and the M granular prototypes $\mathbf{V}_m (m = 1, \dots, M)$ and predict the class label to be the one that is associated with the shortest distance

$$\mathbf{x}^u \in y_t \text{ if } \mathbf{d}(\mathbf{L}(\mathbf{x}^u), \mathbf{V}_t) = \min_{m=1,\dots,M} \mathbf{d}(\mathbf{L}(\mathbf{x}^u), \mathbf{V}_m) \tag{23}$$

The algorithms which summarize the training and classification process of the proposed method are introduced in the Appendix. It is noted that there are two parameters α and β whose values need to be set. Their effect on the classification results will be discussed in the next section.

4. Experimental Studies

4.1. Datasets and Experimental Setting

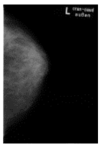
The experiments were carried out using 24 datasets selected from the UCI repository [69]. These datasets were selected as they are often used to validate the performance of various classification systems. To ensure the objectiveness in the comparison between our method and benchmark algorithms, we conducted the experiments on datasets having few hundred (e.g., Hepatitis, Iris, and Wine) and few thousands of observations (e.g., Twonorm, Musk2, and Satimage). The number of attributes also varies from 3 (Haberman) to 649 (Multiple Features). We also conducted the experiment on two additional datasets i.e. a medical imaging dataset and a colon cancer dataset. The medical imaging dataset is selected from the CLEF2009 database which is a large x-ray database collected by Archen University, Germany [70]. Here we chose the 10 class dataset from this database for the experiment. Histogram of Local Binary Pattern (HLBP) was selected as feature vector of the image. The colon cancer dataset [71] includes 62 samples collected from colon cancer patients in which 40 patients suffer from colon cancer and the remaining are normal (see Table 2 and 3).

TABLE 2. UCI DATA: MAIN CHARACTERISTICS

	# of features	# of classes	# of observations	% of observations in each class
Artificial	10	2	700	57.14%, 42.86%
Australian	14	2	690	44.49%,55.51%
Biodeg	41	2	1055	33.74%,66.26%
Blood	4	2	748	23.80%,76.20%
Breast Cancer	9	2	683	65.01%,34.99%
Cleveland	13	5	297	18.18%,11.78%,11.78%,4.38%,53.87%
Colon	2000	2	62	64.51%, 35.49%
Conn Bench Vowel	10	11	528	9.09% for each class label
Contraceptive	9	3	1473	42.70%,22.61%,34.69%
Dermatology	34	6	358	31.01%,16.76%,19.83%,13.41%,13.41%,5.59%
Glass	9	6	214	32.71%,35.51%,7.94%,6.07%,4.21%,13.55%
Haberman	3	2	306	73.53%,26.47%
Heart	13	2	270	55.56%,44.44%
Hepatitis	19	2	80	16.25%,83.75%
Iris	4	3	150	33.33%-33.33%-33.33%
Led7digit	7	10	500	7.4%,10.2%,11.4%,10.4%,10.4%,9.4%,11.4%,10.6%,9.8%,9%
Madelon	500	2	2000	50%,50%
Multiple Features	649	10	2000	10% for each class label
Musk2	166	2	6598	84.59%,15.41%
Satimage	36	6	6435	23.82%,10.92%,21.10%,9.73%,10.99%,23.43%
Texture	40	10	5500	9.09%,9.09%,9.09%,18.18%,9.09%,9.09%,9.09%,9.09%,9.09%,9.09%
Twonorm	20	2	7400	49.96%,50.04%
Vertebral	6	3	310	19.35%,48.39%,32.26%
Wine	13	3	178	33.15%,39.89%,26.97%
Yeast	8	10	1484	31.20%,28.91%,16.44%,10.98%,3.44%,2.96%,2.36%,2.02%,1.35%,0.34%

TABLE 3. 10 CLASS DATASET FROM THE CLEF2009 MEDICAL IMAGE DATABASE

Image					
Description	Abdomen	Cervical	Chest	Facial cranium	Left Elbow
Number of observation	80	81	80	80	69

Image					
Description	Left Shoulder	Left Breast	Finger	Left Ankle Joint	Left Carpal Joint
Number of observation	80	80	66	80	80

We used 3 learning algorithms namely Linear Discriminant Analysis (denoted by LDA), Naïve Bayes, and k-Nearest Neighbors (denoted by kNN) to learn the base classifiers. The choice of these algorithms is to demonstrate that an ensemble system built using just simple learning algorithms can achieve high classification accuracy. Moreover, in a heterogeneous ensemble system, a set of diverse learning algorithms should be used to increase the system diversity. Less diverse learning algorithms usually output hypotheses with similar classification results so that the ensemble has less chance to improve the overall performance [1]. Here, LDA, Naïve Bayes, and kNN are three learning algorithms with significantly different strategies, and they ensure the generation of diverse outputs. For Naïve Bayes classifier we used Gaussian to approximate the likelihood distribution of each feature of the original data. For kNN, the value of k was set to 5, denoted as kNN$_5$. The mean and variance of classification error rates of these learning algorithms are shown in Table 4-6.

For comparison, we choose the benchmark algorithms consisting of:

- Decision Template method: We used the similarity measure S_1 defined by $\mathrm{S}_1(\mathbf{L}(\mathbf{x}), DTem_m) = \frac{\{\mathbf{L}(\mathbf{x})\cap DTem_m\}}{\{\mathbf{L}(\mathbf{x})\cup DTem_m\}}$ where $DTem_m$ is the Decision Template of m^{th}class [5].
- AdaBoost [15]: Decision Tree C4.5 was used as the learning algorithm with 200 iterations as in [4]. We used AdaBoost.M1 (for the binary classification problems) and AdaBoost.M2 (for the multi-class classification problem) from the Statistics and Machine Learning Toolbox of Matlab.
- Random Forest [17]: We used Decision Tree C4.5 as the learning algorithm. 200 trees were created in which the maximum number of features to consider when looking for the best split was set to the square root of the number of features. We used this method from the scikit-

learn library (available at http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html).

- L2-loss Linear Support Vector Machine (denoted by L2LSVM): L2LSVM was introduced by solving the optimization problem including minimizing region bounded by these two hyperplanes (margin) as in SVM plus L2-loss function. We used this method from the package LIBLINEAR [72].
- Decision Tree C4.5: We used this method from the Statistics and Machine Learning Toolbox of Matlab.
- The stacked generalization paradigm in which the three learning algorithms used in the proposed method were used to generate the meta-data of the training set. The unpruned Decision Tree C4.5 learned on the meta-data is used to create the second-level classifier [24]. We called this method the Two States Ensemble System with C4.5 (denoted by TSES).

It is noted that the two benchmark algorithms, i.e. the Decision Template and TSES methods, and the proposed method are all trainable combining methods, and therefore they were constructed with the same learning algorithms in the first-level.

We performed 10-fold Cross Validation and ran the test 10 times to obtain 100 test results for each dataset. To assess the statistical significance of the differences in the classification results produced by different methods, we used Wilcoxon signed rank test [73] to compare the classification results of the proposed approach and each benchmark algorithm. The null hypothesis states that the difference in results produced by the two methods is not statistically significant. The performance scores of two methods are treated as significantly different if the p-value of the test is smaller than a given confidence level. In our experiments, the confidence level was set to 0.05.

4.2. Results and Discussion

4.2.1. The influence of parameters

In the proposed method, we used two parameters, i.e. α and β to control the generation of interval (see (7) and (10)). Figure 4 shows the relationship between classification error rate and values of α and β where $\alpha \in \{0{,}0.1,\ldots,3.9{,}4\}$ and $\beta \in \{0.5, 1, 1.5, 2\}$. Some observations can be made here. First, it can be seen that α could have a significant effect on the classification error rate and its optimal value is somewhat data dependent. For some datasets like Conn Bench Vowel and Glass, the classification error rate reduces sharply and then remain unchanged with the increase of α. For datasets such as Haberman and Musk2, the classification error rate reduces sharply to a minimum before slightly increases. For Iris dataset, the classification error rate is minimum at $\alpha = 0$. Besides, it can be observed that β only have a very slight effect on the classification error rate since the line graphs associated with 4 values of β are nearly the same on the experimental datasets.

In the next experiment, the value of α and β are obtained via a 10-fold cross validation procedure conducting on the meta-data (see Figure 5). We loop through all given values of α and β i.e. $\{0{,}0.1,\ldots,3.9{,}4\}$ and $\{0.5, 1, 1.5, 2\}$, respectively and choose the pairs which minimize the classification error rate on the meta-data of training set.

4.2.2. Comparison with benchmark algorithms

The mean and variance of the classification error rates of the three learning algorithms, the benchmark algorithms, and the proposed method are shown in Table 4, 5 and 6. First, compared to the learning algorithms, the proposed method obtains better results on 16 datasets among 26 datasets. Since we do not know which algorithms are suitable for a given dataset, ensemble method can be a viable solution which generally performs better than using a single classifier. As discussed in the Introduction section, by averaging the results of the base classifiers, we can reduce the risk of choosing a wrong classifier, as well as getting a better approximation for the relationship between the feature vectors and their class labels.

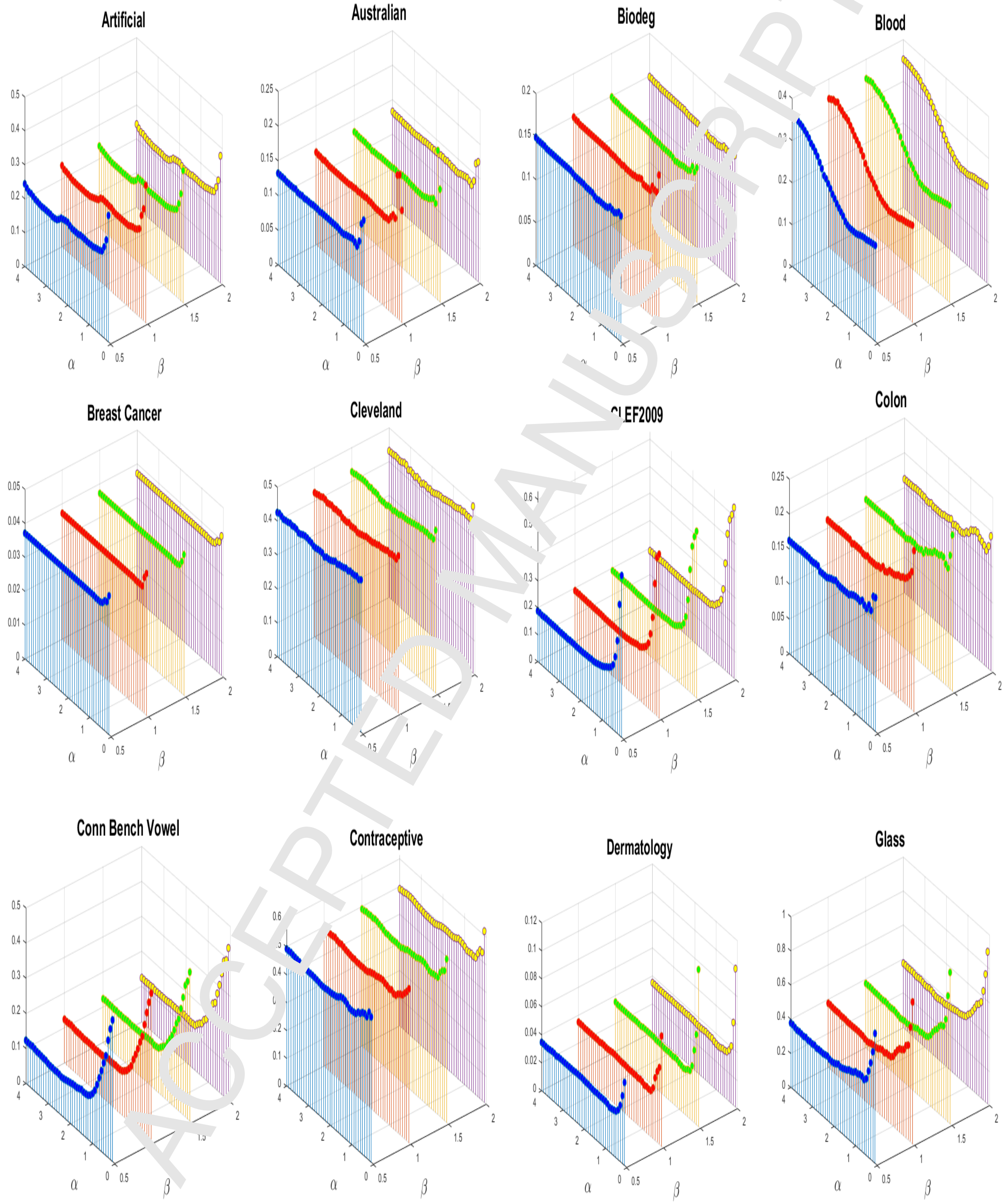
The statistical test result displayed in Figure 6 shows that the proposed method is better than the two trainable combining algorithms. Comparing the proposed method to Decision Template method, we rejected 11 null hypotheses that the two methods perform equally. In all these cases, the classification error rates of the proposed method are smaller than that of Decision Template method. On datasets like Satimage and Texture, the proposed method is significantly better than Decision

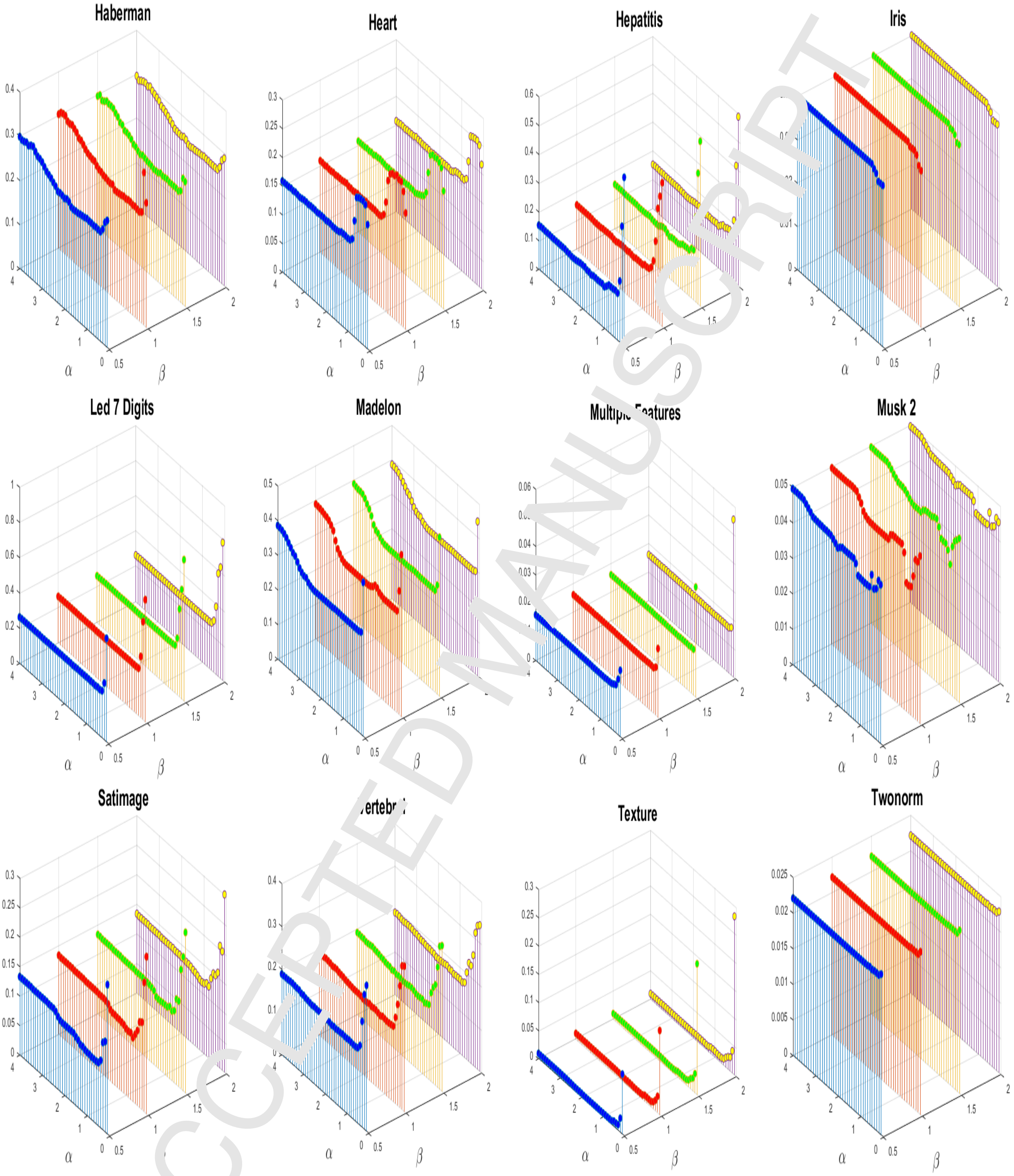
Template method (0.1297 vs. 0.2965 on Satimage, and 0.009 vs. 0.0968 on Texture). Comparing with TSES, we rejected 24 null hypothesis, in which the proposed method is better on 20 datasets and worse on 4 datasets.

The proposed method also outperformed Decision Tree C4.5, L2LSVM, Random Forest, and AdaBoost. Specifically, the proposed method is better than AdaBoost (22 wins and 2 losses), Decision Tree C4.5 (20 wins and 3 losses), Random Forest (16 wins and 8 losses), and L2LSVM (16 wins and 3 losses). The statistical test results clearly demonstrate the advantage of our algorithm.

Table 7 shows the average ranking of the proposed method and the benchmark algorithms. The average ranking was computed based on averaging the rankings of benchmark algorithms and the proposed method on all experimental datasets. These rankings were specified based on the classification error rate: the lower the classification error rate of the method, the higher its ranking. It can be seen that the proposed method clearly ranked first, followed by Decision Template method.

In Table 8, we show the granular prototypes associated with the class labels of several datasets. For datasets like Iris and Twonorm, the intervals of prototype $\mathbf{V}_m$ associated with the m^{th} class predicted by each base classifier are usually very tight, and the intervals of different classes are well separated. Therefore, the discriminative decision model is highly unambiguous, resulting in significantly smaller classification error rate. In contrast, on datasets like Contraceptive and Glass, the intervals are highly overlapped, and causes high ambiguity in the discriminative model which lead to higher classification error rate.


Artificial
Australian
Biodeg
Blood
Breast Cancer
Cleveland
CLEF2009
Colon
Conn Bench Vowel
Contraceptive
Dermatology
Glass
α
β
0
0.5
1
1.5
2
3
4


Haberman
Heart
Hepatitis
Iris
Led 7 Digits
Madelon
Multiple Features
Musk 2
Satimage
Vertebral
Texture
Twonorm
α
β
0
1
2
3
4
0.5
1
1.5
2

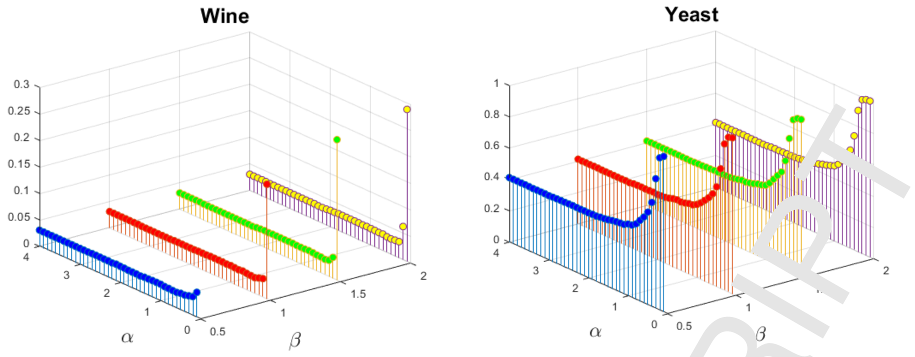


Fig.4. Effect of parameters on the classification error rate

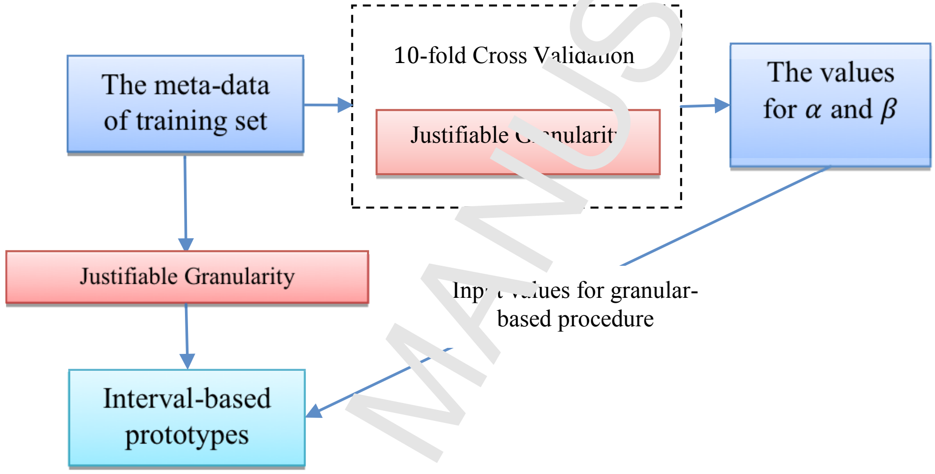

The meta-data of training set
10-fold Cross Validation
Justifiable Granularity
The values for α and β
Justifiable Granularity
Input values for granular-based procedure
Interval-based prototypes

Fig.5 Procedures to find $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

Comparing to the proposed method, some benchmark algorithms like AdaBoost and L2LSVM use different strategies to learn the classifiers. Some of these classifiers can provide a good approximation for the unknown relationship between inputs and labels, resulting in better performance. This is the reason why on some datasets, our method is better than AdaBoost, L2LSVM, Random Forest, and Decision Tree C4.5, and vise verse. Here we further discuss the advantages of our methods in comparison to Decision Template and TSES method. Since they are all heterogeneous ensembles with different combiners, the combining strategy can be used to explain why the proposed method is better on some datasets.

In heterogeneous ensemble, each learning algorithm uses different methodology to learn a base classifier, thereby introducing uncertainty to the meta-data. A combiner which can explicitly represent knowledge with uncertainty is therefore desirable. Some traditional learning algorithms like

Decision Tree C4.5 and Naïve Bayes do not consider the uncertainty when they are used as combiner on the meta-data, as a result, they are less likely to obtain good predictions. Meanwhile, Decision Template method and the proposed method represent the uncertainty in different ways: point estimations and intervals-based prototypes, respectively. It explains why Decision Template method and the proposed method obtain better results than TSES with C4.5 on many datasets.

Decision Template models the meta-data associated with each class label by a vector of point estimations. It is noted that in many scenarios, pointwise statistics such as mean and median are less informative for subsequent reasoning [12]. Figure 7 shows an example of granular prototype and Decision Template associated with each class label for the Vertebral dataset. Clearly, the granular prototype with interval values offers greater flexibility than Decision Template with point values. The proposed method provides a more general and flexible way to describe the meta-data of training observations than Decision Template method, resulting in better classification results on many datasets.

TABLE.4.CLASSIFICATION ERROR RATES OF THE 3 LEARNING ALGORITHMS AND THE PROPOSED METHOD

	LDA		Naïve Bayes		kNN$_5$		Proposed Method	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Artificial	0.3121	1.17E-03	0.3121	1.15E-03	0.2413	2.31E-03	**0.2394**	3.07E-03
Australian	0.1453	1.42E-03	0.1387	1.39E-03	0.3439	2.99E-03	**0.1314**	1.90E-03
Biodeg	0.1465	8.08E-04	0.2068	1.42E-03	0.1828	1.38E-03	**0.1451**	1.19E-03
Blood	**0.2281**	3.05E-04	0.2453	1.11E-03	0.2341	1.56E-03	0.2511	2.92E-03
Breast-cancer	0.0414	4.09E-04	0.0412	5.71E-04	0.0321	4.37E-04	**0.0311**	4.50E-04
CLEF2009	**0.1714**	1.42E-03	0.3684	1.79E-03	0.3583	3.09E-03	0.1861	1.44E-03
Cleveland	0.4228	4.21E-03	0.4328	4.31E-03	0.5521	3.64E-03	**0.4226**	2.80E-03
Colon	0.1845	1.96E-02	0.3717	3.98E-02	0.1740	1.60E-02	**0.1601**	2.07E-02
Conn-bench-vowel	0.3856	3.80E-03	0.4699	4.91E-03	**0.0701**	1.36E-03	0.1179	1.99E-03
Contraceptive	0.4129	1.46E-03	0.5247	1.95E-03	0.4840	1.17E-03	**0.4785**	1.39E-03
Dermatology	**0.0285**	7.05E-04	0.0397	9.84E-04	0.1138	2.63E-03	0.0321	6.30E-04
Glass	0.3574	7.68E-03	0.4019	7.10E-03	**0.3335**	8.59E-03	0.3612	8.90E-03
Haberman	0.2630	2.48E-03	0.2589	2.51E-03	0.2884	3.51E-03	**0.2561**	3.27E-03
Heart	0.1637	4.26E-03	0.1615	4.68E-03	0.3193	6.36E-03	**0.1571**	3.52E-03
Hepatitis	0.1688	1.48E-02	0.1563	1.22E-02	0.1938	6.68E-03	**0.1526**	1.20E-02
Iris	**0.0193**	1.00E-03	0.0400	2.31E-03	0.0393	1.79E-03	0.0400	2.30E-03
Led7digit	0.2778	3.45E-03	0.2706	3.28E-03	0.2970	4.59E-03	**0.2640**	3.92E-03
Madelon	0.4592	1.08E-03	0.4119	1.18E-03	0.2936	9.81E-04	**0.2930**	8.11E-04
Multiple features	0.0199	8.33E-05	0.0389	1.79E-04	0.0511	2.39E-04	**0.0140**	5.20E-05
Musk2	0.0566	6.39E-05	0.2687	2.16E-04	**0.0345**	4.70E-05	0.0417	4.60E-05
Satimage	0.1598	1.28E-04	0.2126	1.76E-04	**0.0910**	1.15E-04	0.1297	1.60E-04
Texture	**0.0053**	7.93E-06	0.2470	2.68E-04	0.0133	2.52E-05	0.0090	1.00E-05
Twonorm	0.0223	2.96E-05	0.0239	3.15E-05	0.0317	3.84E-05	**0.0221**	2.00E-05
Vertebral	0.1965	3.69E-03	0.2565	4.59E-03	0.1845	2.48E-03	**0.1747**	2.62E-03
Wine	**0.0095**	4.45E-04	0.0463	1.98E-03	0.2971	8.24E-03	0.0297	1.23E-03
Yeast	0.4215	1.50E-03	0.4259	1.49E-03	0.4366	1.15E-03	**0.4170**	1.54E-03

**The best results are highlight in bold*

TABLE.5.CLASSIFICATION ERROR RATES OF THE 2 HETEROGENEOUS ENSEMBLE METHODS AND THE PROPOSED METHOD (USING 3 LEARNING ALGORITHMS)

File name	Decision Template		TSES		Proposed Method	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Artificial	0.2433	1.60E-03	0.2789▲	2.74E-03	0.2394	3.07E-03
Australian	0.1346	1.50E-03	0.1771▲	2.41E-03	0.1314	1.90E-03
Biodeg	0.1493▲	9.76E-04	0.1880▲	1.22E-03	0.1451	1.19E-03
Blood	0.272▲	3.06E-03	0.3023▲	2.57E-03	0.2511	2.92E-03
Breast Cancer	0.0374▲	4.15E-04	0.0404▲	5.12E-04	0.0311	4.50E-04
CLEF2009	0.1902	1.51E-03	0.2192▲	1.37E-03	0.1861	1.44E-03
Cleveland	0.4369▲	3.45E-03	0.4763▲	6.04E-03	0.4226	2.80E-03
Colon	0.1598	1.93E-02	0.2319▲	2.17E-02	0.1601	2.07E-02
Conn Bench Vowel	0.1158	2.00E-03	0.0829▼	1.54E-03	0.1179	1.99E-03
Contraceptive	0.4781	1.40E-03	0.5379▲	1.80E-03	0.4785	1.39E-03
Dermatology	0.033	8.86E-04	0.0374▲	7.36E-04	0.0321	6.30E-04
Glass	0.3785▲	1.11E-02	0.3910▲	1.00E-02	0.3612	8.90E-03
Haberman	0.2779▲	5.00E-03	0.3350▲	6.92E-03	0.2561	3.27E-03
Heart	0.1541	4.00E-03	0.2159▲	3.85E-03	0.1571	3.52E-03
Hepatitis	0.1663	1.60E-02	0.2050▲	1.39E-02	0.1526	1.20E-02
Iris	0.040	2.50E-03	0.0313	1.73E-03	0.0400	2.30E-03
Led7digit	0.266	4.18E-03	0.2972▲	4.10E-03	0.2640	3.92E-03
Madelon	0.2941	8.17E-04	0.3697▲	1.14E-03	0.2930	8.11E-04
Multiple Features	0.0148▲	5.90E-05	0.0132▼	5.93E-05	0.0140	5.20E-05
Musk2	0.0455▲	3.89E-05	0.0427▲	5.03E-05	0.0417	4.60E-05
Satimage	0.2965▲	8.20E-05	0.1066▼	1.30E-04	0.1297	1.60E-04
Texture	0.0968▲	9.38E-06	0.0047▼	9.14E-06	0.0090	1.00E-05
Twonorm	0.0221	2.62E-05	0.0225▲	4.05E-05	0.0221	2.00E-05
Vertebral	0.1890▲	3.77E-03	0.1984▲	4.73E-03	0.1747	2.62E-03
Wine	0.0298	1.24E-03	0.0253	1.22E-03	0.0297	1.23E-03
Yeast	0.4186	1.70E-03	0.4854▲	1.19E-03	0.4170	1.54E-03

TABLE.6.CLASSIFICATION ERROR RATES OF THE OTHER BENCHMARK ALGORITHMS

File name	Random Forest		AdaBoost		Decision Tree C4.5		L2LSVM	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Artificial	0.3016 ▲	1.21E-03	0.2197▼	1.90E-03	0.2433	1.60E-03	0.4551▲	1.35E-03
Australian	0.1299	1.74E-03	0.1425▲	1.53E-03	0.1678▲	2.13E-03	0.307▲	2.12E-03
Biodeg	0.2003 ▲	1.29E-03	0.1522▲	1.22E-03	0.1853▲	1.39E-03	0.1328▼	8.65E-04
Blood	0.2304 ▼	1.54E-03	0.2059▼	1.05E-03	0.2595▲	1.68E-03	0.2214▼	6.64E-04
Breast Cancer	0.0269 ▼	3.75E-04	0.0410▲	4.19E-04	0.0526▲	6.94E-04	0.1358▲	1.95E-03
CLEF2009	0.3610▲	2.18E-03	0.5532▲	2.22E-03	0.3664▲	3.12E-03	0.6318▲	1.77E-03
Cleveland	0.3840 ▼	2.54E-03	0.4208	1.88E-03	0.5055▲	6.30E-03	0.4181	1.60E-03
Colon	0.0462▼	5.26E-03	0.2224▲	2.27E-02	0.2588▲	2.74E-02	0.1614	1.92E-02
Conn Bench Vowel	0.3689 ▲	2.53E-03	0.6297▲	3.21E-03	0.2295▲	3.15E-03	0.5485▲	3.74E-03
Contraceptive	0.4912 ▲	1.36E-03	0.4996▲	8.99E-04	0.4830	1.83E-03	0.4949▲	1.33E-03
Dermatology	0.1953 ▲	4.06E-03	0.0436▲	7.25E-04	0.0516▲	1.23E-03	0.0245▼	6.09E-04
Glass	0.3322 ▼	5.25E-03	0.5215▲	2.71E-03	0.3092▼	1.05E-02	0.4067▲	8.82E-03
Haberman	0.2707▲	6.64E-03	0.2743▲	3.60E-03	0.3048▲	5.27E-03	0.2598	1.39E-03
Heart	0.1396 ▼	3.95E-03	0.1896▲	4.67E-03	0.2381▲	6.70E-03	0.1559	3.79E-03
Hepatitis	0.1163 ▼	1.16E-02	0.1363	1.41E-02	0.1663	1.22E-02	0.1588	1.15E-02
Iris	0.0387	1.88E-03	0.0540▲	2.82E-03	0.0507▲	2.40E-03	0.0440	2.33E-03
Led7digit	0.2946 ▲	3.61E-03	0.3474▲	3.91E-03	0.2906▲	2.75E-03	0.2734▲	4.04E-03
Madelon	0.3582 ▲	1.32E-03	0.4056▲	1.09E-03	0.2462▼	1.04E-03	0.4587▲	8.15E-04
Multiple Features	0.0402 ▲	2.26E-04	0.3575▲	2.00E-03	0.0636▲	3.10E-04	0.0260▲	1.05E-04
Musk2	0.1422 ▲	1.56E-05	0.0511▲	5.33E-05	0.0322▼	4.34E-05	0.0473▲	5.41E-05
Satimage	0.3611 ▲	1.92E-04	0.2035▲	1.61E-04	0.1415▲	2.27E-04	0.2292▲	1.67E-04
Texture	0.1082 ▲	1.52E-04	0.3944▲	2.18E-04	0.0761▲	1.13E-04	0.0112▲	2.60E-05
Twonorm	0.0641 ▲	7.34E-05	0.0310▲	3.76E-05	0.1602▲	2.21E-04	0.0221	2.06E-05
Vertebral	0.2003 ▲	3.00E-03	0.2245▲	1.25E-03	0.1984▲	3.75E-03	0.1865▲	3.84E-03
Wine	0.0182 ▼	8.88E-04	0.0378▲	1.77E-03	0.1010▲	4.60E-03	0.0559▲	2.86E-03
Yeast	0.4333 ▲	1.56E-03	0.5880▲	2.44E-04	0.4642▲	1.86E-03	0.4300▲	1.21E-03

▲ or ▼ indicate that proposed method is better or worse than the benchmark algorithm, respectively.

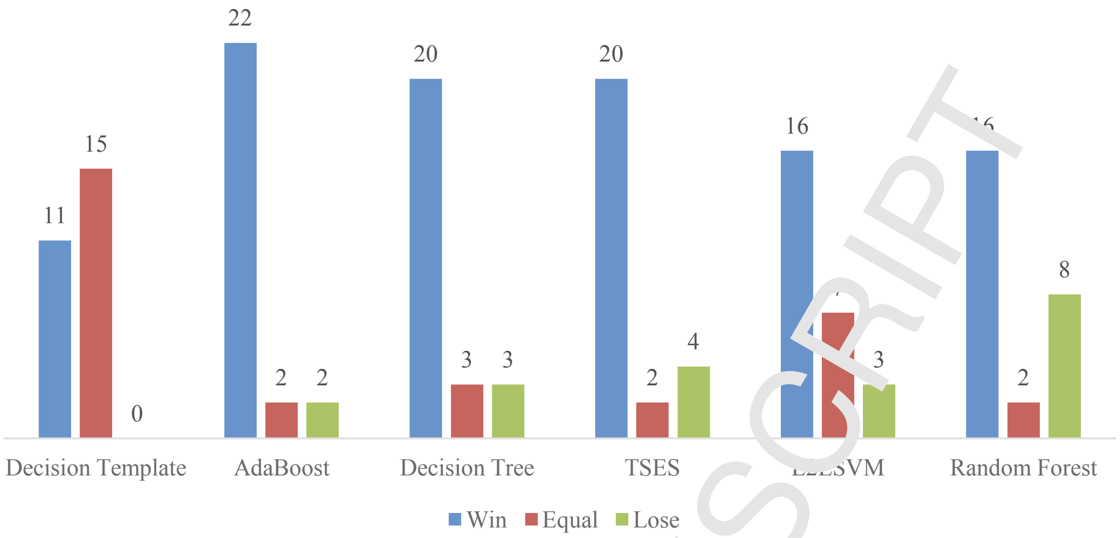


Fig.6. Statistical test results comparing proposed method to the benchmark algorithms (using 3 learning algorithms)

TABLE.7. AVERAGE RANKINGS OF ALL METHODS (USING 3 LEARNING ALGORITHMS)

Algorithm	Ranking
Decision Template	3.37
AdaBoost	5.04
Decision Tree	4.79
TSES	4.44
LLSVM	4.19
Random Forest	3.88
Proposed Method	2.29

TABLE.8. EXAMPLE OF GRANULAR PROTOTYPES FOR SEVERAL DATASETS

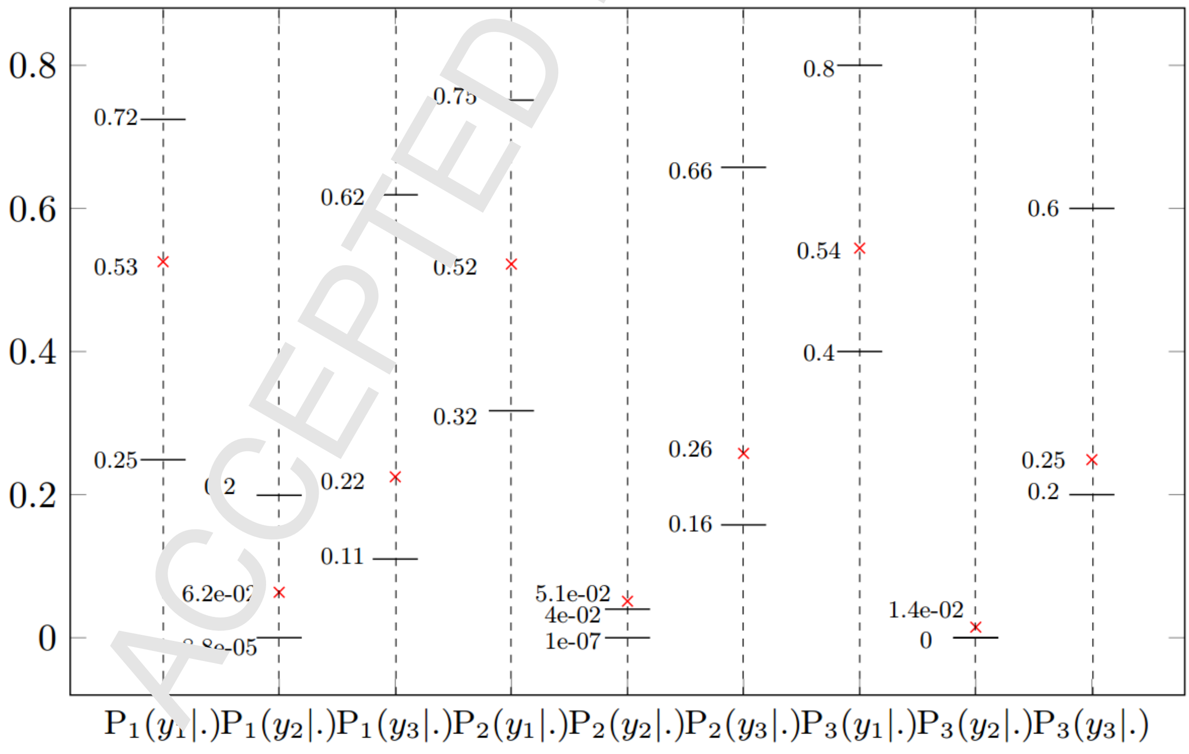
Dataset name	Granular prototypes

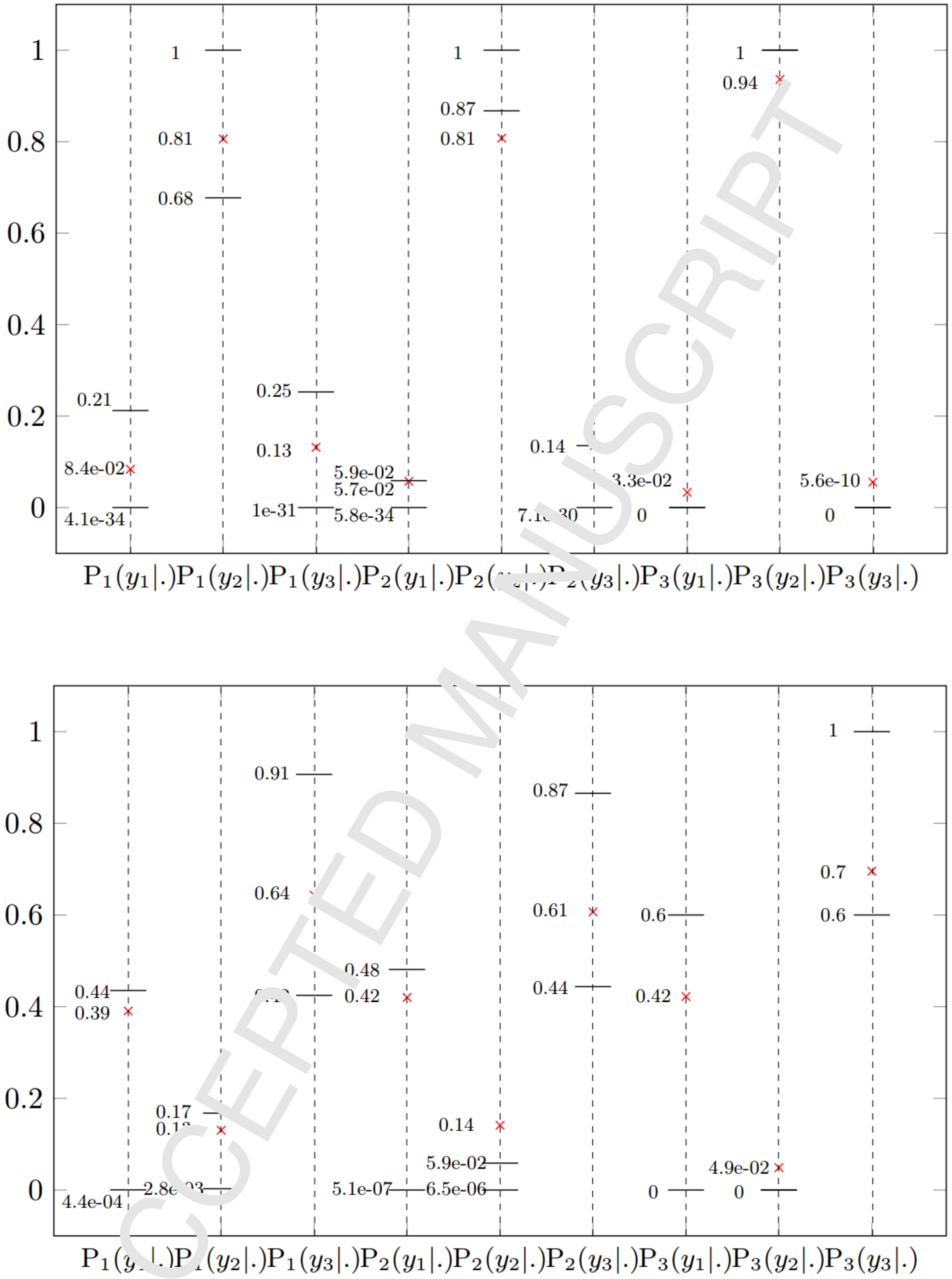
Twonorm	V1={[0.[illegible]560, 1.00000] [0.00000, 0.06440] [0.94109, 1.00000] [0.00000, 0.05891] [1.00000, 1.00000] [0.00000, 0.00000]}
	V2={[0.00000, 0.07[illegible]56] [0.93024, 1.00000] [0.00000, 0.09463] [0.92380, 1.00000] [0.00000, 0.00000] [1.00000, 1.00000]}

Contraceptive	V1={[[illegible]16152, 0.96449] [0.00921, 0.41199] [0.03979, 0.58603] [0.10900, 0.96204] [0.00116, 0.66490] [0.03767, 0.59979] [0.20000, 1.00000] [0.00000, 0.40000] [0.[illegible]00[illegible], 0.40000]}
	V2={[0.[illegible]0282, 0.62354] [0.06022, 0.57610] [0.13421, 0.58900] [0.05647, 0.68147] [0.00232, 0.86498] [0.09485, 0.53928] [0.20000, 0.60000] [0.00000, 0.80000] [0.00000, 0.60000]}
	V3={[0.14256, 0.67763] [0.02580, 0.49724] [0.16817, 0.63002] [0.07716, 0.75685] [0.00103, 0.69072] [0.14702, 0.63025] [0.20000, 0.60000] [0.00000, 0.40000] [0.20000, 0.80000]}

Iris	V1= {[1.00000, 1.00000] [0.00000, 0.00000] [0.00000, 0.00000] [1.00000, 1.00000] [0.00000, 0.00000] [0.00000, 0.00000] [1.00000, 1.00000] [0.00000, 0.00000] [0.00000, 0.00000]}
	V2={[0.00000, 0.00000] [0.91273, 1.00000] [0.00000, 0.08727] [0.00000, 0.00001] [0.94101, 1.00000] [0.00000, 0.05899] [0.00000, 0.00000] [1.00000, 1.00000] [0.00000, 0.00000]}
	V3={[0.00000, 0.00000] [0.00000, 0.19505] [0.80495, 1.00000] [0.00000, 0.00000] [0.00000, 0.06187] [0.93813, 1.00000] [0.00000, [illegible]00000] [0.00000, 0.00000] [1.00000, 1.00000]}

Glass	V1={[0.21048, 0.83217] [0.07672, 0.69066] [0.00343, 0.17914] [0.00000, 0.00061] [0.00000, 0.00544] [0.00000, 0.00[illegible]] [0.[illegible]2167, 0.71802] [0.09053, 0.74780] [0.02996, 0.28073] [0.00000, 0.00036] [0.00000, 0.02064] [0.00000, 0.00012] [0.40000, 1.00000] [0.00000, 0.20000] [0.00000, 0.2[illegible]000] [0.00000, 0.00000] [0.00000, 0.00000] [0.00000, 0.00000]}
	V2={[0.00000, 0.68058] [0.31214, 0.73528] [0.00000, 0.19830] [0.00000, 0.05873] [0.00000, 0.02251] [0.0[illegible]00, 0.[illegible]70[illegible]] [0.00000, 0.61310] [0.28941, 0.74108] [0.00000, 0.16535] [0.00000, 0.02865] [0.00000, 0.01283] [0.00000, 0.00134] [0.00000, 0.40000] [0.40000, 1.00000] [0.00000, 0.00000] [0.00000, 0.00000] [0.00000, 0.00000] [0.00000, 0.00000]}
	V3={[0.20043, 0.83912] [0.06652, 0.71189] [0.00033, 0.33895] [0.00000, 0.00023] [0.00000, 0.0002[illegible]] [0.00000, [illegible]00000] [0.34264, 0.77711] [0.09489, 0.59960] [0.03535, 0.21730] [0.00000, 0.00025] [0.00001, 0.00161] [0.00000, 0.00001] [0.40000, 1.00000] [0.000[illegible], 0.20000] [0.00000, 0.60000] [0.00000, 0.00000] [0.00000, 0.00000] [0.00000, 0.00000]}
	V4={[0.00000, 0.10771] [0.00000, 0.87700] [0.00000, 0.01110] [0.00000, 1.00000] [0.00000, 0.29[illegible]4] [0.[illegible]000, 0.10227] [0.00000, 0.11322] [0.00000, 0.91031] [0.00000, 0.01789] [0.00000, 1.00000] [0.00000, 0.20848] [0.00000, 0.02114] [0.00000, 0.[illegible]000] [0.20000, [illegible]40000] [0.00000, 0.00000] [0.20000, 0.80000] [0.00000, 0.00000] [0.00000, 0.00000]}
	V5={[0.00000, 0.47893] [0.00000, 0.65392] [0.00000, 0.03884] [0.00000, 0.00447] [[illegible]035, 0.98[illegible]6] [0.00000, 0.21954] [0.00001, 0.40464] [0.00000, 0.60271] [0.00003, 0.08869] [0.00000, 0.22933] [0.02167, 0.99452] [0.00002, 0.15650] [0.00[illegible], 0.000[illegible]] [0.20000, 0.40000][0.00000, 0.00000] [0.00000, 0.00000] [0.20000, 0.60000] [0.00000, 0.00000]}
	V6={[0.00000, 0.05812] [0.00000, 0.05386] [0.00000, 0.00986] [0.00000, 0.00[illegible]3] [0.00000, 0.00485] [0.99489, 1.00000] [0.00000, 0.11986] [0.00000, 0.01646] [0.00000, 0.09883] [0.00000, 0.03480] [0.00000, 0.09730] [0.97793, 1.00000] [0.00000, [illegible]00000] [0.00000, 0.00000] [0.00000, 0.00000] [0.00000, 0.00000] [0.00000, 0.00000] [1.00000, 1.00000]}





Note: × marks the Decision Template. Top, middle, and bottom figures are associated with class 1, 2, and 3, respectively.

Fig.7. Decision Templates and Granular Prototypes for the Vertebral Dataset

4.2.3. Different number of learning algorithms

To see the effect of using different number of learning algorithms on the ensemble, we built heterogeneous ensemble system with 10 learning algorithms. The 7 new learning algorithms: two kNN classifiers (with the number of nearest neighbors was set to 25 and 50, denoted by $k\text{NN}_{25}$, and $k\text{NN}_{50}$, respectively), Decision Tree C4.5, Decision Stump, Fisher Classifier [74], Nearest Mean Classifier (denoted by NMC), and Logistic Linear Classifier (denoted by LLC) [75], were added to the previous ensemble system to form the new one. Once again, the learning algorithms were selected as different as possible to promote system diversity. The kNN classifier and Decision Tree C4.5 were obtained from the Statistics and Machine Learning Toolbox of Matlab while the other new learning algorithms was obtained from PRTools (available at http://prtools.org/). It is noted that the classification error rates of AdaBoost, Random Forest, Decision Tree C4.5, and L2LSVM would not change in this experiment so that we only reported the new experimental results of three heterogeneous ensemble methods with these 10 learning algorithms in Table 9.

Table A1 in the Appendix shows the classification error rates of these 10 learning algorithms and the proposed method. Once again, the benefit of using the ensemble is demonstrated since the proposed method obtains the best result on 12 datasets. Based on the statistical test results in Figure 8, it can be seen that proposed method continues to outperform AdaBoost (in 23 cases where the null hypothesis is rejected, the proposed method wins in 21 cases and loses in 2 cases), Decision Tree (in 22 cases where the null hypothesis is rejected, the proposed method wins in 21 cases and loses in 1 case), L2LSVM (in 18 cases where the null hypothesis is rejected, the proposed method wins in 16 cases and loses in 2 cases), TCES (in 23 cases where the null hypothesis is rejected, the proposed method wins in 19 cases and loses in 4 cases), Random Forest (in 23 cases where the null hypothesis is rejected, the proposed method wins in 16 cases and loses in 7 cases) and Decision Template method (in 13 cases where the null hypothesis is rejected, the proposed method wins in 11 cases and loses in 2 cases). The average ranking of the proposed method once again is better than all benchmark algorithms (Table 10).

We note the significant differences in the classification error rate of the proposed method construct by 3 or 10 learning algorithms. First, using 10 learning algorithms obtains better results than using 3 learning algorithms, for example, on Contraceptive (0.4572 vs. 0.4785), Glass (0.3196 vs. 0.3612), Madelon (0.2452 vs. 0.2930), and Vertebral (0.1510 vs. 0.1747). On Conn Bench Vowel, in contrast, the classification error rate reduces 4% when using 3 learning algorithms comparing to using 10 learning algorithms (0.1179 vs. 0.1571). This also happens with other heterogeneous ensemble methods like Decision Template and TSES method. Although the proposed method is better than the benchmark algorithms in both cases, the dependence of choosing the learning algorithms to the ensemble performance is the limitation of all the heterogeneous ensemble methods.

TABLE.9.CLASSIFICATION ERROR RATES OF THE 2 HETEROGENEOUS ENSEMBLE METHODS AND THE PROPOSED METHOD (USING 10 LEARNING ALGORITHMS)

	Decision Template		TSES		Proposed Method	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Artificial	0.2233▲	1.53E-03	0.2509▲	2.39E-03	0.2142	1.73E-03
Australian	0.1274	1.50E-03	0.1832▲	2.01E-03	0.1262	1.25E-03
Biodeg	0.1363	9.89E-04	[illegible]▲	1.12E-03	0.1374	1.10E-03
Blood	0.2754▲	2.51E-03	0.2987▲	2.49E-03	0.2438	1.86E-03
Breast Cancer	0.0362	5.04E-04	0.0455▲	6.31E-04	0.0359	5.01E-04
CLEF2009	0.1666	1.42E-03	0.2245▲	1.88E-03	0.1659	1.45E-03
Cleveland	0.4326	4.94E-03	0.4719▲	5.74E-03	0.4357	2.03E-03
Colon	0.1698	1.79E-02	0.2431▲	1.77E-02	0.1633	2.02E-02
Conn Bench Vowel	0.1750▲	1.91E-03	0.0943▼	2.06E-03	0.1571	2.37E-03
Contraceptive	0.4560	1.69E-03	0.5202▲	1.56E-03	0.4572	1.60E-03
Dermatology	0.0252▲	5.06E-04	0.0352▲	1.11E-03	0.0242	6.10E-04
Glass	0.3198	8.92E-03	0.3630▲	9.58E-03	0.3196	7.10E-03
Haberman	0.2690▲	3.36E-03	0.3373▲	6.82E-03	0.2437	3.75E-03
Heart	0.1559	5.39E-03	0.2204▲	7.50E-03	0.1561	4.75E-03
Hepatitis	0.1725	1.59E-02	0.1975▲	2.07E-02	0.1520	1.38E-02
Iris	0.0440▲	[illegible]	0.0340	2.18E-03	0.0410	2.35E-03
Led7digit*	-	-	-	-	-	-
Madelon	[illegible]▲	9.69E-04	0.2770▲	9.59E-04	0.2452	9.92E-04
Multiple Features	0.0125▲	6.87E-05	0.0144▲	7.38E-05	0.0120	6.35E-05
Musk2	0.0334▼	4.65E-05	0.0276▼	4.57E-05	0.0387	4.31E-05
Satimage	[illegible]▲	6.52E-05	0.1089▼	1.48E-04	0.1222	1.21E-04
Texture	[illegible]▲	1.45E-05	0.0049▼	6.81E-06	0.0096	1.51E-05
Twonorm	0.0219▼	2.29E-05	0.0331▲	3.24E-05	0.0222	2.77E-05
Vertebral	[illegible]	3.39E-03	0.1942▲	3.52E-03	0.1510	3.75E-03
Wine	0.0303▲	2.10E-03	0.0225	1.01E-03	0.0261	1.79E-03
Yeast	0.4056	1.44E-03	0.4944▲	1.62E-03	0.4032	1.38E-03

▲ *or* ▼ *indicate that proposed method is better or worse than the benchmark algorithm, respectively.*

* *several of the learning algorithms cannot be run on this dataset, hence final ensemble outputs are not available*

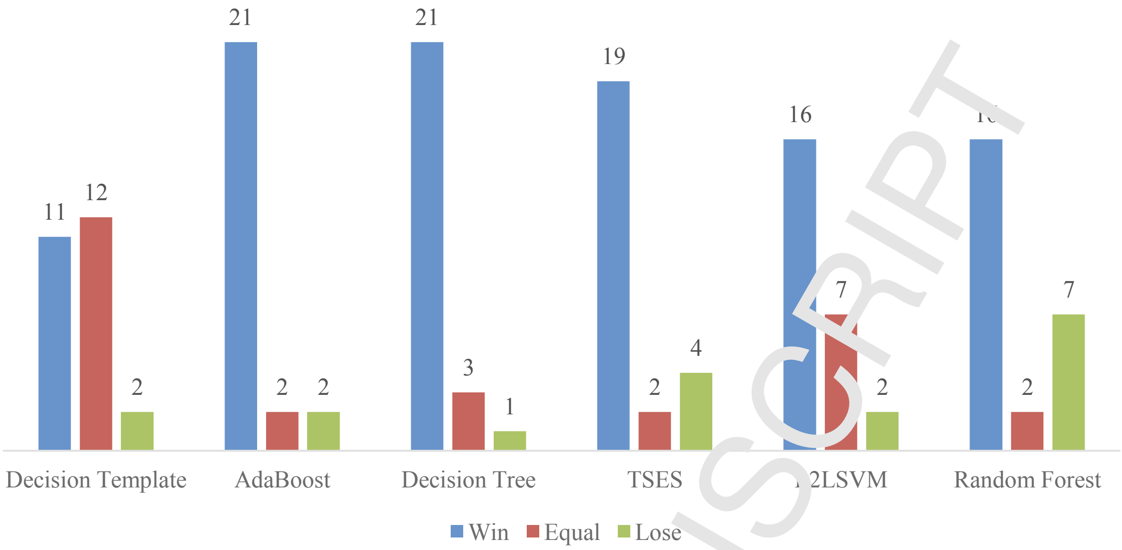


Fig.8. Statistical test results comparing proposed method to the benchmark algorithms (using 10 learning algorithms)

TABLE.10. AVERAGE RANKINGS OF ALL METHODS (USING 10 LEARNING ALGORITHMS)

Algorithm	Ranking
Decision Template	3.18
AdaBoost	5.00
Decision Tree	4.96
TSES	4.32
L2LSVM	4.20
Random Forest	4.08
Proposed Method	2.26

4.2.4. Time complexity analysis

Let $O(\mathcal{K}_k)$ denotes the complexity of the k^{th} learning algorithm $\mathcal{K}_k$, the complexity of the learning process of the proposed method is$O\left(\max\left(T \times \arg\max_{k=1,\dots,K} O(\mathcal{K}_k), (\text{parameters searching}), (\text{combiner})\right)\right)$ in which $O\left(T \times \arg\max_{k=1,\dots,K} O(\mathcal{K}_k)\right)$ is the time complexity of generating meta-data of training set via running T-fold cross validation, $O(\text{parameters searching})$ is the time complexity of finding the parameter α and β from the specific values via 10-fold cross validation (see Figure 5), and

$\mathcal{O}(\text{combiner})$ is the time complexity of combiner working on meta-data of training set to produce the decision model. In the proposed method, we used justifiable granularity to construct the interval for each column of the meta-data of each class. The computation of the median of unsorted posterior probability array with $N_m (m = 1, \dots, M)$ training observations belonging to m^{th} class as well as the bounds of interval class memberships based on (7) and (10) can be done by first applying a sorting algorithm to the array. We can apply a sorting algorithm introduced in [76] to an array with N_m elements in which the time complexity is $\mathcal{O}(N_m \times logN_m)$. The procedure runs though all $M \times K$ columns of meta-data of training observations for each $m = 1, \dots, M$ so that the time complexity of the combiner is $\mathcal{O}\big(M \times K \times \arg\max_{m=1,\dots,M} \mathcal{O}(N_m \times logN_m)\big)$. In the parameter searching procedure, we loop through all given values of α and β to find the specific value that minimize classification error rate on the training set via 10-fold cross validation, as a result the time complexity of the parameters searching procedure is $\mathcal{O}\big(10 \times |\alpha| \times |\beta| \times M \times K \times \arg\max_{m=1,\dots,M} \mathcal{O}(N_m^* \times logN_m^*)\big)$ where $N_m^* < N_m$ is the number of training observations belonging to the m^{th} class in the parts obtained via the 10-fold cross validation procedure. Therefore, the time complexity of the training process of the proposed method is $\mathcal{O}\big(\max\big(T \times \arg\max_{k=1,\dots,K} \mathcal{O}(\mathcal{K}_k)\,,\ M \times K \times \arg\max_{m=1,\dots,M} \mathcal{O}(N_m \times logN_m)\,, 10 \times |\alpha| \times |\beta| \times M \times K \times \arg\max_{m=1,\dots,M} \mathcal{O}(N_m^* \times logN_m^*)\big)\big)$.

For TSES the time complexity of the training process is $\mathcal{O}\left(\max\left(T \times \arg\max_{k=1,\dots,K} \mathcal{O}(\mathcal{K}_k)\,, \mathcal{O}(combiner)\right)\right)$ in which $\mathcal{O}(combiner)$ is the time complexity of the learning algorithm for the combiner. Depending on the learning algorithm for the combiner, TSES could have a longer or shorter training time than the proposed method. In this paper, we used Decision Tree C4.5 (its time complexity is $\mathcal{O}(D \times N)$ via the improvement in [77]) to learn on the meta-data of training observation so that the overall training complexity of TSES method is $\mathcal{O}\left(\max\left(\arg\max_{k=1,\dots,K} \mathcal{O}(\mathcal{K}_k) \times T,\ (D \times N)\right)\right)$. Meanwhile in the combining method of Decision Template, the loop runs through all training observations to compute the average of the meta-data associated with each class label [5] so its time complexity is $\mathcal{O}\big(\max\big(\arg\max_{k=1,\dots,K} \mathcal{O}(\mathcal{K}_k) \times T, N\,\big)\big)$. It is noted that the proposed method can be implemented via parallel mechanism by using T

processors to learn the meta-data, $10 \times |\alpha| \times |\beta| \times M \times K$ processors to search the parameters, and $M \times K$ processors to learn the intervals. The time complexity of the proposed method then becomes:

$$\mathcal{O}\big(\max\big(\arg\max_{k=1,\dots,K} \mathcal{O}(\mathcal{K}_k), \arg\max_{m=1,\dots,M} \mathcal{O}(N_m \times logN_m), \arg\max_{m=1,\dots,M} \mathcal{O}(N_m^* \times logN_m^*)\big)\big) = \mathcal{O}\big(\max\big(\arg\max_{k=1,\dots,K} \mathcal{O}(\mathcal{K}_k), \arg\max_{m=1,\dots,M} \mathcal{O}(N_m \times logN_m)\big)\big), \text{ since } N_m^* < N_m.$$

Table 11 shows the average training and classification time (in seconds) for Decision Template, TSES, and the proposed method, computed on 100 training sets and associated test sets partitioned from each dataset. Although the proposed method generally has longer training time and classification time than Decision Template and TSES methods, the differences are within practical limit.

TABLE.11. TRAINING AND CLASSIFICATION TIME (IN SECONDS) OF DECISION TEMPLATE, TSES, AND PROPOSED METHOD (USING 3 LEARNING ALGORITHMS)

	Decision Template		TSES		Proposed Method	
	Training Time	*Classification Time*	*Training Time*	*Classification Time*	*Training Time*	*Classification Time*
Artificial	0.5414	0.0099	0.5657	0.0744	17.5385	0.4721
Australian	0.5374	0.0101	0.5467	0.0702	29.6069	0.7538
Biodeg	0.7161	0.0170	0.687	0.1141	19.5413	0.7732
Blood	0.5192	0.0083	0.5453	0.0762	13.5489	0.4850
Breast Cancer	0.5915	0.0102	0.5521	0.0693	40.0564	0.8671
CLEF2009	0.7451	0.0099	0.7157	0.0253	8.8935	0.3697
Cleveland	1.1949	0.0128	0.8905	0.0359	52.0346	0.2412
Colon	1.1938	0.0314	1.1808	0.0329	3.1703	0.1781
Conn Bench Vowel	3.1417	0.0216	2.4259	0.0581	95.9051	1.0321
Contraceptive	0.6583	0.0149	0.7007	0.0567	13.2749	0.6792
Dermatology	1.0965	0.0121	1.0225	0.0449	44.6629	0.2011
Glass	0.8978	0.0077	0.9161	0.0249	47.0464	0.1483
Haberman	0.6945	0.0391	0.5203	0.0355	12.6771	0.1212
Heart	0.4976	0.0122	0.4853	0.0316	10.7566	0.0990
Hepatitis	0.5869	0.0080	0.5873	0.0051	8.9105	0.0578
Iris	1.0778	0.0164	0.5311	0.0184	36.4504	0.1599
Led7digit	1.0465	0.0384	1.9646	0.0534	112.4801	0.5317
Madelon	15.6398	0.3284	11.0456	0.4682	227.3102	6.2560
Multiple Features	26.9106	0.4417	30.296	0.7383	1243.4163	13.7829
Musk2	12.1663	0.8991	13.4885	5.0747	152.4641	8.3123
Satimage	3.0802	0.2286	4.8068	3.0047	225.2858	5.4413
Texture	4.0408	0.2359	5.8136	2.2746	212.2031	4.6231
Twonorm	1.9637	0.1587	5.7002	3.0170	100.7969	4.9567
Vertebral	0.5744	0.0078	0.5865	0.0345	12.8961	0.1898
Wine	0.6189	0.0079	0.5316	0.0211	13.1322	0.1874
Yeast	1.9967	0.0298	2.6987	0.1056	80.2426	1.5750
Average	3.1626	0.1007	3.4540	0.5987	109.0116	2.0190

5. Conclusions

In this paper, we have introduced a novel trainable ensemble classifiers system based on the concept of justifiable granularity. In our approach, we construct the granular prototype for each class from the meta-data of training observations with the same class label. Each granular prototype is a vector of intervals, where the intervals reflect the uncertainty in class prediction generated by the base classifiers. The class label of an unlabeled observation is predicted by picking up the class label associated with the granular prototype that is the closest to the meta-data of the unlabeled observation. Extensive experiments were carried out by using an ensemble system of three and ten base classifiers, and performance comparisons were conducted with six benchmark algorithms including AdaBoost, Random Forest, Decision Template, TSES, Decision Tree C4.5, and L2LSVM on 26 datasets. Statistical test results indicated that our method significantly outperforms all the benchmark algorithms.

Some future work can be conducted to further improve the performance of the proposed method. First, to deal with the trade-off between the specificity and the experimental evidence (cardinality), we used the product of these two requirements and maximizing the expression with respect to the bounds of the interval. This simple choice may not provide the best solution in all situations and techniques such as multi-objective optimization can be investigated. Moreover, as mentioned before, the general performance of the proposed method depends on the choice of the learning algorithms to construct the ensemble. A poor selection of learning algorithms may result in the poor performance of the ensemble. The proposed method could be combined with learning algorithm selection [25] to acquire the optimal set of learning algorithms for each specific dataset.

References

[1] Z.-H. Zhou, Ensemble methods: foundations and algorithms, CRC Press, 2012.

[2] T. Dietterich, Ensemble Methods in Machine Learning, in Proceeding of International Workshop on Multiple Classifier Systems, 2000, pp 1-15.

[3] T.T. Nguyen, T.T.T. Nguyen, X.C. Pham, A.W.C. Liew, A Novel Combining Classifier Method based on Variational Inference, Pattern Recognition.49 (2016), 198-212.

[4] T.T. Nguyen, M.P. Nguyen, X.C. Pham, A.W.C. Liew, Heterogeneous classifier ensemble with fuzzy rule-based meta learner, Information Sciences. 422, 2018, pp. 144-160.

[5] L.I. Kuncheva, J.C. Bezdek, R.P.W. Duin, Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recognition. 34 (2001), 299-314.

[6] F.P.A. Coolen, M.C.M. Troffaes, T. Augustin, Imprecise Probability, in: M. Lovric (Eds.), International Encyclopedia of Statistical Science, Springer 2011, pp. 645-648.

[7] P. Walley, Statistical Reasoning with Imprecise Probabilities, Chapman and Hall Press, London, 1991.

[8] L.A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems. 90(2) (1997), 111–117.

[9] L.A. Zadeh, From computing with numbers to computing with words—From manipulation of measurements to manipulation of perceptions, IEEE Trans. on Circuits and Systems. 45(1) (1999), 105–119.

[10] L.A. Zadeh, Toward a generalized theory of uncertainty (GTU) - an outline, Information Sciences. 172 (1,2) (2005), 1–40.

[11] A. Bargiela, W. Pedrycz, Granular Computing: An Introduction, Kluwer Academic Publishers, 2003.

[12] A. Bargiela, W. Pedrycz, Human-Centric Information Processing Through Granular Modelling, Springer Press, 2009.

[13] W. Pedrycz, Knowledge-Based Clustering: From Data to Information Granules, John Wiley Press, 2005.

[14] W. Pedrycz, Granular Computing Analysis and Design of Intelligent Systems, CRC Press 2013.

[15] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of International Conference on Machine Learning (ICML), 1996, pp. 148-156.

[16] L. Breiman, Bagging Predictors, Machine Learning. 24 (1996) 123-140.

[17] L. Breiman, Random Forests, Machine Learning. 45 (2001) 5-32.

[18] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley, 2004.

[19] L. Rokach, Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography, Computational Statistics & Data Analysis. 53 (2009) 4046-4072.

[20] R.P.W. Duin, The combining classifier: to train or not to train?, in: Proceedings of International Conference on Pattern Recognition, 2002, pp. 765-770.

[21] A. Rahman, B. Verma, Novel Layered Clustering-Based Approach for Generating Ensemble of Classifiers, IEEE Transactions on Neural Network. 22 (5) 2011, pp. 781-792.

[22] C.-X. Zhang, R.P.W. Duin, An experimental study of one- and two-level classifier fusion for different sample sizes, Pattern Recogn. Lett. 32 (2011) 1756-1767.

[23] T.T. Nguyen, A.W.C. Liew, M.T. Tran, T.T.T. Nguyen, M.P. Nguyen, Classifier Fusion Based On A Novel 2-Stage Model, in: X. Wang, W. Pedrycz, P. Chan, Q. He (Eds.), Machine Learning and Cybernetics, Springer, 2014, pp. 60-68.

[24] T.T. Nguyen, A.W.C. Liew, X.C. Pham, M.P. Nguyen, A Novel 2-Stage Combining Classifier Model with Stacking and Genetic Algorithm Based Feature Selection, in: D.-S. Huang, K.-H. Jo, L. Wang (Eds.), Intelligent Computing Methodologies, Springer International Publishing, 2014, pp. 33-43.

[25] T.T. Nguyen, A.W.C. Liew, M.T. Tran, X.C. Pham, M.P. Nguyen, A novel genetic algorithm approach for simultaneous feature and classifier selection in multi classifier system, in: IEEE Congress on Evolutionary Computation (CEC), 2014, pp.1698-1705.

[26] T.T. Nguyen, A.W.C. Liew, X.C. Pham, M.P. Nguyen, Optimization of ensemble classifier system based on multiple objectives genetic algorithm, International Conference on Machine Learning and Cybernetics (ICMLC), 2014 (Vol.1), pp. 46 – 51.

[27] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, Rotation Forest: A New Classifier Ensemble Method, IEEE Transactions on Pattern Analysis and Machine Intelligence. 28(10) (2006), 1619-1630.

[28] R. Blaser, P. Fryzlewicz, Random Rotation Ensemble, Journal of Machine Learning Research.2 (2015), 1-15.

[29] O. Wu, Classifier Ensemble by Exploring Supplementary Ordering Information, IEEE Transactions on Knowledge and Data Engineering, 2018, In Press, DOI: 10.1109/TKDE.2018.2818138.

[30] B. Krawczyk, L.L. Minku, J. Gama, J. Stefanowski, M. Woźniake, Ensemble learning for data stream analysis: A survey, Information Fusion. 37(2017), 132-156

[31] X.C. Pham, M.T. Dang, S.V. Dinh, S. Hoang, T.T. Nguyen, A.W.C. Liew, Learning from Data Stream Based on Random Projection and Hoeffding Tree Classifier, in Proceeding of Digital Image Computing: Techniques and Applications (DICTA), 2017.

[32] B. Krawczyk, Alberto Cano, Online ensemble learning with abstaining classifiers for drifting and noisy data streams, Applied Soft Computing. 68(2018), 677-692.

[33] Z. Yu , Y. Zhang, J. You, C.L. P. Chen, H.-S. Wong, G. Han, J. Zhang, Adaptive Semi-Supervised Classifier Ensemble for High Dimensional Data Classification, IEEE Transactions on Cybernetics, 2018, In Press, DOI: 10.1109/TCYB.2017.2761908.

[34] J.M. Moyano, E.L. Gibaja, K.J. Cios, S. Ventura , Review of ensembles of multi-label classifiers: Models, experimental study and prospects, Information Fusion. 44 (2018), 33-45.

[35] Q. Wu, M. Tan, H. Song, J. Chen, M.K. Ng, ML-FOREST: A Multi-Label Tree Ensemble Method for Multi-Label Classification, IEEE Transactions On Knowledge And Data Engineering. 28(10)(2016), 2016.

[36] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On Combining Classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence. 20(3) (1998), 226-239.

[37] X.-Z. Wang, R. Wang, H.-M. Feng, H.-C. Wang, A New Approach to Classifier Fusion Based on Upper Integral, IEEE Transactions On Cybernetics. 44(5) (2014), 620-635.

[38] V.S. Costa, A.D.S. Farias, B. Bedregal, R.H.N. Santiago, A.M. de P.Canuto, Combining multiple algorithms in classifier ensembles using generalized mixture functions, Neurocomputing, 2018, In Press, DOI:10.1016/j.neucom.2018.06.021.

[39] T.K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern 2Analysis and Machine Intelligence. 20(8) (1998), 832-844.

[40] A.S.B. Jr, R. Sabourin, L.E.S. Oliveira, Dynamic selection of classifiers—A comprehensive review, Pattern Recognition. 47 (2014), 3665-3680.

[41] R.M.O.Cruza, R. Sabourin, G.D.C. Cavalcanti, Dynamic classifier selection: Recent advances and perspectives, Information Fusion. 41 (2018), 195-216.

[42] D. H.-Lobato, G.M.-Muoz, A. Suarez, Statistical instance-based pruning in ensembles of independent classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence. 31(2) (2009), 364-369.

[43] L.I. Kuncheva, Switching between selection and fusion in combining classifiers: An experiment, IEEE Transactions on Systems, Man Cybernetics: Part B, Cybernetics. 32(2) (2002), 146-156.

[44] V. Soto, S.G.-Moratilla, G. M.-Muñoz, D. H.-Lobato, A. Suárez, A Double Pruning Scheme for Boosting Ensembles, IEEE Transactions On Cybernetics. 44(12) (2014), 2682 – 2695.

[45] Z. Yu, L. Li, J. Liu, G. Han, Hybrid Adaptive Classifier Ensemble, IEEE Transactions on Cybernetics. 45(2) (2015), 177 - 190.

[46] Q.T. Cai, C.Y. Feng, C.S. Zhang, A weighted subspace approach for improving bagging performance, in Proc. of IEEE ICASSP, USA, 2008, pp. 3341–3344.

[47] Z. Yu , D. Wang, Z. Zhao, C.L.P. Chen, J. You, H.-S. Wong, J. Zhang, Hybrid Incremental Ensemble Learning for Noisy Real-World Data Classification, IEEE Transactions on Cybernetics, 2018, In Press, DOI: 10.1109/TCYB.2017.2774266.

[48] A. Demiriz, K.P. Bennett, J.S. Taylor, Linear Programming Boosting via Column Generation, Machine Learning. 46 (2002), 225-254.

[49] M. Warmuth, J. Liao, G. Ratsch, Totally corrective boosting algorithms that maximize the margin, in Proc. 23rd Int. Conf. on Machine Learning, 2006, pp. 1001–1008.

[50] C. Seiffert, T. Khoshgoftaar, J. Hulse, A. Napolitano, RUSBoost: Improving classification performance when training data is skewed, in Proc. 19rd Int. Conf. on Pattern Recognition, 2008, pp. 1–4.

[51] L.I. Kuncheva, C.J. Whitaker, Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy, Machine Learning. 51 (2003), 181-207.

[52] E. K. Tang, P. N. Suganthan, X. Yao, An analysis of diversity measures, Machine Learning. 65(1) (2006), 247-271.

[53] W. Gao, Z.-H. Zhou, On the doubt about margin explanation of boosting, Artificial Intelligence. 203 (2013), 1–18.

[54] X.-Z. Wang, H.-J. Xing, Y. Li, Q. Hua, C.-R. Dong, W. Pedrycz, A study on relationship between generation abilities and fuzziness of base classifiers in ensemble learning, IEEE Transactions on Fuzzy Systems. 23 (5) (2015), 1638-1654.

[55] L. I. Kuncheva, A Bound on Kappa-Error Diagrams for Analysis of Classifier Ensembles, IEEE Transactions on Knowledge and Data Engineering. 25(3) (2013), 494-501.

[56] L. Li, Q. Hu, X. Wu, D. Yu, Exploration of classification confidence in ensemble learning, 2014.

[57] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, M. Xu, Margin & diversity based ordering ensemble pruning, Neurocomputing 275 (2018), 237–246.

[58] L.I. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Transactions on Pattern Analysis and Machine Intelligence. 24 (2002) 281-286.

[59] D.H. Wolpert, Original Contribution: Stacked generalization, Neural Network. 5(2) (1992) 241-259.

[60] C. Merz, Using Correspondence Analysis to Combine Classifiers, Machine Learning. 36(1) (1999), 33-58.

[61] K.M. Ting, I.H. Witten, Issues in stacked generalization, Journal Of Artificial Intelligence Research. 10 (1999),271-289.

[62] L. Zhang, W.-D. Zhou, Sparse ensembles using weighted combination methods based on linear programming, Pattern Recognition. 44 (2011) 97-106.

[63] M.U. Şen, H. Erdogˇan, Linear classifier combination and selection using group sparse regularization and hinge loss, Pattern Recognition Letters. 34 (2013) 265-274.

[64] L. Todorovski, S. Džeroski, Combining Classifiers with Meta Decision Trees, Machine Learning. 50 (2003) 223-249.

[65] W. Pedrycz, Human Centricity and Perception-Based Perspective and Their Centrality to the Agenda of Granular Computing, in: Y. Wang (Eds.), Cognitive Informatics for Revealing Human Cognition: Knowledge Manipulations in Natural Intelligence, 2013, pp. 178-193.

[66] W. Pedrycz, W. Homenda, Building the fundamentals of granular computing: A principle of justifiable granularity, Applied Soft Computing. 13(10) (2013), 4209-4218.

[67] W. Pedrycz, R. Al-Hmouz, A. Morfeq, A. Balamash, The design of free structure Granular Mappings: The use of the Principle of Justifiable Granularity, IEEE Transactions on Cybernetics. 43(6) (2013), 2105- 2113.

[68] R.E. Moore, R.B. Kearfott, M.J. Cloud, Introduction to Interval Analysis, Society for Industrial and Applied Mathematics Publisher 2009.

[69] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html

[70] Medical imaging CLEF 2009, https://www.imageclef.org/datasets

[71] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, et al, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc Natl Acad Sci USA (1999), pp. 6745-6750.

[72] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A library for large linear classification, Journal of Machine Learning Research. 9(2008), 1871-1874.

[73] J. Demsar, Statistical comparisons of classifiers over multiple datasets, Journal of Machine Learning Research. 7 (2006), 1–30.

[74] S. Raudys and R.P.W. Duin, Expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix, Pattern Recognition Letters. 19(5,6) (1998), 385-392.

[75] J. A. Anderson, Logistic discrimination, in: P. R. Krishnaiah and L. N. Kanal (eds.), Handbook of Statistics 2: Classification, Pattern Recognition and Reduction of Dimensionality,Amsterdam, 1982,pp. 169-191.

[76] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms (3rd edition), MIT Press, 2009.

[77] J. Su, H. Zhang, A Fast Decision Tree Learning Algorithm, in Proceedings of the 21st International conference on Artificial intelligence, Vol 1, 2006, pp 500-505.

Appendix

Proposition 1: If $\left[a_{opt}, b_{opt}\right]$ is the interval built by justifiable granularity on the meta-data associated with the first class label of two class-classification problems, the interval associated with the other class label is $\left[1-b_{opt}, 1-a_{opt}\right]$

Proof: Let denote X and Y as two random variables represented for the meta-data associated with the first and the second class label respectively. Based on the property of meta-data [3], we have:

$$X+Y=1 \tag{A1}$$

Denote $med(X)$ and $med(Y)$ as the median of X and Y. Based on the definition of median, we have:

$$\mathrm{P}\big(X<med(X)\big)=\mathrm{P}\big(X\geq med(X)\big)=1/2 \tag{A2}$$

Replace X by $1-Y$ we obtain:

$$\mathrm{P}\big(1-Y<med(X)\big)=\mathrm{P}\big(1-Y\geq med(X)\big)=1/2$$

$$\Leftrightarrow\ \mathrm{P}(1-med(X)<Y)=\mathrm{P}(1-med(X)\geq Y)=1/2 \tag{A3}$$

$\Longrightarrow med(Y)=1-med(X)$ is the median of the meta-data associated with the second class label.

Since $[a,b]$ is the interval built by justifiable information granularity, based on (6)-(10) we have:

$V(b)=(\mathrm{C}\{med(X)\leq X\leq b\})^{\beta}\times f_2(|med(X)-b|)$and $b_{opt}=\arg\max\ \{V(b)|b\geq med(X)\}$(A4)

$V(a)=(\mathrm{C}\{a\leq X\leq med(X)\})^{\beta}\times f_2(|a-med(X)|)$ and $a_{opt}=\arg\max\ \{V(a)|a\leq med(X)\}$(A5)

Replace X by $1-Y$ and $med(X)$ by $1-med(Y)$ in (A4), we have:

$$V(b)=(\mathrm{C}\{1-med(Y)\leq 1-Y\leq b\})^{\beta}\times f_2(|1-med(Y)-b|)$$

$$\Leftrightarrow V(b) = (\mathrm{C}\{1 - b \le Y \le med(Y)\})^{\beta} \times f_2(|1 - b - med(Y)|) \qquad \text{(A6)}$$

Comparing (A6) and (A5), the lower bound of the interval for Y is $(1 - b_{opt})$. Similarly, the upper bound of the interval for Y is $(1 - a_{opt})$□

Property 1: $d(x, [a, b]) \ge 0$ and $d(x, [a, b]) = 0 \leftrightarrow x = a = b$

Proof: Since $d(x, [a, b]) = \max\{|x - a|, |x - b|\}$, and $|x - a| \ge 0, |x - b| \ge 0 \Longrightarrow d(x, [a, b]) \ge 0$.

If $x = a = b$, it is easy to see that $d(x, [a, b]) = 0$.

On the other hand, in case $d(x, [a, b]) = \max\{|x - a|, |x - b|\} = 0$. since $|x - a| \ge 0, |x - b| \ge 0$ we obtain$|x - a| = |x - b| = 0 \Rightarrow x = a = b$ □

Property 2: $d(x_1, [a, b]) = d(x_2, [a, b])$ iff $x_1 = x_2$ or $x_1 + x_2 = a + b$

Proof: Denote $mid(a, b) = (a + b)/2$. Since $\max\{|x_1 - a|, |x_1 - b|\} = \max\{|x_2 - a|, |x_2 - b|\}$, we have four cases (A7-A10).

$\max\{|x_1 - a|, |x_1 - b|\} = |x_1 - a|, \max\{|x_2 - a|, |x_2 - b|\} = |x_2 - a| \Rightarrow |x_1 - a| = |x_2 - a|$ (A7)

$\max\{|x_1 - a|, |x_1 - b|\} = |x_1 - a|,\ \max\{|x_2 - a|, |x_2 - b|\} = |x_2 - b| \Rightarrow |x_1 - a| = |x_2 - b|$ (A8)

$\max\{|x_1 - a|, |x_1 - b|\} = |x_1 - b|,\ \max\{|x_2 - a|, |x_2 - b|\} = |x_2 - b| \Rightarrow |x_1 - b| = |x_2 - b|$ (A9)

$\max\{|x_1 - a|, |x_1 - b|\} = |x_1 - b|,\ \max\{|x_2 - a|, |x_2 - b|\} = |x_2 - a| \Rightarrow |x_1 - b| = |x_2 - a|$(A10)

Here we only consider (A7) and (A8) (A9 and A10 can be handled similarly). For the case (A7), it means that $|x_1 - a| > |x_1 - b|$ and $|x_2 - a| > |x_2 - b| \Rightarrow x_1 > mid(a, b)$ and $x_2 > mid(a, b) \Rightarrow$ $|x_1 - a| = x_1 - a$ and $|x_2 - a| = x_2 - a \Rightarrow x_1 = x_2$

For case (A8), by the proof above we have $|x_1 - a| = x_1 - a$, and $|x_2 - b| > |x_2 - a| \Rightarrow x_2 < mid(a, b) \Rightarrow |x_2 - b| = b - x_2 \Rightarrow x_1 - a = b - x_2 \Rightarrow x_1 + x_2 = a + b$□

Property 3: If $x_1 \in [a, b]$ and $x_2 \notin [a, b]$ then $d(x_1, [a, b]) < d(x_2, [a, b])$

Proof: Since $x_1 \in [a, b] \Rightarrow d(x_1, [a, b]) = \max\{|x_1 - a|, |x_1 - b|\} \le |b - a|$.

Since $x_2 \notin [a,b] \Rightarrow d(x_2,[a,b]) = \max\{|x_2-a|,|x_2-b|\} = \begin{cases}|x_2-a|+|b-a| \\ |x_2-b|+|b-a|\end{cases} > |b-a| \geq d(x_1,[a,b])$ □

Property 4: $d(x_1,[a,b]) \leq d(x_1,x_2) + d(x_2,[a,b])$

Proof: Since $|x_1-a| = |x_1-x_2+x_2-a| \leq |x_1-x_2|+|x_2-a|$ and $|x_1-b| = |x_1-x_2+x_2-b| \leq |x_1-x_2|+|x_2-b|$

$\Rightarrow d(x_1,[a,b]) = \max\{|x_1-a|,|x_1-b|\} \leq \max\{|x_1-x_2|+|x_2-a|,|x_1-x_2|+|x_2-b|\} = d(x_1,x_2)+d(x_2,[a,b])$□

Property 5: $d(x,[a,b]) = d([a,b],x)$

It is the result of $x = [x,x]$□

Property 6: $d(\alpha x,[\alpha,\alpha][a,b]) = |\alpha| d(x,[a,b])$

Proof: If $\alpha \geq 0$, $[\alpha,\alpha][a,b] = [\alpha a, \alpha b]$

$$d(\alpha x,[\alpha,\alpha][a,b]) = d(\alpha x,[\alpha a,\alpha b]) = \max(|\alpha x-\alpha a|,|\alpha x-\alpha b|) = \max(\alpha|x-a|,|\alpha||x-b|)$$

$$= \alpha\max(|x-a|,|x-b|) = \alpha d(x,[a,b])$$

Doing a similar way, if $\alpha < 0$, $[\alpha,\alpha][a,b] = [\alpha b,\alpha a]$, we have

$d(\alpha x,[\alpha,\alpha][a,b]) = -\alpha d(x,[a,b]) \rightarrow d(\alpha x,[\alpha,\alpha][a,b]) = |\alpha| d(x,[a,b])$□

Property 7: $d(x,[a,b]) = d(x+\alpha,[a,b]+[\alpha,\alpha])$

Proof: $d(x+\alpha,[a,b]+[\alpha,\alpha]) = d(x+\alpha,[a+\alpha,b+\alpha]) = \max(|x+\alpha-a-\alpha|,|x+\alpha-b-\alpha|) = \max(|x-a|,|x-b|) = d(x,[a,b])$□

Property 8: If $\mathbf{t}_1 = \{t_{1j}\}\ t_{1j} \in V_j$ and $\mathbf{t}_2 = \{t_{2j}\}\ t_{2j} \notin V_j \forall j = 1,\dots,|\mathbf{V}|$ then $\mathbf{d}(\mathbf{t_1},\mathbf{V}) < \mathbf{d}(\mathbf{t_2},\mathbf{V})$

Proof: Using Property 3, we have if $t_{1j} \in V_j$ and $t_{2j} \notin V_j$ then $d(t_{1j},V_j) < d(t_{2j},V_j)$

That inequation is true $\forall j = 1, \dots, |\mathbf{V}|$ so $\sum_{j=1}^{|\mathbf{V}|} d(t_{1j}, V_j) < \sum_{j=1}^{|\mathbf{V}|} d(t_{2j}, V_j) \leftrightarrow \mathbf{d}(\mathbf{t_1}, \mathbf{V}) < \mathbf{d}(\mathbf{t_2}, \mathbf{V})$□

Property 9: $\mathbf{d}(\mathbf{t}_1, \mathbf{V}) \leq \mathbf{d}(\mathbf{t}_1, \mathbf{x}_2) + \mathbf{d}(\mathbf{t}_2, \mathbf{V})$ where $\mathbf{d}(\mathbf{t}_1, \mathbf{t}_2)$ is the distance between two vector $\mathbf{t}_1$ and $\mathbf{t}_2$

Proof:

$\mathbf{d}(\mathbf{t}_1, \mathbf{V}) = \sum_{j=1}^{|\mathbf{V}|} d(t_{1j}, V_j) = \sum_{j=1}^{|\mathbf{V}|} \max\left(|t_{1j} - \overline{V_j}|, \left|t_{1j} - \underline{V_j}\right|\right) = \sum_{j=1}^{|\mathbf{V}|} \max\left(|t_{1j} - t_{2j} + t_{2j} - \overline{V_j}|, \left|t_{1j} - t_{2j} + t_{2j} - \underline{V_j}\right|\right) \leq \sum_{j=1}^{|\mathbf{V}|} \max\left(|t_{1j} - t_{2j}| + |t_{2j} - \overline{V_j}|, |t_{1j} - t_{2j}| + \left|t_{2j} - \underline{V_j}\right|\right) = \sum_{j=1}^{|\mathbf{V}|} |t_{1j} - t_{2j}| + \max\left(|t_{2j} - \overline{V_j}|, \left|t_{2j} - \underline{V_j}\right|\right) = \mathbf{d}(\mathbf{t}_1, \mathbf{t}_2) + \mathbf{d}(\mathbf{t}_2, \mathbf{V})$□

Algorithm: Training process

```
Input:    $\mathcal{D}$: training set, $\boldsymbol{\mathcal{K}} = \{\mathcal{K}_k | k = 1,\dots,K\}$ : learning algorithms,
          $\alpha, \beta$: parameters to generate intervals
Output:   $M$ granular prototypes $\mathcal{V} = \{\mathbf{V}_m\}_{m=1,\dots,M}$ and base classifier
          $\{BC_k\}_{k=1,\dots,K}$

$\mathbf{L} = \emptyset$,   $\{\mathcal{D}_1, \dots, \mathcal{D}_T\}$ =T-partition($\mathcal{D}$)
For each $\mathcal{D}_i$
   $\widetilde{\mathcal{D}}_i = \mathcal{D} - \mathcal{D}_i$
   For each $\mathcal{K}_k$
      Classifier $BC_k^{-i}$ = Learn($\mathcal{K}_k$, $\widetilde{\mathcal{D}}_i$)
      $\mathbf{L}$ = $\mathbf{L}$ ∪ Test$(BC_k^{-i}, \mathcal{D}_i)$
   End For
End For
For each $\mathcal{K}_k$
   Base classifier $BC_k$= Learn($\mathcal{K}_k$, $\mathcal{D}$)
End For
$\mathbf{L}_m = \{\mathbf{L}(\mathbf{x}) | y(\mathbf{x}) = y_m\}, (m = 1, \dots, M)$, $\mathcal{V} = \{\mathbf{V}_m\}_{m=1,\dots,M} = \emptyset$
For m=1 to $M$
   For j=1 to $M$x$K$
      Get jth column of $\mathbf{L}_m$ i.e. $\mathbf{L}_{m,j}$
```

```
            Find $med(\mathbf{L}_{m,j})$
            For each $b \in \mathbf{L}_{m,j}, b \geq med(\mathbf{L}_{m,j})$
                Compute $V(b)$ by (7)
            End For
            $\overline{v_{m,j}} = \arg\max_{b} V(b)$
            For each $a \in \mathbf{L}_{m,j}, a \leq med(\mathbf{L}_{m,j})$
                Compute $V(a)$ by (10)
            End For
            $\underline{v_{m,j}} = \arg\max_{a} V(a)$
            $V_{mj} = [\underline{v_{m,j}}, \overline{v_{m,j}}]$
      End For
      $\mathbf{V}_m = \mathbf{V}_m \cup V_{mj}$
    End For
    Return $\mathcal{V} = \{\mathbf{V}_m\}_{m=1,\dots,M}$ and $\{BC_k\}_{k=1,\dots,K}$
```

Algorithm: Classification process

```
Input:   $\mathbf{x}^u$: unlabeled observation, $\mathcal{V}$: set of granular prototypes,
         $\{BC_k\}_{k=1,\dots,K}$: base classifier
Output:  Class label of $\mathbf{x}^u$

    $\mathbf{L}(\mathbf{x}^u) = \emptyset$
    For each $BC_k$
        $\mathbf{L}(\mathbf{x}^u)$ = $\mathbf{L}(\mathbf{x}^u)$ ∪ Test($BC_k, \mathbf{x}^u$)
    End For
    For m=1 to M
        For j=1 to MK
            Compute $d(\mathbf{L}(x_j^u), V_{mj})$ by (12)
        End For
        Compute $\mathbf{d}(\mathbf{L}(\mathbf{x}^u), \mathbf{V}_m)$ by (20)
    End For
    $\mathbf{x}^u \in y_j$ if $\mathbf{d}(\mathbf{L}(\mathbf{x}^u), \mathbf{V}_j) = \min_{m=1,\dots,M} \mathbf{d}(\mathbf{L}(\mathbf{x}^u), \mathbf{V}_m)$
```

TABLE.A1.CLASSIFICATION ERROR RATES OF THE 10 LEARNING ALGORITHMS AND THE PROPOSED METHOD

	LDA		Naïve Bayes		kNN$_5$		kNN$_{25}$		kNN$_{50}$	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Artificial	0.3121	1.17E-03	0.3121	1.15E-03	0.2413	2.31E-03	0.1906	1.03E-03	0.1990	1.94E-03
Australian	0.1453	1.42E-03	0.1387	1.39E-03	0.3439	2.99E-03	0.3286	2.40E-03	0.3316	1.34E-03
Biodeg	0.1465	8.08E-04	0.2068	1.42E-03	0.1828	1.38E-03	0.2265	1.53E-03	0.2472	1.60E-03
Blood	0.2281	3.05E-04	0.2453	1.11E-03	0.2341	1.56E-03	0.2407	4.40E-04	0.2382	2.15E-05
Breast-cancer	0.0414	4.99E-04	0.0412	5.71E-04	0.0321	4.37E-04	0.0369	5.05E-04	0.0407	5.22E-04
CLEF2009	0.1714	1.42E-03	0.3684	1.79E-03	0.3583	3.09E-03	0.4506	2.45E-03	0.4975	2.09E-03
Cleveland	0.4228	4.21E-03	0.4328	4.31E-03	0.5521	3.64E-03	0.4585	1.30E-03	0.4645	3.31E-04
Colon	0.1845	1.96E-02	0.3717	3.98E-02	0.1740	1.69E-02	0.3140	5.69E-03	0.3524	1.45E-03
Conn-bench-vowel	0.3856	3.80E-03	0.4699	4.91E-03	0.0701	1.36E-03	0.4795	3.83E-03	0.5525	3.13E-03
Contraceptive	0.4829	1.46E-03	0.5247	1.95E-03	0.4840	1.17E-03	0.4528	1.26E-03	0.4601	1.20E-03
Dermatology	0.0285	7.05E-04	0.0397	9.84E-04	0.1138	2.63E-03	0.2464	3.39E-03	0.3394	2.46E-03
Glass	0.3574	7.68E-03	0.4019	7.10E-03	0.3333	8.59E-03	0.3793	7.46E-03	0.4195	7.62E-03
Haberman	0.2630	2.48E-03	0.2589	2.51E-03	0.2884	3.51E-03	0.2524	3.20E-03	0.2566	1.66E-03
Heart	0.1637	4.26E-03	0.1615	4.68E-03	0.3193	6.36E-03	0.3156	7.78E-03	0.3552	6.04E-03
Hepatitis	0.1688	1.48E-02	0.1563	1.22E-02	0.1938	6.68E-03	0.1625	3.28E-03	0.1625	3.28E-03
Iris	0.0193	1.00E-03	0.0400	2.31E-03	0.0395	1.79E-03	0.0440	2.33E-03	0.0660	3.51E-03
Led7digit	0.2778	3.45E-03	0.2706	3.28E-03	0.2970	4.59E-03	0.2692	4.27E-03	0.2636	4.17E-03
Madelon	0.4592	1.08E-03	0.4119	1.18E-03	0.2936	9.81E-04	0.2529	7.15E-04	0.2604	7.55E-04
Multiple Features	0.0199	8.33E-05	0.0389	1.79E-04	0.0511	2.39E-04	0.0910	2.97E-04	0.1249	3.63E-04
Musk2	0.0566	6.39E-05	0.2687	2.16E-04	0.0345	4.70E-05	0.0486	6.24E-05	0.0606	6.44E-05
Satimage	0.1598	1.28E-04	0.2126	1.76E-04	0.0910	1.15E-04	0.1067	1.10E-04	0.1230	1.40E-04
Texture	0.0053	7.93E-06	0.2470	2.68E-04	0.0133	2.52E-05	0.0274	4.40E-05	0.0395	5.16E-05
Twonorm	0.0223	2.96E-05	0.0239	2.15E-05	0.0317	3.84E-05	0.0254	4.44E-05	0.0234	3.26E-05
Vertebral	0.1965	3.69E-03	0.2505	4.59E-03	0.1845	2.48E-03	0.1671	3.03E-03	0.1974	3.65E-03
Wine	0.0095	4.45E-04	0.0463	1.98E-03	0.2971	8.24E-03	0.2990	1.13E-02	0.3008	9.96E-03
Yeast	0.4215	1.50E-03	0.4259	1.49E-03	0.4366	1.15E-03	0.4059	1.34E-03	0.4168	1.29E-03

	Decision Tree C4.5		Decision Stump		Fisher		LLC		LMC		Proposed Method	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Artificial	0.2433	1.60E-03	0.4251	1.43E-04	0.3121	1.17E-03	0.3119	1.20E-03	0.4934	2.11E-03	0.2142	1.73E-03
Australian	0.1678	2.13E-03	0.4139	4.42E-04	0.1443	1.42E-03	0.1388	1.24E-03	0.5406	1.91E-03	**0.1262**	1.25E-03
Biodeg	0.1853	1.39E-03	0.3374	1.21E-05	0.1412	7.51E-04	0.1378	8.62E-04	0.3446	2.03E-03	**0.1374**	1.10E-03
Blood	0.2595	1.68E-03	0.2379	1.69E-05	**0.2276**	2.86E-04	0.2281	3.76E-04	0.3330	2.68E-03	0.2438	1.86E-03
Breast Cancer	0.0526	6.94E-04	0.2311	1.62E-03	0.0416	4.93E-04	0.0338	4.[illegible]7E-04	0.0360	5.66E-04	0.0359	5.01E-04
CLEF2009	0.3664	3.12E-03	0.8062	2.06E-04	0.1938	1.38E-03	0.2102	3.49E-03	0.3252	3.28E-02	**0.1659**	1.45E-03
Cleveland	0.5055	6.30E-03	0.4611	7.10E-05	0.4237	1.82E-03	**0.4181**	3.33E-03	0.6100	5.50E-03	0.4357	2.03E-03
Colon	0.2588	2.74E-02	0.3926	2.66E-02	0.2150	2.25E-02	0.1812	1.97E-02	0.6631	1.96E-03	**0.1633**	2.02E-02
Conn Bench Vowel	0.2295	3.15E-03	0.8441	3.84E-04	0.5108	3.77E-03	0.4307	3.84E-03	0.4538	4.75E-03	0.1571	2.37E-03
Contraceptive	0.4830	1.83E-03	0.5730	4.74E-06	0.4959	1.15E-03	0.4880	1.28E-03	0.6196	9.13E-04	**0.4572**	1.60E-03
Dermatology	0.0516	1.23E-03	0.5144	1.64E-03	0.0245	6.67E-04	0.0595	1.70E-03	0.4922	7.52E-03	**0.0242**	6.10E-04
Glass	**0.3092**	1.05E-02	0.4991	4.92E-03	0.3877	5.[illegible]8E-03	0.3626	7.74E-03	0.5578	8.49E-03	0.3196	7.10E-03
Haberman	0.3048	5.27E-03	0.2647	8.92E-05	0.2618	1.96E-03	0.2576	2.07E-03	0.3189	7.04E-03	**0.2437**	3.75E-03
Heart	0.2381	6.70E-03	0.4444	1.37E-04	0.1637	4.26E-03	0.1678	3.85E-03	0.3689	7.73E-03	**0.1561**	4.75E-03
Hepatitis	0.1663	1.22E-02	0.1625	3.28E-03	0.16[illegible]	1.3[illegible]E-02	0.1800	1.73E-02	0.2800	2.69E-02	**0.1520**	1.38E-02
Iris	0.0507	2.40E-03	0.3520	2.05E-03	0.1187	6.45E-03	0.0367	1.99E-03	0.0800	3.73E-03	0.0410	2.35E-03
Led7digit	0.2906	2.75E-03	-	-	0.2690	3.45E-03	0.2666	3.79E-03	0.2634	3.90E-03	-	-
Madelon	0.2462	1.04E-03	0.4998	2.47E-0[illegible]	0.4595	1.24E-03	0.4589	1.30E-03	0.3996	8.91E-04	**0.2452**	9.92E-04
Multiple Features	0.0636	3.10E-04	0.8032	3.98E-05	0.019[illegible]	5.72E-05	0.0127	7.72E-05	0.4499	1.05E-03	0.0120	6.35E-05
Musk2	**0.0322**	4.34E-05	0.1541	3.[illegible]2E-07	0.0607	6.48E-05	0.0473	5.75E-05	0.2757	1.88E-04	0.0387	4.31E-05
Satimage	0.1415	2.27E-04	0.5975	5.93E-05	0.2364	6.90E-05	0.1637	9.45E-05	0.2229	2.00E-04	0.1222	1.21E-04
Texture	0.0761	1.13E-04	0.7737	2.03E-04	0.0134	2.37E-05	0.0969	3.63E-02	0.2419	2.71E-04	0.0096	1.51E-05
Twonorm	0.1602	2.21E-04	0.4924	8.68E-06	0.0223	2.96E-05	0.0223	2.95E-05	**0.0219**	3.15E-05	0.0222	2.77E-05
Vertebral	0.1984	3.75E-03	0.2310	1.37E-03	0.2077	3.29E-03	0.1521	2.60E-03	0.2423	5.03E-03	**0.1510**	3.75E-03
Wine	0.1010	4.60E-03	0.4604	6.45E-03	0.0123	6.00E-04	0.0242	1.22E-03	0.2867	9.76E-03	0.0261	1.79E-03
Yeast	0.4642	1.86E-03	0.6[illegible]99	3.57E-04	0.4658	1.26E-03	0.4173	1.44E-03	0.5028	1.21E-03	**0.4032**	1.38E-03

**The best results are highlight in bold*