NGUYEN, T.T., DANG,M.T., BAGHEL, V.A., LUONG, A.V., MCCALL, J. and LIEW, A.W.-C. 2020 Evolving interval-based representation for multiple classifier fusion. *Knowledge-based systems* [online], 201-202, article ID 106034. Available from: https://doi.org/10.1016/j.knosys.2020.106034

Evolving interval-based representation for multiple classifier fusion.

NGUYEN, T.T., DANG,M.T., BAGHEL, V.A., LUONG, A.V., MCCALL, J. and LIEW, A.W.-C.

2020

Evolving interval-based representation for multiple classifier fusion

Tien Thanh Nguyen[1], Manh Truong Dang[1], Vimal Anand Baghel[2], Anh Vu Luong[3], John McCall[1], Alan Wee-Chung Liew[3]

[1]School of Computing Science and Digital Media, Robert Gordon University, Aberdeen, UK

[2]Department of Computer Science and Engineering, Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur, India

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

Abstract: Designing an ensemble of classifiers is one of the popular research topics in machine learning since it can give better results than using constitute member. Furthermore, the performance of ensemble can be improved using the selection or adaptation approach. In the former, the optimal set of base classifiers, meta-classifier, original features, or meta-data is selected to obtain a better ensemble than using the entire classifiers and features. In the latter, base classifiers or combining algorithms working on the outputs of base classifiers are made to adapt to a particular problem. The adaptation here means that the parameters of these algorithms are trained to be optimal for each problem. In this study, we propose a novel evolving combining algorithm using the adaptation approach for the ensemble systems. Instead of using the numerical value when computing the representation for each class label, we propose to use the interval-based representation for the class label. The optimal value of the representation is found through Particle Swarm Optimization. The class label is assigned to each test instance by selecting the class label associated with the shortest distance between the predictions of the base classifiers on that instance and the interval-based representation. Experiments conducted on a number of popular datasets confirm that the proposed method is better than the well-known combining algorithms for ensemble systems using the combining methods including Decision Template, Sum Rule, L2-loss Linear Support Vector Machine, and Multiple Layer Neural Network, and the selection methods for ensemble systems (GA-Meta-data, META-DES, and ACO).

Keywords: ensemble method; multiple classifiers; classifiers fusion; combining classifiers; ensemble system

1. Introduction

Ensemble of classifiers (EoC, also known as ensemble systems, classifier ensemble) is one of the popular research topics in machine learning in which multiple classifiers collaborate in decision making i.e. their predictions are combined to generate the final prediction, which is expected to be better than the predictions of any of the individual classifier. Many types of ensemble systems have been introduced using one of the six techniques: different initializations, different architectures, different parameters, different learning algorithms, different training sets, and different feature sets [1]. In this study, we are concern with heterogeneous ensemble in which the base classifiers are obtained from different learning algorithms trained on a given training set. A combining algorithm is then used to aggregate the output (called meta-data) of these base classifiers to obtain the final prediction. An important research topic for heterogeneous ensemble is the design of an effective combining algorithm [2, 3]. In the literature, the combining algorithm is also known as the meta-classifier, meta-learner, combiner, or second-level classifier.

It has been observed that by removing some poor-performing base classifiers from the ensemble, a better prediction can be obtained than using the entire EoC. This approach is known as classifier selection (also called ensemble selection, selective ensemble or ensemble pruning) in which a search method is applied to the ensemble to find the optimal set of base classifiers for each specific dataset [4]. Moreover, by searching for a suitable subset of features for each learning algorithm, the obtained classifier can give a better prediction than that trained using the entire feature set. This is known as feature selection for the ensemble [5]. Similar approaches can also be found in the selection of meta-classifier [6] and meta-data [7, 8] for the ensemble system. Finally, we usually use pre-selected parameters for the learning algorithm and the combining algorithm when training the base classifiers and the meta-classifier. These parameters, in fact, can be optimized to adapt to each specific classification task. This is known as ensemble optimization.

Evolutionary Computation (EC) have been widely applied to ensemble optimization [9-11]. Classical optimization methods may be more efficient than EC when solving linear, strongly convex problems. However, for non-differentiable, discontinuous, or multi-model objective functions that appear in many real-life applications, EC may be a better choice. In this study, we use Particle Swarm Optimization (PSO) algorithm [12], a popular EC, to find the optimal parameters for the meta-classifier.

In this work, we introduce a novel meta-classifier for the heterogeneous ensemble. The proposed method origins from the fact that Decision Template method [13], one of the most well-known meta-classifier, does not work well on imbalanced datasets, and for datasets with skewed class distribution. In addition, as classifiers generated from different learning strategies can produce different predictions, it is more informative to model these predictions using the interval [14, 15]. Therefore, we first generate meta-data of the training set through cross validation and then model the meta-data of each class label

by using an interval-based representation i.e. a vector of intervals. The optimal bounds of the representation are obtained by minimizing the empirical 0-1 loss function on the training data based on Partial Swarm Optimization (PSO). Then, in the classification phase, the class label is assigned to a test instance based on the closeness of its meta-data to the interval-based representation.

Table 1. Main Notation

Notation	Description
$\mathcal{D} = \{(\mathbf{x}_i, \hat{y}_i)\}$	The training set
$(\mathbf{x}_i, \hat{y}_i)$	Observation with the true label $\hat{y}_i$
M	The number of class labels
N	The number of training observations $N = \|\mathcal{D}\|$
K	Number of classifiers
$\mathcal{Y} = \{y_m\}_{m=1,\dots,M}$	Set of labels
$\boldsymbol{\mathcal{H}}$	Set of classifiers
$s_{i,j}(\mathbf{x}_i)$	Support of i^{th} classifier for j^{th} class on $\mathbf{x}_i$
$\mathbf{L}$	Meta-data of $\mathcal{D}$
$\mathbf{L}(\mathbf{x}_i)$	Meta-data of $\mathbf{x}_i$
$\|.\|$	Relative cardinality of a set
$\boldsymbol{\mathcal{R}} = \{\mathbf{R}_j\}$	Interval-based representation
$\mathbf{R}_j$	Interval-based representation for the m^{th} class $(m = 1, \dots, M)$
$d(x, [\cdot])$	Distance between scalar x and an interval
$\mathbf{d}(\mathbf{x}_i, \boldsymbol{\mathcal{R}})$	Distance between a vector $\mathbf{x}_i$ and a representation $\boldsymbol{\mathcal{R}}$
$\mathcal{L}_{0-1}(\mathbf{x}_i, \boldsymbol{\mathcal{R}})$	0-1 loss function on $\mathbf{x}_i$ associated with $\boldsymbol{\mathcal{R}}$
$\mathcal{L}_{0-1}(\boldsymbol{\mathcal{R}})$	0-1 loss function on the training data associated with $\boldsymbol{\mathcal{R}}$

The contribution of our work is:

- We performed a review on the development of meta-classifier for heterogeneous ensemble and the evolutionary approaches for ensemble selection and optimization.
- We analysed the limitation of the well-known Decision Template method and proposed a novel meta-classifier algorithm based on the interval-based representation
- We performed extensive experiments to show that the proposed method is better than several well-known meta-classifiers and EC-based ensemble systems.

This paper is organized as follows. Section 2 presents background and related work including meta-classifier design for the ensemble system, the Decision Template method, and the EC-approach for ensemble optimization. Section 3 introduces the formulation and optimization technique of the proposed method. Section 4 presents the experimental design, in which we describe the datasets for the experiments, the baseline methods, the experimental settings, and the statistical tests of significance for performance comparison. Section 5 discusses in detail the comparative study. Section 6 provides the conclusion and future work.

2. Background and related work

a. Meta-classifier for heterogeneous ensemble

We denote $\mathcal{D} = \{(\mathbf{x}_i, \hat{y}_i), i = 1, \dots, N\}$ as the training set in which $\hat{y}_i$ is the true label of $\mathbf{x}_i$, $\hat{y}_i \in \mathcal{Y} = \{y_m\}, m = 1, \dots, M$, and K as the number of base classifiers. For an observation $\mathbf{x}_i$, the k^{th} classifier returns the supports, i.e. the predictions that this observation belongs to the class labels of label set $\mathcal{Y}$. In fact, these supports can be viewed as an estimation of the posterior probabilities that $\mathbf{x}_i$ belongs to the class labels. We denote $s_{k,m}(\mathbf{x}_i)$ as the support of the k^{th} classifier for the m^{th} class on $\mathbf{x}_i$ in which $s(\mathbf{x}_i) \in [0,1]$ and $\sum_m s_{k,m}(\mathbf{x}_i) = 1$.

The meta-classifier works on these predictions to obtain the final prediction. There are two types of meta-classifier for the heterogeneous ensemble: fixed method and trainable method. For the fixed method, the meta-classifier does not use the label information in the training set in its prediction. Besides the simple fixed combining methods like fixed combining rules, e.g. Sum, Product, Max, Min, Median, Majority Vote [16], and weighted multiple rules [17], other fixed methods came up with more complicated designs. One of the popular combining methods is based on the concept of base belief assignment (BBA) in the Dempster-Shafer (D-S) theorem [18]. In detail, this method works on the supports of the base classifiers for a test sample by using BBA and then combines the outputs based on the Dempster's rule or a related modification. There are two approaches here, one focuses on the modification of the Dempster's rule to reallocate and manage evidential conflict in the supports of the base classifiers while the other focuses on the correction of the original supports. Some examples of combining methods based on the D-S theorem can be found in [18, 19].

The trainable methods meanwhile work on the meta-data of the training observations to obtain the class discrimination model. The trainable methods are based on the stacked generalization paradigm (also called stacking algorithm). Stacking algorithm works as follows. The base classifiers are obtained by running different learning algorithms on the original training set. Another algorithm called the combining algorithm is trained on the predictions from the base classifiers to obtain the meta-classifier. Although any supervised algorithm such as Naïve Bayes (NB) and Decision Tree (DT) can be used as

the meta-classifier [8, 20], a specific meta-classifier that can handle and adapt to the characteristics of the meta-data is expected to achieve better performance. There are two main approaches to combine the meta-data, namely weighted trainable combining and representation-based combining. In the first approach, it is assumed that each base classifier puts a different weight on each class label and the meta-classifier tries to find the optimal weights by exploiting the meta-data. Some examples of the weighted trainable method are multiple linear regression (MLR) [21] and hinge loss function with different regularizations with group sparsity [22]. Recently, Yijing et al. [23] proposed a new weighted combining rule in which the weight of each classifier is computed based on its performance on the training data measured by Area under the ROC Curve (AUC). Representation-based combining methods, on the other hand, build the representation for class labels on the meta-data of the training data. A class label is assigned to a test instance based on the maximum similarity (or the minimum dissimilarity) between its meta-data and the class representation. Decision Template method [13], Fuzzy rule-based combining [3], Bayesian-based combining with Gaussian [2] or Gaussian Mixture Model (GMM) [7] approximation, and SCANN [24] are examples in this group.

b. Decision Template method

Among the trainable methods for the heterogeneous ensemble, Decision Template is one of the most popular meta-classifiers because of its simplicity and effectiveness. In this section, we briefly introduce the Decision Template method [13] which is the basis and motivation for our approach. The decision template $\boldsymbol{\mathcal{DT}} = \{\mathbf{DT}_j\}$ where $\mathbf{DT}_j$ is the decision template for the j^{th} class, computed on the meta-data is given by:

$$\mathbf{DT}_j = \begin{bmatrix} dt_j(1,1) & \dots & dt_j(1,M) \\ \dots & \ddots & \dots \\ dt_j(K,1) & \dots & dt_j(K,M) \end{bmatrix} \quad (1)$$

where each element of $\mathbf{DT}_j$ is computed by:

$$dt_j(k,m) = \sum_{i=1}^{N} \mathbb{I}[y_j = \hat{y}_i] s_{k,m}(\mathbf{x}_i) / \sum_{i=1}^{N} \mathbb{I}[y_j = \hat{y}_i] \quad (2)$$

for $k = 1, \dots, K; m = 1, \dots, M; j = 1, \dots, M$

in which $\hat{y}_i$ is the true class label of $\mathbf{x}_i$, $\mathbb{I}[\cdot]$ is the indicator function which returns 1 if the condition is true and 0 otherwise.

In (2), the $dt_j(k,m)$ is the average value computed on the meta-data of the training observations that belong to the j^{th} class (the condition $y_j = \hat{y}_i$ is true for training observations that belong to class y_j) associated with the k^{th} classifier and its predicted class label y_m. In other words, $dt_j(k,m)$ is the average of the supports for label *m* output by classifier *k*, taken over all training data with true label *j*.

In the classification phase, the distance between the meta-data of a test instance $\mathbf{x}$ and $\mathbf{DT}_j$ $(j = 1, \ldots, M)$ are computed. The class label is assigned to $\mathbf{x}$ based on the minimum of the similarity or the maximum of the dissimilarity between $\mathbf{L}(\mathbf{x})$ and $\mathbf{DT}_j$. Eleven measures between $\mathbf{L}(\mathbf{x})$ and each decision template were introduced in [13].

As a trainable combining method, the Decision Template method exploits the label information in the meta-data when constructing the decision templates during training. In the Decision Template method, the mean value is used as a representation for the meta-data distribution of a class. Although the mean is the most popular measure for the central tendency, when the distribution becomes skewed, the mean loses its ability to provide the best central location. Also, for imbalanced datasets, where instances from one class account for the majority of the data, the base classifiers will tend to predict the dominant class. Therefore, the decision templates for the other classes tend to have similar value as the decision template for the dominant class. We illustrate two examples of applying the Decision Template method on two imbalanced datasets Fertility and Hayes Roth (Fig.1). The 2-class Fertility dataset is imbalanced in which 80% of the observations belong to the first class. The 3-class Hayes Roth dataset is also imbalanced in which each of the first two classes accounts for about 40% of the dataset. In these examples, we constructed an ensemble with 3 base classifiers, i.e. Linear Discriminative Analysis (denoted by LDA), Naïve Bayes, and k Nearest Neighbor (with k set to 5, denoted by kNN5) and computed the decision templates on the meta-data to combine the classifiers. Clearly, for the Fertility dataset, the decision templates of the 2 classes are very similar and have low discriminative ability. Meanwhile, on Hayes Roth dataset, the decision templates for class 1 and class 2 are similar, which makes it difficult to assign a test instance to one of them. This makes the Decision Template method poor on these datasets.

c. Evolutionary Computation for ensemble system optimization

In the past years, many ECs for ensemble system optimization have been proposed (see Table 2) and they can be grouped into two categories: *optimization for selection* and *optimization for adaptation.* The optimization for selection searches for the optimal subset of base classifiers, base features, meta-classifiers, or meta-data to obtain better classification accuracy than using the whole set of classifiers or features. Kuncheva and Jain [25] proposed encodings showing which base features are used by different learning algorithms to generate base classifiers, and which base classifiers are selected in the final ensemble. Gabrys and Ruta [26] encoded the base classifiers-base features-combiners triple with a binary scheme in a chromosome and proposed a multidimensional GA including two-stage crossover operation to search for the optimal solution.

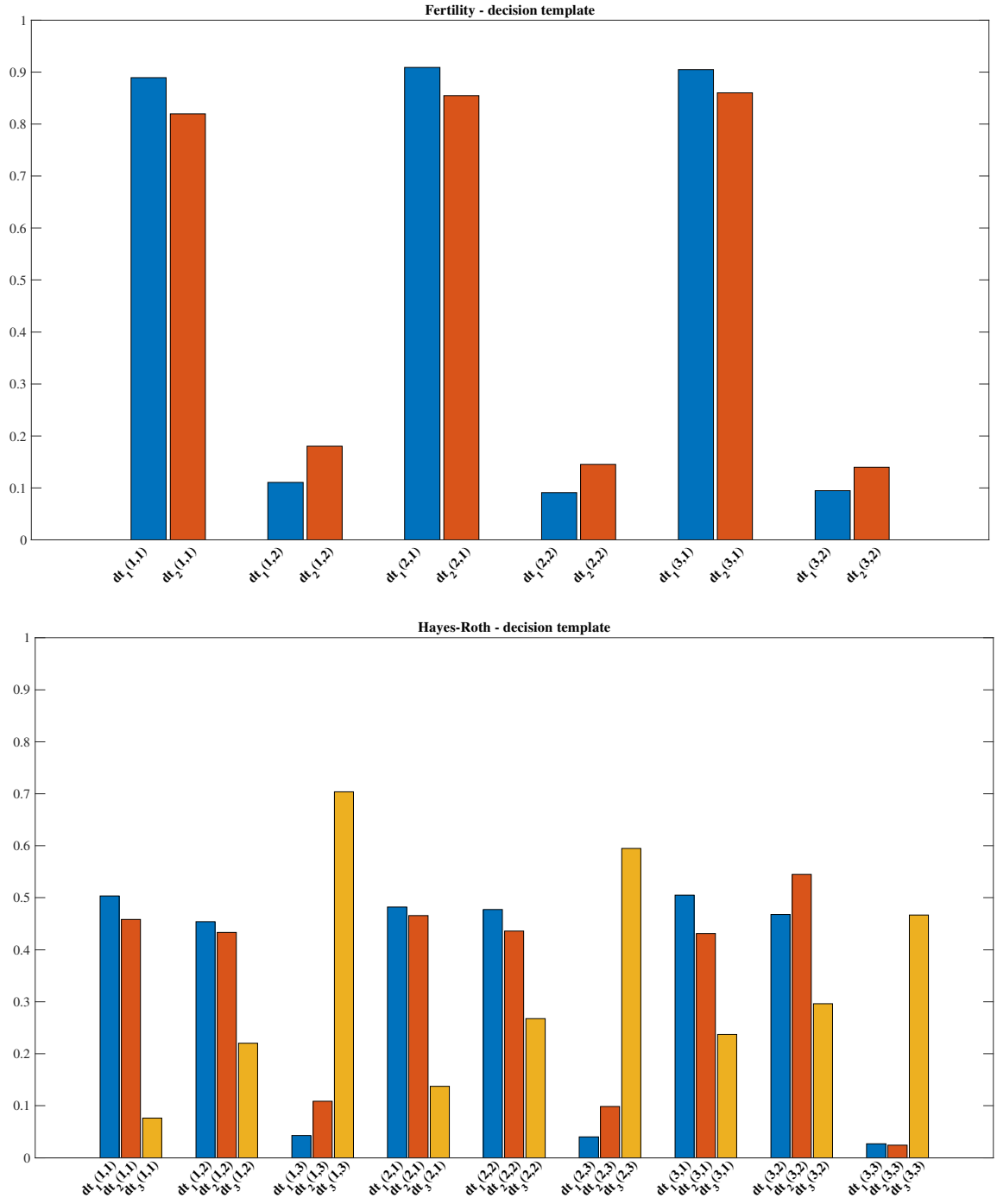


Fig.1. The decision templates generated on meta-data in Fertility and Hayes-Roth dataset

Kim and Oh [27] encoded the base classifiers with a binary scheme and then used hybrid GA to search for the optimal ensemble of classifiers. The hybrid GA contains a local search so that the offspring has a chance to be improved before going to the replacement stage. Kim and Kang [28] applied GA to base

classifier selection for bankruptcy prediction. Shunmugapriya and Kanmani [6] used ABC to find the optimal set of base classifiers and the associated optimal meta-classifier. Nguyen et al. [5] encoded the classifiers and the features in a single chromosome and used GA to simultaneously search for the optimal set of classifiers and the associated features. Nguyen et al. [17] encoded both the base classifiers and six fixed combining rules in a chromosome consisting of a two-part binary vector. The final optimal set of fixed combining rules is combined using the OWA operator. Chen et al. [29] used Ant Colony Optimization (ACO) to find the optimal set of base classifiers in an ensemble system and applied Decision Tree on the meta-data to generate the meta-classifier. Nguyen et al. used GA to search for the optimal subset of meta-data and used it with GMM [7] or Decision Tree [8] meta-classifier. Mendialdua et al. [30] searched for the optimal ensemble of base classifiers using an Estimation of Distribution Algorithm (EDA), a member in the class of EC. In EDAs, new population is generated by sampling the probability distribution estimated from selected individuals of previous generation. Mousavi and Eftekhari [31] considered two objective functions concerning the classification accuracy and ensemble diversity in the selection of base classifier and meta-classifier. The solution obtained by using NSGA-II was combined with a dynamic ensemble selection method with the aim of selecting ensemble of classifiers with the most competencies in a defined region associated with each instance. Haque et al. [32] encoded 20 base classifiers using a binary scheme and then used GA to search for the optimal solution which aims to perform well on the imbalanced datasets. Recently, Wang et al. [33] used NSGA-II to search for the optimal set of base classifiers generated by training a regression tree on 100 new training sets. The new training data was obtained by applying the random subspace and bootstrap resampling techniques on the original training set. Nguyen et al. [34] used ACO to simultaneously select the optimal meta-data and meta-classifier to improve ensemble performance.

Meanwhile, in optimization methods for adaptation, the base classifiers or the combining algorithms are learned to be adapted to the particular classification problem. The adaptation here means that the parameters of these algorithms are trained via an EC-based approach to obtain the specific optimal value for each classification task. Wang and Wang [35] used GA to find the optimal weights that the base learning algorithms put on each training observation. The base classifiers are then trained on the weighted training data. Nanni and Lumini [36] learned the adaptive representation alternative to the orthonormal representation in the SCANN method [24]. Instead of using the crisp label to represent the class label like in SCANN, the proposed chromosome encodes the real value as the new representation. Ali and Majid [9] applied the evolutionary ensemble system to predict human breast cancer using amino acid sequences. The GP technique was used to search for the prediction function as the mapping of meta-data to the target labels. Meanwhile, the PSO algorithm was used to find the optimal threshold for the prediction functions in different feature spaces.

There are also some methods to handle both optimizations for selection and adaptation. For example, Kim and Cho [10] encoded the feature selection methods which will be used for each learning algorithm

Table 2. Evolutionary Computation-based approaches for ensemble system optimization

Year	Author	# of Base Classifiers	Algorithms for Base Classifiers	Algorithms for Meta-Classifiers	Selection/Adaption approach					Ensemble Type	Optimization Method
					Base Classifier	Base Feature	Meta Classifier	Meta-data	Adaptation		
2000	Kuncheva et al. [25]	3	LDA, QDA, Logistic	Decision Template	✓	✓				Heterogeneity	GA
2006	Gabrys et al. [26]	5	LDA, QDA, Pseudo-Fisher SVM, Linear with KL expansion, Levenberg–Marquardt Neural Net	Sum, Majority Vote, Max, Min, Product	✓	✓	✓			Heterogeneity	GA
2006	Wang et al. [35]	100	Ensemble with RBF network	Weight-based average					✓	Homogeneity	GA
2008	Kim et al. [27]	50	KNN(K=1)	Majority Vote	✓					Homogeneity	GA
2008	Kim et al. [10]	6	MLP, SASOM, SVM(Linear and RBF), KNN(Cosine distance), KNN(Pearson correlation)	Majority Vote		✓			✓	Heterogeneity	GA
2009	Bacauskiene et al. [11]	6	SVM-based classifiers	-		✓			✓	Homogeneity	GA
2009	Nanni et al. [36]	25	Random Subspace with KNN(K=1)	Majority Vote					✓	Homogeneity	GA
2012	Kim et al. [28]	vary	Bagging and Boosting with DT, MLP or SVM	Majority Vote	✓					Homogeneity	GA
2013	Shunmugapriya et al. [6]	10	NB, Logistic, IB1, IBk(k=5), KStar, OneR, PART, ZeroR, DStump, J48 DT	NB, Logistic, IB1, IBk (k=5), KStar, OneR, PART, ZeroR, DStump, J48 DT	✓		✓			Heterogeneity	ABC
2014	Nguyen et al. [5]	3	LDA, NB, KNN(K=5)	One in set of (Sum, Majority Vote, Median, Max, Min, Product)	✓	✓				Heterogeneity	GA
2014	Nguyen et al. [17]	3	LDA, NB, KNN(K=5)	Sum, Majority Vote, Median, Max, Min, Product	✓		✓			Heterogeneity	GA*
2014	Nguyen et al. [7]	3	LDA, NB, KNN(K=5)	GMM				✓		Heterogeneity	GA
2014	Nguyen et al. [8]	3	LDA, NB, KNN(K=5)	DT				✓		Heterogeneity	GA
2014	Chen et al. [29]	10	NB, Logistic, IB1, IBk(k=5), KStar, OneR, PART, ZeroR, DStump, C4.5 DT	DT or one in set of (NB, Logistic, IB1, IBk (k=5), KStar, OneR, PART, ZeroR, DStump, C4.5 DT)	✓		✓			Heterogeneity	ACO
2015	Mendialdua et al. [30]	10	1R, KNN, RIPPER, Naive Bayes, C4.5 DT, KStar, Bayesian Networks, NB Tree, RF, SVM	One in set of (1R, KNN, RIPPER, NB, C4.5 DT, KStar, Bayesian Networks, NB Tree, RF, SVM)	✓					Heterogeneity	EDA
2015	Mousavi et al. [31]	46	See the detail in [31]	Sum, Majority Vote, Median, Max, Min, Product	✓		✓			Heterogeneity	NSGA-II*
2015	Ali et al. [9]	4	KNN, SVM, NB, RF	Predictor Function-based GP					✓	Heterogeneity	GP and PSO
2016	Haque et al. [32]	20	See the detail in [32]	Majority Vote	✓					Heterogeneity	GA
2016	Padilha et al. [37]	30	Feature Selection with Least Squares SVM	Weight-based average	✓	✓			✓	Homogeneity	GA
2019	Wang et al. [33]	100	Random Subspace and Bootstrap with Regression Tree	RF	✓					Homogeneity	NSGA-II*
2019	Nguyen et al. [34]	3 and 5	LDA, NB, KNN(K=5), DT, NMC	LDA, NB, KNN(K=5), DT, NMC			✓	✓		Heterogeneity	ACO

LDA: Linear Discriminant Analysis, QDA: Qua Discriminant Analysis, SVM: Support Vector Machine, RBF: Radial Basis Function, KNN: K-Nearest Neighbor, MLP: Multi-Layer Perceptron, SASOM: Structure Adaptive Self-Organizing Map, DT: Decision Tree, NB: Naïve Bayes, RF: Random Forest, NMC: Nearest Mean Classifier, GMM: Gaussian Mixture Model, DStump: Decision Stump
GA: Genetic Algorithm, ABC: Artificial Bee Colony, ACO: Ant Colony Optimization, EDA: Estimation of Distribution Algorithm, NSGA-II: Non-dominated Sorting Genetic Algorithm, GP: Genetic Programming;
* means that the multi-objective optimization was considered

to learn the base classifiers. The optimal solution then is obtained via GA. In the extended version, the chromosomes in GA have encoded the weight for each feature-classifier pair for the weighted sum combining rule. Bacauskiene et al. [11] built an ensemble system of Support Vector Machine (SVM)-based classifiers in which the parameters of each classifier, i.e. the regularization constant and the kernel width, are encoded via the binary encoding scheme. The features used by each classifier are also encoded in the same chromosome. The optimal solution containing the optimal parameters for each classifier and the optimal features used by each of them is obtained after several generations of GA. Padilha et al. [37] encoded base classifiers, base features, parameters of SVM classifiers, and the weight of each classifier in a single chromosome to conduct the selection and adaptation simultaneously.

3. Proposed Method

a. Formulation

As discussed in Section 2, the Decision Template method has limitations by using the single points for the representation and has low discriminative ability on imbalanced datasets. In fact, there are approaches introducing new representations to overcome these weaknesses. For example, [38] introduced a new decision template based on clustering techniques. In detail, a partition ensemble was generated based on the concatenation between the ground truth-based partition and the data partitions outputted by a clustering ensemble on the meta-data. A consensus partition then was built by measuring the weight-based similarity between the partition ensemble and the set of all possible partitions for the meta-data. New decision templates were computed as the centroids of the consensus partition. In this study, we introduce a different approach to overcome these disadvantages of the Decision Template method while inheriting its simplicity. Instead of using the mean value as the representation for each class label, we propose using the interval-based vector as a representation for each class label. It is noted that each learning algorithm uses different approaches to train the base classifiers on the training data, thus introducing uncertainty (i.e. the agreement and disagreement in predictions) in the classifiers' output. On the one hand, several combining methods like the Decision Template method use the precise value built from the supports of classifiers to model the data. In many practical situations, a precise value may not reflect every event in the data, especially when only limited information is available [14]. On the other hand, it is widely recognized that an interval-based model is a suitable tool in representing and handling uncertain knowledge [14]. For applications involving probabilistic and statistical reasoning, the interval models have shown many successes, especially when they are applied to handle the conflict between different sources of information [15].

On each test instance, we compute the distance between its meta-data and the vector of intervals (representation). The class label associated with the shortest distance is assigned to the test instance. By

using the interval vector, we provide a more general representation of the combining method. There are two questions concerning the proposed method:

- How to construct the interval-based vector for the class label representation?
- How to compute the distance between meta-data of a test instance and the vector of intervals?

In detail, we generate the set of base classifiers $\boldsymbol{\mathcal{H}}$ by learning K algorithms on the training set $\mathcal{D}$. The meta-data of the training observations is also generated using the T-fold cross validation. The training set $\mathcal{D}$ is divided into T disjoined parts $\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_T$ in which the number of observations in each part is nearly equal. For $\mathcal{D}_i$, the K algorithms learn on $\widetilde{\mathcal{D}}_i = \mathcal{D} - \mathcal{D}_i$ to obtain classifiers which are used to predict for observations in $\mathcal{D}_i$. In this way, each observation in $\mathcal{D}$ has a unique prediction from each classifier. The meta-data of $\mathcal{D}$ denoted by $\mathbf{L}$ is finally obtained by concatenating all meta-data from each $\mathcal{D}_i$ in a $N \times MK$ matrix (3):

$$\mathbf{L} = \begin{bmatrix} s_{1,1}(\mathbf{x_1}) \ldots s_{1,M}(\mathbf{x_1}) & \ldots & s_{K,1}(\mathbf{x_1}) \ldots s_{K,M}(\mathbf{x_1}) \\ \vdots & \ddots & \vdots \\ s_{1,1}(\mathbf{x_N}) \ldots s_{1,M}(\mathbf{x_N}) & \ldots & s_{K,1}(\mathbf{x_N}) \ldots s_{K,M}(\mathbf{x_N}) \end{bmatrix} \quad (3)$$

Meanwhile, the meta-data of one observation $\mathbf{x}_i$ is given in a $M \times K$ vector of the supports of the base classifiers:

$$\mathbf{L}(\mathbf{x}_i) = [s_{1,1}(\mathbf{x_i}) \ldots s_{1,M}(\mathbf{x_i})\, s_{2,1}(\mathbf{x_i}) \ldots s_{2,M}(\mathbf{x_i}) \ldots \ldots \ldots \; s_{K,1}(\mathbf{x_i}) \ldots s_{K,M}(\mathbf{x_i})] \quad (4)$$

In this study, we propose to use intervals instead of average to represent the meta-data associated with each of the class labels. As mentioned before, different base classifiers would output different supports (possibly with disagreement) for an instance. The meta-classifier, therefore, needs to handle this disagreement, which can be done using interval-based representation [14]. In addition, in many practical situations, observation may be collected under noisy conditions and therefore cannot be associated with a precise value. In this case, intervals provide a flexible way to describe the uncertainty of the underlying knowledge [15]. The proposed interval-based representation $\boldsymbol{\mathcal{R}} = \{\mathbf{R_j}\}$ is given by:

$$\mathbf{R_j} = \left\{ \left[\underline{s_{1,1}^{(j)}(.)}, \overline{s_{1,1}^{(j)}(.)} \right] \ldots \left[\underline{s_{1,M}^{(j)}(.)}, \overline{s_{1,M}^{(j)}(.)} \right] \ldots \ldots \ldots \left[\underline{s_{K,1}^{(j)}(.)}, \overline{s_{K,1}^{(j)}(.)} \right] \ldots \left[\underline{s_{K,M}^{(j)}(.)}, \overline{s_{k,M}^{(j)}(.)} \right] \right\} \quad (5)$$

in which $\overline{s_{k,m}^{(j)}(.)}$ and $\underline{s_{k,m}^{(j)}(.)}$ are the upper and lower bound of the interval-based representation for the training instances in the j^{th} class obtained from the outputs of base classifier k for its prediction for class label m.

By this definition, $\mathbf{R_j}$ is a vector of MK intervals. It is noted that the bounds of each interval satisfy the constraint:

$$0 \le \underline{s_{k,m}^{(j)}(.)} \le \overline{s_{k,m}^{(j)}(.)} \le 1 \; ; k = 1, \ldots, K; m = 1, \ldots, M \quad (6)$$

The vector of intervals will be used as the representation of the class labels. As mentioned before, an instance is assigned to a class based on the closeness between its meta-data and the elements in $\boldsymbol{\mathcal{R}}$. It is noted that the meta-data of an instance is a numeric vector given by (4) whereas each element of the representation $\boldsymbol{\mathcal{R}}$ is a set of a vector of intervals. It is recognized that the distance between a numerical vector and a vector of intervals is a special case of the distance between two interval vectors where one of the vectors has intervals with the same upper bound and lower bound.

In fact, several kinds of distance can be defined for interval data. Irpino and Varde [39] grouped these distances into three groups, namely component-wise approach (which considers some aspects of interval data such as position, span, and content), fuzzy oriented approach (which treats interval data by using fuzzy numbers), and boundary approach (which considers the bounds of interval data only). In this study, we are inspired by the distance between two multivariate intervals defined in [40, 41] which is a sum of the distances between each interval member. In this way, the distance between the vector of meta-data $\mathbf{L}(\mathbf{x}_i)$ and a vector of intervals $\mathbf{R}_{\boldsymbol{j}}$ is given by:

$$\mathbf{d}\big(\mathbf{L}(\mathbf{x}_i), \mathbf{R}_{\boldsymbol{j}}\big) = \sum_{k=1,\ldots,K;m=1,\ldots,M} d\left(s_{k,m}(\mathbf{x}_{\boldsymbol{i}}), \left[\underline{s_{k,m}^{(j)}(.)}, \overline{s_{k,m}^{(j)}(.)}\right]\right) \tag{7}$$

To calculate the distances $d\left(s_{k,m}(\mathbf{x}_{\boldsymbol{i}}), \left[\underline{s_{k,m}^{(j)}(.)}, \overline{s_{k,m}^{(j)}(.)}\right]\right)$ in (7), many formulations for the distance between two intervals can be used. For example, in [40], the distance between two intervals was defined by the sum of three distances concerning the position, span, and content of interval. In this study, we are inspired by the Hausdorff distance between two sets [39] and the distance between two intervals introduced in [42], i.e. $d([x_1, x_2], [a, b]) = \max\{|x_1 - a|, |x_2 - b|\}$, in calculating the distance between a numerical value and an interval. With a note that a numerical value x can be treated as a special interval $x = [x, x]$ [42], the distance between $s_{k,m}(\mathbf{x}_{\boldsymbol{i}})$ and $\left[\underline{s_{k,m}^{(j)}(.)}, \overline{s_{k,m}^{(j)}(.)}\right]$ is given by:

$$d\left(s_{k,m}(\mathbf{x}_{\boldsymbol{i}}), \left[\underline{s_{k,m}^{(j)}(.)}, \overline{s_{k,m}^{(j)}(.)}\right]\right) = \max\left\{\left|s_{k,m}(\mathbf{x}_{\boldsymbol{i}}) - \underline{s_{k,m}^{(j)}(.)}\right|, \left|s_{k,m}(\mathbf{x}_{\boldsymbol{i}}) - \overline{s_{k,m}^{(j)}(.)}\right|\right\} \tag{8}$$

Clearly, the distance in (8) belongs to the boundary approach which only considers two bounds of the interval.

We now introduce an approach to search for the optimal representation from the meta-data and the ground truth of the training set. Assume that we have an arbitrary representation $\boldsymbol{\mathcal{R}}$, we compute the distances between $\mathbf{R}_{\boldsymbol{j}}$ and the meta-data of each training observation. The predicted label is assigned to the shortest one among all these distances. Since the ground truths of all training observations are known in advance, the empirical 0-1 loss function $\mathcal{L}_{0-1}$ computed on a training observation $\mathbf{x}_i$ is given by:

$$\mathcal{L}_{0-1}(\mathbf{x}_i, \boldsymbol{\mathcal{R}}) = \mathbb{I}\left[\arg\min_{y_j, j=1,\ldots,M} \left\{\mathbf{d}\big(\mathbf{L}(\mathbf{x}_i), \mathbf{R}_{\boldsymbol{j}}\big)\right\} \neq \hat{y}_i\right] \tag{9}$$

The empirical 0-1 loss function on the training data is given by:

$$\boldsymbol{\mathcal{L}}_{0-1}(\boldsymbol{\mathcal{R}}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{0-1}(\mathbf{x}_i, \boldsymbol{\mathcal{R}}) \tag{10}$$

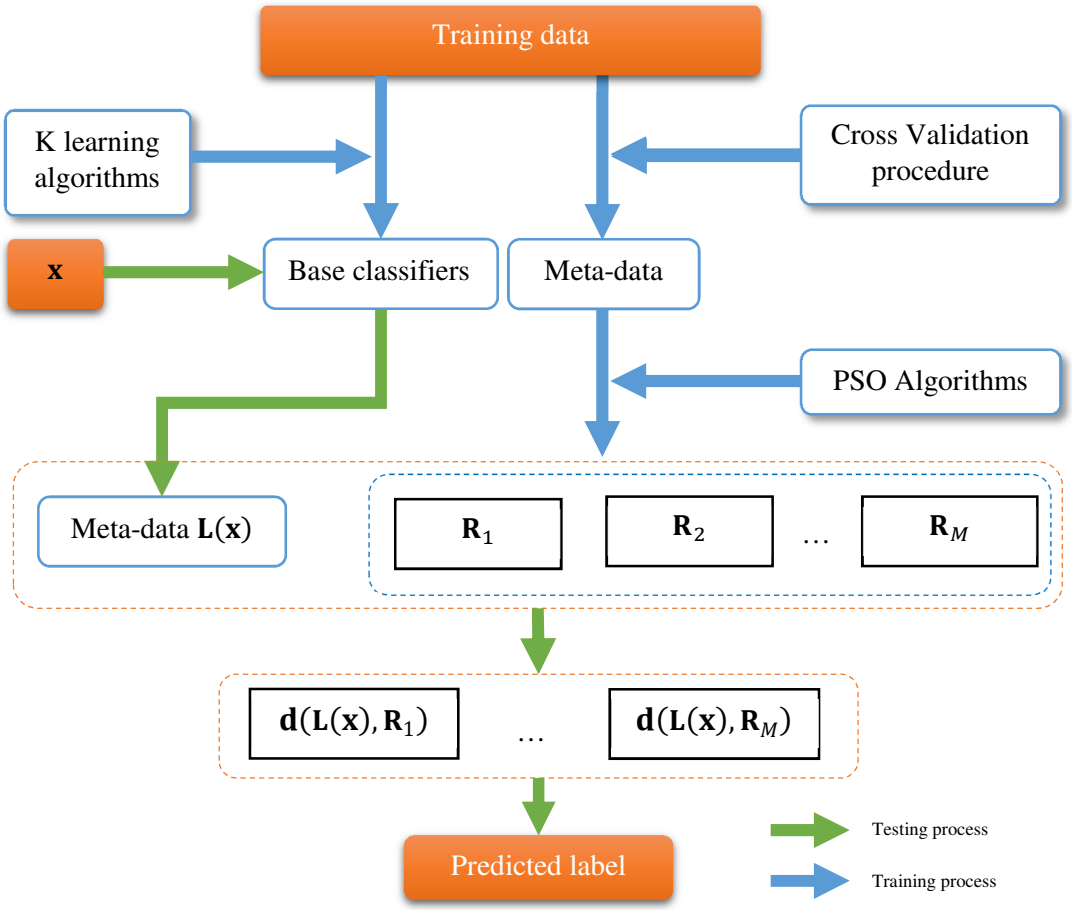


Fig.2. The illustration for the proposed method

The optimal representation is obtained by minimizing the loss function (10) subjects to the constraint in (6). The optimization problem is given by:

$$\begin{cases} \min_{\boldsymbol{\mathcal{R}}} \boldsymbol{\mathcal{L}}_{0-1}(\boldsymbol{\mathcal{R}}) \\ \text{subject to } 0 \le \underline{s_{k,m}^{(j)}(.)} \le \overline{s_{k,m}^{(j)}(.)} \le 1; k = 1, \dots, K; m = 1, \dots, M; j = 1, \dots, M \end{cases} \tag{11}$$

in which $\boldsymbol{\mathcal{R}} = \{\mathbf{R}_j\}$ and $\mathbf{R}_j$ is given in (5). After solving (11) by an optimization method, we obtain the optimal representation $\breve{\boldsymbol{\mathcal{R}}} = \{\breve{\mathbf{R}}_j\}(j = 1, \dots, M)$ for the class labels. During classification for a test instance $\mathbf{x}$, we compute its meta-data $\mathbf{L}(\mathbf{x})$ in (4) by classifying $\mathbf{x}$ with the K base classifiers in $\boldsymbol{\mathcal{H}}$. The class label is assigned to $\mathbf{x}$ by calculating the distance between $\mathbf{L}(\mathbf{x})$ and the optimal representation $\breve{\mathbf{R}}_j$ $(j = 1, \dots, M)$ and then picking the one associated with the smallest distance.

$$\mathbf{x}_i \in y_t \text{ if } y_t = \arg\min_{y_j, j=1,\dots,M} \{\mathbf{d}(\mathbf{L}(\mathbf{x}_i), \breve{\mathbf{R}}_j)\} \tag{12}$$

b. Optimization

In this study, we use the Particle Swarm Optimization (PSO) algorithm to search for the optimal bounds of the intervals in the representation. PSO is a stochastic population-based algorithm originally introduced by Kennedy and Eberhart [12] inspired by the emergent motion of a flock of birds searching for food. PSO has some advantages in comparison to other optimization techniques. It is well-known that PSO is well suited to handle non-linear, non-convex spaces with non-differentiable, discontinuous objective functions. PSO can work with diverse types of variables including continuous, discrete and integer types. In comparison to other evolutionary computation-based optimization methods, PSO requires fewer number of function evaluations, while it can obtain better or similar quality of solutions [43]. Especially, PSO is more effective than GA when solving unconstrained problems with continuous variables [44]. PSO is also effective in exploitation as it can employ the knowledge of the previous good solution to find better solutions. In contrast, other biological-inspired optimization algorithms like the original Artificial Bee Colony is poorer than PSO in term of this ability [45]. Finally, PSO can be efficiently parallelized to reduce computational cost.

There are some variations related to the PSO algorithm that needs to be considered. First, modifications can be made on the PSO algorithm to handle the optimization problem with constraints. The most straightforward way is to keep tracking the feasible solutions only while the particles search the whole space like in the original algorithm [46]. A penalty term can also be added to the objective function value when a position violates the constraints so that the positions do not have any opportunities to become the best position. Parsopoulos and Vrahatis [47] defined the penalty term as the product of a function of iteration number and the sum of multi-stage assignment function of violation of each constraint. Perez and Behdinan [43] introduced the penalty scheme with the attention of the average of the objective function in the current swarm and the violation of each constraint averaged over the current population. Besides, experiments show that the three parameters of PSO, i.e. the inertia weight w which is used to balance the abilities of global and local search, the social and cognitive attraction C_1 and C_2 which indicates how much confidence one candidate has in the swarm or in itself respectively, may have a significant influence on the efficiency and reliability of the PSO algorithm [48]. In the literature, some suggested values for these parameters have been introduced so as to improve the effectiveness of the search procedure. For inertia weight, there are three strategies to determine its value through the iterations: setting it as a constant value and selecting it randomly, changing it with iteration number, and revising it using feedback parameters as summarizing in [49]. In this study, we use the second strategy in which w is updated at the t^{th} iteration as [50]:

$$w = w_{max} - \frac{(w_{max} - w_{min})t}{maxT} \tag{13}$$

Meanwhile, there are also different selections for the social and cognitive attraction when dealing with different optimization problems. In the next section, we examine the influence of different values of (C_1, C_2) on the performance of the proposed method on the experimental datasets.

The training phase of the proposed method is presented in Algorithm 1 and Algorithm 2. Given the training set $\mathcal{D}$, we learn the set of classifiers $\boldsymbol{\mathcal{H}}$ by using K learning algorithms $\{\mathcal{K}_k\}$ (step 1). T-fold cross validation is then used to generate the meta-data of $\mathcal{D}$ (step 2-9). After that, we use PSO algorithm to search for the optimal interval-based representation (step 10). In each population, PSO generates and evaluates the fitness of $nPop$ candidates. Specifically, for each training observation, we compute the distance between its meta-data and each $\mathbf{R}_j$ (vector of intervals) in the candidate $\boldsymbol{\mathcal{R}}$ (step 3 in Algorithm 2). We then compute the value of the loss function of each training observation (step 5 in Algorithm 2) and on the whole training set (step 7 in Algorithm 2). At the end of the PSO algorithm, we reached the optimal solution $\breve{\boldsymbol{\mathcal{R}}}$.

In the classification phase in Algorithm 3, the output of the base classifiers $\boldsymbol{\mathcal{H}}$ on an unlabeled instance $\mathbf{x}$ is obtained as $\mathbf{L}(\mathbf{x})$ (step 1). We then computed the distance between $\mathbf{L}(\mathbf{x})$ and each of $\breve{\mathbf{R}}_j$ of the optimal representation $\breve{\boldsymbol{\mathcal{R}}}$ (step 2-4). Based on the classification rule in (12), we assign class label to $\mathbf{x}^u$ using the shortest distance criterion (step 5).

Algorithm 1: Training phase

Input: Training set $\mathcal{D}$, K learning algorithms $\{\mathcal{K}_k\}$, maximum number of iteration: $maxT$, population size: $nPop$, social attraction: C_1, cognitive attraction: C_2.

Output: The optimal representation of $\breve{\boldsymbol{\mathcal{R}}}$ and $\boldsymbol{\mathcal{H}}$

(Generate the base classifier)

1. Learn K classifiers $\boldsymbol{\mathcal{H}}$ on $\mathcal{D}$ using $\{\mathcal{K}_k\}, k = 1, \ldots, K$

(Generate the meta-data)

2. Meta-data $\mathbf{L} = \emptyset$
3. $\mathcal{D} = \mathcal{D}_1 \cup \ldots \cup \mathcal{D}_T, \quad \mathcal{D}_i \cap \mathcal{D}_j = \emptyset \ (i \neq j)$
4. For each $\mathcal{D}_i$
5. $\quad \widetilde{\mathcal{D}}_i = \mathcal{D} - \mathcal{D}_i$
6. $\quad$ Learn ensemble of classifiers on $\widetilde{\mathcal{D}}_i$ using $\{\mathcal{K}_k\}$
7. $\quad$ Classify instances of $\mathcal{D}_i$ by these classifiers
8. $\quad$ Add outputs on instances in $\mathcal{D}_i$ to $\mathbf{L}$ (3)
9. End

10. Use the PSO method:
For each generated candidate $\boldsymbol{\mathcal{R}}$, compute $\mathcal{L}_{0-1}(\boldsymbol{\mathcal{R}})$ using Algorithm 2
Select the optimal $\breve{\boldsymbol{\mathcal{R}}}$ with the smallest fitness at the end
11. Return $\breve{\boldsymbol{\mathcal{R}}}$ and $\boldsymbol{\mathcal{H}}$

Algorithm 2: Compute the loss value for each candidate generated in PSO algorithm

Input: candidate $\boldsymbol{\mathcal{R}}$

Output: The loss value for $\boldsymbol{\mathcal{R}}$

1. For each $\mathbf{x}_i \in \mathcal{D}$
2. For each $\mathbf{R}_j$ in $\boldsymbol{\mathcal{R}}$
3. Compute distance between $\mathbf{L}(\mathbf{x}_i)$ and $\mathbf{R}_j$ (7)
4. End
5. Compute $\mathcal{L}_{0-1}(\mathbf{x}_i, \boldsymbol{\mathcal{R}})$ by (9)
6. End
7. Compute and return $\mathcal{L}_{0-1}(\boldsymbol{\mathcal{R}})$ by (10)

Algorithm 3: Classification phase

Input: Unlabeled instance $\mathbf{x}$, the optimal representation of $\breve{\boldsymbol{\mathcal{R}}} = \{\breve{\mathbf{R}}_j\}$ and $\boldsymbol{\mathcal{H}}$

Output: Predicted class label for $\mathbf{x}$

1. Obtain the meta-data $\mathbf{L}(\mathbf{x})$ by using $\boldsymbol{\mathcal{H}}$ (4)
2. For each $\breve{\mathbf{R}}_j$
3. Compute distance between $\mathbf{L}(\mathbf{x})$ and $\breve{\mathbf{R}}_j$ (7)
4. End
5. Assign the class label for $\mathbf{x}$ by (12)

4. Experimental Studies

a. Baselines and Settings

We used three learning algorithms, namely LDA, Naïve Bayes, and kNN$_5$, to construct the heterogeneous ensemble. For the PSO algorithm, the maximum number of iterations $maxT$ was set to

100, the number $nPop$ was set to 50 as in the experiments in [7, 8]. Meanwhile, w_{max} and w_{min} in (15) were set to 0.9 and 0.5, respectively.

We compare the classification accuracy and the F1 measure of the proposed method to some benchmark algorithms. On a data file, we used the first 10-fold cross validation procedure to create the training and testing data. The first cross validation procedure was run 3 times to obtain 30 trials in each data file. The mean and variance of the classification accuracy and F1 measure for the 30 results of each method on each data file were computed and reported. We use another 10-fold cross validation on the training data to generate its meta-data (Step 2-9 in Algorithm 1). The fitness value of each individual in the PSO algorithm, i.e. the value of the 0-1 loss function was evaluated on the meta-data by using (9). The details of the experimental procedure can be found in Figure S1 in the Supplement Material.

We selected seven ensemble systems as the benchmark algorithms. The details of the seven benchmark algorithms are:

- Two well-known heterogeneous ensemble methods namely Decision Template [13] and Sum Rule [16]. For these methods, we also used 3 learning algorithms, i.e. LDA, Naïve Bayes, and kNN$_5$, to generate the base classifiers.
- Two EA-based approaches for ensemble selection optimization, namely ACO [29] and GA Meta-data [8]: we also used 3 learning algorithms, i.e., LDA, Naïve Bayes, and kNN$_5$ to generate the base classifiers and the C4.5 Decision Tree was trained on the meta-data to generate the meta-classifier like in the original papers.
- One dynamic ensemble selection method named META-DES [51]. This method aims to select a subset of base classifiers for each test instance based on their performance on the neighbors of the test instances getting from the training set. META-DES obtains top performance among many dynamic ensemble selection methods in experiments.
- Two ensemble methods using L2-loss Linear Support Vector Machine (denoted by L2LSVM) and multi-layer neural network (denoted by MLNN) as the combining algorithms. For these methods, we also used 3 learning algorithms i.e. LDA, Naïve Bayes, and kNN$_5$ to generate the base classifiers. L2-loss Linear Support Vector Machine (denoted by L2LSVM) is a method developed from the traditional Support Vector Machine for large scale datasets [52]. We used L2LSVM from the package LIBLINEAR (https://www.csie.ntu.edu.tw/~cjlin/liblinear/). For MLNN, since it is widely recognized that its performance depends on the configuration, we experimented on a wide range of parameter values and then reported the best results for the comparison. In detail, the number of hidden layers was set to 2 while the size of the hidden layer was chosen from the set of {20, 30, 50, 70, 100}. We used MLNN from the Pattern recognition network package of Matlab.

b. Statistical Test of Significance

The proposed method was compared to the benchmark algorithms in terms of classification accuracy and F1 measure. We conducted statistical tests to evaluate the difference between the performance of the proposed method and the benchmark algorithms. In this study, we used two different statistical tests for the comparison purpose. The first test is to compare the performance of all experimental methods on all experimental datasets. Here the Friedman test was used to test the null hypothesis that "all methods perform equally". If the null hypothesis is rejected, Nemenyi post-hoc test was used to compare all pairwise combinations of the experimental methods. The level of significance was set to 0.05.

In the second test, we aim to compare performance of the proposed method and each benchmark algorithm on a specific dataset. The Wilcoxon signed-rank test [53] was used with the null hypothesis that "two methods perform equally on the dataset". 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 α (set to 0.05). We used the software package provided in the Matlab library for the Wilcoxon signed-rank test.

c. Datasets

All methods are run on the 29 datasets extracted from UCI (https://archive.ics.uci.edu/ml/index.php) and the GM4 dataset from [3]. The datasets were selected to ensure an objective comparison: diverse in the number of observations, the number of class labels, and the number of features. Table 3 presents a description of the datasets used in this analysis.

5. Results and Discussions

a. Evaluations under different values of parameters

We first evaluate the influence of the two parameters of the PSO algorithm, i.e. social attraction C_1 and cognitive attraction C_2, on the performance of the proposed method. As mentioned in section 3, there are different choices for the values of these parameters. In this study, we examine C_1 and C_2 from the set $\{0, 0.2, 0.4, \dots, 1.8, 2\}$. The relationships between (C_1, C_2) and the classification accuracy and F1 score on 10 datasets are shown in Fig 3 and 4, respectively. Clearly, these parameters only have a slight effect on the performance of the proposed method except when C_1 is set to 0, as the classification accuracy and F1 score only change slightly with the changes of parameters' values. For example, the classification accuracy varies only by 3.1% on Artificial (0.7519 to 0.7829), 1.13% on Breast-Cancer (0.9565 to 0.9678), and 1.44% on Iris (0.9678 to 0.9822) with the changes of C_1 and C_2. A similar pattern is found in the figures for F1 score. For instance, F1 score only shows small changes on Artificial (3.79%, from 0.7371 to 0.7705) and Iris (2.53%, from 0.9569 to 0.9820). For datasets such as Hayes-Roth and Balance, the proposed method performs poorly when $C_1 = 0$, meanwhile, its performance is

nearly similar to other values. Since the performance of the proposed method depends only slightly on the parameters of the PSO algorithm and its optimal value is somewhat data-dependent, we choose the value $C_1 = C_2 = 1.494$ when comparing to the benchmark algorithms. These values are also suggested in [48, 49].

Table 3. The experimental datasets

Dataset name	# of observations	# of classes	# of dimension
Appendicitis	106	2	7
Artificial	700	2	10
Balance	625	3	4
Banana	5300	2	2
Biodeg	1055	2	41
Blood	748	2	4
Breast-Cancer	683	2	9
Cleveland	297	5	13
Contraceptive	1473	3	9
Dermatology	358	6	34
Fertility	100	2	9
GM4	1000	3	1000
Haberman	306	2	3
Hayes-Roth	160	3	4
Heart	270	2	13
Hill-Valley	2424	2	100
Iris	150	3	4
Led7digit	500	10	7
Madelon	2000	2	500
Magic	19020	2	10
Musk2	6598	2	166
Newthyroid	215	3	5
Ring	7400	2	20
Skin_NonSkin	245057	2	3
Spambase	4601	2	57
Twonorm	7400	2	20
Vehicle	846	4	18
Waveform_w_Noise	5000	3	40
Waveform_wo_Noise	5000	3	21
Wdbc	569	2	30

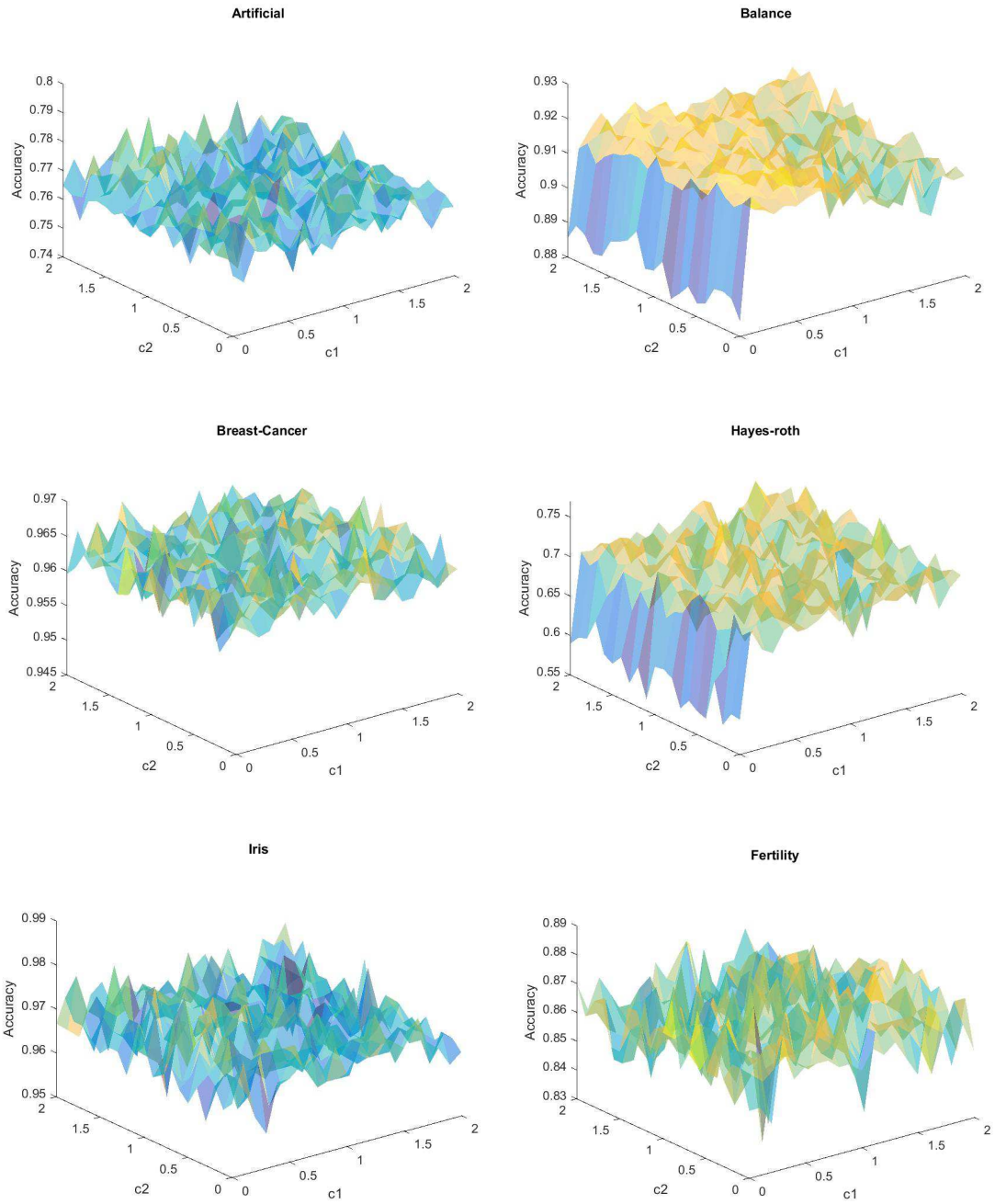


Fig. 3. Relationship between social and cognitive attraction parameters and classification accuracy

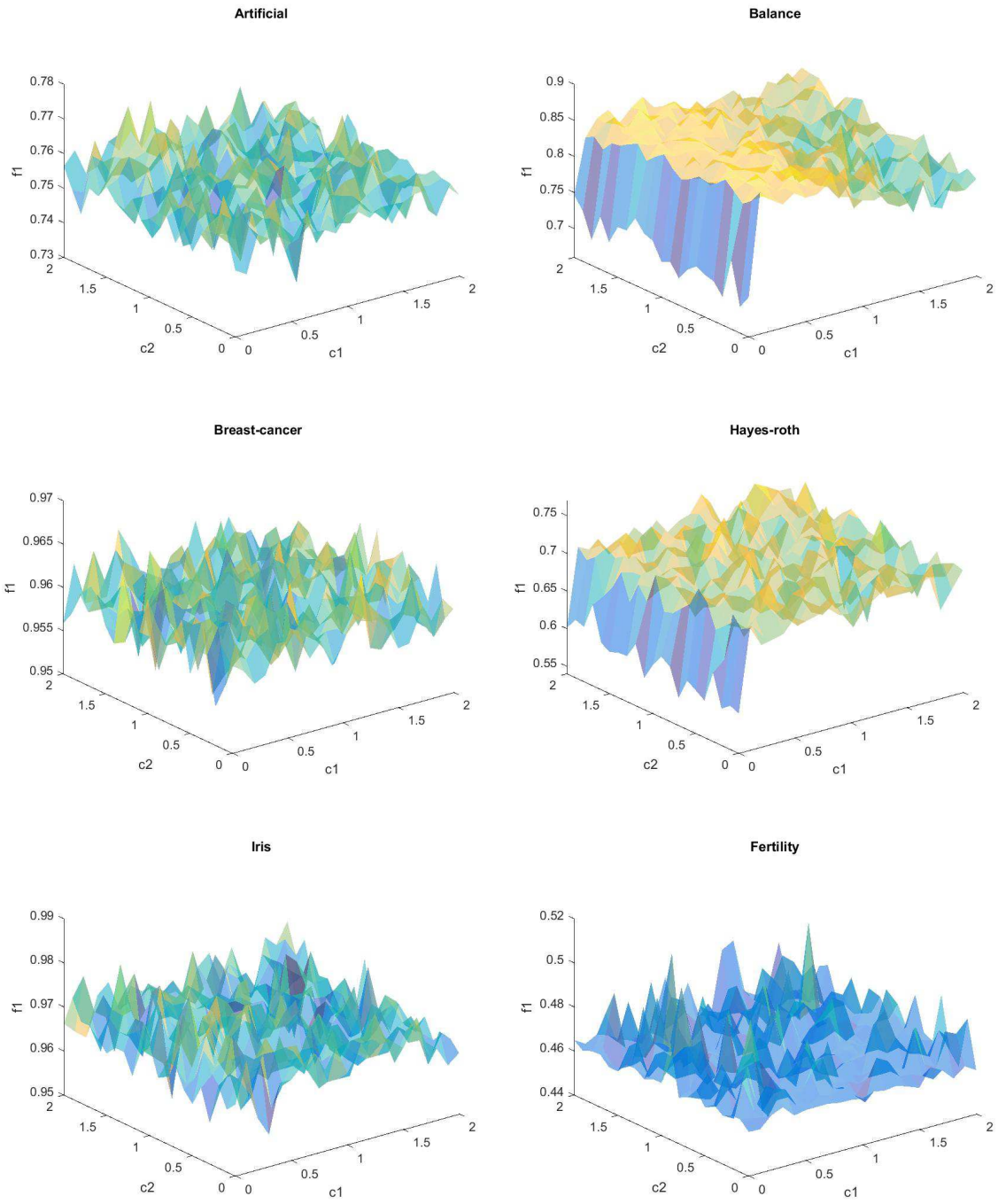


Fig. 4. Relationship between social and cognitive attraction parameters and F1 score

b. Comparison based on classification accuracy

The mean and variance of the classification accuracy of the proposed method and the benchmark algorithms are shown in Table 4. First, based on the Friedman test, we rejected the null hypothesis that all methods perform equally. We, therefore, conducted the Nemenyi post-hoc test to compare all pairwise combinations of all methods. The results of this test are shown in Fig 5. The post-hoc test

shows that the proposed method is better than META-DES, GA Meta-data, and ACO. Concerning the ranking computed on the classification accuracy, the proposed method is better than all benchmark algorithms, ranking first with rank value 2.97. Ensemble (L2LSVM) is the second-best method with a rank value of 3.23. GA Meta-data and ACO are the two poorest methods in the experiment, ranked at values 5.42 and 5.53, respectively. The detail can be found in Table S3 in the Supplement Material.

Fig 6 shows the results of Wilcoxon signed-rank test when comparing the proposed method and each benchmark algorithm. Clearly, the proposed method is competitive to Ensemble (L2LSVM) while it performs better than the other benchmark algorithms on the experimental datasets. In comparison to META-DES, the proposed method is better on 14 datasets while worse only on 4 datasets. On the two datasets where the proposed method is poorer than META-DES, the difference of classification accuracy is only significant on the Hill-Valley (0.7856 vs. 0.7076) and Ring (0.9787 vs. 0.8923) dataset. Meanwhile, on 14 datasets that our method is better than META-DES, some differences are very significant, for example, on Cleveland (0.5774 vs. 0.6184), Madelon (0.6545 vs. 0.7110), and Vehicle (0.7226 vs. 0.7762). The proposed method also outperforms Sum Rule, winning on 18 datasets and losing on only 2 datasets. On the two datasets that our method loses, i.e. Appendicitis and Artificial, the differences in classification accuracy are only 2.21% and 2.57%, respectively. The proposed method performs significantly better than the Decision Template method since ours does not lose on any datasets while wins on 14 datasets. On datasets such as Biodeg and Breast-Cancer, the classification accuracy of the proposed method is around 1% higher than that of the Decision Template method. On datasets such as Cleveland, Blood, and Haberman, the proposed method is nearly 4% better than the Decision Template method. Especially, on Hayes-Roth, Ring, and Fertility, the proposed method significantly outperforms the Decision Template method and the differences in classification accuracy of the two methods are 12.3%, 8.55%, and 28%, respectively.

The proposed method is better than Ensemble (MLNN), winning on 8 datasets and losing on 2 datasets. In the cases where ours are poorer than Ensemble (MLNN), the classification accuracies are not too much different, 0.7771 vs. 0.7614 on the Artificial dataset and 0.9970 vs. 0.9840 on the GM4 dataset. On the other hand, on the datasets where the proposed method is better than Ensemble (MLNN), the differences in the classification accuracies are significant, for example, 0.8509 vs. 0.8730 on the Biodeg dataset, 0.7540 vs. 0.7746 on the Blood dataset, and 0.9278 vs. 0.9683 on the Wdbc dataset. Finally, the proposed method is competitive to Ensemble (L2LSVM) as the win/loss ratio is 5/4. In the cases where the Wilcoxon test shows significant differences, the differences in classification accuracy of the two methods are from 0.5% to 1.5%. These outstanding results show the superiority of the proposed method on the experimental datasets.

We compare the proposed method to two selection methods namely GA Meta-data and ACO. Once again, our method is significantly better than both selection methods since it wins on 15 and 17 datasets

for GA Meta-data and ACO, respectively. In detail, GA Meta-data and ACO are better than the proposed method on 2 datasets, namely GM4 (1 and 1 vs. 0.9840) and Newthyroid (0.9629 and 0.9582 vs. 0.9408). In contrast, we achieved significantly better results than GA Meta-data and ACO on 15 and 17 datasets, respectively. Significant differences are found on Blood (0.7180 of ACO vs. 0.7746 of the proposed method), Contraceptive (0.4764 of GA Meta-data vs. 0.5347 of the proposed method), and Heart (07605 of GA-Meta-data vs. 0.8272 of the proposed method).

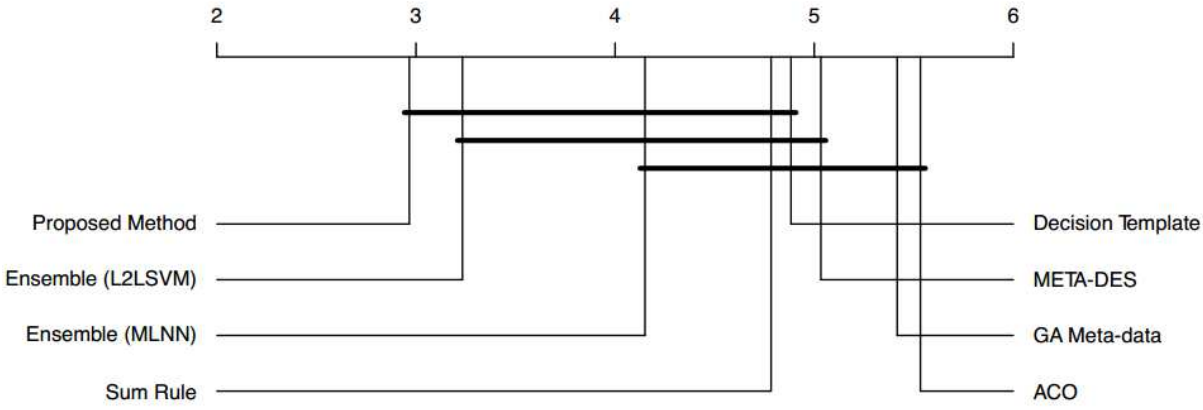


Fig. 5. Results of Nemenyi post-hoc test concerning classification accuracy

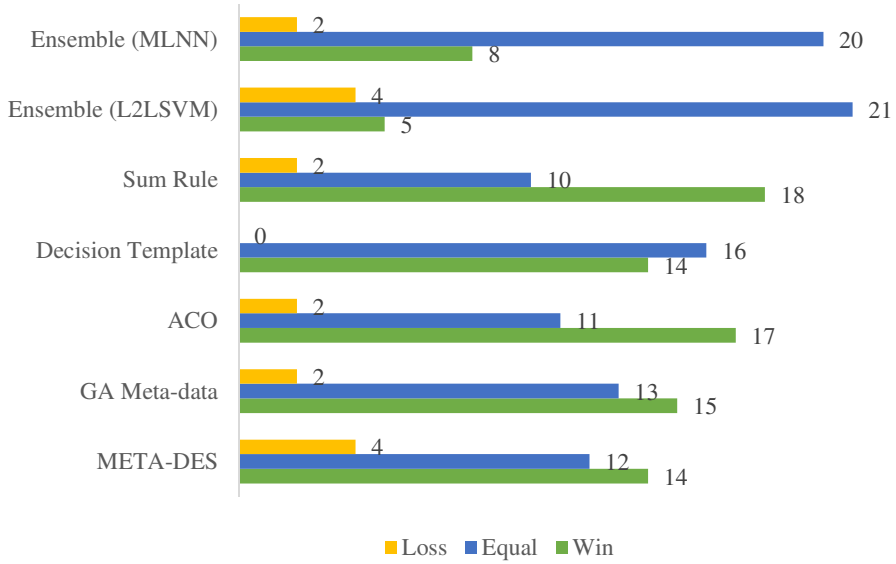


Fig.6. Results of Wilcoxon signed-rank test concerning classification accuracy

Table 4. Mean and standard deviation of classification accuracy of proposed method and benchmark algorithms

Dataset	META-DES		GA Meta-data		ACO		Decision Template		Sum Rule		Ensemble (MLNN)		Ensemble (L2LSVM)		Proposed Method	
	Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Appendicitis	0.8770▼	0.0927	0.8400	0.1020	0.8173	0.1229	0.8703	0.0908	0.8803 ▼	0.0940	0.7906	0.1852	0.8682	0.0924	0.8582	0.1034
Artificial	0.7638	0.0464	0.7705	0.0489	0.7743	0.0491	0.7557	0.0402	0.7871 ▼	0.0327	0.7771▼	0.0362	0.7557▲	0.0387	0.7614	0.0410
Balance	0.8875▲	0.0240	0.9148	0.0332	0.9040	0.0285	0.9040	0.0306	0.8901 ▲	0.0221	0.9190	0.0349	0.8912▲	0.0232	0.9072	0.0291
Banana	0.8875	0.0096	0.8884	0.0111	0.8871	0.0152	0.8883	0.0117	0.8903	0.0097	0.8903	0.0094	0.8883	0.0117	0.8906	0.0093
Biodeg	0.8572▲	0.0295	0.8164 ▲	0.0349	0.8200 ▲	0.0348	0.8610 ▲	0.0312	0.8575 ▲	0.0313	0.8509▲	0.0268	0.8787	0.0237	0.8730	0.0300
Blood	0.7679	0.0335	0.7656	0.0260	0.7180 ▲	0.0535	0.7326 ▲	0.0484	0.7830	0.0349	0.7540▲	0.0390	0.7853	0.0319	0.7746	0.0330
Breast-Cancer	0.9624	0.0223	0.9580	0.0285	0.9595	0.0264	0.9541 ▲	0.0267	0.9536 ▲	0.0260	0.9566▲	0.0254	0.9614	0.0256	0.9658	0.0209
Cleveland	0.5774▲	0.0750	0.5567 ▲	0.0637	0.5357 ▲	0.0781	0.5794 ▲	0.0796	0.5782 ▲	0.0560	0.5925	0.0747	0.6274	0.0650	0.6184	0.0555
Contraceptive	0.5282	0.0321	0.4764 ▲	0.0361	0.4972 ▲	0.0442	0.5338	0.0416	0.5440	0.0432	0.5069▲	0.0405	0.5424	0.0318	0.5347	0.0387
Dermatology	0.9525▲	0.0281	0.9617 ▲	0.0321	0.9663 ▲	0.0285	0.9739	0.0240	0.9739	0.0241	0.9674	0.0096	0.9739	0.0240	0.9739	0.0289
Fertility	0.8533	0.0562	0.8100 ▲	0.1136	0.8533	0.0562	0.5800 ▲	0.1513	0.8667	0.0538	0.8533	0.0763	0.8800	0.0400	0.8600	0.0611
GM4	0.9923▼	0.0117	1.0000 ▼	0.0000	1.0000 ▼	0.0000	0.9747 ▲	0.0146	0.9540 ▲	0.0214	0.9970▼	0.0069	1.0000▼	0.0000	0.9840	0.0123
Haberman	0.7298	0.0538	0.7036 ▲	0.0475	0.7016 ▲	0.0415	0.6898 ▲	0.0624	0.7405	0.0470	0.7317	0.0504	0.7188	0.0344	0.7243	0.0424
Hayes-Roth	0.6708	0.1237	0.7083	0.0959	0.7292	0.1145	0.5958 ▲	0.1432	0.6583 ▲	0.1449	0.7083	0.1063	0.7167	0.1204	0.7188	0.1082
Heart	0.8173	0.0496	0.7605 ▲	0.0832	0.7815 ▲	0.0909	0.8259	0.0559	0.8296	0.0609	0.7926	0.0945	0.8272	0.0544	0.8272	0.0600
Hill-Valley	0.7856▼	0.0420	0.7256	0.0497	0.7215	0.0595	0.7122	0.0256	0.6352 ▲	0.0292	0.7160	0.0515	0.7112	0.0210	0.7076	0.0541
Iris	0.9689	0.0374	0.9667	0.0412	0.9600 ▲	0.0443	0.9667	0.0375	0.9667	0.0375	0.9556	0.0737	0.9667	0.0375	0.9711	0.0373
Led7digit	0.7013▲	0.0630	0.7027 ▲	0.0671	0.6987▲	0.0778	0.7300	0.0661	0.7273	0.0670	0.7260	0.0743	0.7227	0.0694	0.7267	0.0662
Madelon	0.6545▲	0.0297	0.7130	0.0263	0.7130	0.0263	0.7148	0.0284	0.6320 ▲	0.0332	0.6935▲	0.0288	0.7132	0.0270	0.7110	0.0248
Magic	0.8031▲	0.0070	0.8080 ▲	0.0117	0.8098 ▲	0.0069	0.8098 ▲	0.0078	0.8091 ▲	0.0061	0.8109▲	0.0086	0.8132	0.0068	0.8132	0.0067
Musk2	0.9600▲	0.0070	0.9650	0.0059	0.9645	0.0059	0.9537 ▲	0.0070	0.9503 ▲	0.0078	0.9665	0.0067	0.9639▲	0.0065	0.9651	0.0058
Newthyroid	0.9317	0.0510	0.9629 ▼	0.0349	0.9582 ▼	0.0369	0.9316	0.0492	0.9052 ▲	0.0507	0.9396	0.0434	0.9347	0.0505	0.9408	0.0454
Ring	0.9787▼	0.0052	0.8768 ▲	0.0135	0.8789 ▲	0.0114	0.8068 ▲	0.0127	0.7912 ▲	0.0110	0.8916	0.0122	0.8901	0.0114	0.8923	0.0122
Skin_NonSkin	9.9835E-01▲	3.07E-04	9.9957E-01	1.19E-04	9.9957E-01	1.13E-04	9.6696E-01 ▲	1.06E-03	9.5884E-01 ▲	1.10E-03	9.9958E-01	1.22E-04	9.9953E-01▲	1.15E-04	9.9958E-01	1.08E-04
Spambase	0.9039▲	0.0126	0.8815 ▲	0.0140	0.8776 ▲	0.0172	0.9078	0.0108	0.9031 ▲	0.0116	0.9087	0.0123	0.9055▲	0.0104	0.9091	0.0112
Twonorm	0.9783	0.0049	0.9670 ▲	0.0060	0.9690 ▲	0.0059	0.9789	0.0044	0.9789	0.0043	0.9761▲	0.0047	0.9788▼	0.0045	0.9782	0.0047
Vehicle	0.7226▲	0.0422	0.7373 ▲	0.0436	0.7403 ▲	0.0379	0.7833	0.0335	0.7395 ▲	0.0427	0.7644	0.0367	0.7904▼	0.0307	0.7762	0.0310
Waveform_w_Noise	0.8468▲	0.0121	0.8213 ▲	0.0143	0.8230 ▲	0.0149	0.8366 ▲	0.0141	0.8329 ▲	0.0127	0.8571	0.0170	0.8567	0.0148	0.8539	0.0160
Waveform_wo_Noise	0.8537▲	0.0163	0.8262 ▲	0.0211	0.8295 ▲	0.0166	0.8440 ▲	0.0183	0.8354 ▲	0.0186	0.8601	0.0169	0.8651▼	0.0164	0.8579	0.0163
Wdbc	0.9584▲	0.0205	0.9648	0.0249	0.9543 ▲	0.0292	0.9654	0.0190	0.9648 ▲	0.0188	0.9278▲	0.1225	0.9724	0.0202	0.9683	0.0258

▼ : *The benchmark algorithm is better to the proposed method*, ▲: *The benchmark algorithm is worse than the proposed method*

c. Comparison based on F1 score

We compare the proposed method to the benchmark algorithms on the experimental datasets based on F1 score. Once again, the advantage of the proposed method is demonstrated although the result is not as significant as the comparison based on classification accuracy. First, the P-Values of the Friedman test concerning F1 score is smaller than 0.05 so that we reject the null hypothesis that all methods perform equally. The Nemenyi post-hoc test result in Fig. 7 shows that the proposed method is better than ACO. The proposed method ranks first among all methods (rank value 3.45), closely followed by Ensemble (L2LSVM) (rank value 3.48). ACO and GA Meta-data once again are the two poorest methods for F1 measure. The detail of ranking of all methods on each dataset can be found in Table S4 in the Supplement Material.

Concerning the Wilcoxon signed-rank test (Fig 8), the proposed method is competitive to Ensemble (L2LSVM) and continues to be better than the other benchmark algorithms on the experimental datasets. In comparison to META-DES, we rejected 17 null hypotheses that our method and META-DES perform equally among the 30 datasets. In these cases, the proposed method wins on 14 datasets and loses on 3 datasets. The proposed method is better than Sum Rule on F1 score since from the 18 null hypotheses rejected, ours is better than Sum Rule on 17 cases. The significance in the difference between the Decision Template method and the proposed method, however, is reduced in this case comparing to the results for classification accuracy. Among the 30 cases, our method outperforms Decision Template methods on 9 datasets while underperforms on 4 datasets while for classification accuracy, the winning and losing cases are 14 and 0. The proposed method is outstanding compared to GA Meta-data and ACO for F1 score. Among 16 rejected null hypothesis, we win on 14 cases and lose on only 2 cases. A similar result is obtained when comparing ACO and the proposed method as the win/loss ratio is 15/2. The proposed method continues to be slightly better than Ensemble (MLNN) since our method wins on 6 cases among 8 rejected null hypothesis. Finally, although, the proposed method performs competitively to Ensemble (L2LSVM) with 5 wins and 4 losses, the differences in the cases where ours wins are very significant on the Balance (0.6184 vs. 0.8063) and the Haberman dataset (0.4823 vs. 0.5400).

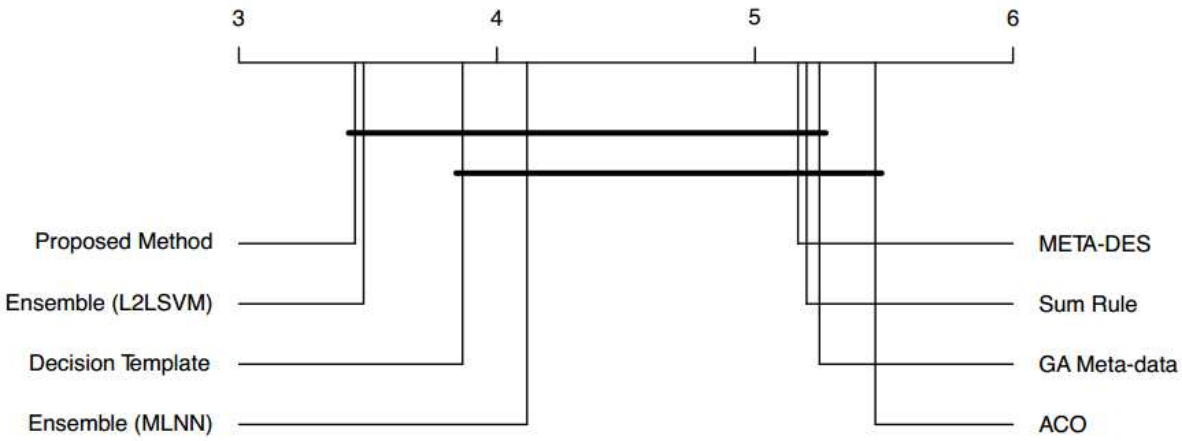


Fig.7. Results of Nemenyi post-hoc test concerning F1 score

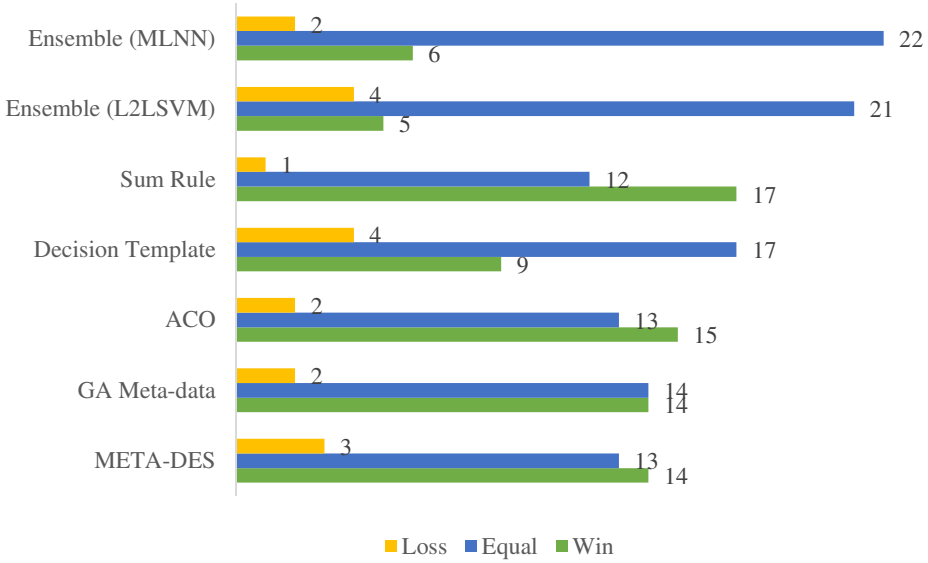


Fig8. Results of Wilcoxon signed-rank test concerning F1 score

Table 5. Mean and standard deviation of F1 score of proposed method and benchmark algorithms

Dataset	META-DES		GA Meta-data		ACO		Decision Template		Sum Rule		Ensemble (MLNN)		Ensemble (L2LSVM)		Proposed Method	
	Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Appendicitis	0.7490	0.1931	0.7057	0.1947	0.6786	0.2002	0.7786 ▼	0.1652	0.7554 ▼	0.2010	0.6243	0.2313	0.7279	0.1985	0.6970	0.2265
Artificial	0.7444	0.0522	0.7517	0.0496	0.7546	0.0503	0.7503	0.0414	0.7651	0.0391	0.7604	0.0381	0.7500	0.0401	0.7509	0.0420
Balance	0.6233▲	0.0349	0.8241	0.0730	0.7951	0.0752	0.8380	0.0431	0.6178 ▲	0.0150	0.8268	0.0941	0.6184▲	0.0158	0.8063	0.0876
Banana	0.8842▲	0.0101	0.8863	0.0115	0.8852	0.0152	0.8869	0.0118	0.8877	0.0100	0.8883	0.0096	0.8869	0.0118	0.8886	0.0096
Biodeg	0.8440▲	0.0312	0.7963 ▲	0.0378	0.7941 ▲	0.0415	0.8479	0.0333	0.8435 ▲	0.0342	0.8317▲	0.0308	0.8627	0.0270	0.8566	0.0339
Blood	0.5874	0.0668	0.5201 ▲	0.0748	0.5294 ▲	0.0646	0.6597 ▼	0.0593	0.5915	0.0778	0.5778	0.0648	0.5803	0.0767	0.5884	0.0722
Breast-Cancer	0.9587	0.0241	0.9540 ▲	0.0312	0.9559	0.0285	0.9488▲	0.0296	0.9483 ▲	0.0288	0.9523▲	0.0276	0.9572	0.0283	0.9630	0.0223
Cleveland	0.3399	0.1030	0.3193	0.0896	0.3070	0.0721	0.3550	0.0951	0.3065	0.0753	0.3474	0.0934	0.3487	0.0814	0.3365	0.0637
Contraceptive	0.5096	0.0329	0.4504 ▲	0.0356	0.4649 ▲	0.0438	0.5236	0.0405	0.5247	0.0453	0.4783▲	0.0502	0.5159	0.0381	0.5151	0.0438
Dermatology	0.9473▲	0.0313	0.9558 ▲	0.0385	0.9570 ▲	0.0395	0.9716	0.0257	0.9717	0.0259	0.9608	0.0427	0.9716	0.0257	0.9715	0.0315
Fertility	0.4599	0.0167	0.4672	0.0973	0.4790	0.0746	0.4191	0.1393	0.4638	0.0161	0.4953	0.1208	0.4678	0.0117	0.4618	0.0185
GM4	0.9920▼	0.0124	1.0000 ▼	0.0000	1.0000 ▼	0.0000	0.9745 ▲	0.0148	0.9543 ▲	0.0215	0.9970▼	0.0068	1.0000▼	0.0000	0.9836	0.0126
Haberman	0.5358	0.0859	0.4999	0.0894	0.4659 ▲	0.0627	0.5791 ▼	0.0794	0.5563	0.0874	0.5467	0.0931	0.4823▲	0.0639	0.5400	0.0843
Hayes-Roth	0.6943	0.1187	0.7156	0.0976	0.7389	0.1192	0.6176 ▲	0.1353	0.6728 ▲	0.1418	0.6920	0.1334	0.7269	0.1217	0.7189	0.1229
Heart	0.8115	0.0533	0.7547▲	0.0889	0.7765 ▲	0.0961	0.8224	0.0572	0.8250	0.0633	0.7809	0.1192	0.8235	0.0558	0.8230	0.0622
Hill-Valley	0.7817▼	0.0448	0.7156	0.0538	0.7102	0.0653	0.7117	0.0256	0.6283 ▲	0.0342	0.7112	0.0566	0.7109	0.0211	0.6851	0.0716
Iris	0.9684	0.0382	0.9662	0.0420	0.9594	0.0452	0.9662	0.0382	0.9662	0.0382	0.9508	0.0919	0.9662	0.0382	0.9709	0.0375
Led7digit	0.6936▲	0.0616	0.6959 ▲	0.0720	0.6939 ▲	0.0787	0.7266	0.0660	0.7248	0.0675	0.7207	0.0738	0.7175	0.0697	0.7219	0.0678
Madelon	0.6538▲	0.0299	0.7119	0.0272	0.7119	0.0272	0.7142	0.0291	0.6317 ▲	0.0330	0.6926▲	0.0289	0.7122	0.0278	0.7101	0.0252
Magic	0.7603▲	0.0093	0.7749 ▲	0.0115	0.7764 ▲	0.0084	0.7853 ▼	0.0092	0.7706 ▲	0.0079	0.7804	0.0100	0.7842▼	0.0084	0.7811	0.0083
Musk2	0.9179▲	0.0154	0.9299	0.0123	0.9280	0.0132	0.9058 ▲	0.0149	0.8957 ▲	0.0179	0.9322▼	0.0137	0.9276	0.0132	0.9286	0.0127
Newthyroid	0.8859	0.0928	0.9482 ▼	0.0548	0.9374 ▼	0.0580	0.8818	0.0904	0.8346 ▲	0.1025	0.9048	0.0720	0.8885	0.0917	0.9000	0.0817
Ring	0.9787▼	0.0052	0.8754▲	0.0137	0.8774 ▲	0.0118	0.8047 ▲	0.0131	0.7816 ▲	0.0125	0.8909	0.0124	0.8891▲	0.0117	0.8917	0.0124
Skin_NonSkin	9.9750E-01▲	4.65E-04	9.9934E-01	1.80E-04	9.9934E-01	1.72E-04	9.5249E-01 ▲	1.45E-03	9.3936E-01 ▲	1.57E-03	9.9936E-01	1.86E-04	9.9929E-01▲	1.75E-04	9.9936E-01	1.64E-04
Spambase	0.8993▲	0.0131	0.8757 ▲	0.0148	0.8714 ▲	0.0178	0.9023	0.0115	0.8963 ▲	0.0126	0.9036	0.0132	0.8997▲	0.0111	0.9044	0.0117
Twonorm	0.9783	0.0049	0.9670 ▲	0.0060	0.9690 ▲	0.0059	0.9789	0.0044	0.9789	0.0043	0.9761▲	0.0047	0.9788	0.0045	0.9782	0.0047
Vehicle	0.7095▲	0.0475	0.7376 ▲	0.0447	0.7414 ▲	0.0363	0.7768	0.0369	0.7279 ▲	0.0467	0.7630	0.0359	0.7867▼	0.0311	0.7692	0.0322
Waveform_w_Noise	0.8463▲	0.0122	0.8212 ▲	0.0144	0.8229 ▲	0.0148	0.8332 ▲	0.0148	0.8286 ▲	0.0135	0.8568	0.0171	0.8563	0.0148	0.8536	0.0160
Waveform_wo_Noise	0.8524▲	0.0166	0.8259 ▲	0.0213	0.8292 ▲	0.0167	0.8386 ▲	0.0197	0.8281 ▲	0.0204	0.8593	0.0173	0.8643▼	0.0166	0.8568	0.0167
Wdbc	0.9551▲	0.0222	0.9626	0.0264	0.9514 ▲	0.0309	0.9622 ▲	0.0211	0.9614 ▲	0.0209	0.9138▲	0.1597	0.9700	0.0220	0.9658	0.0276

▼ : *The benchmark algorithm is better to the proposed method*, ▲: *The benchmark algorithm is worse than the proposed method*

d. Discussions

The proposed method is better than Sum Rule for both classification accuracy and F1 score. As mentioned before, Sum Rule is a fixed combining method which does not use the label information in the meta-data of the training observations to train the combiner. This, therefore, makes Sum Rule very fast in training since only the base classifiers are trained. However, because Sum Rule does not exploit information in the meta-data of the training observations, it performs poorer compared to trainable combining methods such as the Decision Template method and the proposed method. As we also optimized the upper and lower bound of each interval in the representation by minimizing the 0-1 loss on the training data, the proposed method is therefore significantly better than Sum Rule.

Surprisingly, the two selection methods perform the poorest in our experiment, taking the last positions in ranking. In GA Meta-data and ACO method, we used C4.5 Decision Tree as the combining algorithm as in the original studies. The poor performance of Decision Tree when working as a combiner is a reason why these two methods underperform on experimental datasets. Theoretically, any learning algorithms can be used to combine classifiers in an ensemble system. However, algorithms that can handle and adapt to the characteristics of the meta-data like Decision Template and the proposed method are expected to achieve better performance.

META-DES ranks fourth for both performance measures. In fact, META-DES is a dynamic ensemble selection method that selects a subset of base classifiers for each test instance based on their competency on the neighborhood of the test instance. The performance of the dynamic ensemble selection method thus mainly depends on the technique to define the region to compute competency [51]. In META-DES, the K Nearest Neighbors obtained from a validation set are used to construct the region of competence associated with each test instance. On some datasets, the base classifiers which perform well on the validation instances may misclassify test instance because of the difference in distribution between the validation data and the test data. This explains why META-DES obtains low classification accuracy on some datasets.

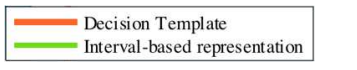
Finally, the proposed method is better than the Decision Template method, especially for classification accuracy. As mentioned in Section 2.2, the Decision Template method has a limitation when applying to imbalanced data as the prediction is mainly belonged to the majority label, resulting in the similarity in the decision templates. The Decision Template method, therefore, performs poorly on imbalanced datasets in experiments such as Fertility and Hayes-Roth. The proposed method meanwhile uses intervals to model the class label. The representation is learned to adapt to each dataset by minimizing the 0-1 loss function on the training data, resulting in better results than the Decision Template method. Moreover, interval is an effective way to handle reasoning and it has more advantages than the single point representation in the Decision Template method. Fig 9 shows the comparison between the representation of the Decision Template method and the proposed method on the Fertility dataset

(Another figure for Hayes-Roth is shown in the Supplement Material). In detail, we draw the histogram to show the distribution of data in each column of the meta-data. For each class label, we will have $M \times K$ histograms, $M \times K$ decision templates, and $M \times K$ intervals. The red line here shows the value of decision template while the yellow range between two green lines shows the interval of representation. It is observed that the interval approach provides a more general representation by capturing the range in predictions than just a single point value as in the Decision Template method. Moreover, the decision templates of this dataset are very similar, resulting in an ambiguous decision when assigning instances to the class labels (see Fig. 1 for more detail). The elements in the interval-based representation, in contrast, are significantly different between the two class labels. These make the proposed method performs better than the Decision Template method on some datasets.

Fig. 10 shows the average value of loss function vs iterations in PSO. The value was obtained from the loss values of all candidates in the population in each generation. In this work, along with candidates generated by PSO algorithm, we initialize a special candidate based on the decision template by adding/subtracting a value $\varepsilon = 0.01$ to the decision template to obtain the interval i.e. $\left[dt_j(k,m) - \varepsilon, dt_j(k,m) + \varepsilon\right]$. This candidate will be evolved through iterations in PSO algorithms. In this way, we also search intervals around the decision template for the optimal solution. It is observed from Fig. 10 that the average value of the loss value quickly coverages to the optimal value after a small number of iterations.

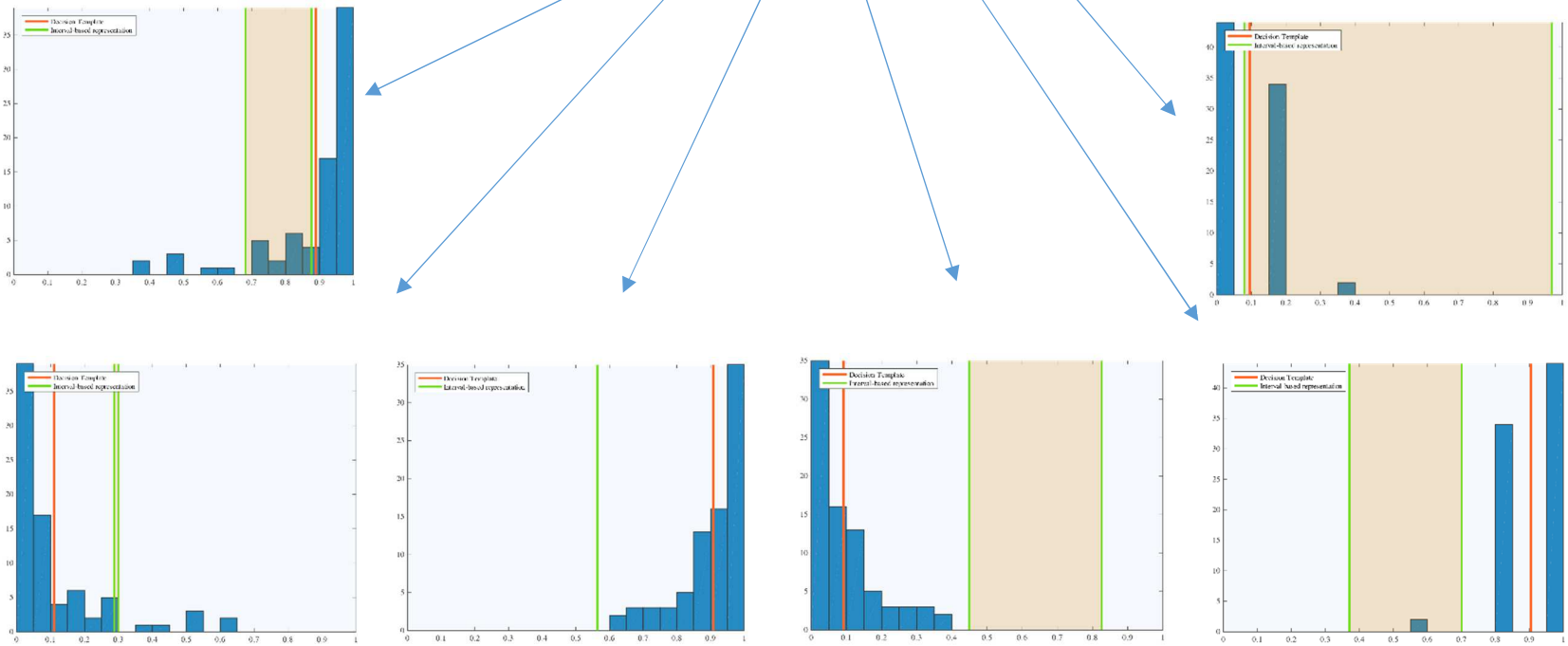
6. Conclusions

We have introduced a novel combining method for heterogeneous ensemble systems via the interval-based representation. The proposed method originates from the observation that the Decision Template method does not perform well on imbalanced datasets and datasets with skewed class distribution. In this study, instead of using numerical value to model the representation like in the Decision Template method, we used the interval values which provide a more general representation as well as capturing the uncertainty in the classifiers' predictions. Here each class label is modeled by an optimal vector of intervals which is found by minimizing the 0-1 loss function on the training data using PSO algorithm. It is noted that the bounds of each interval belong to $[0,1]$ and the lower bound is smaller than or equal to the upper bound so that the original PSO algorithm was modified to handle the constraints of the bounds of intervals. Classification is done by assigning a test instance to the label that is associated with the shortest distance between the meta-data of the instance and the interval-based representation.



Meta-data associated with class 1

0.8338	0.1662	0.9170	0.0830	1.0000	0.0000
0.9873	0.0127	0.9838	0.0162	1.0000	0.0000
0.8105	0.1895	0.8926	0.1074	0.8000	0.2000
…	…	…	…	…	…
0.9986	0.0014	0.9962	0.0038	1.0000	0.0000
0.9980	0.0020	0.9908	0.0092	1.0000	0.0000
0.9875	0.0125	0.9246	0.0754	1.0000	0.0000



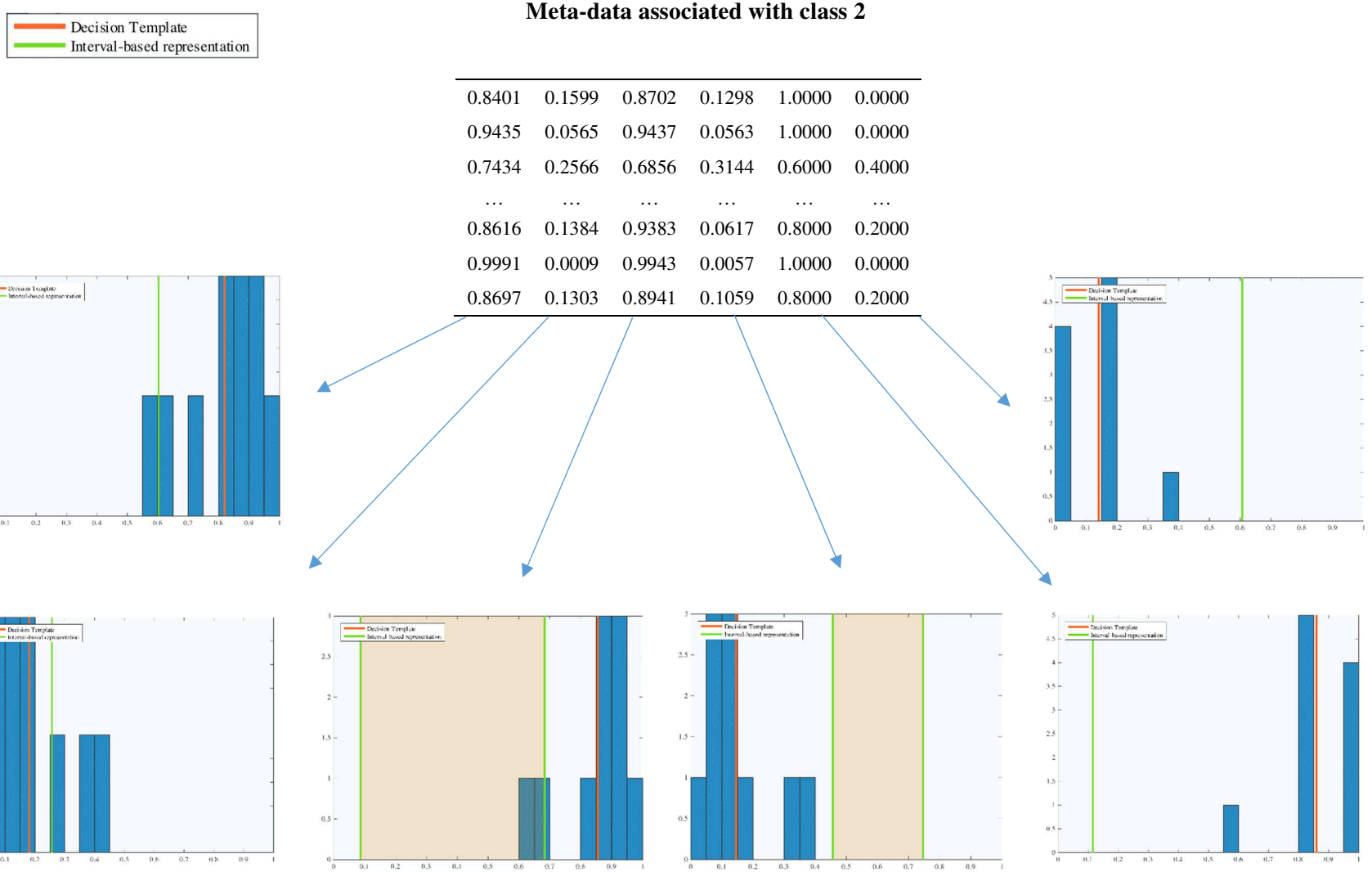


Fig. 9. Comparison between the representation of Decision Template method and proposed method on the Fertility dataset

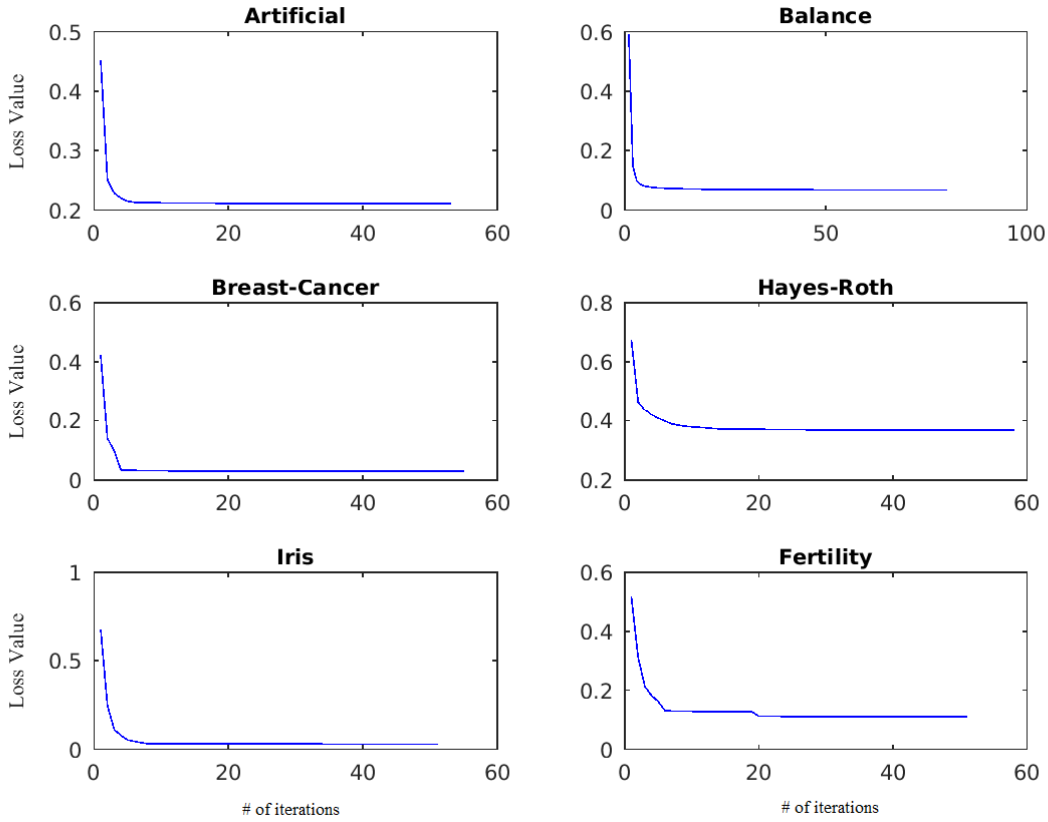


Fig. 10. The value of loss function in the PSO *iterations* on six datasets

We conducted the experiments on the proposed methods and 7 benchmark algorithms on 30 datasets. Based on the results of multiple methods – multiple datasets test, the proposed method is better than META-DES, ACO and GA Meta-data for classification accuracy. The proposed method also ranks first for both performance measures. Meanwhile, via the results of Wilcoxon signed-rank test when comparing to each benchmark algorithm, proposed method is better than all the benchmark algorithms except Ensemble (L2LSVM) for classification accuracy and F1 score. We also noted that the social and cognitive attraction parameters of the PSO algorithm have slight influences on the performance of proposed method except on some datasets when social attraction is set to 0.

The proposed method is an evolutionary computation-based approach for the adaptation of combining algorithm. In the future, the proposed method can be expanded to handle the selection problem which could further improve the classification performance. In this study, we only addressed a single objective optimization problem i.e. minimizing the 0-1 loss function to obtain the optimal representation. The searching process therefore only focuses on finding the candidate which has the smallest loss value while ignoring other performance measures such as F1 score. The F1 score or other performance measures of the proposed method can be enhanced by considering multi-objective optimization when

searching for the optimal candidates. Besides, although the mean loses its capability to provide the best central location in some situations, it is still a very important statistic. One potential research is to combine both the mean value-based distance and the interval-based distance to assign the predicted label in (12). Moreover, in this study, we used the boundary approach in calculating the distance between two vectors of intervals [39, 42]. Other distances like in [39-41] can be used in this method to measure the similarity (or dissimilarity) between the meta-data and the interval-based representation. The analysis of the influence of using different distances to the ensemble performance will be a potential future work. Finally, the proposed method can be combined with other techniques to handle imbalanced data. In [54], synthetic minority over-sampling technique (SMOTE) and sampling with replacement technique in Bagging were combined with differentiated sampling rates (DSR) to generate positive and negative samples respectively to obtain balanced training data. [55] meanwhile handles imbalanced data streams by using embedded SMOTE in each iteration of AdaBoost to generate minority samples. The sampling process focuses more on generating new samples around new and difficult minority samples to make the training data balanced. All these approaches are useful for further investigations.

References

[1] R.P.W. Duin, The combining classifier: to train or not to train?, in Conference of Object recognition supported by user interaction for service robots, 2002, DOI: 10.1109/ICPR.2002.1048415

[2] 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.

[3] 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), 144-160.

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

[5] 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, IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1698-1705.

[6] P. Shunmugapriya, S. Kanmani, Optimization of stacking ensemble configuration through Artificial Bee Colony algorithm, Swarm and Evolutionary Computation. 12 (2013), 24-32.

[7] T.T. Nguyen, A.W.C. Liew, M.T. Tran, M.P. Nguyen, Combining Multi Classifiers Based on a Genetic Algorithm – A Gaussian Mixture Model Framework, in: D.-S. Huang, K.-H. Jo, L. Wang (Eds.), Intelligent Computing Methodologies, Springer International Publishing, 2014, pp. 56-67.

[8] 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.

[9] S. Ali, A. Majid, Can-Evo-Ens: Classifier stacking based evolutionary ensemble system for prediction of human breast cancer using amino acid sequences, Journal of Biomedical Informatics. 54 (2015), 256-269.

[10] K.-J. Kim, S.-B. Cho, An Evolutionary Algorithm Approach to Optimal Ensemble Classifiers for DNA Microarray Data Analysis, IEEE Trans. of Evolutionary Computation. 12(3) (2008), 377 - 388

[11] M. Bacauskiene, A. Verikas, A. Gelzinis, D. Valincius, A feature selection technique for generation of classification committees and its applications to categorization of laryngeal images, Pattern Recognition. 42 (2009), 645-654.

[12] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: Proceedings of IEEE International Conference on Neural Networks, Vol IV, pp. 1942–1948.

[13] 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.

[14] 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.

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

[16] 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

[17] 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.

[18] Z. Wang, R. Wang, J. Gao, Z. Gao, Y. Liang, Fault recognition using an ensemble classifiers based on Dempster-Shafer Theory, Pattern Recognition. 99 (2020), 1070-1079.

[19] K. Guo, W. Li, Combination rule of D–S evidence theory based on the strategy of cross merging between evidences, Expert Systems with Applications. 38 (2011) 13360–13366.

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

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

[22] M.U. Şen, H. Erdog˘an, Linear classifier combination and selection using g8roup sparse regularization and hinge loss, Pattern Recognition Letters. 34 (2013) 265-274.

[23] L. Yijing, G. Haixiang, L. Xiao, L. Yanan, L. Jinling, Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data, Knowledge-Based Systems. 94 (2016), 88-104.

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

[25] L.I. Kuncheva, L.C. Jain, Designing Classifier Fusion Systems by Genetic Algorithms, IEEE Trans. of Evolutionary Computation. 4(4) (2000), 327-336.

[26] B. Gabrys, D. Ruta, Genetic algorithms in classifier fusion, Applied Soft Computing. 6 (2006), 337-347.

[27] Y.-W. Kim, I.-S. Oh, Classifier ensemble selection using hybrid genetic algorithms, Pattern Recognition Letters.29 (2008), 796-802.

[28] M.-J. Kim, D.-K. Kang, Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction, Expert System with Applications. 39 (2012), 9308-9314.

[29] Y. Chen, M-L. Wong, H. Li, Applying Ant Colony Optimization to configuring stacking ensembles for data mining, Expert System with Applications. 41 (2014), 2688-2702.

[30] I. Mendialdua, A. Arruti, E. Jauregi, E. Lazkano, B. Sierra, Classifier Subset Selection to construct multi-classifiers by means of estimation of distribution algorithms, Neurocomputing. 157 (2015), 46-60.

[31] R. Mousavi, M. Eftekhari, A new ensemble learning methodology based on hybridization of classifier ensemble selection approach, Applied Soft Computing.37 (2015), 652-666.

[32] M.N. Haque, N. Noman, R. Berretta, P. Moscato, Heterogeneous Ensemble Combination Search Using Genetic Algorithm for Class Imbalanced Data Classification, PLOS1, 2016, 10.1371/journal.pone.0146116.

[33] Y. Wang, D. Wang, N. Geng, Y. Wang, Y. Yin, Y. Jin, Stacking-based ensemble learning of decision trees for interpretable prostate cancer detection, Applied Soft Computing. 77 (2019), 188-204.

[34] T.T. Nguyen, A.V. Luong, T.M.V. Nguyen, T.S. Ha, A.W.C. Liew, J. McCall, Simultaneous Meta-Feature and Meta-Classifier Selection in Multiple Classifier System, in Genetic and Evolutionary Computation Conference (GECCO), 2019.

[35] X. Wang, H. Wang, Classification by evolutionary ensembles, Pattern Recognition. 39 (2006), 595-607.

[36] L. Nanni, A. Lumini, A genetic encoding approach for learning methods for combining classifiers, Expert Systems with Applications. 36 (2009), 7510-7514.

[37] C.A.A. Padilha, D.A.C. Barone, A.D.D. Neto, A multi-level approach using genetic algorithms in an ensemble of Least Squares Support Vector Machines, Knowledge-based Systems. 106 (2016), 85-95.

[38] M.A. Duval-Poo, J. Sosa-Garcia, A. Guerra-Gandon, S. Vega-Pons, J. Ruiz-Schulcloper, A New Classifier Combination Scheme Using Clustering Ensemble, in Proceeding of CIARP 2012, pp. 154-161.

[39] A. Irpino, R.Verde, Dynamic Clustering of Interval Data Using a Wasserstein-based Distance. Pattern Recognition Letters. 29 (2008), 1648-1658.

[40] K.C. Gowda, E. Diday, Symbolic clustering using a new dissimilarity measure, Pattern Recognition. 24 (1991), 567-578.

[41] M. Ichino, H. Yaguchi, Generalized Minkowsky metrics for mixed feature-type analysis, IEEE Trans. on System Man Cyber. 24 (1994) (4), 698-708.

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

[43] R.E. Perez, K. Behdinan, Particle swarm approach for structural design optimization, Computers and Structures. 85 (2007), 1579-1588.

[44] R. Hassan, B. Cohanim, O. Weck, G. Venter, A comparison of particle swarm optimization and the genetic algorithm, in AIAA, 2005.

[45] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Applied Mathematics and Computation. 217 (2010), 3166-3173.

[46] X. Hu, R. Eberhart, Solving constrained nonlinear optimization problems with particle swarm optimization, In Proceedings of the sixth World multiconference on Systemics, Cybernetics and Informatics, Vol. 5, pp. 203-206, 2002.

[47] K.E. Parsopoulos, M.N. Vrahatis, Particle swarm optimization method for constrained optimization problems, Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies, 76(1) (2002), 214-220.

[48] I.C. Trelea, The particle swarm optimization algorithm: Convergence analysis and parameter selection, Information Processing Letters, 85(6) (2003), 317–325.

[49] L. Zhang, Y. Tang, C. Hua, X. Guan, A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques, Applied Soft Computing. 28 (2015), 138-149.

[50] Y. Shi, E. Eberhart, A modified particle swarm optimizer, IEEE Congress on Evolutionary Computation (CEC), 1998, pp. 69-73.

[51] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, T.I. Ren, META-DES: A dynamic ensemble selection framework using meta-learning, Pattern Recognition. 48(5) (2015), 1925-1935.

[52] 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.

[53] T.T. Nguyen, M.T. Dang, A.W.C. Liew, J.C. Bezdek, A weighted multiple classifier framework based on Random Projection, Information Science. 490 (2019), 36-58.

[54] J. Sun, J. Lang, H. Fujita, H. Li, Imbalanced Enterprise Credit Evaluation with DTE-SBD: Decision Tree Ensemble Based on SMOTE and Bagging with Differentiated Sampling Rates", Information Sciences. 425 (2018), 76-91.

[55] J. Sun, H. Li, H. Fujita, B. Fu, W. Ai, Class-imbalanced dynamic financial distress prediction based on Adaboost-SVM ensemble combined with SMOTE and time weighting, Information Fusion. 54 (2020), 128-144.

Supplement Material

Evolving interval-based representation for multiple classifier fusion

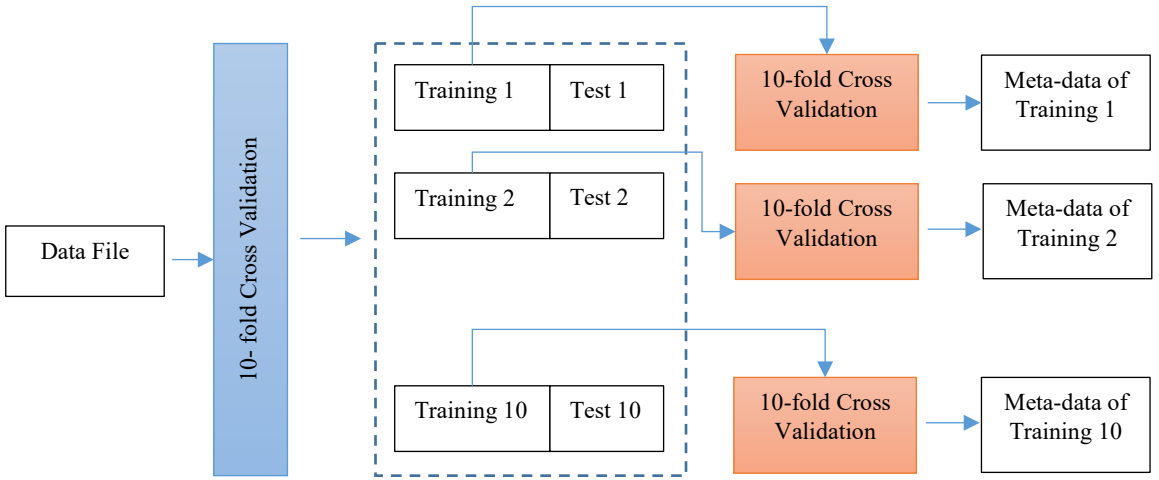
Tien Thanh Nguyen[1], Manh Truong Dang[1], Vimal Anand Baghel[2], Anh Vu Luong[3], John McCall[1], Alan Wee-Chung Liew[3]

[1]School of Computing Science and Digital Media, Robert Gordon University, Aberdeen, UK

[2]Department of Computer Science and Engineering, Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur, India

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

Fig.S1. The experimental procedure



**From a data file, we used the first 10-fold Cross-Validation procedure (the blue box) to create the training and testing data. The first Cross-Validation procedure was run 3 times to obtain 30 trials in each data file. The second 10-fold Cross-Validation procedure (the orange box) was used in each training set to generate its meta-data*

Table.S1. Wilcoxon signed rank test result concerning classification accuracy

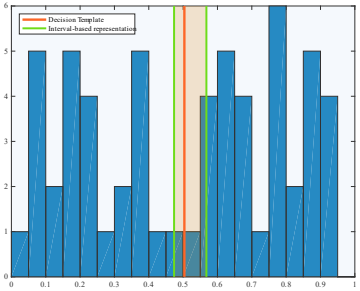
Dataset	META-DES		vs. GA Meta-data		vs. ACO-S1		vs. Decision Template		vs. Sum Rule		vs. Ensemble (L2LSVM)		vs. Ensemble (MLNN)	
	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?
Appendicitis	2.34E-02	Yes▼	4.20E-01		5.27E-02		5.39E-01		3.13E-02	Yes ▼	1.56E-01		1.12E-01	
Artificial	7.12E-01		2.09E-01		1.23E-01		7.12E-02		2.05E-03	Yes ▼	4.06E-02	Yes ▲	3.60E-02	Yes ▼
Balance	1.24E-03	Yes▲	1.33E-01		3.38E-01		4.44E-01		3.28E-03	Yes ▲	4.53E-03	Yes ▲	8.02E-02	
Banana	1.10E-01		6.32E-02		1.35E-01		9.58E-02		6.73E-01		9.58E-02		5.59E-01	
Biodeg	1.12E-02	Yes▲	2.53E-06	Yes ▲	8.04E-06	Yes ▲	4.03E-02	Yes ▲	9.90E-03	Yes ▲	1.74E-01		5.75E-04	Yes ▲
Blood	1.62E-01		1.34E-01		3.03E-05	Yes ▲	9.26E-05	Yes ▲	1.81E-01		1.12E-01		9.99E-03	Yes ▲
Breast-Cancer	2.83E-01		8.96E-02		7.56E-02		2.69E-03	Yes ▲	1.78E-03	Yes ▲	2.64E-01		2.67E-02	Yes ▲
Cleveland	2.72E-02	Yes▲	1.25E-03	Yes ▲	3.23E-04	Yes ▲	5.72E-03	Yes ▲	1.39E-02	Yes ▲	2.92E-01		6.53E-02	
Contraceptive	3.00E-01		4.54E-06	Yes ▲	7.54E-05	Yes ▲	9.32E-01		1.13E-01		2.16E-01		4.45E-03	Yes ▲
Dermatology	2.79E-03	Yes▲	4.58E-02	Yes ▲	4.22E-02	Yes ▲	1.00E+00		1.00E+00		1.00E+00		2.21E-01	
Fertility	7.54E-01		2.83E-02	Yes ▲	7.81E-01		3.42E-06	Yes ▲	7.66E-01		6.25E-02		7.45E-01	
GM4	4.36E-03	Yes▼	1.44E-05	Yes ▼	1.44E-05	Yes ▼	6.10E-05	Yes ▲	4.95E-06	Yes ▲	1.44E-05	Yes ▼	4.23E-04	Yes ▼
Haberman	6.39E-01		1.97E-02	Yes ▲	2.39E-02	Yes ▲	1.63E-02	Yes ▲	1.06E-01		2.72E-01		6.47E-01	
Hayes-Roth	5.57E-02		5.73E-01		5.22E-01		2.84E-05	Yes ▲	7.69E-03	Yes ▲	9.53E-01		5.15E-01	
Heart	4.27E-01		3.98E-05	Yes ▲	6.84E-03	Yes ▲	9.52E-01		3.61E-01		8.83E-01		1.76E-01	
Hill_Valley	2.16E-05	Yes▼	2.29E-01		2.70E-01		7.19E-01		6.01E-05	Yes ▲	6.29E-01		3.36E-01	
Iris	8.13E-01		6.88E-01		2.97E-02	Yes ▲	5.00E-01		5.00E-01		5.00E-01		4.06E-01	
Led7digit	1.62E-03	Yes▲	1.06E-04	Yes ▲	5.56E-04	Yes ▲	4.61E-01		8.84E-01		2.83E-01		8.64E-01	
Madelon	4.05E-06	Yes▲	3.62E-01		3.62E-01		1.77E-01		1.98E-06	Yes ▲	3.01E-01		3.81E-03	Yes ▲
Magic	2.83E-06	Yes▲	1.48E-03	Yes ▲	3.38E-03	Yes ▲	5.14E-04	Yes ▲	4.03E-05	Yes ▲	5.09E-01		4.99E-02	Yes ▲
Musk2	1.35E-05	Yes▲	8.59E-01		4.16E-01		1.71E-06	Yes ▲	1.72E-06	Yes ▲	3.89E-02	Yes ▲	6.18E-02	
Newthyroid	2.86E-01		3.06E-02	Yes ▼	1.36E-02	Yes ▼	2.32E-01		6.16E-04	Yes ▲	4.69E-01		7.75E-01	
Ring	1.73E-06	Yes▼	2.46E-06	Yes ▲	2.46E-06	Yes ▲	1.71E-06	Yes ▲	1.72E-06	Yes ▲	6.54E-02		3.94E-01	
Skin_NonSkin	1.73E-06	Yes▲	6.35E-01		1.02E-01		1.73E-06	Yes ▲	1.73E-06	Yes ▲	8.49E-03	Yes ▲	5.69E-01	
Spambase	3.71E-02	Yes▲	2.35E-06	Yes ▲	1.92E-06	Yes ▲	1.35E-01		1.47E-04	Yes ▲	9.64E-03	Yes ▲	7.40E-01	
Twonorm	7.53E-01		2.03E-06	Yes ▲	4.27E-06	Yes ▲	1.23E-01		1.15E-01		4.29E-02	Yes ▼	1.54E-02	Yes ▲
Vehicle	4.81E-05	Yes▲	5.24E-04	Yes ▲	2.42E-04	Yes ▲	1.51E-01		1.28E-04	Yes ▲	2.41E-03	Yes ▼	1.03E-01	
Waveform_w_Noise	1.34E-02	Yes▲	3.45E-06	Yes ▲	2.33E-06	Yes ▲	5.47E-06	Yes ▲	3.42E-06	Yes ▲	8.25E-02		1.61E-01	
Waveform_wo_Noise	2.77E-02	Yes▲	2.11E-06	Yes ▲	3.48E-06	Yes ▲	4.26E-05	Yes ▲	3.85E-06	Yes ▲	2.10E-04	Yes ▼	3.99E-01	
Wdbc	5.22E-03	Yes▲	4.07E-01		9.40E-03	Yes ▲	7.81E-02		3.27E-02	Yes ▲	4.94E-01		1.15E-02	Yes ▲

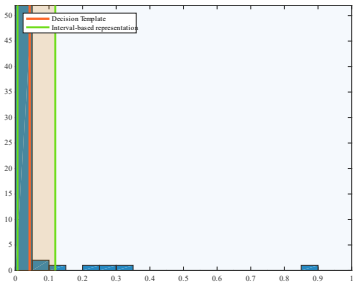
** Yes means we rejected the null hypothesis because P-Value of the test is smaller than 0.05*
** ▲ and ▼ mean the proposed method is better or worse than the benchmark algorithm*

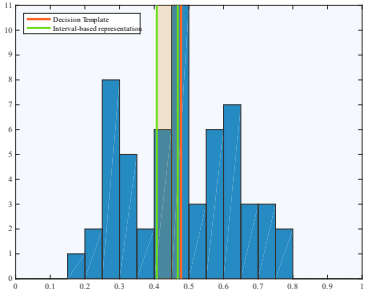
Table.S2. Wilcoxon signed rank test result concerning F1 score

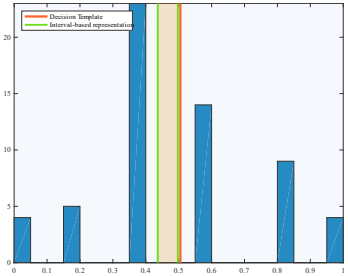
Dataset	META-DES		vs. GA Meta-data		vs. ACO-S1		vs. Decision Template		vs. Sum Rule		vs. Ensemble (L2LSVM)		vs. Ensemble (MLNN)	
	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?	P-Value	Reject?
Appendicitis	5.08E-02		1.00E+00		6.73E-01		3.39E-02	Yes ▼	2.34E-02	Yes ▼	1.64E-01		2.70E-01	
Artificial	4.91E-01		9.54E-01		7.70E-01		8.41E-01		1.91E-01		6.79E-01		3.16E-01	
Balance	3.88E-06	Yes▲	8.94E-01		2.30E-01		3.36E-01		6.28E-06	Yes ▲	4.72E-06	Yes ▲	3.82E-01	
Banana	1.57E-02	Yes▲	5.45E-02		1.20E-01		2.37E-01		4.69E-01		2.37E-01		7.54E-01	
Biodeg	2.85E-02	Yes▲	2.56E-06	Yes ▲	9.32E-06	Yes ▲	1.59E-01		3.32E-02	Yes ▲	2.58E-01		8.35E-04	Yes ▲
Blood	5.96E-01		5.29E-04	Yes ▲	4.68E-03	Yes ▲	7.51E-05	Yes ▼	8.48E-01		4.69E-01		5.04E-01	
Breast-Cancer	6.73E-02		1.90E-02	Yes ▲	9.87E-02		6.34E-04	Yes ▲	3.63E-04	Yes ▲	1.16E-01		1.12E-02	Yes ▲
Cleveland	8.45E-01		2.99E-01		1.11E-01		2.89E-01		1.65E-01		3.04E-01		7.50E-01	
Contraceptive	4.91E-01		4.73E-06	Yes ▲	1.49E-05	Yes ▲	2.06E-01		2.37E-01		8.13E-01		2.58E-03	Yes ▲
Dermatology	7.29E-04	Yes▲	1.13E-02	Yes ▲	3.09E-02	Yes ▲	1.00E+00		1.00E+00		1.00E+00		8.34E-02	
Fertility	1.00E+00		4.85E-01		6.08E-01		9.75E-02		7.66E-01		6.25E-02		5.45E-01	
GM4	3.78E-03	Yes▼	1.76E-05	Yes ▼	1.76E-05	Yes ▼	6.10E-05	Yes ▲	3.18E-06	Yes ▲	1.76E-05	Yes ▼	3.13E-04	Yes ▼
Haberman	6.48E-01		8.19E-02		2.84E-03	Yes ▲	4.76E-02	Yes ▼	2.47E-01		2.23E-03	Yes ▲	7.92E-01	
Hayes-Roth	2.64E-01		8.29E-01		5.85E-01		5.56E-05	Yes ▲	3.25E-02	Yes ▲	5.10E-01		3.47E-01	
Heart	2.05E-01		4.80E-05	Yes ▲	4.17E-03	Yes ▲	9.66E-01		8.72E-01		8.95E-01		6.74E-02	
Hill_Valley	1.64E-05	Yes▼	1.20E-01		1.27E-01		1.99E-01		2.96E-03	Yes ▲	1.85E-01		8.22E-02	
Iris	5.00E-01		4.53E-01		1.56E-01		3.13E-01		3.13E-01		3.13E-01		2.50E-01	
Led7digit	2.56E-03	Yes▲	9.71E-05	Yes ▲	9.63E-04	Yes ▲	2.00E-01		5.39E-01		4.56E-01		7.01E-01	
Madelon	4.73E-06	Yes▲	3.49E-01		3.49E-01		1.77E-01		2.13E-06	Yes ▲	3.05E-01		4.68E-03	Yes ▲
Magic	1.73E-06	Yes▲	2.61E-04	Yes ▲	8.94E-04	Yes ▲	1.71E-03	Yes ▼	2.60E-06	Yes ▲	1.59E-03	Yes ▼	8.13E-01	
Musk2	1.64E-05	Yes▲	2.99E-01		6.29E-01		1.73E-06	Yes ▲	1.73E-06	Yes ▲	4.53E-01		1.69E-02	Yes ▼
Newthyroid	6.12E-02		6.07E-03	Yes ▼	2.28E-02	Yes ▼	9.20E-02		7.77E-05	Yes ▲	2.72E-01		8.72E-01	
Ring	1.73E-06	Yes▼	2.13E-06	Yes ▲	2.13E-06	Yes ▲	1.73E-06	Yes ▲	1.73E-06	Yes ▲	2.70E-02	Yes ▲	3.99E-01	
Skin_NonSkin	1.73E-06	Yes▲	5.84E-01		9.45E-01		1.73E-06	Yes ▲	1.73E-06	Yes ▲	2.97E-02	Yes ▲	4.81E-01	
Spambase	3.16E-02	Yes▲	2.35E-06	Yes ▲	1.92E-06	Yes ▲	6.45E-02		1.49E-05	Yes ▲	1.48E-03	Yes ▲	7.66E-01	
Twonorm	5.16E-01		1.92E-06	Yes ▲	4.28E-06	Yes ▲	1.63E-01		1.75E-01		8.92E-02		2.11E-02	Yes ▲
Vehicle	3.11E-05	Yes▲	1.71E-03	Yes ▲	2.77E-03	Yes ▲	1.36E-01		1.36E-04	Yes ▲	8.31E-04	Yes ▼	5.04E-01	
Waveform_w_Noise	1.40E-02	Yes▲	3.52E-06	Yes ▲	2.35E-06	Yes ▲	1.73E-06	Yes ▲	1.73E-06	Yes ▲	9.78E-02		1.85E-01	
Waveform_wo_Noise	2.85E-02	Yes▲	2.35E-06	Yes ▲	3.18E-06	Yes ▲	1.02E-05	Yes ▲	2.13E-06	Yes ▲	2.22E-04	Yes ▼	3.82E-01	
Wdbc	4.72E-03	Yes▲	6.37E-01		3.86E-02	Yes ▲	1.13E-02	Yes ▲	4.82E-03	Yes ▲	4.73E-01		1.50E-02	Yes ▲

** Yes means we rejected the null hypothesis because P-Value of the test is smaller than 0.05*
** ▲ and ▼ mean the proposed method is better or worse than the benchmark algorithm*



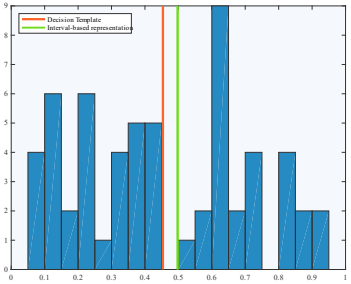


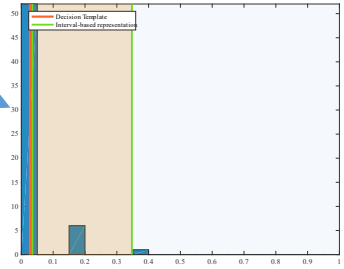


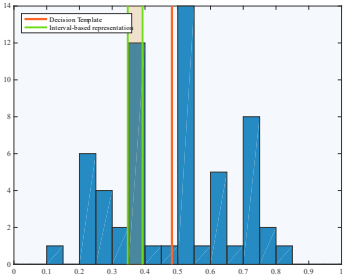


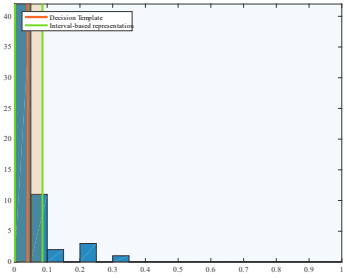
0.7507	0.2493	0.0000	0.5449	0.4534	0.0017	0.6000	0.4000	0.7507
0.6991	0.3009	0.0000	0.5452	0.4528	0.0020	0.6000	0.4000	0.6991
0.8994	0.1006	0.0000	0.7237	0.2748	0.0015	0.8000	0.2000	0.8994
…	…	…	…	…	…	…	…	..
0.3445	0.6299	0.0256	0.4196	0.5493	0.0312	0.4000	0.6000	0.3445
0.3169	0.6368	0.0463	0.3954	0.5465	0.0581	0.2000	0.4000	0.3169
0.7952	0.2048	0.0001	0.6471	0.3514	0.0016	1.0000	0.0000	0.7952

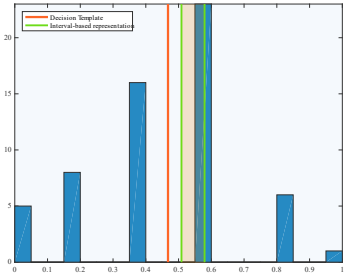
Meta-data associated with class 1

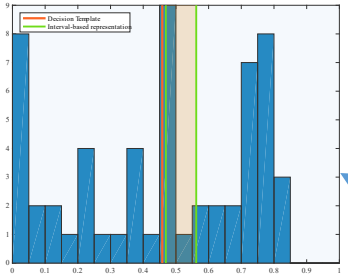


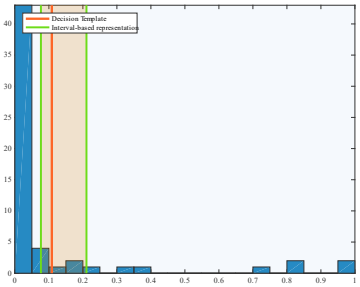


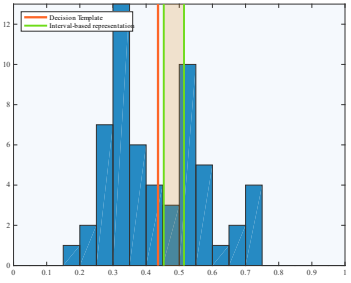


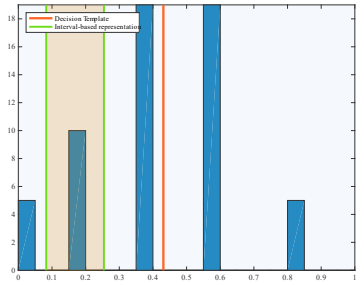






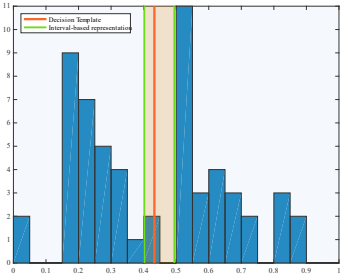


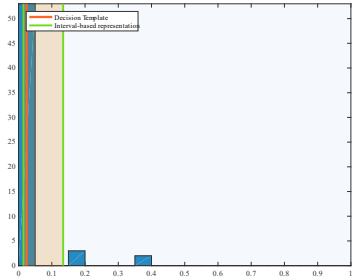


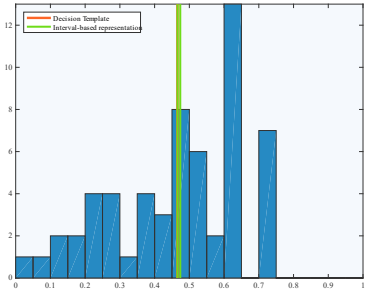


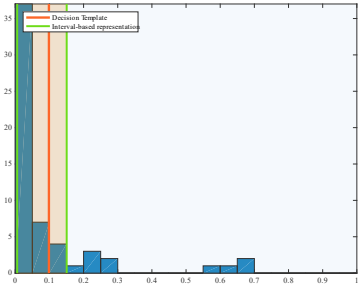
0.3917	0.5377	0.0706	0.4049	0.4539	0.1413	0.6000	0.4000	0.3917
0.2314	0.6850	0.0836	0.3009	0.5888	0.1103	0.6000	0.4000	0.2314
0.5793	0.3686	0.0521	0.5065	0.3253	0.1682	0.8000	0.0000	0.5793
…	…	…	…	…	…	…	…	..
0.2065	0.7459	0.0477	0.2687	0.6682	0.0631	0.4000	0.6000	0.2065
0.4114	0.5855	0.0032	0.5466	0.4177	0.0357	0.4000	0.6000	0.4114
0.4963	0.5023	0.0014	0.5433	0.4097	0.0470	0.0000	1.0000	0.4963

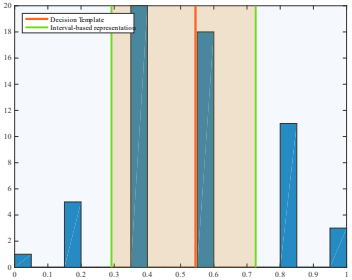
Meta-data associated with class 2











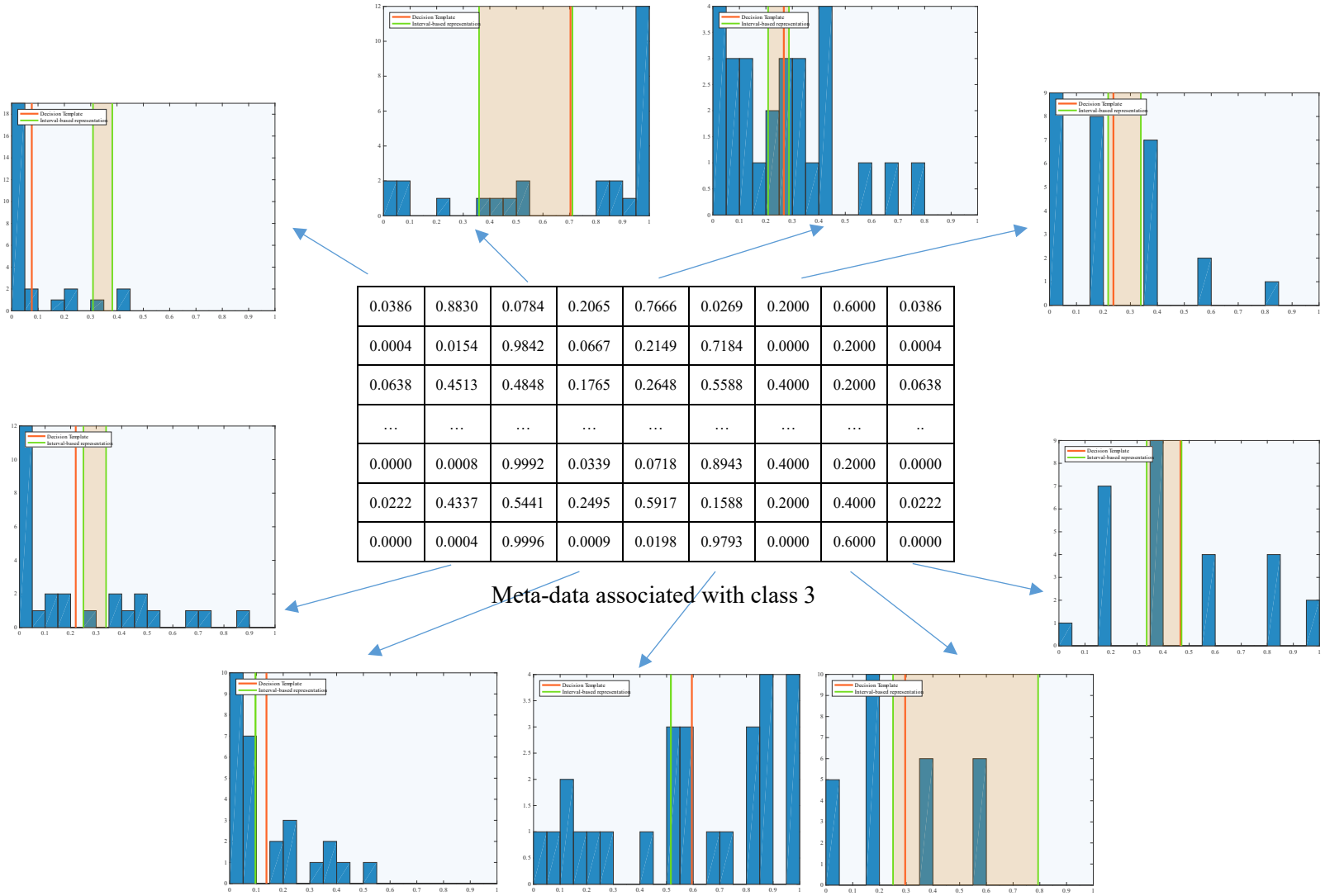


Fig.S2.Comparison between the representation of Decision Template method and proposed method on Hayes-Roth dataset

Table.S3. Ranking of benchmark algorithms and proposed method on each dataset for classification accuracy

	META-DES	GA Meta-data	ACO	Decision Template	Sum Rule	Ensemble (MLNN)	Ensemble (L2LSVM)	Proposed Method
Appendicitis	2	6	7	3	1	8	4	5
Artificial	5	4	3	7.5	1	2	7.5	6
Balance	8	2	4.5	4.5	7	1	6	3
Banana	7	4	8	5.5	2.5	2.5	5.5	1
Biodeg	5	8	7	3	4	6	1	2
Blood	4	5	8	7	2	6	1	3
Breast-Cancer	2	5	4	7	8	6	3	1
Cleveland	6	7	8	4	5	3	1	2
Contraceptive	5	8	7	4	1	6	2	3
Dermatology	8	7	6	2.5	2.5	5	2.5	2.5
Fertility	5	7	5	8	2	5	1	3
GM4	5	2	2	7	8	4	2	6
Haberman	3	6	7	8	1	2	5	4
Hayes-Roth	6	4.5	1	8	7	4.5	3	2
Heart	5	8	7	4	1	6	2.5	2.5
Hill-Valley	1	2	3	5	8	4	6	7
Iris	2	4.5	7	4.5	4.5	8	4.5	1
Led7digit	7	6	8	1	2	4	5	3
Madelon	7	3.5	3.5	1	8	6	2	5
Magic	8	7	4.5	4.5	6	3	1.5	1.5
Musk2	6	3	4	7	8	1	5	2
Newthyroid	6	1	2	7	8	4	5	3
Ring	1	6	5	7	8	3	4	2
Skin_NonSkin	6	3.5	3.5	7	8	1.5	5	1.5
Spambase	5	7	8	3	6	2	4	1
Twonorm	4	8	7	1.5	1.5	6	3	5
Vehicle	8	7	5	2	6	4	1	3
Waveform_w_Noise	4	8	7	5	6	1	2	3
Waveform_wo_Noise	4	8	7	5	6	2	1	3
Wdbc	6	4.5	7	3	4.5	8	1	2
Average	5.03	5.42	5.53	4.88	4.78	4.15	3.23	2.97

Table.S4. Ranking of benchmark algorithms and proposed method on each dataset for F1 score

	META-DES	GA Meta-data	ACO	Decision Template	Sum Rule	Ensemble (MLNN)	Ensemble (L2LSVM)	Proposed Method
Appendicitis	3	5	7	1	2	8	4	6
Artificial	8	4	3	6	1	2	7	5
Balance	6	3	5	1	8	2	7	4
Banana	8	6	7	4.5	3	2	4.5	1
Biodeg	4	7	8	3	5	6	1	2
Blood	4	8	7	1	2	6	5	3
Breast-ancer	2	5	4	7	8	6	3	1
Cleveland	4	6	7	1	8	3	2	5
Contraceptive	5	8	7	2	1	6	3	4
Dermatology	8	7	6	2.5	1	5	2.5	4
Fertility	7	4	2	8	5	1	3	6
GM4	5	2	2	7	8	4	2	6
Haberman	5	6	8	1	2	3	7	4
Hayes-Roth	5	4	1	8	7	6	2	3
Heart	5	8	7	4	1	6	2	3
Hill-Valley	1	2	6	3	8	4	5	7
Iris	2	4.5	7	4.5	4.5	8	4.5	1
Led7digit	8	6	7	1	2	4	5	3
Madelon	7	3.5	3.5	1	8	6	2	5
Magic	8	6	5	1	7	4	2	3
Musk2	6	2	4	7	8	1	5	3
Newthyroid	6	1	2	7	8	3	5	4
Ring	1	6	5	7	8	3	4	2
Skin_NonSkin	6	3.5	3.5	7	8	1.5	5	1.5
Spambase	5	7	8	3	6	2	4	1
Twonorm	4	8	7	1.5	1.5	6	3	5
Vehicle	8	6	5	2	7	4	1	3
Waveform_w_Noise	4	8	7	5	6	1	2	3
Waveform_wo_Noise	4	8	6	5	7	2	1	3
Wdbc	6	3	7	4	5	8	1	2
Average	5.17	5.25	5.47	3.87	5.20	4.12	3.48	3.45