IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 11, 2021, accepted January 25, 2021, date of publication March 2, 2021, date of current version March 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063254

DDES: A Distribution-Based Dynamic Ensemble Selection Framework

YE-RIM CHOI AND DONG-JOON LIM
Department of Industrial Engineering, Sungkyunkwan University, Suwon 16419, South Korea
Corresponding author: Dong-Joon Lim (tgno3@skku.edu)

This work was supported in part by the National Research Foundation of Korea (NRF), South Korea, under Grant 2020R1F1A1066629, and in part by the Korea Institute for Advancement of Technology (KIAT), South Korea, under Grant N0002429.

ABSTRACT Dynamic Ensemble Selection (DES) is a special type of ensemble modeling that selects different subsets of base classifiers for different test sample cases. In this process, multiple base classifiers are compared in terms of their competence as an ensemble to the best possible prediction for a given test sample case. Traditional DES methods rely on the Euclidean distance-based k-Nearest Neighbor (kNN) algorithm to identify the most relevant reference data points in such a way that the base classifiers correctly classifying them can be considered for an optimal ensemble. However, this approach is sensitive to the local structure of the data and the presence of noisy or irrelevant attributes. This study proposes a novel distribution-based DES (DDES) framework that takes the data structure into consideration when selecting reference data points. The experimental results for 30 classification problems indicate that the proposed approach yielded the best accuracy among the competing DES methods. Additionally, we discuss the correlation between the data complexity and the improvement in classification performance.

INDEX TERMS Multiple classifier systems, dynamic ensemble selection, region of competence, feature variance, data complexity.

I. INTRODUCTION

In multiple classifier systems, ensemble modeling seeks to make more accurate predictions by combining multiple classifiers in an attempt to reduce the potential bias from reliance on a single classifier. The typical ensemble approaches, such as bagging [1], boosting [2], and stacked generalization [3], achieve this by fostering diversity in the ensemble, and they generally lead to better decisions complementing the vulnerability of weak classifiers.

There are two distinct strategies in ensemble selection: static ensemble selection (SES) and dynamic ensemble selection (DES). SES builds an optimal ensemble for all test data by exploring multiple combinations of base classifiers, while DES assigns an individual ensemble for each test sample point by customizing the base classifiers. A key assumption in DES is that varying the decision boundaries of the base classifiers may result in their own regions of competence (RoC), i.e., confident regions for prediction in the feature space. Therefore, it is essential to properly evaluate base classifiers for a given test sample case.

In DES, the predictive performance of the resulting ensemble is significantly affected by the choice of reference data points. It should be pointed out that most DES methods employ the Euclidean distance-based k-Nearest Neighbor (kNN) algorithm as a means of identifying the reference data points. Consequently, the data structure is not accounted for when estimating the RoC. This may be trivial in cases where classes are readily separable and hence decision boundaries are constructed equally well by classifiers. However, as will be discussed in the following sections, an indecision region issue [4] occurs when data points are highly dispersed and/or overlapped. As a result, RoC estimation becomes highly ambiguous, which eventually undermines the overall ensemble performance.

This study presents a novel DES framework, referred to as distribution-based DES (DDES). Specifically, we introduce two DDES methods, DDES-I (Independent dispersion based DDES) and DDES-M (Mahalanobis distance based DDES), that can effectively estimate the RoC such that more

The associate editor coordinating the review of this manuscript and approving it for publication was Shunfeng Cheng.

relevant classifiers are involved in the construction of an ensemble. In particular, feature variances are accounted for independently in DDES-I, whereas covariance among features is utilized in DDES-M. Both methods are intended to enhance the predictive performance of the resulting ensemble by properly selecting reference data points for a given test sample point, especially in the presence of an indecision region.

The remainder of this paper is organized as follows. Section II reviews the related studies on ensemble methods. Section III highlights the motivation of this research. Section IV describes the computational procedures of the proposed methods. Section V presents the performance benchmarks for the proposed methods against other state-of-the-art dynamic selection methods. Section VI provides a discussion focusing on the relationship between data complexity and prediction performance. Finally, Section VII summarizes the findings and suggests directions for future work.

II. RELATED WORKS

A. STATIC ENSEMBLE SELECTION

SES employs certain searching methods in conjunction with selection criteria to identify the most suitable ensemble across the given data [5]. The searching methods include greedy search [6] and heuristic algorithms [7]. Recently, searching techniques such as the greedy iterative optimization method [8] have been proposed for balancing the diversity and the individual classifier accuracy, as well as for searching the optimal number of classifiers and feature subsets using various metaheuristic algorithms [9].

As selection criteria in SES, diversity in the ensemble has attracted much research attention. In particular, data diversity [10], [12], model structure diversity [12], and hyperparameter diversity [13] can be considered to improve the performance of the ensemble [14], [15]. Two different aspects of diversity were also addressed in the literature: pairwise diversity and overall diversity. [16] The former is focused on a pair of classifiers [17]–[19], while the latter seeks diversity presented by a whole ensemble [20], [21]. Various attempts were made to obtain the best possible diversity [22] such as tuning learning algorithms [23], [24], sampling methods [25], [26], and feature selection [27]–[30].

Recent advancements utilizing the concept of diversity also include applying classifier overlapping indexes [31], attentional mechanism-based explicit measures [32], multimodal perturbation-based ensemble algorithms with progressive kNN classifiers [33], ensemble clustering algorithms with diversity based on the normalized mutual information [34], and the ensemble pruning method using margin and diversity [35].

However, it is debatable whether a certain selection criterion can be used to yield a better ensemble performance in SES. For example, some researchers argued that combining heterogeneous classifiers does not guarantee that they complement each other [36]–[38]. Moreover, one argued that the impact may exist, but an immediate cause may not [39], or the performance does not monotonically improve in proportion to the diversity measures [40], [41]. See more discussion in [16], [22], [42]–[44].

B. DYNAMIC SELECTION

Unlike SES, dynamic selection (DS) approaches assign a (set of) classifier(s) to a given test sample case by estimating the RoC where a group of data points identified to be relevant to the test sample case lies. Specifically, two distinct DS approaches exist: Dynamic Classifier Selection (DCS) and Dynamic Ensemble Selection (DES). The former chooses the most competent classifier, whereas the latter discovers a subset of base classifiers most suitable for a given test sample case. Table 1 lists the most well-known DS methods.

Local Classifier Accuracy (LCA) utilizes a posteriori-probability to obtain the percentage of correct classifiers within the local region [45]. Specifically, this method compares base classifiers by how well they predict data points in RoC whose class label is the same as the test sample case. In contrast, Overall Local Accuracy (OLA) adopts an a priori-probability to choose the most suitable classifier based on how each classifier correctly predicts all points in the RoC.

Multiple Classifier Behavior (MCB) takes a behavior-based DCS approach that calculates the value of the similarity function of the output profiles of the classifiers [46]. Within this process, data points in RoC are evaluated according to the similar behavior of classifiers with a test sample case. The most competent classifier is then selected by applying OLA to the qualified data points.

In the DES approach, RoC defines a validation space where multiple classifiers are examined. K-Nearest-ORAcles-Union (KNORA-U) and K-Nearest-ORAcles-Eliminate (KNORA-E) are two popular Oracle-based methods that discover a subset of base classifiers to correctly predict the unknown pattern in RoC [47]. In particular, KNORA-U involves all classifiers that correctly predict at least one data point in RoC to an ensemble and each classifier submits the votes by the number of points that it predicts correctly, while KNORA-E only chooses ones that correctly predict all data points in RoC. They both employ kNN to define the RoC. If a test sample case is in an indecision region (that is, no classifier is selected), KNORA-E adjusts the value of k until a certain number of classifiers is chosen.

Meta learning-DES (META-DES) introduces meta-classifiers to further assess the level of competence of a classifier based on meta-features extracted from the training data especially when there is low consensus among base classifiers [48]. The meta-features include a neighbor's hard classification, posterior probability, OLA, output profile classification, and classifier confidence. During the generalization phase, the meta-classifier estimates whether a base classifier is competent enough to be added to the ensemble.

TABLE 1. Dynamic selection (DS) method variants.

Method	Type	Characteristic	Reference
LCA	DCS	Accuracy-based	[45]
OLA	DCS	Accuracy-based	[45]
MCB	DCS	Behavior-based	[46]
KNORA-U	DES	Oracle-based	[47]
KNORA-E	DES	Oracle-based	[47]
META-DES	DES	Meta learning-based	[48]
DES-P	DES	Probability-based	[49]

DES-Performance (DES-P) is a probability-based method using weighted kNN that considers an absolute standard expected from a random classifier that draws a class label using a uniform distribution [49]. The competence of a base classifier is defined as how much improvement is made as compared to a random classification. A base classifier is qualified to be included in the ensemble only when the competence value is positive.

See more details for the DES techniques and their categorization describing the definition of RoC, competence estimation, and selection criteria in [50], [51].

C. REGION OF COMPETENCE

In recent years, various methodological advancements have been made to enhance the DES approaches by modifying the RoC definition. There are two primary topics to better define the RoC: noisy points and overlapped decision boundaries.

In [52], a new way of estimating the competence of classifiers was proposed to address the class imbalance problem. Moreover, to address the problem of borderline samples in the local region, the Frienemy Indecision REgion-DES (FIRE-DES) method introduces a notion of ''frienemy'' by pre-selecting classifiers that correctly classify at least one pair of samples from different classes [53].

FIRE-DES++ is an extension of FIRE-DES to ameliorate the noise sensitivity and indecision region problems [54]. It removes the noises and reduces the overlap of classes in the validation set and then applies K-Nearest Neighbors Equality (KNNE) to define a more balanced RoC.

DES-hesitant utilizes the concept of a hesitant fuzzy set as an RoC [55]. It constructs the Hesitant Fuzzy Decision Matrix (HFDM) consisting of several selection criteria and techniques, such as the accuracy, fraction-based method, ranking-based method, and potential function method. In dynamic selection procedure, the classifiers whose competence levels are more than a threshold are selected.

An online local pool generation method deals with the problem of overlapped samples [56]. After the online phase obtains the RoC by kNN to the training set, it defines whether a test sample point is in an indecision region or not by considering the hardness value calculated from the offline phase. If yes, it generates a local pool using Self-Generating Hyperplanes (SGH) method, and the kNN rule is used otherwise.

The oracle-based DES method using a discriminant index, which is used in the Item and Test Analysis (ITA), shows promising results [57]. To compose the RoC, it selects the k most discriminant instances among the validation data by ranking them, starting with the double size of kNN.

In [58], a novel dynamic ensemble outlier detection model was proposed to ensemble one-class classifiers with the adaptive kNN rule using Support Vector Data Description (SVDD) to estimate the RoC. The competence level of classification is estimated by the posterior probabilities, and then two non-parametric statistical tests, the Friedman test and Nemenyi test, are used to make a final decision.

Graph-based Dynamic Ensemble Pruning (GDEP) addresses the problem of sensitivity in the classifier selection process by building the must-link and cannot-link graphs whereby the level of competence is measured [59]. To better choose the proper neighborhood, this approach considers the statistics of classifiers' behavior toward data as a form of the Behavior-based Geodesic Matrix (BGM).

To tackle the noisy data problem, a prototype selection technique was proposed by applying novel kNN algorithms [60]. This approach effectively reduces the high degree of overlap between two classes, making decision boundaries more discriminative. The performance using both Edited Nearest Neighbor (ENN) and adaptive kNN has proven to yield competent results.

In [61], the Discriminant Adaptive Nearest Neighbor (DANN) was developed to take into account the shape and size of the RoC. It adopts an ellipsoidal RoC whose secondary axis is nearly orthogonal to the decision boundary between the two classes. The number of neighborhood points is dynamically chosen by using a posteriori probability.

An enhanced Differential Evolution (DE) algorithm with KNORA-E was shown to improve the classification performance [62]. This approach automatically generates a pool of diverse classifiers, while removing noisy samples in the validation data.

In addition, there is a context-based framework that exploits the output profiles from the validation set to improve the performance of Dos Santos Approach (DSA) [63]. In this approach, the best ensemble of classifiers is dynamically chosen and then the switch mechanism identifies whether the decisions are confident enough after generating RoC with k most similar output profiles.

III. MOTIVATION

DES is based on the premise that the closer data points are to one another in the feature space, the more likely are they to share aptly applicable classifiers. This is intuitive, especially when classes are readily separable, and hence decision boundaries are constructed equally well by classifiers. However, when data points are highly dispersed and/or overlapped, the problem of an indecision region may arise owing to the ambiguous decision boundaries, which may even be

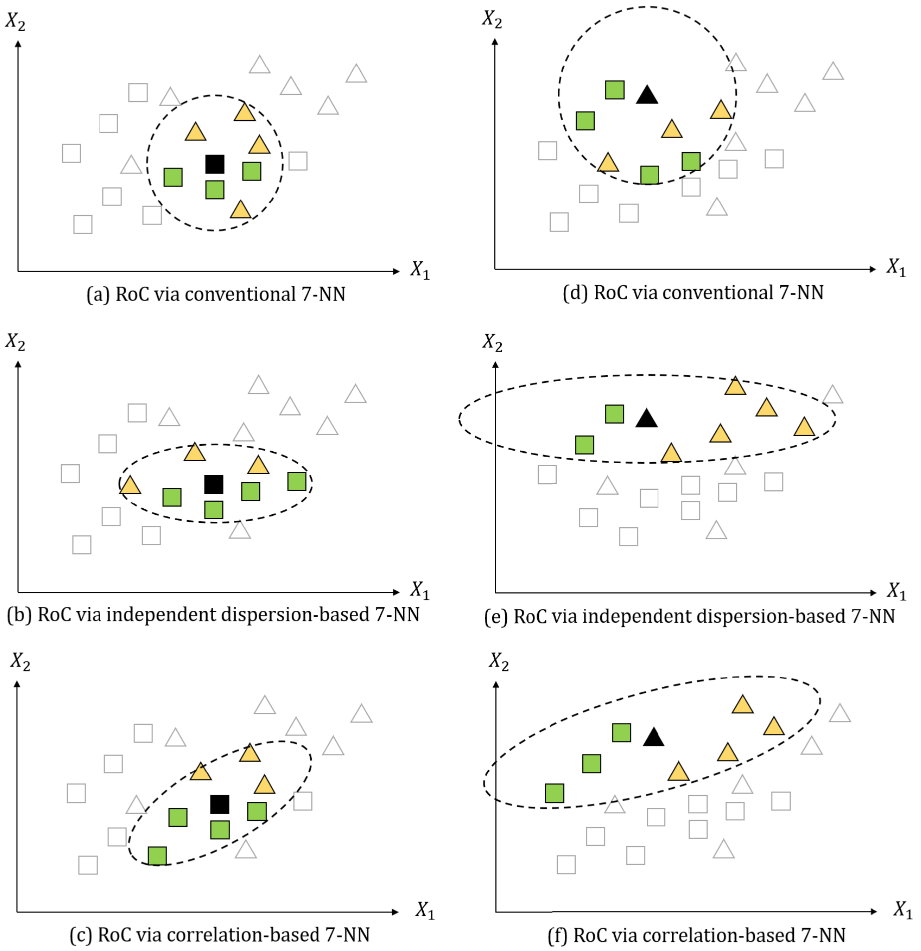

X_2
X_1
(a) RoC via conventional 7-NN
(d) RoC via conventional 7-NN
(b) RoC via independent dispersion-based 7-NN
(e) RoC via independent dispersion-based 7-NN
(c) RoC via correlation-based 7-NN
(f) RoC via correlation-based 7-NN

FIGURE 1. **Illustration of the differences between the RoC estimated by the conventional 7-NN, independent dispersion-based 7-NN, and correlation-based 7-NN.**

inconsistent across classifiers [64]. In the context of DES, this implies that the predictive performance is largely affected by the choice of reference data points.

It should be recognized that most DES methods employ kNN as a means of identifying reference data points for the test sample case. More importantly, the Euclidean distance is commonly used as a distance metric and, consequently, the data structure is not accounted for when estimating the RoC. Moreover, the existing DES methods inherit the other drawbacks of kNN; namely, they are sensitive to the local structure of the data [65] as well as to the presence of noisy and/or irrelevant attributes [66], [67].

Figure 1 illustrates the differences between the RoC estimated by the conventional 7-NN, i.e., (a) and (d), the RoC estimated by independent dispersion-based 7-NN, i.e., (b) and (e), and the RoC estimated by correlation-based 7-NN, i.e., (c) and (f). The two class labels of the validation data are marked with "Δ" and "□"; the markers filled with black represent the test sample cases. Those with colors are reference data points within RoC which are indicated as a dotted outline.

The figure demonstrates typical situations of indecision regions consisting of dispersed data. Note that the unequal as well as correlated variances presented in X_1 and X_2 make key differences in the similarity assessment of the test sample case relative to the validation data. Specifically, it is reasonable to expect that the reference data points with the same class labels as the test sample case are more likely to be included in the RoC when feature variances are considered in the similarity measure. With more representative reference data points, this approach is more likely to construct an ensemble that can correctly classify the test sample case without engaging classifiers fitted to heterogeneous data points. In contrast, the conventional kNN based on the Euclidean distance can mislead the RoC because this distance by its nature does not always correspond to the likelihood of the appearance of homogeneous data points. What is needed is a more versatile RoC estimation process that can consider both how distant a test sample case is from each validation point, and how the rest of the data points vary. It should also be noted that the distribution-based RoC (either the independent dispersion-based consideration or the correlation-based consideration) can be readily integrated into the existing DES framework. That is, the conventional kNN-based RoC can be viewed as a special case when features are assumed to have equal and independent variance.

In this study, we propose a DDES framework by introducing two distinct RoC estimation approaches: one with independent dispersion consideration (DDES-I) and the other that considers the correlation by adopting the Mahalanobis distance (DDES-M). Both of these can effectively estimate the RoC such that more relevant classifiers are involved in the construction of an ensemble. As discussed in the following sections, this is enabled by taking the data distribution into consideration when selecting the reference data points for the test sample case. The aim of this study is both to present algorithmic procedures of the proposed methods and their comparative classification performances against the existing methods, and to provide a discussion of the relationship between data complexity and the behavior of the new methods to give insights into possible extensions of the current specifications.

IV. METHODOLOGY

Following the conventions of the DES literature, let us first define notation as follows:

- The original dataset is partitioned into a training set (D^{Train}), a dynamic selection set (D^{Sel}), and a test set (D^{Test}).
- R_i denotes an RoC identified for a test sample case (x_i^{Test}) consisting of reference data points (x_j^{Sel}) in D^{Sel}.
- C_i indicates a set of competent classifiers for a test sample case (x_i^{Test}) in D^{Test}.
- K specifies the minimum size of R_i, which can also be dynamically adjusted by the algorithm.
- $Acc_{Threshold}$ is the minimum accuracy of a classifier required to be included in C_i.

The DDES framework is divided into four phases as summarized in Figure 2: 1) the training phase where a pool of classifiers (C) is trained using D^{Train}; 2) the RoC estimation phase where R_i is determined as a set of x_j^{Sel}; 3) the DES phase where C_i is composed of classifiers that yield classification accuracy greater than or equal to $Acc_{Threshold}$ on R_i; and 4) the prediction phase where C_i is applied to x_i^{Test} and a majority vote gives the classification.

1) *Training phase:* In this phase, D^{Train} is used to generate a pool of classifiers, among which some will be adopted as competent classifiers for x_i^{Test}. A diverse set of classifiers is generally sought to take advantage of divergent decision boundaries. To achieve this, various kinds of base classifiers in conjunction with a wide range of hyperparameters may be trained using D^{Train}. Additionally, the feature variances (σ) presented in D^{Train} are obtained to account for the data distribution in the RoC estimation phase.
2) *RoC estimation phase:* This phase determines an RoC for each x_i^{Test}. As discussed already, DDES is characterized by incorporating the data distribution into the selection of reference data points. There can be different strategies to achieve this; in this paper, we introduce two approaches: DDES-I and DDES-M.

Algorithm 1 RoC Estimation for x_i^{Test} in DDES-I

```
Input
    Data: $D^{Sel}$, and $x_i^{Test} \in D^{Test}$
    Parameters: $K$ and $\Delta l_i$
    Standard deviations of features observed in $D^{Train}$: $\sigma$
Procedure
    Set initial lengths of axes: $l_i = \sigma$
    while $l_i < 3\sigma$ do
        Set an initial RoC: $R_i = \emptyset$
        for all $x_j^{Sel} \in D^{Sel}$ do
            if $\sum_i (\frac{x_j^{Sel} - x_i^{Test}}{l_i})^2 \leq 1$ then
                $R_i = R_i \cup \{x_j^{Sel}\}$
            end if
        end for
        if $|R_i| < K$ then
          $l_i = l_i + \Delta l_i$
        end if
    end while
Output
    RoC: $R_i$
```

- *DDES-I:* In this approach, feature variances are accounted for independently, and RoC is essentially defined as an ellipsoidal form with axes represented by the corresponding variances of D^{Train} along each feature (see Algorithm 1). To elaborate, the initial size of the ellipsoid is determined by the feature variances (σ), and the ellipsoid is enlarged by updating the lengths of the axes at the learning rate (Δl_i) until the required number (K) of reference data points (x_j^{Sel}) is included in the RoC (R_i) (see Figure 3). The updating procedure terminates once the required number of reference points (K) are included in the RoC, that is the number of points larger than K can be included in R_i. One may notice that the resulting RoC will retain axes parallel to the feature axes. That is, it is presumed that features may have unequal variances, but they are not correlated. It is also of note that reference data points collectively falling within three standard deviations are only considered to minimize the impact of outlying data points.
- *DDES-M:* In practice, many datasets exhibit correlations as well as differing levels of variance between features. DDES-M attempts to address this by accounting for the variance structure (see Algorithm 2). In particular, the Mahalanobis distance [68] is employed so that the covariance among features is reflected in the distance calculation. Consequently, the resulting RoC is typically an ellipsoid constructed in a principal component

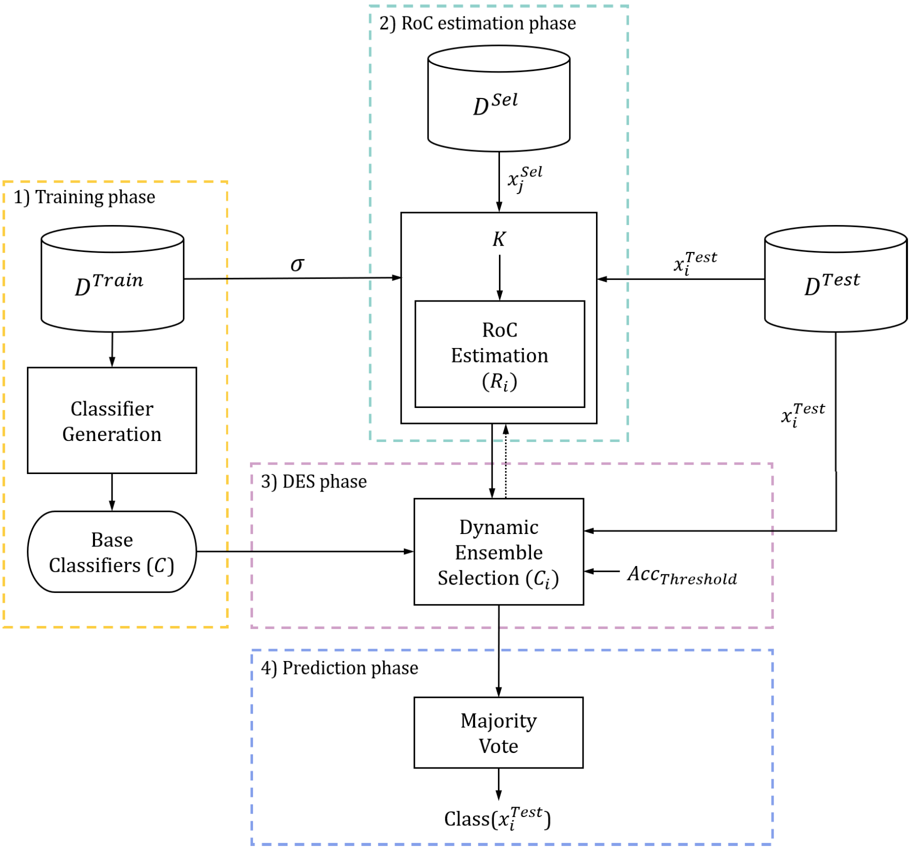


FIGURE 2. Overview of the DDES framework.

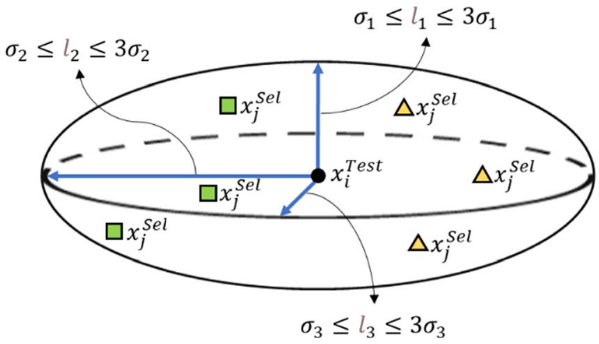

$\sigma_1 \leq l_1 \leq 3\sigma_1$
$\sigma_2 \leq l_2 \leq 3\sigma_2$
x_j^{Sel}
x_j^{Sel}
x_i^{Test}
x_j^{Sel}
x_j^{Sel}
x_j^{Sel}
x_j^{Sel}
$\sigma_3 \leq l_3 \leq 3\sigma_3$

FIGURE 3. Schematic illustration of the RoC estimated in DDES-I.

subspace. When the features are completely uncorrelated, i.e., $S^{-1} = 1$, the principal component axes will correspond to the feature axes, thus this method becomes identical to the conventional kNN-based DES method.

3) *DES phase:* Next, we proceed to the DES phase. As summarized in Algorithm 3, a set of competent classifiers (C_i) is dynamically selected for each test sample point (x_i^{Test}). Whether a classifier is competent is determined by its accuracy when applied to R_i. If the accuracy of a classifier is greater than or equal to $Acc_{Threshold}$, it is included in the ensemble (C_i). For example, suppose there are 10 reference data points in R_i and the $Acc_{Threshold}$ is 0.8. The C_i comprises base classifiers that correctly classify at least eight points in the R_i. In a case where no classifier meets the accuracy threshold, the value of K is increased and re-estimate the RoC until there is at least one classifier in C_i.

Algorithm 2 RoC Estimation for x_i^{Test} in DDES-M

```
Input
    Data: $D^{Sel}$ and $x_i^{Test} \in D^{Test}$
    Parameters: $K$
    Covariance matrix of $D^{Sel}$: $S$
Procedure
    Set an initial distance set: $Dist_i = \emptyset$
    for all $x_j^{Sel} \in D^{Sel}$ do
        $Dist_i = Dist_i \cup \sqrt{(x_i^{Test} - x_j^{Sel})S^{-1}(x_i^{Test} - x_j^{Sel})}$
    end for
    Arrange $Dist_i$ in ascending order
    Assign $x_j^{Sel}$ to $R_i$ where $j$ is in the first $K$ elements of $Dist_i$
Output
    RoC: $R_i$
```

4) *Prediction phase:* Lastly, in the prediction phase, multiple predictions from the set of competent classifiers (C_i) on x_i^{Test} are collected, and then a simple majority

Algorithm 3 Dynamic Ensemble Selection

```
Input
    Data: $R_i$
    Parameters: $Acc_{Threshold}$
    Classifiers: $C$
Procedure
    for all $i$ do
        Set an initial set of competent classifiers: $C_i = \emptyset$
        for all $c_k \in C$ do
            Compute the accuracy of $c_k$ on $R_i$: $Acc(c_k)$
            if $Acc(c_k) \geq Acc_{Threshold}$ then
                $C_i = C_i \cup \{c_k\}$
            end if
        end for
        if $C_i$ is empty then
            $K = K + 1$
            Go to the RoC estimation phase
        end if
    end for
Output
    Set of competent classifiers: $C_i$
```

vote is applied to determine the final prediction. In the case of a tie, a random prediction is made as in the conventional DES specification [52], [54]. Obviously, this setting can be revised, e.g., by a weighted vote, as in recent studies [74]–[76].

V. EXPERIMENTAL EVALUATION

In this section, we present the comparative classification performances of the proposed methods against well-known DS methods by applying them to multiple datasets.

A total of 30 datasets were taken from the UCI machine learning repository and Kaggle benchmark repository. The datasets consist of 23 binary classification problems and seven multiclass problems (see Table 2). The last two columns contain the class separability metric, namely the N2 measure, and the level of unequal variance.

The class separability metric represents the ratio of the sum of the intra-class distances to the sum of the inter-class distances [72]. As the measure compares the intra-class dispersion with the inter-class separability, a larger value generally indicates a higher level of difficulty in classification owing to a lower degree of separability.

To verify the impact of unequal variance to the classification performance across the datasets, we first applied a min-max rescaling so that the feature ranges were consistently scaled to the range between 0 and 1, and then we obtained the ratio of the maximum variance to the minimum variance across the features in each dataset. The resulting values therefore indicated how representative unequal variances are in the datasets.

The pool of base classifiers was composed of total 51 varying classifiers: 11 Neural Networks, 12 Support Vector Machines, six Decision Trees, five Logistic Regressions, five Gradient Boosts, six kNNs, five XGBoosts, and a Naïve Bayes model (see Table 3).

TABLE 2. Description of the dataset.

Data name	Instances	Features	No. of class	N2	Unequal Variance
Mammographic	961	5	2	0.27	2.01
Somerville	143	6	2	0.43	0.30
Heart	303	13	2	0.40	1.20
Monk	1711	6	2	0.37	0.26
B.C. Coimbra	116	9	2	0.46	0.45
Banknote	1371	4	2	0.10	0.15
Liver	345	6	2	0.46	0.31
B.C. Diagnostic	569	30	2	0.36	0.94
Hungarian heart disease	294	13	2	0.38	1.68
B.C. Wisconsin	683	9	2	0.17	0.65
Pima	768	8	2	0.45	0.45
Biodegradation	1055	41	2	0.35	2.08
Spine	310	12	2	0.48	1.06
Vertebral	310	6	2	0.41	0.62
Phoneme	5404	5	2	0.25	0.22
Plrx	181	12	2	0.46	0.32
Ilpd	579	9	2	0.44	1.72
Haberman	306	3	2	0.40	0.66
Parkinson	195	22	2	0.32	0.58
Wpbc	194	33	2	0.48	0.85
Hepatitis	155	19	2	0.36	1.12
Spect	267	22	2	0.40	0.34
Admission	500	7	2	0.26	0.82
Phishing	1353	9	3	0.17	0.37
Wine	178	13	3	0.36	0.50
Iris	150	4	3	0.20	0.49
Newthyroid	215	5	3	0.25	0.48
Vehicle	846	18	4	0.44	0.91
Glass	214	9	6	0.39	0.97
Yeast	1484	8	10	0.47	0.58

The experiment was conducted in a Python environment, and the pool of base classifiers was generated using the Scikit-learn package [73] except for XGBoost. Additionally, the five DS methods were implemented using DESlib [74].

We compared the proposed methods with seven state-of-the-art dynamic selection methods: three DCS methods (LCA, OLA, and MCB) and four DES methods (KNORA-U, KNORA-E, META-DES, and DES-P). Following the conventions in the literature, we designed the experiment as follows.

- The standard stratified random split was conducted to perform the holdout method, and each dataset was randomly partitioned into three sets, a training set (50%), a dynamic selection set (20%), and a test set (30%) in such a way that the ratios of each class were maintained. Validation was repeated 100 times to obtain an average accuracy to mitigate variations in the random splits.
- For the sake of simplicity, hyperparameters for base classifiers were randomly selected from Table 3 rather than tuned in each iteration. This makes it possible for diverse weak classifiers to be involved in the ensemble across the benchmark methods. See similar experimental designs in [49], [75], [76].

TABLE 3. Base classifiers and hyperparameter pools.

Base classifier	Hyperparameter pool

Neural Network	Activation function ∈ {'relu', 'tanh', 'logistic'}
	Hidden layer size ∈{(10), (10, 10), (10, 10, 10), (100), (100, 100)}
	Solver ∈ {'lbfgs', 'sgd', 'adam'}

Support Vector Machines	C = [1, 15]
	γ = [0.01, 1]
	Kernel ∈ {'linear', 'rbf'}

Decision Tree	Max features ∈ {'auto', 'sqrt', 'log2'}
	Criterion ∈ {'entropy', 'gini'}
	Min samples split = 2
	Min samples leaf = 1

Logistic Regression	Solver ∈ {'lbfgs', 'newton-cg', 'liblinear', 'sag', 'saga'}

kNN	Neighbors ∈ {3, 5, 7, 9}
	Weights ∈ {'uniform', 'distance'}

Gradient Boost	Learning rate = 0.1 Min samples split = 2 Min samples leaf = 1
XGBoost	Booster ∈ {'gbtree', 'gblinear', 'dart'} $\eta \in \{0, 1\}$ $\gamma \in \{0.01, 1\}$
Naïve Bayes	None

- For the sake of consistency, common parameters were selected based on the best overall performance: K was set to 7 and $Acc_{Threshold}$ was set to 0.3, both of which yielded the best average classification accuracy obtained across the datasets. Likewise, method specific parameters were selected on the basis of the resulting performance. K_p and h_c are META-DES parameters that were set to 5 and 1, respectively, and Δl_i is a DDES-I parameter that was set to $0.1 \times \sigma$. See similar settings in [48], [53], [54].
- Following guidelines in the literature [53], [54], [78], the Wilcoxon signed rank test with an adjusted alpha risk of 0.0014 ($\approx 0.05/36$) was used to determine whether the differences in classification performance of competing methods were statistically significant.

Table 4 presents the comparative classification accuracies of the seven benchmark methods in conjunction with those of the proposed methods.[1] Note that the accuracies were obtained by averaging the results. The best results for each dataset are highlighted in bold, and statistically significant differences compared with the other methods are marked with the symbol •. The last two rows contain the average accuracies and average ranks of each method across 30 datasets.

On average, DDES methods outperformed the others. In particular, DDES-M was found to be most compatible with our datasets, achieving the highest mean accuracy of 0.8309 along with the mean rank of 2.0333. Specifically, DDES-I showed more accurate classification results than other methods in 10 datasets, among which half of them exhibited statistically significant differences. DDES-M was superior to the others in 14 datasets, among which eight of them were found to show statistically significant differences. In fact, DDES methods were surpassed (in a statistical sense) only in four datasets: Banknote, Phoneme, Haberman, and Phishing.

It is seen that DCS methods were generally outperformed by DES methods, which is consistent with the literature [50], [51]. The accuracy gap was particularly wider when the dataset was hard to classify, with B.C. Coimbra, Liver, and Glass being typical cases in point. In contrast, the DDES showed good performance for those datasets as well. We will investigate the relationship between data complexity and classification performance in the following section.

To further examine the classification performance of competing methods, the Friedman omnibus test was first conducted, and then a *post-hoc* Wilcoxon rank test was employed. It was verified from the results that nine methods indeed performed unequally (p-value $< 2.2e^{-16}$), and, inter alia, DDES methods outperformed the others while there was no statistically significant difference between DDES-I and DDES-M (see Table 5). Figure 4 is a critical difference diagram summarizing the *post-hoc* test results. Note that the mean rank is plotted on the horizontal axis, and methods connected by a solid line are on par with each other in a statistical sense. In conclusion, the data distribution considered in the DES framework was effective, while the conventional DES methods, particularly DES-P, META-DES, and KNORA-U yielded relatively more accurate classification performances than the DCS methods (LCA, OLA, and MCB) in our experiment.

VI. DISCUSSION

In this section, we carry the analysis a stage further by examining the relationship between the data complexity and the classification performance. We specifically verified the impact of the unequal variance and class separability metric (N2) on the classification performance across the datasets.

Figure 5 illustrates the distribution of winning methods corresponding to the degree of unequal variance. The area indicates the proportion that each method yielded the best classification performance, e.g. DDES-M outperformed the other methods about 20% in a repeated experiment on a dataset whose degree of unequal variance was 0.20 (scaled). It is generally seen that DDES methods were dominant even in the presence of a high level of unequal variances; they cumulatively accounted for more than 60% of dominance even in the most extreme case of the Biodegradation dataset whose feature variances range from 0.001 to 141.378 (scaled to 1.90 in the figure).

It is interesting to point out that DDES methods surpassed the other methods particularly when the dataset features high-dimensionality, with B.C. Diagnostic, Biodegradation, Wpbc, Hepatitis, Spect, and Vehicle being typical cases in point (see Table 2). This indicates that DDES methods are capable of handling complex data in terms of dimensionality as well as variance structure.

[1] The results presented here can be reproduced at https://github.com/yerimchoi/IEEE.2020.DDES

TABLE 4. **Performance benchmarks.**

Dataset	LCA	OLA	MCB	KNORA-U	KNORA-E	META-DES	DES-P	DDES-I	DDES-M
Mammographic	0.8122	0.8092	0.8139	0.8193	0.8166	0.8232	0.8327	**0.8356**	0.8239
Somerville	0.5708	0.5733	0.5714	0.5907	0.5894	0.5711	0.5928	0.5969	**0.6114**
Heart	0.7734	0.7833	0.7699	0.8185	0.7770	0.7893	0.7974	**0.8457 •**	0.8387
Monk	0.6382	0.6413	0.6245	0.6369	0.6127	0.6252	**0.6461**	0.6451	0.6432
B.C. Coimbra	0.6103	0.6541	0.6555	0.7108	0.6514	0.6610	0.6697	0.7669	**0.7676 •**
Banknote	0.9831	0.9953	0.9961	0.9816	0.9994	**0.9999 •**	0.9990	0.9998	0.9998
Liver	0.6241	0.6583	0.6666	0.6830	0.6326	0.6907	0.7114	0.7340	**0.7413 •**
B.C. Diagnostic	0.9653	0.9654	0.9538	0.9766	0.9656	0.9687	0.9661	**0.9819 •**	0.9756
Hungarian heart disease	0.8059	0.8005	0.7935	0.8227	0.7900	0.7991	0.8123	0.8132	**0.8274**
B.C. Wisconsin	0.9661	0.9660	0.9615	0.9662	0.9650	0.9681	0.9691	0.9692	**0.9724**
Pima	0.7277	0.7291	0.7379	0.7677	0.7306	0.7552	0.7634	**0.7733**	0.7693
Biodegradation	0.8413	0.8680	0.8572	0.8642	0.8614	0.8602	0.8665	**0.8830 •**	0.8808
Spine	0.7600	0.7885	0.7932	0.8289	0.7997	0.8260	0.8056	0.8453	**0.8513 •**
Vertebral	0.8078	0.8174	0.8056	0.8558	0.8038	0.8572	0.8267	0.8686	**0.8759 •**
Phoneme	0.7814	0.8675	0.8716	0.7496	0.8768	**0.8841 •**	0.8761	0.8701	0.8752
Plrx	0.7037	0.6320	0.6311	0.6857	0.6413	0.6763	0.7037	**0.7061**	0.7020
Ilpd	0.7062	0.6923	0.6594	0.7106	0.6871	0.6979	0.7139	0.7286	**0.7350 •**
Haberman	0.7395	0.7168	0.7027	**0.7494 •**	0.7097	0.7179	0.7305	0.7318	0.7461
Parkinson	0.8655	0.9067	0.8986	0.8454	0.9139	**0.9190**	0.9114	0.9080	0.9039
Wpbc	0.7359	0.7373	0.7220	0.7836	0.7445	0.7651	0.7671	0.8122	**0.8157 •**
Hepatitis	0.7713	0.7915	0.7759	0.8155	0.8369	0.7885	0.8015	**0.8508 •**	0.8431
Spect	0.7709	0.7973	0.7863	0.8226	0.8127	0.8261	0.8212	**0.8499 •**	0.8319
Admission	0.9316	0.9311	0.9285	0.9459	0.9254	0.9438	0.9486	0.9484	**0.9494**
Phishing	0.8308	0.8696	0.8765	0.8323	0.8834	**0.8871 •**	0.8792	0.8788	0.8781
Wine	0.9800	0.9707	0.9582	0.9750	0.9862	0.9784	0.9784	**0.9876**	0.9807
Iris	0.9303	0.9558	0.9397	0.9630	0.9418	0.9447	0.9479	**0.9684**	0.9661
New thyroid	0.9085	0.9394	0.9317	0.9488	0.9435	0.9481	0.9478	0.9709	**0.9720 •**
Vehicle	0.7580	0.7999	0.7809	0.7831	0.7956	0.7967	0.7874	0.7891	**0.8008**
Glass	0.6193	0.6887	0.6998	0.6265	0.7030	0.7181	0.7204	0.7237	**0.7339**
Yeast	0.5736	0.5724	0.5612	0.5923	0.5673	0.5876	0.6107	0.6089	**0.6154 •**
Mean Accuracy	0.7831	0.7973	0.7908	0.8051	0.7988	0.8092	0.8135	0.8294	**0.8309**
Mean Rank	7.4167	6.4667	7.6667	4.7333	6.1000	4.5500	3.8000	2.2333	**2.0333**

TABLE 5. ***Post-hoc* test (Wilcoxon) results (*p*-value).**

	LCA	OLA	MCB	KNORA-U	KNORA-E	META-DES	DES-P	DDES-I	DDES-M
LCA	-	0.0030	0.1226	0.0002	0.0088	0.0001	0.0000	0.0000	0.0000
OLA		-	0.0011	0.0214	0.2264	0.0003	0.0002	0.0000	0.0000
MCB			-	0.0059	0.0014	0.0000	0.0000	0.0000	0.0000
KNORA-U				-	0.0489	0.1747	0.2652	0.0000	0.0000
KNORA-E					-	0.0003	0.0003	0.0000	0.0000
META-DES						-	0.0209	0.0000	0.0000
DES-P							-	0.0001	0.0000
DDES-I								-	0.2142
DDES-M									-

As discussed earlier, the problem of an indecision region arises particularly when the data are highly dispersed, and consequently, heterogeneous classes are mixed within RoC.

Figure 6 illustrates the distribution of the winning methods corresponding to the class separability metric, i.e., the N2 measure. Likewise, predominance of the DDES methods is apparent, and it is particularly notable that this superiority is more prominent in more complex datasets. That is, the proposed methods provided a robust framework to properly select reference data points even in the presence of an indecision region which, in contrast, generally undermined the classification performance of the other competing methods.

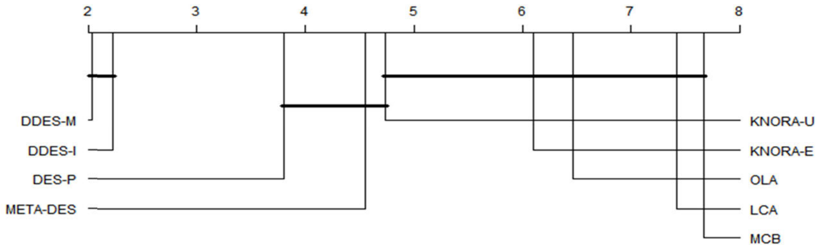


FIGURE 4. Critical difference diagram.

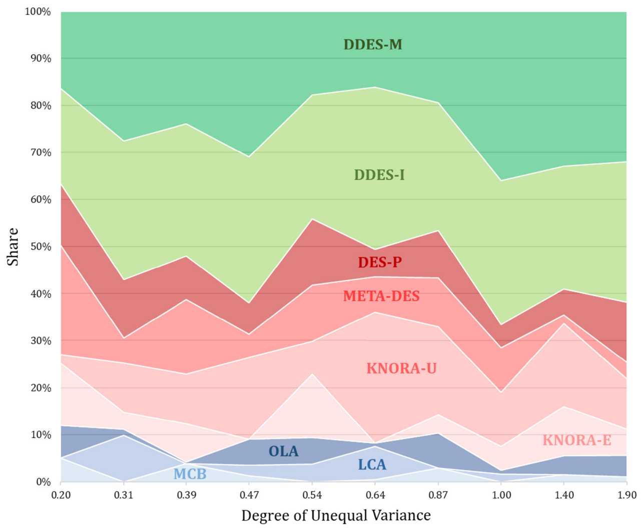


FIGURE 5. Distribution of the winning methods over the degree of unequal variance.

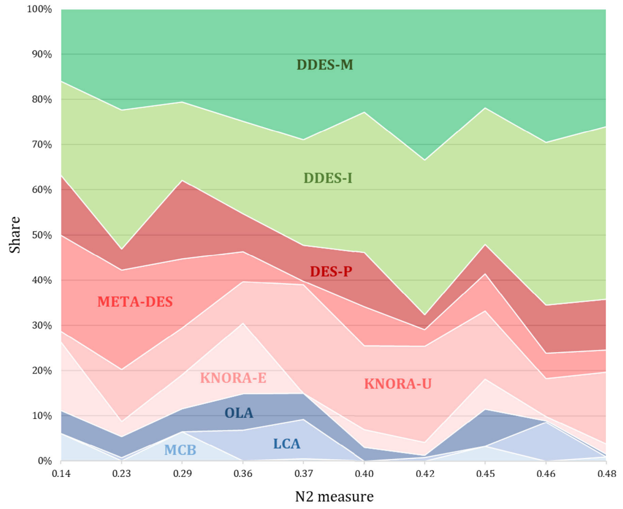


FIGURE 6. Distribution of the winning methods over the class separability metric (N2).

This also implies that the DDES methods are better able to deal with complex classification problems given the fact that there is a negative association ($\rho = -0.7216$, p-value $= 6.7900e^{-6}$) between the average classification accuracy and the N2 measure.

There was no discernible performance difference between DDES-I and DDES-M in our experiment. It is interesting to note that DDES-I provided better classification performances over DDES-M when datasets were highly dispersed. This suggests that additional characteristics of data other than covariance could be further investigated to improve the current specifications of DDES-M, while computational procedures of DDES-I would need to be streamlined.

VII. CONCLUSION

In this paper, we proposed two novel DES methods, namely DDES-I and DDES-M, that can effectively construct the RoC by accounting for data distribution. The performance benchmark showed that the proposed methods yielded the best accuracy on average, providing statistically significant improvements over the competing DS methods. It was further verified that the improvements correlated with both the difference in feature variances and the class separability. That is, more unequal variances and dispersion present in the dataset resulted in better selection of the reference data points by the proposed methods, which eventually provided more accurate classification performance.

The current specifications of the proposed methods raise the possibility of further elaboration. This can be driven by: 1) incorporating additional stochastic factors into the RoC construction such as bias and density of data along with principal components; 2) maximizing the information gain such as redundant reference data elimination; and 3) realizing an adaptive ensemble selection by considering combinatorial DES approaches optimized for a given test case. Lastly, the proposed approaches could be integrated with other dynamic selection frameworks to investigate the combinatorial impact on the classification performance.

REFERENCES

[1] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[2] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[3] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Jan. 1992.

[4] S. García, J. Luengo, and F. Herrera, "Dealing with noisy data," in *Data Preprocessing in Data Mining*. Cham, Switzerland: Springer, 2015, pp. 107–145.

[5] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Inf. Fusion*, vol. 6, no. 1, pp. 63–81, Mar. 2005.

[6] I. Partalas, G. Tsoumakas, E. V. Hatzikos, and I. Vlahavas, "Greedy regression ensemble selection: Theory and an application to water quality prediction," *Inf. Sci.*, vol. 178, no. 20, pp. 3867–3879, Oct. 2008.

[7] Y.-W. Kim and I.-S. Oh, "Classifier ensemble selection using hybrid genetic algorithms," *Pattern Recognit. Lett.*, vol. 29, no. 6, pp. 796–802, Apr. 2008.

[8] S. Mao, L. C. Jiao, L. Xiong, and S. Gou, "Greedy optimization classifiers ensemble based on diversity," *Pattern Recognit.*, vol. 44, no. 6, pp. 1245–1261, Jun. 2011.

[9] A. A. Feitosa Neto and A. M. P. Canuto, "An exploratory study of mono and multi-objective metaheuristics to ensemble of classifiers," *Int. J. Speech Technol.*, vol. 48, no. 2, pp. 416–431, Feb. 2018.

[10] L. Yu, S. Wang, and K. K. Lai, "Testing of diversity strategy and ensemble strategy in SVM-based multiagent ensemble learning," in *Applications of Soft Computing*. Berlin, Germany: Springer, 2009, pp. 431–440.

[11] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognit.*, vol. 36, no. 6, pp. 1291–1302, 2003.

[12] T. Sun and Z.-H. Zhou, "Structural diversity for decision tree ensemble learning," *Frontiers Comput. Sci.*, vol. 12, no. 3, pp. 560–570, 2018.

[13] F. Wenzel, J. Snoek, D. Tran, and R. Jenatton, "Hyperparameter ensembles for robustness and uncertainty quantification," 2020, *arXiv:2006.13570*. [Online]. Available: http://arxiv.org/abs/2006.13570

[14] L. Yu, S. Wang, and K. K. Lai, "Investigation of diversity strategies in SVM ensemble learning," in *Proc. 4th Int. Conf. Natural Comput.*, vol. 7, 2008, pp. 39–42.

[15] Y. Ren, P. Suganthan, and N. Srikanth, "Ensemble methods for wind and solar power forecasting—A state-of-the-art review," *Renew. Sustain. Energy Rev.*, vol. 50, pp. 82–91, Dec. 2015.

[16] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Mach. Learn.*, vol. 51, no. 2, pp. 181–207, 2003.

[17] T. Windeatt, "Accuracy/diversity and ensemble MLP classifier design," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1194–1211, Sep. 2006.

[18] G. D. C. Cavalcanti, L. S. Oliveira, T. J. M. Moura, and G. V. Carvalho, "Combining diversity measures for ensemble pruning," *Pattern Recognit. Lett.*, vol. 74, pp. 38–45, Apr. 2016, doi: 10.1016/j.patrec.2016.01.029.

[19] H. Zouari, L. Heutte, and Y. Lecourtier, "Controlling the diversity in classifier ensembles through a measure of agreement," *Pattern Recognit.*, vol. 38, no. 11, pp. 2195–2199, Nov. 2005.

[20] L. Masisi, V. Nelwamondo, and T. Marwala, "The use of entropy to measure structural diversity," in *Proc. IEEE Int. Conf. Comput. Cybern.*, Nov. 2008, pp. 41–45.

[21] L. I. Kuncheva, M. Skurichina, and R. P. W. Duin, "An experimental study on diversity for bagging and boosting with linear classifiers," *Inf. Fusion*, vol. 3, no. 4, pp. 245–258, Dec. 2002.

[22] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Inf. Fusion*, vol. 6, no. 1, pp. 5–20, Mar. 2005.

[23] J. G. Carney and P. Cunningham, "Tuning diversity in bagged ensembles," *Int. J. Neural Syst.*, vol. 10, no. 4, pp. 267–279, Aug. 2000.

[24] S. Mao, J.-W. Chen, L. Jiao, S. Gou, and R. Wang, "Maximizing diversity by transformed ensemble learning," *Appl. Soft Comput.*, vol. 82, Sep. 2019, Art. no. 105580.

[25] F. Yang, X. Li, Q. Li, and T. Li, "Exploring the diversity in cluster ensemble generation: Random sampling and random projection," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4844–4866, Aug. 2014.

[26] C. Jian, J. Gao, and Y. Ao, "A new sampling method for classifying imbalanced data based on support vector machine ensemble," *Neurocomputing*, vol. 193, pp. 115–122, Jun. 2016.

[27] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, "Diversity in search strategies for ensemble feature selection," *Inf. Fusion*, vol. 6, no. 1, pp. 83–98, Mar. 2005.

[28] P. Cunningham and J. Carney, "Diversity versus quality in classification ensembles based on feature selection," in *Proc. Eur. Conf. Mach. Learn.*, 2000, pp. 109–116.

[29] A. Tsymbal, S. Puuronen, and D. W. Patterson, "Ensemble feature selection with the simple Bayesian classification," *Inf. Fusion*, vol. 4, no. 2, pp. 87–100, Jun. 2003.

[30] V. Bolón-Canedo and A. Alonso-Betanzos, "Ensembles for feature selection: A review and future trends," *Inf. Fusion*, vol. 52, pp. 1–12, Dec. 2019.

[31] B. Krawczyk and M. Woániak, "Diversity measures for one-class classifier ensembles," *Neurocomputing*, vol. 126, pp. 36–44, Feb. 2014.

[32] H. Liu, Y. Du, and Z. Wu, "AEM: Attentional ensemble model for personalized classifier weight learning," *Pattern Recognit.*, vol. 96, Dec. 2019, Art. no. 106976.

[33] Y. Zhang, G. Cao, B. Wang, and X. Li, "A novel ensemble method for k-nearest neighbor," *Pattern Recognit.*, vol. 85, pp. 13–25, Jan. 2019.

[34] X. Zhao, J. Liang, and C. Dang, "Clustering ensemble selection for categorical data based on internal validity indices," *Pattern Recognit.*, vol. 69, pp. 150–168, Sep. 2017.

[35] H. Guo, H. Liu, R. Li, C. Wu, Y. Guo, and M. Xu, "Margin & diversity based ordering ensemble pruning," *Neurocomputing*, vol. 275, pp. 237–246, Jun. 2018.

[36] E. K. Tang, P. N. Suganthan, and X. Yao, "An analysis of diversity measures," *Mach. Learn.*, vol. 65, no. 1, pp. 247–271, 2006.

[37] A. Ulaş, M. Semerci, O. T. Yłldız, and E. Alpaydın, "Incremental construction of classifier and discriminant ensembles," *Inf. Sci.*, vol. 179, no. 9, pp. 1298–1318, Apr. 2009.

[38] M. Lanes, E. N. Borges, and R. Galante, "The effects of classifiers diversity on the accuracy of stacking.," in *Proc. SEKE*, 2017, pp. 323–328.

[39] T. K. Ho, "Nearest neighbors in random subspaces," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR)*, 1998, pp. 640–648.

[40] S. Bian and W. Wang, "On diversity and accuracy of homogeneous and heterogeneous ensembles," *Int. J. Hybrid Intell. Syst.*, vol. 4, no. 2, pp. 103–128, Jun. 2007.

[41] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 5, pp. 730–742, May 2010.

[42] G. Tsoumakas, I. Partalas, and I. Vlahavas, "A taxonomy and short review of ensemble selection," in *Proc. Workshop Supervised Unsupervised Ensemble Methods Their Appl.*, 2008, pp. 1–6.

[43] D. Guan, W. Yuan, Y.-K. Lee, K. Najeebullah, and M. K. Rasel, "A review of ensemble learning based feature selection," *IETE Tech. Rev.*, vol. 31, no. 3, pp. 190–198, May 2014.

[44] A. Jurek, Y. Bi, S. Wu, and C. Nugent, "A survey of commonly used ensemble-based classification techniques," *The Knowl. Eng. Rev.*, vol. 29, no. 5, p. 551, 2014.

[45] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, Apr. 1997.

[46] G. Giacinto and F. Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognit.*, vol. 34, no. 9, pp. 1879–1882, 2001.

[47] A. H. R. Ko, R. Sabourin, and A. S. Britto, "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognit.*, vol. 41, no. 5, pp. 1718–1731, May 2008.

[48] R. M. O. Cruz, R. Sabourin, G. D. C. Cavalcanti, and T. Ing Ren, "META-DES: A dynamic ensemble selection framework using meta-learning," *Pattern Recognit.*, vol. 48, no. 5, pp. 1925–1935, May 2015.

[49] T. Woloszynski, M. Kurzynski, P. Podsiadlo, and G. W. Stachowiak, "A measure of competence based on random classification for dynamic ensemble selection," *Inf. Fusion*, vol. 13, no. 3, pp. 207–213, Jul. 2012.

[50] A. S. Britto, R. Sabourin, and L. E. S. Oliveira, "Dynamic selection of classifiers—A comprehensive review," *Pattern Recognit.*, vol. 47, no. 11, pp. 3665–3680, Nov. 2014.

[51] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Inf. Fusion*, vol. 41, pp. 195–216, May 2018.

[52] S. Sukhanov, C. Debes, and A. M. Zoubir, "Dynamic selection of classifiers for fusing imbalanced heterogeneous data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5361–5365.

[53] D. V. R. Oliveira, G. D. C. Cavalcanti, and R. Sabourin, "Online pruning of base classifiers for dynamic ensemble selection," *Pattern Recognit.*, vol. 72, pp. 44–58, Dec. 2017.

[54] R. M. O. Cruz, D. V. R. Oliveira, G. D. C. Cavalcanti, and R. Sabourin, "FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection," *Pattern Recognit.*, vol. 85, pp. 149–160, Jan. 2019.

[55] J. Elmi and M. Eftekhari, "Dynamic ensemble selection based on hesitant fuzzy multiple criteria decision making," *Soft Comput.*, vol. 2, pp. 1–13, Jan. 2020.

[56] M. A. Souza, G. D. C. Cavalcanti, R. M. O. Cruz, and R. Sabourin, "Online local pool generation for dynamic classifier selection," *Pattern Recognit.*, vol. 85, pp. 132–148, Jan. 2019.

[57] M. Pereira, A. Britto, L. Oliveira, and R. Sabourin, "Dynamic ensemble selection by K-Nearest local oracles with discrimination index," in *Proc. IEEE 30th Int. Conf. Tools with Artif. Intell. (ICTAI)*, Nov. 2018, pp. 765–771.

[58] B. Wang and Z. Mao, "A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule," *Inf. Fusion*, vol. 63, pp. 30–40, Nov. 2020.

[59] D. Li, G. Wen, X. Li, and X. Cai, "Graph-based dynamic ensemble pruning for facial expression recognition," *Int. J. Speech Technol.*, vol. 49, no. 9, pp. 3188–3206, Sep. 2019.

[60] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "Prototype selection for dynamic classifier and ensemble selection," *Neural Comput. Appl.*, vol. 29, no. 2, pp. 447–457, 2018.

[61] L. Didaci and G. Giacinto, "Dynamic classifier selection by adaptive *K*-nearest-neighbourhood rule," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2004, pp. 174–183.

[62] T. P. F. De Lima, A. T. Sergio, and T. B. Ludermir, "Improving classifiers and regions of competence in dynamic ensemble selection," in *Proc. Brazilian Conf. Intell. Syst.*, 2014, pp. 13–18.

[63] P. R. Cavalin, R. Sabourin, and C. Y. Suen, "Dynamic selection of ensembles of classifiers using contextual information," in *Proc. Int. Workshop Multiple Classifier Syst.*, 2010, pp. 145–154.

[64] K. Napierala, J. Stefanowski, and S. Wilk, "Learning from imbalanced data in presence of noisy and borderline examples," in *Proc. Int. Conf. Rough Sets Current Trends Comput.*, 2010, pp. 158–167.

[65] I. A. Abu Amra and A. Y. A. Maghari, "Students performance prediction using KNN and Naïve Bayesian," in *Proc. 8th Int. Conf. Inf. Technol. (ICIT)*, May 2017, pp. 909–913.

[66] K. Gayathri and A. Marimuthu, "Text document pre-processing with the KNN for classification using the SVM," in *Proc. 7th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2013, pp. 453–457.

[67] M. S. Aldayel, "K-nearest neighbor classification for glass identification problem," in *Proc. Int. Conf. Comput. Syst. Ind. Informat.*, Dec. 2012, pp. 1–5.

[68] P. C. Mahalanobis, "On the generalized distance in statistics," Nat. Inst. Sci. India, 1936.

[69] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[70] J. Xia, S. Zhang, G. Cai, L. Li, Q. Pan, J. Yan, and G. Ning, "Adjusted weight voting algorithm for random forests in handling missing values," *Pattern Recognit.*, vol. 69, pp. 52–60, Sep. 2017.

[71] M. Sabzevari, G. Martínez-Muñoz, and A. Suárez, "Vote-boosting ensembles," *Pattern Recognit.*, vol. 83, pp. 119–133, Dec. 2018.

[72] J.-R. Cano, "Analysis of data complexity measures for classification," *Expert Syst. with Appl.*, vol. 40, no. 12, pp. 4820–4831, 2013.

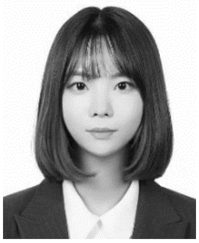
[73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and V. J. Scikit, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[74] R. M. Cruz, L. G. Hafemann, R. Sabourin, and G. D. Cavalcanti, "DESlib: A Dynamic ensemble selection library in Python," *J. Mach. Learn. Res.*, vol. 21, no. 8, pp. 1–5, 2020.

[75] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks," 2017, *arXiv:1707.06799*. [Online]. Available: http://arxiv.org/abs/1707.06799

[76] C. Mellema, A. Treacher, K. Nguyen, and A. Montillo, "Multiple Deep Learning Architectures Achieve Superior Performance Diagnosing Autism Spectrum Disorder Using Features Previously Extracted from Structural and Functional MRI," in *Proc. IEEE 16th Int. Symp. Biomed. Imag. (ISBI)*, Mar. 2019, pp. 1891–1895.

[77] R. A. Armstrong, "When to use the B onferroni correction," *Ophthalmic Physiol. Opt.*, vol. 34, no. 5, pp. 502–508, 2014.



YE-RIM CHOI received the bachelor's degree in systems management engineering from Sungkyunkwan University, South Korea. She is currently a Junior Researcher with the Department of Industrial Engineering, Sungkyunkwan University. Her current research interests include data mining, machine learning, ensemble modeling, and optimization modeling.

DONG-JOON LIM received the B.S. and M.S. degrees in industrial engineering from Sungkyunkwan University, South Korea, and the Ph.D. degree in engineering and technology management from Portland State University, USA. He is currently an Assistant Professor with the Department of Systems Management Engineering, Sungkyunkwan University. He is also a developer of an open source R package DJL which implements various decision support tools related to econometrics and technometrics. His current research interests include technological forecasting, optimization modeling, productivity analysis, and data mining. His academic honors include the Emerald Literati Network Award (outstanding author), the ENI Award (finalist for renewable and non-conventional energy), the Marie Brown Award, and various fellowships from PSU, SKKU, and A&P.