Journal of Theoretical and Applied Information Technology
31st October 2019. Vol.97. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DESIGNING MULTIPLE CLASSIFIER COMBINATIONS A SURVEY

[1]ABDULLAH HUSIN, [2]KU RUHANA KU-MAHAMUD

[1]*Department of Information System, Universitas Islam Indragiri Tembilahan, 29213, Indonesia*

[2]*School of Computing, Universiti Utara Malaysia 06010 Sintok, Kedah, Malaysia*

ABSTRACT:

Classification accuracy can be improved through multiple classifier approach. It has been proven that multiple classifier combinations can successfully obtain better classification accuracy than using a single classifier. There are two main problems in designing a multiple classifier combination which are determining the classifier ensemble and combiner construction. This paper reviews approaches in constructing the classifier ensemble and combiner. For each approach, methods have been reviewed and their advantages and disadvantages have been highlighted. A random strategy and majority voting are the most commonly used to construct the ensemble and combiner, respectively. The results presented in this review are expected to be a roadmap in designing multiple classifier combinations.

Keywords: *Multiple Classifier Combination, Classifier Ensemble Construction, Combiner Construction*

1. INTRODUCTION

The combination of several classification algorithms is considered as a new direction to solve classification problems [1]–[4]. Several approaches have been proposed in designing multiple classifier combinations. All of these approaches attempt to generate diversity in the classifier ensemble. There are four general approaches to design multiple classifier systems [5] as cited by Guo and Nuagu [6]. This can be achieved by using different (i) combination schemes, (ii) classifier models, (iii) feature subsets, and (iv) training sets. These approaches for designing multiple classifier combinations are depicted in Figure 1.

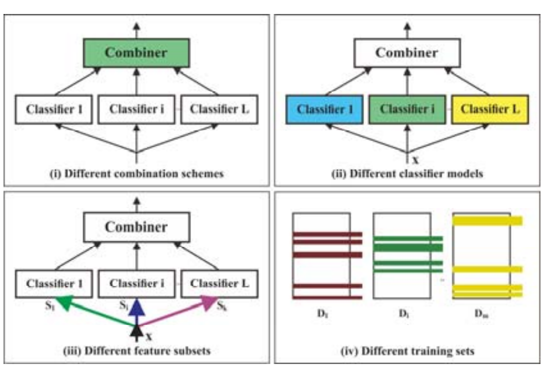

Figure 1: Four Approaches for Designing a Multiple Classifier Combination (Kuncheva, 2001)

According to Roli and Giancito [7], three (3) main steps in designing a multiple classifier combination i) constructing classifier ensemble, ii) constructing combiner, and iii) evaluating performance. The two main problems in designing multiple classifier combinations are the classifier ensemble construction algorithm and combination algorithm. The two main problems in designing a multiple classifier combination are: (i) there is no standard guideline for constructing an accurate and diverse classifier ensemble, and (ii) how to combine the classifier outputs [8]–[10]. Thus, the aim of this paper is to review (i) the approaches in constructing classifier ensembles, and (ii) strategies to combine classifiers.

2. METHOD

This survey has been conducted systematically based based on the guidelines as proposed by Kitchenham et al. [11]. The stages in the systematic survey method are (i) formulating the research questions, (ii) implementing the searching process, (iii) determining the inclusion criteria, and (iv) determining the quality assessment.

2.1 Research Questions

Two research questions have been formulated to focus on the retrieval of the articles.

RQ1: What are the currently and widely used classifier ensemble construction methods?

RQ2: What are the currently and widely used combiner construction methods?

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.2 Search Process

To determine the search sources, only relevant libraries with significant publications in the field of multiple classifier combinations are employed. The search process was a manual search of relevant articles from journals and conference proceedings. Selected journals and conferences proceedings are shown in Table 1.

Table 1 Selected Journals and Conference Proceedings

No	Source	No	Source
1	Advances in Neural Information Processing Systems	31	Knowledge and Information Systems
2	Artificial Intelligence	32	Pattern Recognition Letters
3	Bulletin of Mathematical Biophysics	33	Soft Computing
4	Computational Statistics and Data Analysis	34	Journal of Information and Communication Technology
5	Current Bioinformatics	35	Advances in Modelling and Analysis B
6	Decision Support Systems	36	International Journal of Engineering, Transactions A: Basics
7	Expert System	37	Pattern Analysis and Applications,
8	Expert Systems with Applications	38	Journal of Neuroscience Methods
9	Information and Software Technology	39	Transaction on Electrics Packaging Manufacturing
10	Information Fusion	40	Transaction on Evolutionary Computation
11	Intelligent Data Analysis	41	Transaction on Image Processing
12	Hybrid Intelligent System	42	Transaction on Neural Networks
13	Journal Knowledge and Information Systems	43	Transaction on Pattern Analysis and Machine Intelligence
14	Korean Statistical Society	44	Transaction on Systems, Man and Cybernetics
15	Machine Learning	45	International Conference on Artificial Intelligence
16	Neural Computation	46	Int. Conf.on Computer Inf.System & Industrial Management
17	Neurocomputing	47	Int. Conf.on Computing and Informatics
18	Pattern Analysis and Application	48	Int. Conf.on Fuzzy System and Knowledge Discovery
19	Pattern Recognition	49	Int. Conf.on Machine Learning
20	Pattern Recognition and Artificial Intelligence	50	Int. Conf.on Machine Learning and Cybernetics
21	International Journal of Electrical and Computer Engineering	51	Int. Conf.on Multimedia and Expo
22	Studies in Computational Intelligence	52	Int. Conf.on Multiple Classifier System
23	Medical and Biological Engineering and Computing	53	Int. Conf.on Neural Network
24	International Journal of Business Intelligence and Data Mining	54	Int. Conf.on Pattern Recognition
25	Computers in Industry,	55	Int. Conf.on System Man Cybernetics
26	Journal of Applied Statistics	56	International Joint Conference on Neural Networks
27	Computational Intelligence	57	Asia Joint Conference on Information System
28	Natural Hazard	58	International Conference on Information Reuse and Integration
29	Pocedia Computer Science	58	International Conference on Web Information System and Tech
30	International Engineering an International Journal	60	Int. Conf on Pattern Recognition Application and Methods

2.3 Inclusion Criteria

The goal of this stage is to select appropriate and relevant articles to answer the research questions. Inclusion: (i) Journals and conference proceedings related to multiple classifier combination methods; (ii) Journals and conference proceedings selected that matches with the search keywords; (iii) Books that are related to multiple classifier combinations.

2.4 Quality Assessment

Table 2 Summary of selected articles from digital libraries

Sources	Number of articles selected
IEEE	33
Google Scholar	10
Science Direct	16
Acm Digital Library	3
Scopus	61
Book	6
Total Relevant Articles	129

Based on the six existing sources, Scopus journals are preferred more than others. This is then followed by IEEE, Science Direct, Google Scholar, Book and ACM digital library. Retrieval of articles from sources produced 104 articles from the six sources.

3. CLASSIFIER ENSEMBLE CONSTRUCTION

Four approaches to construct an ensemble of neural networks (NN) has been summarized by Wanas and Kamel [12] are (i) using different initial values: a set of neural networks was constructed by different initial values such as initial weight, (ii) using different topology of neural networks: several architectures and topologies of neural networks were used to determine the number of nodes and hidden layers, (iii) using different learning algorithm: several learning algorithms, were used to train each neural network, and iv) using different data training: each member of an ensemble was trained by various data training. Three ways to construct a diverse classifier ensemble have been summarized by Canuto et al. [13] as follows: (i) different parameters of classifiers, such as weights and topology of neural network models, (ii) varying learning algorithms, such as decision tree, neural

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

networks or naïve Bayes, and (iii) varying training sets, obtained from the original training set by resampling. Finally Roli [14] summarizes five approaches to construct a diverse classifier ensemble as follows: (i) different classifiers construction, (ii) randomness injection, (iii) output labels manipulation, (iv) training data manipulation, and (v) by input features manipulation.

In the first approach, a classifier ensemble is constructed by using several different individual classifiers. The effect of ensemble members has been analysed by Canuto et al. [13], [15]. This approach may good for domains where distinct representations of patterns are possible or complementary information sources are available [14]. The base classifiers apply in their study are: k nearest neighbour (k-NN), Radial basis Function (RBF), multi-layer perceptron (MLP), Fuzzy MLP, J48 decision tree, support vector machine (SVM) and JRip. A small number of classifiers have a strong effect on diversity in the ensemble if less than ten classifiers are used [16]. The study showed that the input features and initial parameters of the classifiers are important to be considered. Multiple types of a classifier which are obtained by randomisation of parameters can be implemented to create multiple classifiers.

In the second approach, a classifier ensemble can be constructed via injecting randomness. An identical neural network, randomly trained with different initial weight values, has been presented by Ranawana and Palade [17]. They combine similar methodologies by training diverse neural networks on the same dataset. In the third approach, the classifier ensemble is constructed by output label manipulaton. This approach is particularly useful when the number of output classes is large. A multiclass problem can be divided into a set of sub-problems. This allows the designer to divide the output classes into a smaller number of class subsets which are used for the training of the classifier. All classes in a subset are relabeled with the same label before training. Thus, the trained classifier would be able to differentiate among the small number of classes it was trained on. By repeating this process, the diverse classifier ensemble will be constracted. A technique known as an error correcting output code for constructing a classifier ensemble has been introduced by Dietterich and Bakiri [18]. The idea is to break the multiclass problem into several dichotomies. Each classifier is assigned with a different dichotomy.

In the fourth approach, classifier ensembles are constructed by using training data manipulation. In this approach, a classifier ensemble is constructed by training the base classifier using different datasets. Several methods for constructing diverse classifiers using different training sets have been presented. One of the subsample methods to produce a set of diverse classifiers is called Boosting [15], [19]. In boosting, the wrongly predicted samples by the previous classifiers are chosen more often than the correctly predicted samples. Hence, the performance of new classifiers is better than previous classifiers. Boosting is more effective for learning problems where the prediction samples have different levels of difficulty. A new boosting method known as AdaBoost has been introduced by Freund and Schapire [20]. In AdaBoost a series of classifiers is constructed sequentially. A training sample which is incorrectly classified by the classifier will gain higher priority in the next training. Prediction of classifiers is combined through weighted voting.

Bootstrap aggregating (Bagging) has been presented by Breiman [21], which also uses the training data manipulation approach to construct the classifier ensemble. Bagging produces multiple bootstrap sample training sets from the original training set by sampling with a replacement. A classifier is created and trained on the training set from the bootstrap at each iteration. With each new training set, one new classifier is obtained. Breiman found that a small change in the training set will produce different predictions. According to the experiment, this method is efficient when data are limited in the training process, because it uses random sampling with replacement. Bagging also works well for unstable classifiers, where the classifier gives various results when there is a small change in the training set. Bagging as a function only provides the training data to the ensemble classifier. Therefore, bagging performance is related to the selected classifier. In this method, sampling with replacements was used. Thus several of the original samples may be assigned more than once and several of the original samples may not be asigned to train the ensemble. In bagging, prediction of classifiers is combined through majority voting [22].

A method to construct ensemble classifiers called random forest has been developed by Breiman [23]. Random forest contains a lot of decision trees. This method is suitable to be applied when the number of features is much greater than the number of samples. In the random forest, if there are many input features, usually there is no need for pre-processing, normalizing the data or feature selection. The advantage of the random forest is it is suitable for many datasets, does not over-fit, and can handle very large numbers of feature inputs. Other advantages of random forest are the speed, parallelism, that it is robust to noise and is flexible. The disadvantage is that the random forest requires large amounts of data in order to fully exploit its potential.

Diverse creation by oppositional relabeling of artificial training examples (DECORATE) as a meta-learner for constructing diverse classifier ensembles by using artificial training examples has been introduced by Melville and Mooney [24]. The experiment shows that this method is better than bagging and random forest. DECORATE also obtains higher accuracy than boosting on small training sets, and achieves comparable performance on larger training sets.

A method to generate ensemble classifiers called ensemble-based artificially generated training samples (EBAGTS) has been developed by Jamalinia et al. [25]. It manipulates training examples in three ways to generate diverse classifier ensembles. This is done by drawing a sub-sample from the training set, reducing error-prone training instances, and reducing local instances around error-prone regions. EBAGTS also obtains higher accuracy than bagging and boosting.

In the fifth approach, the classifier ensemble is constructed by manipulating the input feature set. The idea of an input feature manipulation approach is to simply train each classifier with a different projection of the training set. The feature decomposition method manipulates the input feature in constructing the ensemble. In this method, vertical partitioning on the training sets is performed in order to build an ensemble. Feature space may be partitioned by: random selection, genetic algorithm (GA), input decimation, or other statistical approaches [26]. There are three strategies for feature decomposition. These are i) random-based strategy, ii) reduct-based strategy, and iii) performance-based strategy. In the random-based strategy, the feature subset is created randomly. In the reduct-based strategy, the smallest feature subset that has the same predictive ability as the overall feature set is performed, thus the size of the ensemble is limited to the number of features. In the performance-based strategy, the feature subset is performed based on the classifier's performance.

Random-based Strategy

The simplest technique in the feature subset based ensemble is to assign randomly featured subsets of the original training set as training sets to ensemble members. The random subspace method (RSM) [27], constructs diverse classifier ensembles by training them with different feature subsets which are randomly selected from the original feature set. There are two ways to manipulate input features: sampling with replacement and without replacement. In sampling with replacements, given features can be replicated to train the classifier, whereas in sampling without replacement, the given features cannot be used more than once to train the classifier.

The number of features that are used in each classifier are as many as half of all the features available. This process is repeated to produce an ensemble with one hundred classifiers using all the available features. Majority voting was used to combine classifier outputs. According to the experiments, if the size of training set is relatively small compared to the dimension of the data, the RSM gives good results. According to Skurichina et al. [28], the RSM is suitable for high dimensional data. The evaluation has been performed by comparing this technique with bagging and boosting in constructing classifier ensembles. The RSM showed good accuracy compared to the other two techniques.

A Multiple Feature Subset for nearest neighbor has been proposed by Bay [29]. With similar ideas, Bay suggested another learning ensemble when bagging and boosting are not able to increase the ability of the stable k-NN classifier. In this method, random subsets of the original features are projected to each *k*-NN. Each *k*-NN is trained randomly with the same feature number. Final prediction is obtained by aggregating predictions using majority voting.

Attribute Bagging (AB), introduced by Bryll et al. [30], is a technique in which each classifier in the ensemble is trained on a randomly selected subset of attributes (or features) without replacement. AB establishes an appropriate feature subset size and then randomly selects subsets of features, creating projections of the training set on which the ensemble classifiers are constructed. Majority voting was used to combine classifier outputs. A relevance feedback algorithm [31] also used the random subspace method to overcome instability and overfitting problems. The goal is to construct a solid classifier ensemble with a set of classifiers, which are trained on different feature subset. Multiple classifiers without overfitting problems are successfully developed in this study. Predictions of classifiers are combined with majority voting. In general, this method is proven to increase the accuracy of classifiers.

The techniques described above are almost the same. All of them assign features randomly to each ensemble member. Differences exist only in the parameter determination of feature subset size and the size of the ensemble. Additionally, the experiment they conduct in order to implement the method uses different datasets and different classifiers.

Performance-based Strategy

While the aim of common feature selection algorithms is to select the best feature subset, the task of ensemble feature selection has the additional goal of searching a set of feature subsets that will promote diversity among members of the ensemble. The idea of this method was to apply an algorithm to find different feature subsets. Feature subsets search algorithms directly to support diversity and accuracy in constructing classifier ensembles. In principle, any feature selection algorithm can be used.

The random feature selection strategy to establish the initial ensemble was used by Zenobi and Cunningham [32]. An iterative refinement was performed to induce diversity and enhance the accuracy of the classifier ensemble. In their approach, the selection of features is performed using hill climbing search based on accuracy and diversity. In this method, an attempt to insert or remove any features within the subset of features is performed. If the selected feature subset provides better performance, then the feature subset is not changed. This process is repeated until no more improvement is needed.

Ensemble construction methods based on feature selection algorithms have been proposed by Günter and Bunke [33]. They tested and applied their methods to handwriting recognition problems [34]. In their proposed method, each classifier is assigned a set of features using certain feature selection algorithms. Günter and Bunke [35] used two algorithms which are backward search and sequential floating forward [36]. Each classifier uses one of the two algorithms to search a subset of features.

Tumer and Oza [37] presented Input Decimation Ensembles technique where the input decimation method is applied to select feature subsets with the ability to discriminate among the classes and reduce correlation between classifiers. In their proposed approach, classifiers were trained on an equal number of feature partitions. Input decimation applies principle component analysis to the feature space to generate subsets in which each of them corresponds to a specific class. This method uses a certain number of classifiers based on the number of classes. The number of classifiers is constructed based on the class labels. Each feature subset is used to train the base classifier, such that the feature subset is highly correlated with the class. The final decision of the ensemble is the mean of the individual classifier outputs. They provided a summary of the benefits of correlation reduction. They conducted an experiment on datasets, from the University of California at Irvine (UCI) repositories, and two synthetic datasets. The results indicated that input decimated ensembles outperform ensembles whose base classifiers use all of the input features; simple random feature subset selection; and features created using principal components' analysis.

Comparison of the use of several diversity measures in the context of the search strategy for ensemble construction has been performed by Tsymbal et al. [38]. They constructed an ensemble based on several diversity measures. They considered four search strategies for feature selection which include: genetic search, hill-climbing, ensemble forward and backward sequential selections. They showed that, in some cases, the ensemble feature selection process can be sensitive to the choice of the diversity measure, and on the data being used. In many cases, the combination of disagreement diversity measure, genetic search and dynamic voting provide the best performance.

A multiple classifier combination construction method called rotation forest was proposed by Rodríguez et al. [39]. The application of rotation forest as a classifier ensemble to several application domains has significant enhancements [40], [41]. An improved rotation forest was proposed [42] to increase diversity among base classifiers. Experimental results indicate the algorithm has higher classification and better stability than rotation forest. Similar to input decimation, rotation forest utilizes principal component analysis to obtain diverse and accurate classifiers. This method consists of generating several random feature subsets, in the first step, and regenerating these feature subsets by applying principal component analysis on each of them separately. Instead of using feature subset selection techniques, some researchers have suggested a reconstruction scheme for the feature space by the addition of new features.

Kim and Cho [43] have adopted correlation analysis for feature selection in a study on classification of DNA microarrays for cancer management. Their study aims to establish an effective ensemble classifier based on correlation analysis. This was done because if the feature set provides the same information, a combination of classifiers cannot improve its performance because it will make the same error and there is no possibility of compensation. Therefore, a different set of features providing various amounts of information is necessary to improve classifier combination performance. The feature subset ensemble approach was performed by considering the limited amount of sample DNA microarray with a large number of features.

Constructing an ensemble classifier using feature selection and diversity measures namely Attribute Selection and Diversity Measure (ASDM) has been proposed by Shi and Lv [44]. An individual classifier was trained on different feature subsets which produce diversity among ensemble members. The Naïve Bayes classifier was chosen as a base classifier in the experiment because it is stable and difficult to improve by boosting or bagging. The final decision was obtained by majority voting. The experiments were conducted using Reuters' dataset. ASDM, subsequently, was evaluated using a 10-fold cross validation method. The experimental results show that the classifier ensemble exceeded the original single best classifier. Constructing ensemble classifier using diversity measures has also been proposed by Abdullah et al. [45]. The idea is to partition the original feature set in such a way that the training process will produce a diverse ensemble. The results showed that the method can be used to create a better nearest mean classifier ensemble.

The use of a genetic algorithm [46] for constructing an ensemble by feature selection was also presented in a previous study. Opitz and Shavlik [47] presented a technique known as Accurate and Diverse Ensemble-Maker giving United Prediction (ADDEMUP) that uses genetic algorithms to seek and find a set of feature subsets to create an accurate and diverse classifier ensemble. ADDEMUP works by first creating an initial population with a random sample strategy, and then genetic operator is used to create new ensemble members, the ensemble maintains as much accuracy and diversity as possible. Their studies focused only on the neural network [48]; however, this method can be applied to other classifiers. Based on the results of their evaluation, this method can improve the classification performance of a knowledge-based neural network which exceeds the capabilities of the bagging method.

Inspired by ADDEMUP, Opitz [49] presented the Genetic Ensemble Feature Selection (GEFS) for ensemble construction. Although both of these techniques use the genetic algorithm, ADDEMUP is more complex. GEFS works by first constructing an initial classifier ensemble where each classifier is constructed randomly with replacement selecting a different feature subset. The next new candidate classifiers are produced using crossover and mutation on a subset of features. The authors claimed that the GEFS approach is straightforward, simple, accurate and quick. Based on their evaluation, this method has successfully enhanced the neural network classification performance which exceeds the capabilities of AdaBoost and Bagging.

Ensemble construction based on GA has been proposed by Guerra-Salcedo and Whitley [50], [51] where table-based prediction models and euclidean decision table have been used sequentially. Oliveira et al. [52] also proposed GA as a technique to find a good feature subset in order to construct an ensemble. They used two-stage hierarchical search for ensemble construction. In the first stage, a set of significant classifiers are produced using Multi-Objective Genetic Algorithm search [53]. The neural networks are used as base classifiers, although all types of classifiers can be used. The second stage searches different combinations of these good classifiers and, again, a Multi-Objective Genetic Algorithm is used to find the best ensemble.

The utilization of an ant system (AS) algorithm for classifier ensemble construction (ASFSP) has been proposed by Abdullah and Ku-Mahamud [54], [55]. The idea is to optimize the partition of input feature set for ensemble training by using AS. The proposed method was evaluated on several benchmark datasets and the results showed that the method can be used to construct better homogeneous classifier ensembles.

Reduct-based Strategy

The smallest feature subset that has a similar predictive ability with the ability of the overall feature is called a reduct. In order to obtain a good reduct and to find a significant set of reducts, Wu et al. [56] introduced the Worst Attribute Drop First algorithm. This algorithm was developed through the analysis of decision system properties using the rough set theory. Hu et al. [57] introduced a decision forest using the reduct-based strategy to improve the performance of classification. They proposed methods to build a diverse set of rough decision forests. This method produces significant reducts recursively. Thus, there are no common features in all reducts. This ensures a decision tree is trained by different reducts. The final decision is made based on the output of the decision tree [58] by the majority-voting rule. Bao and Ishii [59] presented a multiple reducts approach to improve the performance of k-NN for text classification. Later, the method was enhanced by Ishii et al. [60] which uses multiple reducts with confidence to classify documents with higher accuracy. To select multiple reducts, they developed a greedy algorithm, which is based on the selection of significant features. A summary of classifier ensemble construction is depicted in Table 3.

Table 3: A summary of classifier ensemble construction

Strategy	Technique	Characteristics
Random-based	Random Subspace Method [28],	Classifiers are trained on randomly elected feature subset. The output is usually combined by a simple majority vote.
	Multiple Feature Subset [29]	Simple voting is used to combine the outputs of multiple NN classifiers, each having access only to random subsets of features.
	Attribute Bagging [30]	Each classifier in the ensemble is trained on a randomly selected subset of attributes without replacement and integrates the outputs by majority voting.
	Relevance Feedback [31]	Constructing a good classifier ensemble with a set of classifiers, which are trained randomly on a different feature subset.
Performance-based	Random Feature Selection [61]	Insert or remove any features within the subset of features is performed and repeated until no more improvement is needed.
	Feature Selection Algorithm [33]	Each classifier is trained on a set of selected features using existing feature selection algorithms.
	Input Decimation Ensembles [37]	A certain number of classifiers are trained on an equal number of feature partitions using a certain number of classifiers based on the number of classes.
	Rotation Forest [39]	Classifier ensemble is constructed by using independently trained decision trees. The feature set is randomly partitioned into k subsets and principal component analysis is applied to each subset.
	Attribute Selection and Diversity Measure [44]	Constructing ensemble classifier by using feature selection and diversity measures.
	Accurate and Diverse Ensemble-Maker giving United Prediction [47]	The genetic algorithm is used to seek and find a set of feature subsets to create an accurate and diverse classifier ensemble.
	Genetic Ensemble Feature Selection [49], [53]	The genetic algorithm is also used but is more straightforward, simple, accurate and quick compared with ADDEMUP.
	Ant-based feature set partitioning [54], [55], [62]	The ant-based algorithm is used to seek and find a set of feature subsets to create an accurate and diverse classifier ensemble.
	Deterministic Subspace (DS) approach. [63]	Classifier ensemble is constructed based on the idea of creating subspaces incrementally, in a certain way guided by both the quality of feature subspaces and the ensemble diversity.
Reduct-based	Worst attribute drop first [56]	An attempt to create a set of optimal reducts and integrate them by using Naive Bayes.
	Decision forest [57]	Building a diverse set of rough decision forests based on reduction.
	Multiple Reducts [60]	An attempt to select multiple reducts, by using a greedy algorithm, which is based on the selection of significant features.
	Discernibility matrix simplification with genetic algorithm [64]	Genetic algorithm is used for feature reduction.

A special case of feature subset-based ensemble is known as feature set partitioning. This method does not search for single useful subsets. The original set is decomposed into several subsets and then classifier ensemble is trained on a different feature subset. This methodology is suitable for the problem with a big number of features [61], [65], [66]. Figure 3 shows subset-based ensemble covers the feature set partitioning and also feature selection.

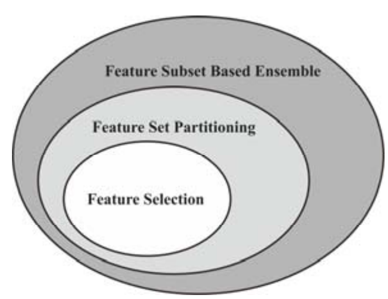

Figure 3 Venn diagram of the search space of feature orientation (Rokach, 2008)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Liao and Moody [67] proposed feature set partitioning by pair-wise mutual information feature grouping whereby similar features are assigned to the same partition. Therefore, a hierarchical clustering algorithm is used. Furthermore, artificial neural networks are constructed for each group and run to achieve the final decision. A partition searches algorithm using incremental oblivious decision trees known as Decomposed Oblivious Gain (DOG) has been proposed by Rokach and Maimon [68]. The DOG algorithm starts the search from an empty partition, which can lead to a relatively small subset of features. Furthermore the DOG has no backtracking capabilities.

Ahn et al. [69] proposed the randomly partitioned input features involve several subsets. Each classifier is assigned with different subsets, which is suitable for unbalanced data and high-dimensional datasets. Combining the results from different features selection has been proposed by Rokach et al. [70]. They combined several feature selection methods which succesfully improve the classification accuracy significantly. Earlier, Rokach and Maimon [68] developed a general framework for disjointing feature set partitions. This framework nests many algorithms, two of which were empirically tested using more than one dataset. This framework shows that the performance of decision tree can be improved by using feature set decomposition. Rokach et al. [70] applied genetic algorithms for feature set partitioning. This algorithm has been tested with different datasets and the results show advantages compared to other methods and this algorithm accelerates the execution.

4. COMBINER CONSTRUCTION

The combiner (or fuser) aims to create a combination rule that can utilise the diversity of classifiers and optimize in combining the classifiers. Xu et al. [71] and Kuncheva [72] categorize the operating level of classifier combination based on the output which is produced by the classifier, into three levels namely the abstract level, rank level and measurement level. At the abstract level each classifier generates an output class label for any unknown object to be classified. The top candidate which is produced by each classifier is used. At the ranked-level, the output of each classifier is a subset of the class label set, the ranking list of candidates which is produced by each classifier is used. At the measurement-level combination method, both ranked-level and similarity measurement or confidence value of each candidate are used. The combination method that works at abstract-level can be applied to any classifier. In contrast, the ranked-level and the measurement-level can cause other difficulties when they are used in which each classifier individually gives the top candidate, or when combined, the classifier provides a value of similarity measurements.

Woods et al. [73] divided the combination scheme into two (2) major types which are classifier fusion and classifier selection. This scheme has been further enhanced by Kuncheva et al. [74], where dynamic selection and static selection are differentiated. Furthermore this scheme has been improved into the selection-fusion [75], [76]. A summary of combiner construction is depicted in Table 4.

Classifier-fusion Scheme

The classifier fusion scheme assumes that the entire classifier has the same capability, and the prediction of the entire classifier has been considered. Sharkey [77] further extended the classifier fusion into fixed classifier fusion and trained classifier fusion. Static weight is used for the fixed classifier fusion scheme. Thus there is no training process to learn the weight of each classifier. The simplest way is to use operators such as sum-rule and product-rule. The final prediction is either the maximum or minimum value of the outputs. This scheme is simple to implement and has low computational cost. However, it is infective in combining the outputs [78].

Lincoln and Skrzypek [79] have proposed a Simple Averaging approach as a fixed classifier fusion. In this combination rule, the final output of classifier combination is the average of each classifier output value. Some experiments have shown that the simple average is an effective approach [21], [22], [71]. However, the weakness of this technique is the equal treatment to each member of the classifiers and there is no emphasis on the performance of classifiers.

Shilen [80] proposed one combination rule at the abstract level namely Dempster-Shafer (DS). This rule used a priori knowledge of information about the performance of each individual classifier. DS combines several different classifiers using a level of recognition and substitution rates as a priori knowledge [81]. Generally, if given an input pattern, all the classifiers that have the same output are collected into a group. Thus, after this combination, each group is equivalent to a new

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

classifier with the recognition rate and substitution rate that is new. The next step is to combine the recognition and substitution rates to calculate the confidence true output and confidence through the equivalent output of one classifier. However, this integration method requires heavy computation and gives low generalization performance.

Xu [71] proposed a Bayesian method which is based on applying Bayes theorem by error consideration of each classifier. In this method, the probabilistic summary for each class is defined earlier. However, one of the significant disadvantages of this method is that the mutual independencies between classifiers are ignored, but this does not always happen in the real application [82].

Ho et al. [83] have presented the Borda Count as a vote technique on the rank level. In this technique, each class which is produced by individual classifiers is ranked. The first rank is given the highest value and last rank is given the lowest value. The output is the class with the highest number of rank overall. The advantages of the Borda Count technique are its simplicity, lower computational cost and that it requires no training. This technique provides better results than majority voting, particularly on classification problems with more than two classes because more information is used. However, the weakness of this technique is the equal treatment for each classifier, so there is no emphasis on the classifier that gives more contribution to the output.

In the trained classifier fusion, the weight of classifier is learned through training. The advantages of the trained classifier fusion approach are its flexibility and potentially better performances than fixed classifier fusion; however, the disadvantages are high memory and time requirements.

Huang et al. [84] proposed a combination method using the data transformation and Neural Network (NN). The output value of each classifier is first converted into a likelihood measurement. The measurement value that has been transformed is inputted into the neural network layer, and then the neural network produces the final classification decision.

A neural network that consists of multi-layer perceptrons trained continuously until the required accuracy is achieved for the combination classifier has been proposed by [85]. Breve et al. [86] combined classifiers by NN for noisy data classification. However, one weakness of using the artificial neural network is expensive computational cost [87], [88]. Jacobs [89] proposed the Weighted Averaging as another variant of Simple Averaging approach. This technique gives weight to each classifier before calculating the average amount of each output from the classifier. In this technique, a weight is attached to each individual classifier. The final classification result is calculated based on the performance of each member of classifiers. The total weight is 1 and each classifier member is given a part of the total weight according to their performance. Therefore, the strength of each classifier is considered, but the weakness of this technique is sensitive to biased classifiers.

A combination scheme known as Behavior Knowledge Space (BKS) which can combine the outputs of individual classifiers has been proposed by Huang and Suen [90]. The BKS is the combination technique on the abstract level that combines the decisions generated by each classifier. The BKS is followed by two stages, namely the learning stage and decision stage. During the learning stage, the training set is given to the K classifier to gather a priori knowledge information that is required on the decision stage. Experiments on handwritten numerals have proven that this method outperforms the Simple Voting, Bayesian, and Dempster-Shafer approaches. However, to proof that BKS is good, it needs to be trained on large training dataset.

Kuncheva and Jain [91] have presented two simple ways to use the genetic algorithm on multiple classifier combinations. The genetic algorithm is used as a combination scheme to optimize the weights connection. The starting process involves randomizing the weight values gradually. The weight reflects the importance of each classifier. The classifiers used were Quadratic Discriminant Classifier, Linear Discriminant Classifier, and Logistic Classifier. Kim et al. [82] proposed a technique to combine multiple classifiers based on the genetic algorithm. The classifier used was the neural network. The method shows better performance than majority voting, Bayesian, behavior knowledge space, Borda count, weighted Borda count, sum and neural network.

Aslam and Montague [92] proposed the Weighted Borda Count as a variant of the Borda Count which gives weight to the individual classifier. Weights are intended to address the performance of each individual classifier. An advantage of the weighted Borda count is that it does not require training. The weighted Borda Count still requires classifiers that are able to give ratings on the potential class

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

although it considers individual classifier performance.

Classifier Selection Scheme

In the classifier selection scheme, only one classifier is needed to correctly classify the input pattern. Select a single "best" classifier from base classifiers for the final decision. In order to do this, it is important to define a procedure to choose a member of the classifier ensemble to obtain the final classification output. Kuncheva [75] further extended the selection scheme into the static classifier selection and dynamic classifier selection. In static classifier selection, the selection of the best classifier is specified during a training phase [93]. In dynamic classifier selection, the choice of classifier is made during the classification phase. One individual classifier among ensemble classifiers is chosen. It is called "dynamic" because the classifier used critically depends on the test pattern itself. Only the output of the selected classifier is considered in the final decision.

There are several type of dynamic classifier selection method. Woods et al. [73] introduced dynamic classifier selection by local accuracy. Other types are the posterior selection, prior selection and dynamic classifier selection which are based on multiple classifier behavior [94], [95]. The main idea in the dynamic classifier selection is that the choice of one individual classifier must exceed any other classifier, so it depends on the ability of the estimated generalization from the classifier [75]. The superiority of dynamic selection method is that error-dependency can be omitted [78]. The advantage of selecting the dynamic ensemble is the ability to estimate distribution to a group of classifiers rather than a single individual classifier. So far, this scheme seems to work [96].

Selection-fusion Scheme

In this scheme, the selection and fusion process is used to provide the most suitable way to combine multiple classifiers. Usually, there is a criterion to decide whether to select the best classifier or to combine the classifiers. The idea is to use the selection method if and only if the best one classifier can be determined, otherwise the combination method is used. The dynamic classifier selection based on multiple classifier behavior [97] and the dynamic classifier selection using decision templates [98] are two application samples of selection-fusion scheme. Yang and Browne [99] have proposed a hybrid method which combined classifier selection simultaneously with particle swarm optimization. The experiment showed that the proposed method gives good performance and averagely outperforms other rule such as max, min, mean, median, product and majority voting.

Table 4: Summary of combiner construction

Scheme	Technique	Characteristics
Classifier fusion	Simple Averaging [79]	The final output of classifier combination is the average of each classifier output value.
	Majority Voting [12][29][100]–[109]	The final output is the most votes from a set of classifier.
	Weigted Voting [62], [110]–[120]	A variant of majority voting, where each classifier is weighted before voting.
	Unanimous Voting [121]	The final prediction is based on choosing the class for which all classifiers agree, whereas the majority voting is based on choosing the class with the largest number of votes.
	Dempster-Shafer [80]	Combining several different classifiers using a level of recognition and substitution rates as a priori knowledge.
	Bayesian Method [71]	Applying Bayes theorem by error consideration of each classifier.
	Borda Count [83]	A vote technique on the rank level, where the output is the class with the highest number of rank overall.
	Data Transformation & Neural Network [84]	Multi-layer perceptrons trained continuously until the required accuracy is achieved for the combination classifier.
	Weighted Averaging (WA) [89]	Another variant of simple averaging, which gives weight to each classifier before calculating the average amount of each output from the classifier.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	Behavior Knowledge Space (BKS) [90]	The final output is estimated by calculating high order distribution of classifier outputs from the frequencies of occurrence in the training set.
	Genetic Algorithm [91]	The genetic algorithm is used as a combination scheme to optimize the weights, which it reflects in the importance of each classifier.
Classifier selection	Dynamic Classifier Selection by Local Accuracy [73], [122]–[125]	The choice of one individual classifier by local accuracy must exceed any other classifier.
	Dynamic Classifier Selection by Multiple Classifier Behaviour [94], [95], [126]	The choice of one individual classifier by multiple classifier behaviour must exceed any other classifier.
Selection-fusion	Dynamic Classifier Selection based on Multiple Classifier Behavior [97]	The selection and fusion techniques are used based on multiple classifier behaviour in order to provide the most suitable method.
	Dynamic Classifier Selection using Decision Templates [98]	The selection and fusion techniques are used based on decision template.
	Classifier Selection and Combination using Particle Swarm Optimisation [99]	Classifier selection and combination are implemented simultaneously using particle swarm optimization.

The application of majority voting for classifier combiner was first proposed by Hansen and Salamon [22]. Several popular ensemble methods such as bagging and boosting, random subset of random forest method used majority voting as a combiner. Figure 3 shows three popular ensemble methods, namely (i) bagging [21], (ii) boosting [20] and (iii) random forest [23] using voting in combining classifier outputs [127].

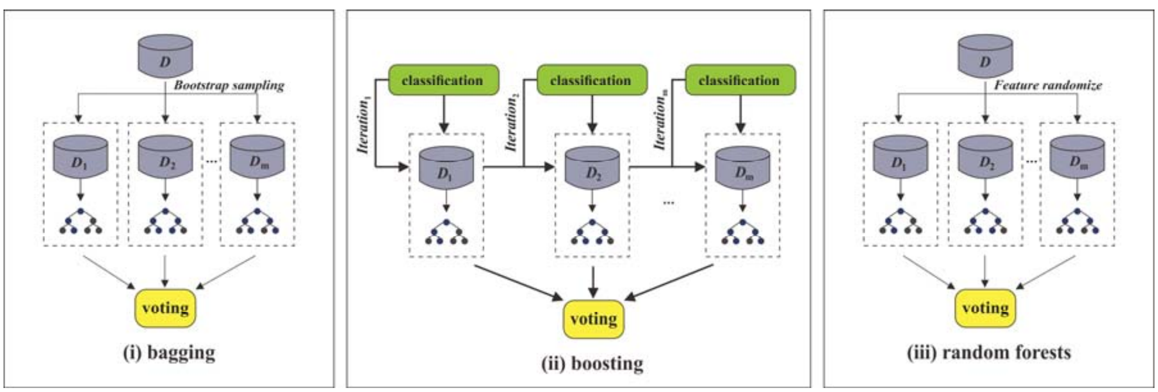

Figure 3: Voting Combiner on Three Popular Ensemble Methods (Yang et al., 2010)

This technique considers the most frequent class among the existing class labels. In order to overcome the draw voting problem, the number of classifiers used is usually odd. Here, each classifier votes for one class label. The class label that is the most frequently chosen is the final decision. One of the advantages of majority voting is the ability to combine the output of each classifier regardless of the classifier is used. The weakness of this combiner is that it does not consider the strength of classifier, in other words, the strength of each classifier is considered equal in the vote.

Weighted voting is a trainable version of majority voting proposed by Littlestone and Warmuth [128]. This technique gives weight to each classifier before voting. The weight for each classifier is obtained through the training process. To make an overall prediction, a weighted vote of the classifier prediction is performed to predict the most weighted class. Although this technique considers the strength of each classifier, the lack of this technique is that it only considers the first rank class or classes most probably found in each classifier.

Wanas and Kamel [12] presented a feature based approach as well as training algorithm. In the feature based approach, each classifier is trained independently. This algorithm is based on the adaptive training algorithm for training neural network ensembles. This training approach helps optimize the weights to achieve better overall classification. Based on the experiment, the results of two benchmark problems and comparison to a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

single classifier show that the approach obtained a better classification accuracy.

A novel multiple classifier combination that incorporates global optimization based on genetic algorithms to develop multiple classifiers was introduced by Stefano et al. [113]. The multiple classifier combination adopts the weighted voting approach to combine the output of the classifiers. The weights are obtained by maximizing the performance of the ensemble. This multiple classifier combination has been tested on a handwritten digit recognition problem. Based on the results of an experiment conducted on 30,000 digits from the NIST database, it shows good performance.

A modified approach to weighted majority voting was presented by Gangardiwala and Polikar [114] where the classifiers are dynamically weighted. The idea of this approach is that the classifier, whose training dataset is closest to the given instance, has more information about the instance. Therefore, it is more likely to classify the instance correctly. The proposed algorithm provides improved performance compared to Adaboost based experiments over benchmark dataset.

Zhang et al. [115] proposed a parallel multiple classifier, using the maximum of posterior probability average, with self-adaptive weight based on output vectors and decision template (MASWOD). Self-adaptive weight calculated by the confusion matrix using decision templates and error punishment factor (EPF). Each of the classifier outputs are treated as inputs to the next level, which combines the results of each level classifier using a self-adaptive weight. Experiments were performed on the UCI datasets to compare the MASWOD with the classical Bayesian algorithm in order to combine several classifier outputs. The experimental results show the MASWOD algorithm can efficiently improve the performance of classification, where the classical Bayesian cannot always enhance classification performance. This proves that the method is efficient because the self-adaptive weight can improve the classification performance.

The weighted majority voting has also been used as a combiner in predicting financial distress [129]. The voting weight was specified by a priori performance measure which was calculated from the confusion matrix. In the experiment, 135 pairs of Chinese listed companies and 35 financial ratios were initially used. The stepwise discriminant analysis method was used in feature selection. After voting weight determination, the performance of financial distress prediction was compared with a single classifier approach. It was concluded that the use of weighted majority voting for financial distress prediction give higher accuracy and lower variance than any single classifiers.

Valdovinos and Sánchez [116] introduced another weighted voting system in classifier fusion which corresponds to the average distance weight. The goal of this weighting technique is to reward (by assigning the highest weight) the individual classifier with the nearest neighbor to the input pattern. The effectiveness of this approach is empirically tested over a number of data sets. Experimental results with several real-problem datasets from the UCI Machine Learning Database Repository demonstrated the advantages of this weighted voting technique over simple majority voting.

An experiment analyzing the ability of a weighted voting combiner, to combine the output of multiple classifiers, has been conducted by Mu et al. [117]. The weighted voting was applied to human face and voice recognition. The effectiveness of the weighted voting methodology was tested on images and voice benchmark database. The weighted voting successfully achieved high performance and outperformed majority voting. It can be concluded that the weighted voting strategy can be used to combine any independent classifiers.

Huang and Wang [118] proposed the multi-weighted majority voting strategy to improve the performance of classification task for complex facial security application. A support vector machine (SVM) was used as the classification algorithm. The hierarchical classification method and the multi-weighted majority voting strategy are two important parts in this strategy. Experimental results indicate that the proposed algorithm improves the performance of face authentication when tested with a massive number of training and testing data.

Wozniak [119] presented the evolutionary approach to produce a classifier ensemble based on weighted voting. Several classifier fusion methods were discussed and evaluated through computer experiments on seven benchmark databases from the UCI Machine Learning Repository. The aim of the experiment was to evaluate the performance of fuser of discriminants based on weights which depended on classifier and class number. The results justified the use of weighted voting combiner. Unfortunately, it is not possible to

Journal of Theoretical and Applied Information Technology
31st October 2019. Vol.97. No 20

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

determine the weights in an analytic way; therefore, using a heuristic optimization method (like evolutionary algorithms) seems a promising research direction.

A weighted voting classification ensemble method, called WAVE was proposed by Kim et al. [110]. The instance weight vector assigns higher weights to observations that are hard to classify. The classifier weight vector puts larger weights to classifiers that perform better on hard-to-classify instances. The final prediction of the ensemble is obtained by voting using the optimal weight vector of classifiers. Both majority voting and the proposed weighted voting have been applied for comparison. The results showed that, in general, the proposed weighted voting performs significantly better than majority voting.

Hamzeloo et al. [120] introduced a weighted ensemble technique (WNNE) to improve the nearest neighbor classifier. WNNE assigns a weight to ensemble member, and uses a rule to determine the output of the ensemble. This method was compared with the single nearest neighbour classifier and random subspace method on some datasets from UCI. Experimental results indicate that the performance of WNNE exceeded the two approaches that have been used for comparison.

5. DISCUSSION

Several approaches to construct classifier ensembles attempted to create differences among ensemble members and try to induce classifier diversity, that make errors on different patterns. Figure 4 shows the trend of strategies to construct the ensemble from 2012 to 2108. It can be seen that random-based is the most often used approach in constructing classifier ensembles. The weakness of this approach is that it does not construct an optimal diverse classifier ensemble.

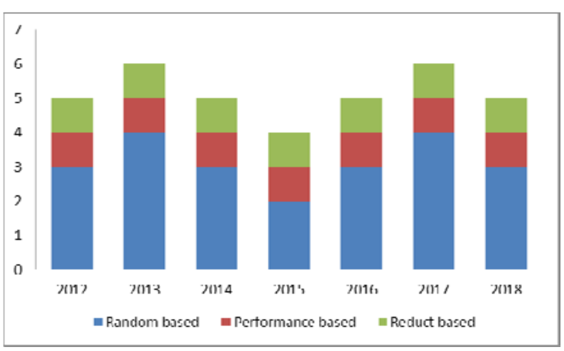

Figure 4 Strategy in constructing ensemble

There are several schemes that have been introduced to combine classifiers. Figure 4 shows the trend of schemes in combining classifiers from 2012 to 2018. It can be shown that the classifier fusion scheme is the most popular strategy in constructing the combiner. The classifier's strength is not considered and this posed as the combiner's weakness. The classifier selection scheme is also gaining in popularity.

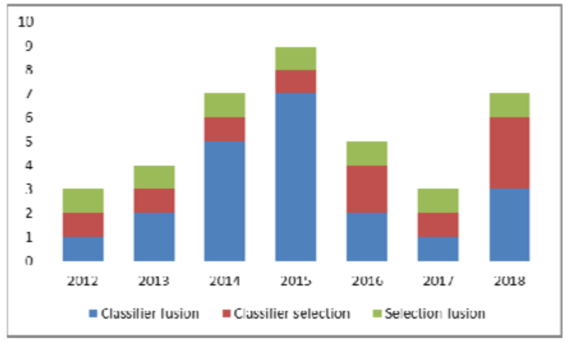

Figure 5 Scheme in combining classifier

6. CONCLUSION

The random based approach is the most commonly used to construct classifier ensembles. The advantage of a random based approach is its simplicity. The majority voting combiner, as one of the fixed classifier fusion schemes, is the most popular, fundamental and straightforward strategy to combine classifiers. The advantage of majority voting is the ability to combine the output of each classifier regardless of the classifier is used. Nevertheless, this does not mean that majority voting is the best. The effectiveness of weighted voting was tested on several application domains. Weighted voting successfully achieved high performance and outperformed the majority voting combiner. The weighted voting combiner can be used as a new direction to combine any individual classifiers.

ACKNOWLEDGEMENT

This study is under the Fundamental Research Grant Scheme, S/O code 12490 provided by The Ministry of Higher Education of Malaysia.

REFERENCES

[1] R. A. D. Silva, A. M. D. P. Canuto, J. C. Xavier Junior, and T. B. Ludermir, "No Title," in *Proceedings of the International Joint Conference on Neural Networks*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

(IJCNN), 2018.

[2] E. Santucci, L. Didaci, G. Fumera, and F. Roli, “A Parameter Randomization Approach for Constructing Classifier Ensembles,” *Pattern Recognit.*, vol. 69, pp. 1–13, 2017.

[3] Y. Yan, Q. Zhu, M. Shyu, and S. Chen, “A Classifier Ensemble Framework for Multimedia Big Data Classification,” in *7th International Conference on Information Reuse and Integration (IRI)*, 2016, pp. 615–622.

[4] R. Soleymani, E. Granger, and G. Fumera, “Classifier Ensembles with Trajectory Under-Sampling for Face Re-Identification,” in *ICPRAM 2016 Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods*, 2014, pp. 97–108.

[5] L. I. Kuncheva, “Combining classifiers: Soft computing solution,” in *Pattern Recognition: From Classical to Modern Approaches*, S. K. Pal and A. Pal, Eds. Singapore: World Scientific Publishing Co, 2001, pp. 427–452.

[6] G. Guo and D. Neagu, “Similarity-based classifier combination for decision making,” in *IEEE International Conference on Systems, Man and Cybernetics*, 2005, vol. 1, pp. 176–181.

[7] F. Roli and G. Giancito, “Design of multiple classifier systems,” in *Hybrid Methods in Pattern Recognition*, H. Bunke and A. Kandel, Eds. Singapore: World Scientific Publishing, 2002, pp. 199–226.

[8] A. H. R. Ko, R. Sabourin, A. Britto, and L. Oliveira, “Pairwise fusion matrix for combining classifiers,” *Pattern Recognit.*, vol. 40, pp. 2198–2210, 2007.

[9] M. Wozniak and M. Zmyslony, “Chosen problems of designing effective multiple classifier systems,” in *IEEE International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, 2010, pp. 42–47.

[10] P. Thomas, H. Bril El Haouzi, M. C. Suhner, A. Thomas, E. Zimmermann, and M. Noyel, “Using a Classifier Ensemble for Proactive Quality Monitoring and Control,” *Comput. Ind.*, vol. 99, pp. 193–204, 2018.

[11] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering – A systematic literature review,” *Inf. Softw. Technol.*, vol. 51, no. 2009, pp. 7–15, 2009.

[12] N. M. Wanas and M. S. Kamel, “Weighted combination of neural network ensembles,” in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN’02 (Cat. No.02CH37290)*, 2002, vol. 2, pp. 1748–1752.

[13] A. M. P. Canuto, M. C. C. Abreu, L. D. M. Oliveira, C. X. Jr, and A. D. M. Santos, “Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles,” *Pattern Recognit. Lett.*, vol. 28, pp. 472–486, 2007.

[14] F. Roli, “Multiple classifier system,” in *Encyclopedia of Biometrics*, S. Z. Li and A. K. Jain, Eds. New York: Springer, 2009, pp. 981–986.

[15] U. R. Salunkhe, “Classifier Ensemble Design for Imbalanced Data Classification : A Hybrid Approach,” *Procedia Comput. Sci.*, vol. 85, no. Cms, pp. 725–732, 2016.

[16] L. I. Kuncheva and S. T. Hadjitodorov, “Using diversity in cluster ensembles,” *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 2, pp. 1214–1219, 2004.

[17] R. Ranawana and V. Palade, “Multi-Classifier systems: Review and a roadmap for developers,” *Int. J. Hybrid Intell. Syst.*, vol. 3, no. 1, pp. 35–61, 2005.

[18] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error correcting output codes,” *J. Artif. Intell.*, vol. 2, pp. 263–286, 1995.

[19] R. E. Schapire, “The strength of weak learnability,” *Mach. Learn.*, vol. 5, pp. 197–227, 1990.

[20] Y. Freund and R. R. E. Schapire, “Experiments with a new boosting algorithm,” in *International Conference on Machine Learning*, 1996, pp. 148–156.

[21] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 421, pp. 123–140, 1996.

[22] L. Hansen and P. Salamon, “Neural network ensembles,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

993–1001, 1990.

[23] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[24] P. Melville and R. J. Mooney, “Creating diversity in ensembles using artificial data,” *Inf. Fusion*, vol. 6, no. 1, pp. 99–111, 2005.

[25] H. Jamalinia, S. Khalouei, V. Rezaie, S. Nejatian, K. Bagheri-Fard, and H. Parvin, “Diverse classifier ensemble creation based on heuristic dataset modification,” *J. Appl. Stat.*, vol. 45, no. 7, pp. 1209–1226, 2018.

[26] L. Rokach, *Pattern Classification Using Ensemble Methods*. Singapore: World Scientific Publishing Co. Pte. Ltd., 2010.

[27] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

[28] M. Skurichina, L. I. Kuncheva, and R. P. . Duin, “Bagging and boosting for the nearest mean classifier: Effect of sample size on diversity and accuracy,” in *Multiple Classifier Systems, Lecture Notes in Computer Science, Vol. 2364*, F. Roli and J. Kittler, Eds. Berlin Heidelberg: Springer-Verlag, 2002, pp. 62–71.

[29] S. D. Bay, “Combining nearest neighbor classifiers through multiple feature subsets,” in *Proceedings of the 17th International Conference on Machine Learning*, 1998, pp. 37–45.

[30] 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.

[31] D. Tao and X. Tang, “SVM-based relevance feedback using random subspace method,” in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME) Vol. 1*, 2004, pp. 269–272.

[32] G. Zenobi and P. Cunningham, “Using diversity in preparing ensembles of classifiers based on dfferent feature subsets to minimize generalization error,” in *Proceedings of 12th European Conference on Machine Learning (EMCL)*, 2001, pp. 576–587.

[33] S. Günter and H. Bunke, “Creation of classifier ensembles for handwritten word recognition using feature selection algorithms,” in *Proceedings of the 8th IEEE International Workshop on Frontiers Handwriting Recognition*, 2002, pp. 183–188.

[34] U. Marti and H. Bunke, “Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 01, pp. 65–90, 2001.

[35] S. Günter and H. Bunke, “Feature selection algorithms for the generation of multiple classifier systems,” *Pattern Recognit. Lett.*, vol. 25, pp. 1323–1336, 2004.

[36] P. Pudil, F. J. Ferri, J. Novovi Cova, and K. J, “Floating search methods for feature selection with nonmonotonic criterion functions,” in *Proceedings of the 12th IEEE International Conference on Pattern Recognition*, 1994, pp. 279–283.

[37] K. Tumer and N. Oza, “Input decimated ensembles,” *Pattern Anal. Appl.*, vol. 6, no. 1, pp. 65–77, 2003.

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

[39] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, “Rotation forest: A New classifier ensemble method,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, 2006.

[40] B. A. Tama and K. H. Rhee, “Classifier Ensemble Design with Rotation Forest to Enhance Attack Detection of IDS in Wireless Network,” *Proc. - 11th Asia Jt. Conf. Inf. Secur. AsiaJCIS 2016*, vol. 11, no. August, pp. 87–91, 2016.

[41] B. T. Pham, D. T. Bui, I. Prakash, and M. B. Dholakia, “Rotation forest fuzzy rule-based classifier ensemble for spatial prediction of landslides using GIS,” *Nat. Hazards*, no. October 2017, 2016.

[42] T. Chen, “An Improved Rotation Forest Algorithm Based on Heterogeneous Classifiers Ensemble for Classifying Gene Expression Profile,” *Adv. Model. Anal. B*, vol. 60, no. 1, pp. 1–24, 2017.

[43] K.-J. Kim and S.-B. Cho, “Ensemble classifiers based on correlation analysis for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DNA microarray classification," *Neurocomputing*, vol. 70, pp. 187–199, 2006.

[44] H. Shi and Y. Lv, "An ensemble classifier based on attribute selection and diversity measure," in *Proceedings of the 5th IEEE International Conference on Fuzzy Systems and Knowledge Discovery*, 2008, vol. 2, pp. 106–110.

[45] Abdullah, K. R. Ku-Mahamud, and A. Sediyono, "A new feature set partitioning method for nearest mean classifier ensembles," in *Proceedings of the 4th International Conference on Computing and Informatics (ICOCI)*, 2013, pp. 39–44.

[46] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. New York: Addison-Wesley, 1989.

[47] D. W. Opitz and J. W. Shavlik, "Generating accurate and diverse members of a neural-network ensemble," *Adv. Neural Inf. Process. Syst.*, vol. 8, pp. 535–541, 1996.

[48] W. McCulloch and W. Pitts, "A logical calculus of the ideas imminent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 115–133, 1943.

[49] D. W. Opitz, "Feature selection for ensembles," in *Proceedings of the 16th National Conference on Artificial Intelligence*, 1999, pp. 379–384.

[50] C. Guerra-Salcedo and D. Whitley, "Genetic approach to feature selection for ensemble creation," in *Proceedings of Genetic and Evolutionary Computation Conference*, 1999, pp. 236–243.

[51] W. Dillon and M. Goldstein, *Multivariate Analysis Methods and Applications*. New York: John Wiley and Sons, 1984.

[52] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Feature selection for ensembles: A hierarchical multi-objective genetic algorithm approach.," in *Proceedings of the 7th ICDAR*, 2003, pp. 676–680.

[53] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen, "Feature selection using multi-objective genetic algorithms for hand written digit recognition," in *Proceedings of the 16th IEEE International Conference on Pattern Recognition*, 2002, pp. 568–571.

[54] Abdullah and K. R. Ku-Mahamud, "Ant system-based feature set partitioning algorithm for k-NN and LDA ensembles construction," in *Proceedings of the 5th International Conference on Computing and Informatics (ICOCI)*, 2015, pp. 326–332.

[55] Abdullah and K. R. Ku-Mahamud, "Ant system-based feature set partitioning algorithm for classifier ensemble construction," *Int. J. Soft Comput.*, vol. 11, no. 3, pp. 176–184, 2016.

[56] Q. X. Wu, D. Bell, and M. McGinnity, "Multi-knowledge for decision making," *J. Knowl. Inf. Syst.*, vol. 7, no. 2, pp. 246–266, 2005.

[57] Q. H. Hu, M. Y. Wang, and D. R. Yu, "Construct rough decision forests based on sequentially data reduction," in *Proceedings of 5th IEEE International Conference on Machine Learning and Cybernetics*, 2006, pp. 2284–2289.

[58] D. T. Larose, *Discovering Knowledge in Data, An Introduction to Data Mining*. New Jersey: Wiley Interscience, 2005.

[59] Y. Bao and N. Ishii, "Combining multiple k-nearest neighbor classifiers for text classification by reducts," in *Proceedings of 5th International Conference on Discovery Science, LNCS 2534*, 2002, pp. 340–347.

[60] N. Ishii, Y. Morioka, H. Kimura, and Y. Bao, "Classification by Partil Data of Multiple Reducts-kNN with Confidence," in *Proceeding of Ithe 22nd IEEE International Conference on Tools with Artificial Intelligence*, 2010, pp. 94–101.

[61] A. Kusiak, "Decomposition in data mining: an industrial case study," *IEEE Trans. Electron. Packag. Manuf.*, vol. 23, no. 4, pp. 345–353, 2000.

[62] A. Husin and K. R. Ku-Mahamud, "Ant system and weighted voting method for multiple classifier systems," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 6, pp. 4705–4712, 2018.

[63] M. Koziarski, B. Krawczyk, and M. Woźniak, "The deterministic subspace method for constructing classifier ensembles," *Pattern Anal. Appl.*, vol. 20, no. 4, pp. 981–990, 2017.

[64] M. Heidari, "Fault detection of bearings

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

using a rule-based classifier ensemble and genetic algorithm," *Int. J. Eng. Trans. A Basics*, vol. 30, no. 4, pp. 604–609, 2017.

[65] L. Rokach, "Decomposition methodology for classification tasks: A meta decomposer framework," *Pattern Anal. Appl.*, vol. 9, no. 2–3, pp. 257–271, 2006.

[66] O. Maimon and L. Rokach, "Improving supervised learning by feature decomposition," in *Proceedings of the 2nd International Symposium on Foundations of Information and Knowledge Systems*, 2002, pp. 178–196.

[67] Y. Liao and J. Moody, "Constructing heterogeneous committees via input feature grouping: Application to economic forecasting," in *Proceedings of the 12th International Conference on Neural Information Processing Systems*, 1999, pp. 921–927.

[68] L. Rokach and O. Maimon, "Feature set decomposition for decision trees," *J. Intell. Data Anal.*, vol. 9, pp. 131–158, 2005.

[69] H. Ahn, H. Moon, M. J. Fazzari, N. Lim, J. J. Chen, and R. L. Kodell, "Classification by ensembles from random partitions of high-dimensional data," *Comput. Stat. Data Anal.*, vol. 51, no. 12, pp. 6166–6179, 2007.

[70] L. Rokach, B. Chizi, and O. Maimon, "A methodology for improving the performance of non-ranker feature selection filters," *Pattern Recognit. Artif. Intell.*, vol. 21, no. 5, pp. 809–830, 2007.

[71] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans Syst. Man Cybern.*, vol. 22, no. 3, pp. 418–435, 1992.

[72] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms: Second Edition*. New Jersey: John Wiley and Sons, 2004.

[73] K. Woods, W. P. J. Kegelmeyer, and K. Bowyer, "Combination of multiple classifier using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intellegence*, vol. 19, no. 4, 1997.

[74] L. I. Kuncheva, J. C. Bezder, and R. P. W. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognit.*, vol. 34, pp. 299–314.

[75] L. I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 2, pp. 281–286, 2002.

[76] V. S. Costa, A. D. S. Farias, B. Bedregal, R. H. N. Santiago, and A. M. P. Canuto, "Combining Multiple Algorithms in Classifier Ensembles using Generalized Mixture Functions," *Neurocomputing*, vol. 313, pp. 402–414, 2018.

[77] A. J. C. Sharkey, "Types of multinet systems," in *Proceedings of 3rd International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science 2364*, 2002, pp. 108–117.

[78] L. Chen and M. S. Kamel, "A new design of multiple classifier system and its application to the classification of time series data," in *IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 385–391.

[79] W. Lincoln and J. Skrzypek, "Synergy of clustering multiple back propagation networks," *Adv. Neural Inf. Process. Syst.*, vol. 2, pp. 650–659, 1989.

[80] S. Shlien, "Multiple binary decision tree classifiers," *Pattern Recognit.*, vol. 23, no. 7, pp. 757–763, 1990.

[81] F. Lu and F. Yamaoka, "Integration of handwritten digit recognition results using evidential reasoning," in *Proceedings of 4th International Workshop on Frontiers in Handwriting Recognition*, 1994, pp. 456–463.

[82] W. Kim, E., Kim and Y. Lee, "Combination of multiple classifiers for the customer's purchase behavior prediction," *Decis. Support Syst.*, vol. 34, pp. 167–175, 2002.

[83] T. K. Ho, J. J. Hull, S. N. Srihari, and D. Analysis, "On multiple classifier systems for pattern recognition," in *Proceedings of the 11th International Conference on Pattern Recogition*, 1992, pp. 84–87.

[84] Y. . Huang, K. Liu, and C. . Suen, "A neural network approach for multi-classifier recognition systems," in *Proceeding of 4th International Workshop on Frontiers in Handwritten Recognition*, 1994, vol. 235–

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

244.

[85] D. S. Lee and S. . Srihari, "A theory of classifier combination : The neural network approach," in *The 1st Proceedings of the Third International Conference on Document Analysis and Recognition*, 1995, pp. 42–45.

[86] F. A. Breve, M. P. Ponti-Jr, and N. D. A. Mascarenhas, "Neural-network combination for Noisy data classification," in *Proceedings of 2nd Workshop de Visao Computacional*, 2006, pp. 87–92.

[87] C. M. Bishop, *Neural networks for pattern recognition*. Oxford: Clarendon Press, 1995.

[88] G. Lera and M. Pinzolas, "Neighborhood based levenberg-marquardt algorithm for neural network training," *IEEE Trans. Neural Networks Mach. Learn.*, vol. 13, no. 5, pp. 1200–1204, 2002.

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

[90] Y. S. Huang and C. Y. Suen, "The behavior-knowledge space method for combination of multiple classifiers," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 1993, pp. 347–352.

[91] L. I. Kuncheva and L. C. Jain, "Designing classifier fusion Systems by genetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 4, no. 4, pp. 327–336, 2000.

[92] J. A. Aslam and M. Montague, "Models for metasearch," in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001, pp. 276–284.

[93] C. Schaffer, "Selecting a classification method by cross-validation," *Mach. Learn.*, vol. 13, pp. 135–143, 1993.

[94] G. Giancito and F. Roli, "Methods for dynamic classifier selection," in *Proceedings of International Conference on Image Analysis and Processing*, 1999, pp. 659–664.

[95] G. Giancito and F. Roli, "Dynamic classifier selection," in *Proceedings of 1st International Workshop Multiple Classifier Systems*, 2000, pp. 177–189.

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

[97] G. Giancito and F. Roli, "Dynamic classifier selection based on multiple classifier behaviour," *Pattern Recognit.*, vol. 34, pp. 1879–1881, 2001.

[98] L. . Kuncheva, "Switching Between Selection and Fusion in Combining Classifiers: An Experiment," *IEEE Trans. Syst. Man. Cybern.*, vol. 32, no. 2, p. 2002, 2002.

[99] S. Yang and A. Browne, "Neural network ensembles: combining multiple models for enhanced performance using a multistage approach," *Expert Syst.*, vol. 21, no. 5, pp. 279–288, 2004.

[100] S. D. Bay, "Nearest neighbor classification from multiple feature subsets," *Intell. Data Anal.*, vol. 3, pp. 191–209, 1999.

[101] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition : An analysis of its behavior and performance," *IEEE Trans. Syst. Man. Cybern.*, vol. 27, no. 5, pp. 553–568, 1997.

[102] G. Brown and L. I. Kuncheva, "'Good' and 'bad' diversity in majority vote ensembles," in *Proceedings of the 9th International Workshop Multiple Classifier System*, 2010, pp. 124–133.

[103] A. Hajdu, L. Hajdu, A. Jonas, L. Kovacs, and H. Toman, "Generalizing the majority voting scheme to spatially constrained voting," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4182–4194, 2013.

[104] H. Li. and J. Sun, "Majority voting combination of multiple case-based reasoning for financial distress prediction," *Expert Syst. Appl.*, vol. 36, pp. 4363–4373, 2009.

[105] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "An ensemble of filters and classifiers for microarray data classification," *Pattern Recognit.*, vol. 45, no. 1, pp. 531–539, 2012.

[106] D. Albashish, S. Sahran, A. Abdullah, M. Alweshah, and A. Adam, "A Hierarchical Classifier for Multiclass Prostate Histopathology Image Gleason Grading," *J.*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Inf. Commun. Technol., vol. 17, no. 2, pp. 323–346, 2018.

[107] A. M. Anter and A. E. Hassenian, "Normalized Multiple Features Fusion Based on PCA and Multiple Classifiers Voting in CT Liver Tumor Recognition," pp. 113–129, 2018.

[108] S. Amiri, M. M. Movahedi, K. Kazemi, and H. Parsaei, "3D cerebral MR image segmentation using multiple‐classifier system," *Med. Biol. Eng. Comput.*, vol. 55, no. 3, pp. 353–364, 2016.

[109] I. Perikos and I. Hatzilygeroudis, "A Classifier Ensemble Approach to Detect Emotions Polarity in Social Media," in *Special Session on Social Recommendation in Information Systems*, 2016, vol. 1, no. January, pp. 363–370.

[110] H. Kim, H. Kim, H. Moon, and H. Ahn, "Iterative weight-adjusted voting algorithm for ensemble of classifiers," *J. Korean Stat. Soc.*, vol. 40, no. 4, pp. 437–449, 2011.

[111] D. K. Renuka and P. Visalakshi, "Weighted-based multiple classifier and F-SGO algorithm for email spam classification," *Int. J. Bus. Intell. Data Min.*, vol. 12, no. 3, pp. 274–298, 2017.

[112] S. Bashir, U. Qamar, and H. F. Khan, "A Multicriteria Weighted Vote-Based Classifier Ensemble for Heart Disease Prediction," *Comput. Intell.*, vol. 32, no. 4, pp. 615–645, 2015.

[113] C. . Stefano, A. D. Cioppa, and A. Marcelli, "An adaptive weighted majority vote rule for combining multiple classifiers," in *Proceedings. 16th International Conference on Pattern Recognition*, 2002, pp. 192–195.

[114] A. Gangardiwala and R. Polikar, "Dynamically weighted majority voting for incremental learning and comparison of three boosting based approaches," in *Proceedings of the International Joint Conference on Neural Networks*, 2005, pp. 1131–1136.

[115] J. Zhang, L. Cheng, and M. Jun, "A new multiple classifiers combination algorithm," in *Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*, 2006, pp. 287–291.

[116] R. M. Valdovinos and J. S. Sánchez, "Combining multiple classifiers with dynamic weighted voting," in *Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems (HAIS '09)*, 2009, pp. 510–516.

[117] X. Mu, J. Lu, P. Watta, and M. H. Hassoun, "Weighted voting-based ensemble classifier with application to human face recognition and voice recognition," in *Proceedings of International Joint Conference on Neural Networks*, 2009, pp. 2168–2171.

[118] C. Huang and J. Wang, "Multi-weighted majority voting algorithm on support vector machine and its application," in *TENCON 2009 IEEE Region 10 Conference*, 2009, pp. 1–4.

[119] M. Wozniak, "Evolutionary approach to produce classifier ensemble based on weighted voting," in *Proceedings of World Congress on Nature and Biologically Inspired Computing, NaBIC.*, 2009, pp. 648–653.

[120] S. Hamzeloo, H. Shahparast, and M. Zolghadri Jahromi, "A novel weighted nearest neighbor ensemble classifier," in *Proceedings of the 16th International Symposium on Artificial Intelligence and Signal Processing*, 2012, pp. 413–416.

[121] D. Liang, C. Tsai, A.-J. Dai, and W. Eberle, "A novel classifier ensemble approach for financial distress prediction," *Knowl. Inf. Syst.*, vol. 54, no. 2, pp. 437–462, 2018.

[122] R. M. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection," *Inf. Fusion*, vol. 41, no. C, pp. 195–216, 2018.

[123] A. Jain, B. Verma, and J. L. Rana, "Classifier Selection Models for Intrusion Detection System (IDS)," *Int. Eng. an Int. J.*, vol. 4, no. 1, pp. 1–11, 2016.

[124] L. Nanni, A. Lumini, and N. Zaffonato, "Ensemble based on static classifier selection for automated diagnosis of Mild Cognitive Impairment," *J. Neurosci. Methods*, vol. 302, pp. 42–46, 2018.

[125] C. Ballard and W. Wang, "Dynamic ensemble selection methods for heterogeneous data mining," in *12th World Congress on Intelligent Control and Automation (WCICA)*, 2016, pp. 1021–2026.

[126] A. L. Brun, A. S. Britto, L. S. Oliveira, F. Enembreck, and R. Sabourin, “A framework for dynamic classifier selection oriented by the classification problem difficulty,” *Pattern Recognit.*, vol. 76, pp. 175–190, 2018.

[127] P. Yang, Y. H. Yang, B. B. Zhou, and A. Y. Zomaya, “A review of ensemble methods in bioinformatics,” *Curr. Bioinform.*, vol. 5, no. 4, pp. 296–308, 2010.

[128] J. T. Littlestone and M. K. Warmuth, “The weighted majority algorithm,” *Inf. Comput.*, vol. 108, no. 2, pp. 212–261, 1994.

[129] J. Sun and H. Li, “Listed companies' financial distress prediction based on weighted majority voting combination of multiple classifiers,” *Expert Syst. Appl.*, vol. 35, no. 3, pp. 818–827, 2008.