World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:14, No:5, 2020

Rank-Based Chain-Mode Ensemble for Binary Classification

Chongya Song, Kang Yen, Alexander Pons, Jin Liu

***Abstract*—**In the field of machine learning, the ensemble has been employed as a common methodology to improve the performance upon multiple base classifiers. However, the true predictions are often canceled out by the false ones during consensus due to a phenomenon called "curse of correlation" which is represented as the strong interferences among the predictions produced by the base classifiers. In addition, the existing practices are still not able to effectively mitigate the problem of imbalanced classification. Based on the analysis on our experiment results, we conclude that the two problems are caused by some inherent deficiencies in the approach of consensus. Therefore, we create an enhanced ensemble algorithm which adopts a designed rank-based chain-mode consensus to overcome the two problems. In order to evaluate the proposed ensemble algorithm, we employ a well-known benchmark data set NSL-KDD (the improved version of dataset KDDCup99 produced by University of New Brunswick) to make comparisons between the proposed and 8 common ensemble algorithms. Particularly, each compared ensemble classifier uses the same 22 base classifiers, so that the differences in terms of the improvements toward the accuracy and reliability upon the base classifiers can be truly revealed. As a result, the proposed rank-based chain-mode consensus is proved to be a more effective ensemble solution than the traditional consensus approach, which outperforms the 8 ensemble algorithms by 20% on almost all compared metrices which include accuracy, precision, recall, F1-score and area under receiver operating characteristic curve.

***Keywords*—**Consensus, curse of correlation, imbalanced classification, rank-based chain-mode ensemble.

I. Introduction

IN the field of Machine Learning (ML), ensemble is an effective approach to improve the accuracy upon multiple base classifiers. Significant efforts on ensemble algorithms have been made in recent years [1], which have been applied to various domains such as classification [2], regression [3] and clustering [4]-[6]. The overwhelming majority of the practices are adopting the same framework named Multiple Classifier System (MCS) [7] which consists of a defective procedure called consensus.

Since the consensus is responsible for making the final prediction (i.e., labeling) via combining multiple mutual-conflicted predictions made by the base classifiers, such a combination process will introduce four issues in practice: (1) The Curse of Correlation (CoC): the accuracy improvement cannot be guaranteed because the correct predictions are often cancelled out by the wrong ones [8]. (2) The magnitude of accuracy improvement is erratic due to unstable performances on diverse datasets resulting from various unreliable consensus techniques [9]. (3) The selection criterion of the base classifiers is often confined to the accurate base models (i.e., achieve an accuracy of above 50%) for reducing the generalization error [10]. (4) Making the final prediction by predicting every sample multiple time increases the computing overhead. Although the researchers have created numerous intelligent consensus approaches [11]-[22] to overcome the aforementioned four issues, the solutions are still not able to achieve the desired goal "significant accuracy improvement with guarantee." Therefore, this paper will present an ensemble algorithm with an improved Rank-Based Chain-Mode (RBCM) base model invoking methodology to resolve the four issues that come with the consensus.

Chongya Song is with the Florida International University, Miami, Florida 33174, U.S. (corresponding author, phone: 786-670-5380; e-mail: ysong024@fiu.edu).

Kang Yen is with the Florida International University, Miami, Florida 33174, U.S. (phone: 305-348-3037; e-mail: yenk@fiu.edu).

Alexander Pons is with the Florida International University, Miami, Florida 33174, U.S. (phone: 305-348-7253; e-mail: apons@fiu.edu).

Jin Liu is with the Florida International University, Miami, Florida 33174, U.S. (phone: 786-509-2174; e-mail: jliu048@fiu.edu).

II. MCS-Series Ensemble Algorithms

There are a lot of successful cases achieved by the MCS-series ensemble models. For instance, in order to resolve the poor relevance feedback issue that results from the Support Vector Machine (SVM) models when tackle the content-based image retrieval problem, the authors propose a comprehensive model (ABRS-SVM) via combining an asymmetric Bagging-based SVM (AB-SVM) and a Random Sub-space SVM (RS-SVM). As a result, the ABRS-SVM is not been a successful research project, but has also, directly or indirectly resulted in 17 industry patents [23]. Consequently, both of the accomplished ensemble algorithms in the aforementioned case (i.e., Bagging and Random Sub-space) as well as other 6 mature and successful ensemble algorithms (i.e., Randomizable Filter, Random Committee, Random Forest, Extra Trees, AdaBoost and Majority Vote) will be involved in the experiments as the competitors of the RBCM ensemble algorithm.

III. Rank-Based Chain-Mode Ensemble Algorithm

A. Motivation: The Deficiency of the Consensus

Essentially, the consensus of the MCS is trying to mitigate the CoC by making the correct predictions dominate the final result. There are numerous techniques such as voting, weighted voting, Bayesian formalism, belief/certainty, Dempster-Shafer's evidence theory [24]-[30] that have been developed to achieve this goal. However, it is the fact that the CoC is an unavoidable issue once the final prediction is obtained through combining multiple mutual-conflicted predictions no matter how intelligent consensus technique is employed, so the consensus procedure should be abandoned due to this inevitable architectural deficiency. Instead, the CoC can be completely resolved only when every sample is predicted by one of the base models.

B. Workflow

The RBCM ensemble algorithm works in the following steps: (1) Splitting the dataset into three subsets: (a) training set, (b) sorting set (or splitting from the training/evaluating set) and (c) evaluating set. (2) Training the base classifiers by the training set. (3) Testing the base models on the sorting set. (4) Sorting the base models in descending order based on the precisions on two classes, respectively. (5) Constructing two precision-ranked model chains. (6) Summing up the sorted precisions of each model chain: if the base classifiers in a model chain are sorted by the precision on class A, then only calculating the sum of the precisions on this class. (7) Selecting the model chain with a minor sum of the precision. (8) Orderly invoking the first N classifiers (starts from 1 for the best result) and adopting the following three criteria to make prediction on the sorting set: (a) if the model chain in use is ranked by the precision on the class A, then every base classifier is responsible for labeling a sample only when predicting it as the class A; (b) if a sample is not labeled by the current base classifier, it would be labeled by the next base model only when it is predicted as the class A; (c) if a sample is still not labeled until the last classifier, it will be labeled by the class B. A simplified demonstration of the above prediction methodology/ process can be found in Fig. 1. (9) Recurrently produce new model chains by involving the next lower-ranked base classifier into the current ones and adopting the same methodology to test the new chains on the sorting set. (10) Monitoring the accuracies of all the model chains produced in the previous step and identifying the one with the highest accuracy. (11) Employing the aforementioned methodology to evaluate the best model chain on the evaluating set.

Sample IDs →	1	2	3	4	5	6	7	8	9	10
True Labels →	**A**	**B**	**A**	**A**	**A**	**B**	**B**	**A**	**A**	**A**
Model No.1 →	A			A					A	
Model No.2 →					A			A		A
Model No.3 →			A							
Auto Labeling →		B				B	B			
Ensemble Results →	**A**	**B**	**A**	**A**	**A**	**B**	**B**	**A**	**A**	**A**

Fig. 1 The prediction methodology/process of the RBCM (simplified demonstration)

C. Principles

The imbalance classification issue is represented as a classifier which achieves (much) higher precision on the major/ bias class than the minor/non-bias class, hence the accuracy reduction should be mainly attributed to the misclassification on the non-bias class. According to this conclusion, the RBCM ensemble algorithm is able to improve the accuracy due to the following reasons. (1) The objective of determining the model chain with a higher precision on a specific class is to figure out which class the trained classifiers bias to (refer to the steps 1-7 of the workflow). (2) The misclassification on the non-bias class can be minimized as long as we invoke the first classifier to label the samples that are predicted as the bias class (refer to the step 8 of the workflow). (3) The subsequent classifiers will label more biased-class samples with minimum misclassification on the non-bias class. (4) Only a few unlabeled samples belong to the bias class. (5) Labeling the rest as the non-bias class can minimize the misclassification on the bias class (refer to the step 8 of the workflow). In conclusion, the RBCM ensemble classifier improves the accuracy via taking the advantage of each base classifier and then minimizes the misclassification result from every step.

D. Advantages

The RBCM ensemble algorithm can overcome the aforementioned four issues that come with the MCS-series ensemble algorithms. (1) The CoC can be completely avoided (i.e., the accuracy improvement can be guaranteed) because the consensus procedure is no longer employed and the accuracy must be at least as high as the highest-ranked/most precise base classifier in the worst case (i.e., all the subsequent base models misclassify all the unlabeled samples). (2) The magnitude of accuracy improvement would be always significant due to the fact that the aforementioned worst case is (almost) impossible in practice. (3) The selection criterion of the base classifiers becomes much looser for training a classifier with a high precision on either one of the two classes is the prerequisite of training a classifier with high precisions on both classes (i.e., high accuracy). (4) There is a huge advantage on the computing overhead as every base classifier is only make predictions on the unlabeled samples. For instance, there is a dataset with 10,000 samples and the created model chain consisted of 10 base classifiers. (1) For the MCS-series ensemble classifiers, there are 100,000 (i.e., $10 \times 10{,}000$) predictions will be made because every one of the 10 base classifiers need to predict the 10,000 samples. (2) For the RBCM ensemble classifiers, the number of necessary predictions is reduced to only 19,980 (i.e., $10{,}000 \times \left(2-\frac{1}{2^9}\right)$) even if under a conservative assumption that every base model just labels 50% samples it predicts.

IV. Experiments

A. Goal and Methodology

The goal of the two experiments is to make a comparison between the MCS-series and the RBCM ensemble algorithms in terms of the improvement of accuracy and reliability when employing the same group of base classifiers. Since the purpose of achieving the highest accuracy is out of the scope of this paper, every base classifier is trained with the default configuration (set the random state to 0 as well) and the dataset is not pre-processed. In addition, in order to make a comparison between the RBCM and the consensus, a hybrid model will be created by combining the features/principles of the MCS-series and the RBCM ensemble algorithms.

In order to make a comprehensive comparison, there are 22 base models and 20 MCS-series ensemble classifiers (i.e., trained by 8 common ensemble algorithms: Randomizable Filter, Random Sub-space, Random Committee, Random Forest, Extra Trees, AdaBoost, Bagging and Majority Vote, refer to Tables V and VI before Section Conclusion) that adopt the consensus procedure involved in the two experiments. Particularly, for some MCS-series ensemble algorithms (e.g. Bagging) which can only consent upon the same type of base classifiers, the combined base models are trained by 2-4 of the most accurate algorithms (e.g. Perceptron) to maximize the accuracy and competitiveness. A well-known benchmark dataset NSL-KDD [31] which has two classes (i.e., Benign: B, Malicious: M) and comes with two challenging test sets (1) test+ and (2) test-21 is employed to evaluate the models in terms of accuracy, precision (P), recall (R), F1-score (F1) and the Area under the Receiver Operating Characteristic (AUROC).

B. Comparison of the Performances

Tables I and II show the comparisons of the performances among (1) the 20 MCS-series ensemble models, (2) the 22 base classifiers and (3) the RBCM ensemble classifier on the test sets test-21 and test+, respectively. Specifically, there are four groups of performances in each table. (a) The performances of the 20 classifiers are averaged based on (grouped by) the 8 ensemble algorithms and sorted by the accuracy (refer to the rows above the third line from the bottom). (b) The simple-averaged performances of the 20 classifiers (refer to the third row from the bottom). (c) The simple-averaged performances of the 22 base classifiers (refer to the second row from the bottom). (d) The performances of the RBCM ensemble classifier (refer to the last row).

Overall, Tables I and II show that the RBCM ensemble model significantly outperforms the 22 base classifiers and the 20 MCS-series ensemble classifiers. For example, its accuracies are improved by 28.83% (i.e., 82.51%–53.68%) and 26.07% (i.e., 88.84%–73.95%) upon the base classifiers and the MCS-series ensemble classifiers, respectively (refer to the last values in the last two rows of the tables). Most importantly, the RBCM ensemble classifier is able to resolve or at least mitigate the imbalance classification issue. For instance (refer to Table II), it can be indicated that the base classifiers are seriously bias to the class M due to the huge difference (almost 30%) between P(B) and P(M). However, the RBCM ensemble model is able to improve the P(B) by 22.24% (i.e., 85.78%–63.54%) and remain the P(M) within the same level (only reduce 1.94% = 93.25%–91.31%). In the meantime, the R(M) is improved by 30.19% (i.e., 88.86%–58.67%) and the difference of the two recalls is reduced to only 0.004% (i.e., 88.86%–88.82%), which makes an extremely balanced ensemble classifier. The reason of this prominent improvement is that the RBCM ensemble classifier orderly invokes the base models to only label the bias-class (i.e., the class M) samples; hence more M-class samples are labeled with the minimum misclassification on the class B during this process. As a result, labeling the rest samples as the class B will result in the minimum misclassification on the class M and then every metrices will obtain a huge improvement. Therefore, these achievements should be attributed to the RBCM base model invoking methodology because all the tested ensemble classifiers are employing the same group of base classifiers. On the other hand, since the test-21 and test+ are somewhat varied in terms of data pattern, it is unnecessary that the classifiers perform differently on the two test sets. However, if an ensemble classifier presents a great disparity between the performances on the two test sets, it indicates that this model is vulnerable to the pattern variation because the two test sets share the same data source after all. Employing such an ensemble classifier in practice would cause unexpected and/or severe damages, especially in some sensitive organization that requires high-level protections such as the intrusion detection domain. According to the results of the experiments, the aforementioned performance difference results from the MCS-series ensemble classifiers is almost 20% (i.e., 75.89%-56.44%), whereas the same metric is only around 6% (i.e., 88.84%-82.51%) on the RBCM ensemble model. This comparison shows that the RBCM ensemble classifier will perform much stable on the unseen data/diversity data patterns, which means that the RBCM ensemble has a much stronger generalization ability. Furthermore, since the aforementioned performance difference produced by the base classifiers is also around 20% (i.e., 73.95%-53.68%), it shows that the erratic performance achieved by the MCS-series ensemble classifiers is derived from the instability of the base classifiers and the formers completely inherent this disadvantage. Moreover, since the base models are shared by all the tested ensemble classifiers, it can be concluded that the RBCM ensemble is an issue-isolated algorithm that is able to successful resolve the disadvantages that come with the base classifiers.

C. Analysis of the Rank-Based Chain-Mode Base Model Invoking and the Consensus Methodologies

There are only three differences between the RBCM and the MCS-series ensemble algorithms: (1) Utilizing the capabilities of every base classifier fully (i.e., labeling both classes) or partially (i.e., only labeling one class). (2) Employing the base classifiers with or without a certain order (i.e., invoking based on the ranking or not). (3) Labeling each sample via a consensus approach (i.e., combining multiple base predictions) or not. Consequently, a combination of RBCM and majority voting ensemble algorithms is created to reveal the crucial factor that affects the ensemble accuracy. From the respective of prediction, this hybrid model performs in exactly the same way as a RBCM ensemble classifier: orderly invoking the base classifiers based on the ranking and employing every model to only predict the bias class. From the respective of labeling, the hybrid model performs in exactly the same way as a majority voting ensemble classifier: every base model is only responsible for adding one more vote (instead of direct labeling) to a sample that is predicted as the bias class and the final prediction is resulting from combining multiple votes. Namely, the hybrid model should be considered as a special RBCM classifier that employing the voting consensus.

We evaluate the hybrid ensemble classifiers on the test sets test-21 and test+, respectively (refer to Tables III and IV), and draw the following conclusions from the two tables. Due to the fact that the hybrid classifiers are greatly outperformed by the RBCM classifiers and the only difference between the hybrid and the RBCM classifiers is the adoption of consensus, it can be concluded that the consensus is a crucial factor for all the ensemble algorithms and the accuracy reduction result from the hybrid classifier is because of the adoption of consensus. Furthermore, the RBCM model invoking is a more effective methodology in terms of accuracy improvement compared with the consensus.

TABLE I
COMPARISON OF THE PERFORMANCES ON THE TEST-21

Ensemble Classifier	Precision (B)	Recall (B)	F1-score (B)	Precision (M)	Recall (M)	F1-score (M)	AUROC	Accuracy
Randomizable Filter	22.38%	71.80%	34.10%	74.48%	43.33%	53.90%	0.5253	49.85%
Bagging	24.81%	73.00%	36.98%	89.13%	50.21%	64.08%	0.6161	54.35%
AdaBoost	24.12%	63.99%	34.65%	87.53%	54.70%	66.66%	0.5935	56.39%
Random Forest	29.12%	86.90%	43.62%	94.80%	53.06%	68.04%	0.6998	59.21%
Random Sub-space	28.87%	80.17%	42.33%	93.10%	56.13%	69.73%	0.7893	60.52%
Majority Vote	25.94%	62.55%	36.68%	87.90%	60.38%	71.59%	0.6147	60.78%
Random Committee	30.60%	87.70%	45.35%	95.35%	55.75%	70.40%	0.7985	61.58%
Extra Trees	30.51%	86.43%	45.10%	94.93%	56.32%	70.70%	0.7138	61.79%
MCS-series	25.93%	74.35%	38.30%	87.61%	52.13%	64.94%	0.6466	56.44%
Base	25.30%	78.75%	38.20%	91.46%	48.12%	62.50%	0.6344	53.68%
RBCM	51.81%	53.07%	52.43%	89.53%	89.05%	89.29%	0.7106	82.51%

TABLE II
COMPARISON OF THE PERFORMANCES ON THE TEST+

Ensemble Classifier	Precision (B)	Recall (B)	F1-score (B)	Precision (M)	Recall (M)	F1-score (M)	AUROC	Accuracy
Randomizable Filter	62.68%	92.55%	74.73%	87.45%	58.30%	71.10%	0.7915	73.05%
Bagging	64.69%	91.39%	75.70%	90.51%	61.99%	73.45%	0.7669	74.66%
AdaBoost	64.69%	93.73%	76.52%	92.70%	61.10%	73.59%	0.7741	75.15%
Random Forest	64.86%	92.38%	76.21%	91.51%	62.13%	74.01%	0.7726	75.16%
Random Sub-space	65.01%	94.79%	77.13%	93.97%	61.40%	74.27%	0.7809	75.78%
Majority Vote	66.91%	97.10%	79.22%	96.66%	63.66%	76.76%	0.8038	78.06%
Random Committee	68.73%	95.60%	79.90%	95.43%	66.90%	78.50%	0.8557	79.25%
Extra Trees	68.75%	97.25%	80.55%	97.00%	66.60%	78.95%	0.8545	79.80%
MCS-series	65.44%	93.81%	77.06%	92.25%	62.33%	74.55%	0.7982	75.89%
Base	63.54%	94.14%	75.76%	93.25%	58.67%	71.58%	0.7640	73.95%
RBCM	85.78%	88.82%	87.54%	91.31%	88.86%	90.47%	0.8907	88.84%

TABLE III
COMPARISON OF THE PERFORMANCES BETWEEN THE HYBRID AND THE RBCM ENSEMBLE CLASSIFIERS ON THE TEST-21

Ensemble Classifier	Precision (B)	Recall (B)	F1-score (B)	Precision (M)	Recall (M)	F1-score (M)	AUROC	Accuracy
RBCM	51.81%	53.07%	52.43%	89.53%	89.05%	89.29%	0.7106	82.51%
Hybrid	25.86%	80.86%	39.19%	91.96%	48.57%	63.56%	0.6471	54.43%

TABLE IV
COMPARISON OF THE PERFORMANCES BETWEEN THE HYBRID AND THE RBCM ENSEMBLE CLASSIFIERS ON THE TEST+

Ensemble Classifier	Precision (B)	Recall (B)	F1-score (B)	Precision (M)	Recall (M)	F1-score (M)	AUROC	Accuracy
RBCM	85.78%	88.82%	87.54%	91.31%	88.86%	90.47%	0.9544	88.84%
Hybrid	63.99%	95.53%	76.64%	94.61%	59.31%	72.91%	0.7742	74.91%

V. CONCLUSION

The rank-based chain-mode ensemble is an improved ensemble algorithm which is able to avoid the CoC issue through abandoning the traditional consensus procedure used by the MCS-series ensemble algorithms. In addition, it can tackle the imbalance classification issue and provide a strong generalization ability on unseen data via invoking the base classifiers in the RBCM approach and then minimizing the misclassification on every step. Although the current version is dedicated for the binary classification, there are mature solutions such as the pre-processing algorithm LabelBinarizer (converting the multi-class labels to binary labels) that can be used to apply to the proposed ensemble algorithm on multiclass classification problems.

TABLE V
THE PERFORMANCES OF THE 20 COMMON ENSEMBLE CLASSIFIERS ON THE TEST-21

Ensemble Classifier (Base Classifier)	Precision (B)	Recall (B)	F1-score (B)	Precision (M)	Recall (M)	F1-score (M)	AUROC	Accuracy
Random Forest	29.12%	86.90%	43.62%	94.80%	53.06%	68.04%	0.6998	59.21%
Extra Trees	30.51%	86.43%	45.10%	94.93%	56.32%	70.70%	0.7138	61.79%
AdaBoost (Decision Tree)	26.51%	68.77%	38.27%	89.28%	57.70%	70.10%	0.6324	59.71%
AdaBoost (Linear SVM)	23.56%	79.74%	36.37%	90.45%	42.59%	57.91%	0.6116	49.33%
AdaBoost (Gaussian NB)	25.53%	45.63%	32.74%	85.38%	70.47%	77.21%	0.5805	65.96%
AdaBoost (Perceptron)	20.89%	61.80%	31.22%	85.01%	48.05%	61.40%	0.5493	50.55%
Bagging (Decision Tree)	30.95%	85.97%	45.51%	94.86%	57.43%	71.55%	0.717	62.62%
Bagging (Linear SVM)	21.23%	68.40%	32.40%	86.17%	43.68%	57.97%	0.5604	48.17%
Bagging (Gaussian NB)	25.26%	66.91%	36.68%	88.42%	56.07%	68.63%	0.6149	58.04%
Bagging (Perceptron)	21.79%	70.72%	33.31%	87.05%	43.66%	58.15%	0.5719	48.57%
Majority Voting (Decision Tree, Linear SVM, Gaussian NB, Perceptron)	25.94%	62.55%	36.68%	87.90%	60.38%	71.59%	0.6147	60.78%
Randomizable Filter (Decision Tree)	27.40%	87.80%	41.70%	94.70%	48.30%	64.00%	0.6790	55.49%
Randomizable Filter (Bayes Network)	24.20%	80.90%	37.30%	37.30%	37.30%	37.30%	0.3730	50.65%
Randomizable Filter (SGD)	18.80%	53.60%	27.90%	82.60%	48.80%	61.30%	0.5120	49.64%
Randomizable Filter (Perceptron)	19.10%	64.90%	29.50%	83.30%	38.90%	53.00%	0.5370	43.62%
Random Sub-space (Decision Tree)	27.60%	88.20%	42.10%	94.90%	48.60%	64.30%	0.8000	55.84%
Random Sub-space (REP Tree)	28.00%	64.30%	39.00%	88.90%	63.30%	73.90%	0.7680	63.46%
Random Sub-spacer (Random Forest)	31.00%	88.00%	45.90%	95.50%	56.50%	71.00%	0.8000	62.26%
Random Committee (Random Tree)	30.10%	87.50%	44.80%	95.20%	54.80%	69.60%	0.7870	60.77%
Random Committee (Random Forest)	31.10%	87.90%	45.90%	95.50%	56.70%	71.20%	0.8100	62.39%

TABLE VI
THE PERFORMANCES OF THE 20 COMMON ENSEMBLE CLASSIFIERS ON THE TEST+

Ensemble Classifier (Base Classifier)	Precision (B)	Recall (B)	F1-score (B)	Precision (M)	Recall (M)	F1-score (M)	AUROC	Accuracy
Random Forest	66.91%	97.10%	79.22%	96.66%	63.66%	76.76%	0.8038	78.06%
Extra Trees	65.01%	94.79%	77.13%	93.97%	61.40%	74.27%	0.7809	75.78%
AdaBoost (Decision Tree)	68.16%	92.92%	78.63%	92.61%	67.15%	77.85%	0.8003	78.25%
AdaBoost (Linear SVM)	62.33%	91.17%	74.04%	89.71%	58.31%	70.68%	0.7474	72.47%
AdaBoost (Gaussian NB)	67.20%	88.61%	76.44%	88.64%	67.27%	76.50%	0.7794	76.47%
AdaBoost (Perceptron)	61.08%	92.85%	73.69%	91.08%	55.23%	68.76%	0.7404	71.44%
Bagging (Decision Tree)	69.03%	96.91%	80.63%	96.63%	67.10%	79.21%	0.8201	79.94%
Bagging (Linear SVM)	61.56%	92.67%	73.98%	91.02%	56.21%	69.50%	0.7444	71.92%
Bagging (Gaussian NB)	66.47%	92.40%	77.32%	91.84%	64.73%	75.94%	0.7857	76.65%
Bagging (Perceptron)	61.70%	92.92%	74.16%	91.31%	56.35%	69.69%	0.7463	72.10%
Majority Voting (Decision Tree, Linear SVM, Gaussian NB, Perceptron)	64.86%	92.38%	76.21%	91.51%	62.13%	74.01%	0.7726	75.16%
Randomizable Filter (Decision Tree)	65.30%	97.10%	78.10%	80.90%	61.00%	74.70%	0.8090	76.51%
Randomizable Filter (Bayes Network)	63.00%	95.60%	76.00%	94.60%	57.60%	71.60%	0.8120	73.98%
Randomizable Filter (SGD)	62.90%	87.30%	73.10%	86.40%	61.00%	71.50%	0.7420	72.36%
Randomizable Filter (Perceptron)	59.50%	90.20%	71.70%	87.90%	53.60%	66.60%	0.8030	69.36%
Random Sub-space (Decision Tree)	65.50%	97.40%	78.30%	96.90%	61.20%	75.00%	0.8470	76.78%
Random Sub-space (REP Tree)	71.50%	92.10%	80.50%	92.30%	72.30%	81.10%	0.8560	80.79%
Random Sub-spacer (Random Forest)	69.20%	97.30%	80.90%	97.10%	67.20%	79.40%	0.8640	80.16%
Random Committee (Random Tree)	68.30%	97.20%	80.20%	96.90%	65.90%	78.40%	0.8460	79.38%
Random Committee (Random Forest)	69.20%	97.30%	80.90%	97.10%	67.30%	79.50%	0.8630	80.23%

REFERENCES

[1] A. C. Braun, U. Weidner, and S. Hinz, "Classification in high-dimensional feature spaces assessment using SVM, IVM and RVM with focus on simulated EnMap data," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 5, no. 2, pp. 436-443, Apr. 2012.

[2] M. Asafuddoula, B. Verma, and M. Zhang, "A Divide-and-Conquer-Based Ensemble Classifier Learning by Means of Many-Objective Optimization," IEEE Transactions on Evolutionary Computation, vol. 22, no. 5, pp. 762-777, Dec. 2017.

[3] S. Li, P. Wang, and L. Goel, "Wind Power Forecasting Using Neural Network Ensembles With Feature Selection," IEEE Transactions on Sustainable Energy, vol. 6, no. 4, pp. 1447-1456, Oct. 2015.

[4] D. Huang, J. Lai, and C. Wang, "Robust Ensemble Clustering Using Probability Trajectories," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 5, pp. 1312-1326, May. 2016.

[5] D. Huang, J. Lai, and C. Wang, "Locally Weighted Ensemble Clustering," IEEE Transactions on Cybernetics, vol. 48, no. 5, pp.

1460-1473, May. 2018.
[6] Z. Yu, P. Luo, J. You, H. Wong, H. Leung, S. Wu, J. Zhang, and G. Han, "Incremental Semi-Supervised Clustering Ensemble for High Dimensional Data Clustering," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 3, pp. 701-714, Mar. 2016.
[7] 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, Jun. 1992.
[8] L. I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms. USA: NY: New York: Wiley-Interscience, 2004, ch. 6.
[9] L. I. Kuncheva, "Diversity in multiple classifier systems," Information Fusion, vol. 6, no. 1, pp. 3-4, Mar. 2005.
[10] P. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, Pearson, 2006.
[11] A. Bagnall, J. Lines, J. Hills and A. Bostrom, "Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 9, pp. 2522-2535, Sept. 2015.
[12] L. Breiman, Bagging Predictors, Springer, Aug. 1996, pp. 123-140.
[13] E. Bauer, and R. Kohavi, An Empirical Comparison Of Voting Classification Algorithms: Bagging, Boosting, And Variants, Springer, July. 1999, pp. 105-139.
[14] S. Merler, B. Caprile, and C. Furlanello, "Parallelizing AdaBoost by weights dynamics," Journal of Computational Statistics & Data Analysis, vol. 51, no. 5, pp. 2487-2498, Feb, 2007.
[15] C. Zhang, and J. Zhang, "A local boosting algorithm for solving classification problems," Journal of Computational Statistics & Data Analysis, vol. 52, no. 4, pp. 1928-1941, Jan, 2008.
[16] T. K. Ho, "The random subspace method for constructing decision forests," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832-844, Aug. 1998.
[17] L. Breiman, Random forests. Springer, Oct. 2001, pp. 5-32.
[18] P. Geurts, D. Ernst, and L. Wehenke, Extremely Randomized Trees. Springer, Apr. 2006, pp. 3-42.
[19] Z. Yu, L. Li, and J. Liu, and G. Han, "Hybrid Adaptive Classifier Ensemble," IEEE Transactions on Cybernetics, vol. 45, no. 2, pp. 177-190, Feb. 2015.
[20] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online Ensemble Learning of Data Streams with Gradually Evolved Classes," IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 6, pp. 1532-1545, June. 2016.
[21] Y. Sun, L. L. Minku, S. Wang, and X. Yao, "A learning framework for online class imbalance learning," *Conf. 2013* IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), pp. 36-45.
[22] Y. Sun, L. L. Minku, S. Wang, and X. Yao, "Resampling-Based Ensemble Methods for Online Class Imbalance Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 5, pp. 1356-1368, May. 2015.
[23] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 7, pp. 1088-1099, July 2006.
[24] B. G. Buchanan, and E.G. Shortliffe, Rule based expert systems, USA: MA: Addison-Wesley, 1984.
[25] S. Shilen, Multiple binary tree classifiers, Pattern Recognition, 1990, pp. 757-763.
[26] W. L. Buntine, "A theory of learning classification rules. Doctoral Dissertation," PhD Dissertation, School of Computing Science, University of Technology. Sydney, Australia, Nov, 1992.
[27] P. Derbeko, R. E. Yaniv, and R. Meir, "Variance optimized bagging," *Conf.* 2002 European Conference on Machine Learning, pp. 60-72.
[28] J. V. Hansen, "Combining predictors: Meta machine learning methods and bias, variance & ambiguity decompositions," PhD Dissertation. Aurhus University, 2000.
[29] A. F. R, Rahman, and M. C. Fairhurst, "A new hybrid approach in combining multiple experts to recognize hand-written numerals," Pattern Recognition Letters, vol. 18: no. 8, pp. 781-790, Aug, 1997.
[30] K. Tumer, and J. Ghosh, "Robust Order Statistics based Ensembles for Distributed Data Mining," Advances in distributed and parallel knowledge discovery. AAAI/MIT Press, Cambridge, pp 185-210.
[31] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.