JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
http://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Sainin, M. S., & Alfred, R., & Ahmad, F. (2021). Ensemble meta classifier with sampling and feature selection for data with imbalance multiclass problem. *Journal of Information and Communication Technology, 20*(2), 103-133. https://doi.org/10.32890/jict2021.20.2.1

Ensemble Meta Classifier with Sampling and Feature Selection for Data With Multiclass Imbalance Problem

[1]Mohd Shamrie Sainin, [2]Rayner Alfred & [3]Faudziah Ahmad
[1&2]Faculty of Computing and Informatics
Universiti Malaysia Sabah, Malaysia
[3]School of Computing, Universiti Utara Malaysia, Malaysia

shamrie@ums.edu.my
ralfred@ums.edu.my, fudz@uum.edu.my

Received: 31/3/2020 Revised: 26/8/2020 Accepted: 13/9/2020 Published: 23/2/2021

ABSTRACT

Ensemble learning by combining several single classifiers or another ensemble classifier is one of the procedures to solve the imbalance problem in multiclass data. However, this approach still faces the question of how the ensemble methods obtain their higher performance. In this paper, an investigation was carried out on the design of the meta classifier ensemble with sampling and feature selection for multiclass imbalanced data. The specific objectives were: 1) to improve the ensemble classifier through data-level approach (sampling and feature selection); 2) to perform experiments on sampling, feature selection, and ensemble classifier model; and 3) to evaluate the performance of the ensemble classifier. To fulfil the objectives, a preliminary data collection of Malaysian plants' leaf images was prepared and experimented, and the results were compared. The ensemble design was also tested with three other high imbalance ratio benchmark data. It was found that the design using sampling, feature selection, and ensemble classifier method via AdaboostM1 with random forest (also an ensemble classifier) provided improved performance throughout the investigation. The result of this study is important to the on-going problem of multiclass imbalance where specific structure and its performance can be improved in terms of processing time and accuracy.

Keywords: Imbalance, multiclass, ensemble, feature selection, sampling.

INTRODUCTION

With the advancement of the industrial revolution 4.0, more data are being captured, stored, processed, and analysed. Imbalance problem is still the leading challenge in real-world data and most of the recent works proposed to address this problem. In machine learning, a multiclass classification problem refers to assigning one of the several class labels with an input object. Unlike binary classification, learning a multiclass problem is a more complex task since each example can only be assigned to exactly one class label. Numerous attempts at using binary classification methods have failed to perform well in multiclass classification problems. There are three categories of methods proposed for learning multiclass classification problems, namely 1) direct multiclass classification technique using a single classifier; 2) binary conversion classification techniques; and 3) hierarchical classification techniques. A direct classifier is any algorithm that can be applied naturally to solve multiclass classification problems directly, such as neural network, decision tree, k-Nearest neighbour (k-NN), Naive Bayes (NB), and support vector machine (SVM) (Mehra & Gupta, 2013). In contrast, if the process requires several steps to change, select, and preprocess certain data before the classification, then it is called an indirect method, or also identified as a hybrid approach.

Many scholars in the related domain research argue that although a direct classifier algorithm can easily be applied to solve multiclass classification problems, the performance of a single classifier is not encouraging when solving real-world problems. One of the possible effects of this problem is the existence of multiclass imbalanced data. A multiclass imbalance problem refers to a dataset with a target class that is skewed in distribution and poses a significant effect on classifier performance. Ensemble classifiers offer a distinctive arrangement approach by joining the quality of a few single classifiers. This classifier has been suggested as the promising trend in machine learning with different ensemble methods discussed in Ren et al. (2016) and Feng et al. (2018).

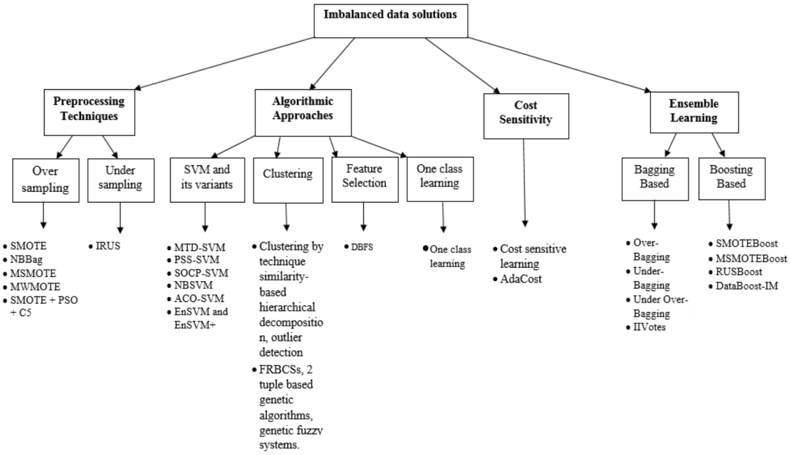
This paper aims to find the possible hybrid of sampling, feature selection, and ensemble classifiers as well as the order of which process (feature selection and sampling) that may increase the performance of the overall process. The approach introduces the diversity in training data and ensemble design. The investigation flow begins from the dataset, the effect of sampling and feature selection using single classifiers and then moves to ensemble classifiers. Fine-tuning the ensemble design at the end of the phase is another crucial point as specific settings and parameter tuning could improve the performance and execution time.

LITERATURE REVIEW

A recent review by Ali et al. (2019) related to the imbalance problem in data mining showed that the issue of imbalance is still prevalent. In the paper, the issues are to distinguish between the imbalance ratio measurement and lack of information. Normally, the ratio is measured as the number of minorities divided by the number of majorities. However, lack of information was not discussed. Other factors that were also considered when discussing imbalance are data overlapping, overfitting, small disjoint, small data, and high dimensionality. The detailed taxonomy of imbalance as proposed in the work is depicted in the following Figure 1. According to the taxonomy, this work focused on preprocessing (data-level method) and algorithmic method (feature selection) in a sequential process before ensemble learning is applied.

Figure 1

Taxonomy of Imbalanced Data, Solution, and Example Methods by Ali et al. (2019).



Ensemble Methods

An ensemble method is defined as an approach that applies several single classifiers or may combine more diverse learners where the classification will be identified using a method (known as a committee of expert decision) for classifying new unseen instances. As mentioned by Ren et al. (2016), Weka offers conventional ensemble methods including random forest, boosting and bagging, rotation forest, DECORATE, END, and stacking. Adaptive boosting (AdaBoost) and bagging (bootstrap aggregation) are the most popular techniques to construct ensembles, which leads to significant improvement in some applications (Galar et al., 2012). Adaptive boosting or AdaBoost was proposed by Freund and Schapire (1996). It is a popular ensemble learning framework with good classification performance over general datasets. However, when the imbalance problem exists within the dataset, the algorithm cannot be applied directly to the training sample without much attention to the minority class (Li et al., 2019). Therefore, it is important to investigate the data level manipulation of the dataset before the algorithm can be applied.

Bagging is an ensemble method introduced by Breiman (1996), where some base classifiers are induced by a similar learning algorithm and certain samples by bootstrapping. In this method, the ensemble operates on a bootstrapped portion of training data using different N classifiers. Different random datasets (with replacement) from the original dataset are constructed for training each classifier to achieve the ensemble diversity through resampling. Prediction by the classifiers is finalised based on the equal weight majority voting (Jerzy et al., 2013). MultiBoostAB is the extension of the boosting method, specifically the AdaBoost algorithm, which constructs strong decision committees (Webb, 2000). The algorithm combines AdaBoost and wagging together by reducing the AdaBoost's bias and variance. It was reported that by using the decision tree of C4.5, the method demonstrated a lower error more often when tested on a large representative of the University of California Irvine (UCI) datasets.

Diverse ensemble creation by oppositional relabelling of artificial training examples (DECORATE) is the ensemble method introduced by Melville and Mooney (2004), which manipulates and generates diverse hypotheses with syntactically produced training samples. The main advantage of DECORATE is the concept of diversity in the ensemble constructed during the creation of artificial training instances. An ensemble of nested dichotomies or END is constructed using standard statistical techniques to address polytomous classification problems with logistic regression (Gu & Jin, 2014). It was originally represented using binary trees that iteratively split multiclass data into a system of dichotomies. Recent studies using END include hydraulic brake health monitoring (Jegadeeshwaran & Sugumaran, 2015), adaptive nested dichotomies (Leathart et al., 2016), and evolving nested dichotomies classifier with genetic algorithm for optimisation (Wever et al., 2018).

In other ensemble designs, rotation forest is a classifier ensemble based on feature extraction. It was introduced by Rodríguez and Kuncheva (2006) with the aim to acquire distinct accuracy and diversity in the ensemble design. It works by preparing the portion of training samples for the base classifier, where a certain set of features are based on *K* random subsets. Then, the principal component analysis is performed, retaining each of the components to preserve the dataset information variability. Furthermore, the key factor for the algorithm to succeed is the application of a transformation matrix for calculating the extracted features that are sparse (Kuncheva & Rodrıguez, 2007). Since then, the ensemble algorithm has been used in various studies such as ensemble model in spatial modelling of groundwater potential (Naghibi et al., 2019), comparing meta classifier approach based on rotation forest (Tasci, 2019), and applying rotation forest for imbalance problems (Guo et al., 2019). The sample works indicated that the performance was improved on various evaluations.

Stacking was first introduced by Wolpert (1992) based on the stack generalisation, where it tries to reduce the error rate using one or more classifiers. It is said to learn by induction of biases of the classifier according to the single training dataset and finally voting is applied as the baseline method for combining the result of the classifier. One of the recent uses of this method is shown in Rajagopal et al. (2020) that discussed network intrusion detection. Based on the study, the stacking method is able to achieve superior performance as indicated by the reported accuracy.

Multiclass Imbalance Problem

When a dataset exists with an unequal number of examples between its classes, it is called an imbalanced data problem. Analysing and learning from such distribution is a challenge. The general definition of imbalanced data is that the classes sample is not 50:50 and can be viewed from minor (2:1 to 18:1) to highly imbalanced (19:1 or more). The multiclass classification problem becomes more difficult to be solved with the existence of a highly imbalanced dataset. Ding (2011) stated that if the imbalance ratio in a general classification problem is no less than 19:1 with a minority class size of only 5 percent of the entire data size, then it is called the learning problem of a highly imbalanced classification problem (imbalance learning).

A high imbalance ratio is defined from the problem-solving point of view as any imbalance of 19:1 and more than 50:1 is considered a severe high imbalance problem. This will make complex and challenging modelling of the smaller class sample (Triguero et al., 2015). In extreme cases, academic scholars characterise the imbalance problem as a majority class where the ratio is replicated due to the dominant part to minority class proportion varieties from 100:1 to 10,000:1 (Leevy et al., 2018). Imbalance in data has been discovered as a challenging issue in machine learning problems (Bia & Zhang, 2017) and is considered a long-standing problem to date. The dominant class will generally blind the traditional classifier and almost disregard the marginal class, giving unsuitable classification performance (Dong et al., 2019). Consequently, to some extent, the accuracy of the imbalanced class can be improved by using resampling or a sensitive based learning technique (Krawczyk, 2016).

Imbalance problem solution approaches are characterised by algorithm-level and data-level. Data-level methods consist of row-based (e.g. sampling) or column-based (e.g. feature selection, which is also an algorithmic–level method). This categorisation was first mentioned by Garcia et al. (2007) and a new categorisation was constructed as shown in Figure 1. While data-level methods can produce balanced data for the classifier to work, however, the methods could lead to duplicates or potential data loss; thus, overfitting may occur. Algorithm–level methods, on the other hand, have two possible utilisations; either a specific new algorithm construction or tuning the existing algorithm to produce high performance on imbalance problems.

Sampling

Oversampling and undersampling are known as traditional resampling approaches. Although sampling has a significant contribution to classifier performance, its drawbacks were reviewed in Wang and Yao (2012). Oversampling is inclined towards producing more samples and may cause overfitting to the minority classes (it can be shown by the low recall and high precision or F-measure). In contrast, undersampling suffers from performance loss in majority classes due to its sensitivity to the number of minority classes. Resample and SMOTE are two of the sampling methods in Weka that were used in this study.

Feature Selection

Another significant research in machine learning that can be considered as preprocessing is feature selection (otherwise called attribute subset selection or reduction of attribute). The feature selection that is investigated in this work is the attempt to find the solution to the class imbalance problem and to discover the significant feature subset that improves classification accuracy. There are several methods in feature selection techniques that are explicitly applied to reduce the dimensionality of features in data. There are three general methods involved: filter, wrapper, and embedded (Ladha & Deepa, 2011). These strategies vary fundamentally on how the search works on available features. Filter techniques focus on the issue of determining the features as an autonomous procedure from the model choice (e.g. when induction generalisation is not required).

Interestingly, wrapper strategies match the search hypothesis with the classifiers to gain feedback on whether the model is suitable or needs improvement. According to this strategy, different mixtures of subset features are produced and then it will be evaluated whether the model has improved. In contrast, embedded techniques will scan for an ideal subset and are structured inside the classifier development. A few examples of applied feature selection are bio-inspired feature selection (Basir et al., 2018), information gain, Gain Ratio, etc. (Mohsin et al., 2014), and wrapper-based genetic algorithm (Barati et al., 2013).

Filter-based Attribute Evaluators

Correlation-based feature subset selection strategy or CfsSubsetEval (CFS) aims at the hypothesis containing subset features that are exceptionally related with the class, yet have no relationship with one another (Hall, 1999). In that sense, each feature is the test that measures traits related to the class using merit evaluation. It then will calculate the relationship between properties by discretisation and pursue by measuring the uncertainty symmetrical value. In this paper, the wrapper feature selection technique is found to be comparable to CFS. Nevertheless, CFS is better on a small dataset and general runtime execution.

ConsistencySubsetEval depends on statistical probabilistic estimation in dealing with feature selection that is fast and straightforward, thus ensured to get the ideal feature subset given the appropriate resources (Liu & Setiono, 1996). An explanatory correlation of filter-based techniques was investigated on CSE and CFS utilising performance estimation using decision tree classifiers in Onik et al. (2015), where CFS gave less feature subsets. However, CSE with the BestFirstSearch method has a better accuracy rate. Essentially, a filter-based subset selection based on FilteredSubsetEval (FSE) requires running in a flexible feature evaluator on the training data to get the best possible feature subset.

Wrapper-based Attribute Evaluator

The subset evaluation known as the WrapperSubsetEval method originally presented by Kohavi and John (1997) is a feature subset selection strategy that utilises inductive learning as an evaluator for subset selection. It uses built-in five-fold-cross validation (5-cv) for evaluating the subset's worth. The accuracy estimation of this method needs to quantify the significance of the selected features. In the investigation, the technique indicates noteworthy improvement using two algorithms: decision tree and Naïve Bayes. This method was used in studies such as attribute selection in hepatitis patients (Samsuddin et al., 2019), feature selection to improve diagnosis and classification of neurodegenerative disorders (Álvarez et al., 2019), feature selection for gene expression (Hameed et al., 2018), and data attribute selection approach for drought modelling (Demisse et al., 2017).

Samsuddin et al. (2019) performed a study on the attribute selection to hepatitis data (a two-class dataset acquired from UCI repository) using various attribute evaluators (CfsSubsetEval, WrapperSubsetEval, Gain Ratio Subset Eval, and Correlation Attribute Eval) and classifiers (Naive Bayes Updatable, SMO, KStar, random tree, and SimpleLogistic). The study concluded that CfsSubsetEval was the best selector while SMO was the best classifier. The performance of SMO with feature selection as reported in the paper was 85 percent, while other reported results achieved 84 percent with Naïve Bayes without feature selection (Karthikeyan & Thangaraju, 2013), and 93.06 percent using feature selection and an ensemble of neuro-fuzzy (Nilashi et al., 2019). In comparison to the studies, the present work is specifically applied to a problem where there exists an imbalance in the multiclass data (more than two classes), and at the same time has many features (high dimensionality). Thus, the complexity of finding the solution to combine sampling, feature selection, and ensemble classifier design is an important step to the investigation.

METHODOLOGY

This study performed several experiments and systematic comparisons, and at the end suggested an improved ensemble classifier design for imbalanced multiclass data, which were indicated in three phases, namely Phase 1: Theoretical study, Phase 2: Data collection, and Phase 3: Hybrid ensemble design. In the previous section, several ensemble design methods were described. In support of this ensemble design, two data-level methods (sampling and feature selection) that were also discussed in the literature review were proven to enhance the classification performance in multiclass imbalance problems. Thus, to improve the ensemble classifier, this paper aims to develop the ensemble classifier with sampling, feature selection, and ensemble method as a hybrid ensemble design. In the data collection phase, preliminary dataset construction was started on the collection of Malaysian medicinal leaf images. A total of 65 leaf samples were arbitrarily chosen from five indicated species. This dataset was obtained from a village in the state of Perlis, Malaysia. The list of leaf species samples is listed in Table 1 and the dataset description in Table 2.

Table 1

Five Species of Malaysian Medicinal Plant

Class	Sample Image	#Training Samples	#Testing Samples
Mengkudu		6	4
Lakom		5	4
Cemumar		11	4
Kapal Terbang		12	4
Kemumur Itik		11	4
		45	20

Table 2

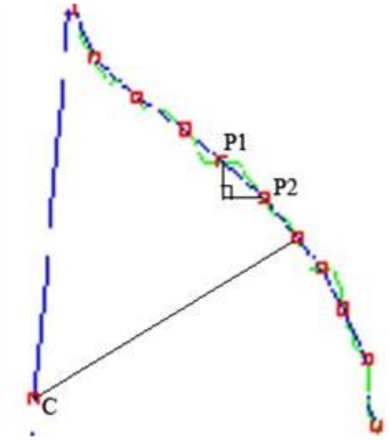
Detailed Information of Dataset

Description	Value
Number of Samples	65
Number of Features/Attributes	564
Training Samples	45
Testing Samples	20
Majority Samples	12
Minority Samples	5

There were two steps in preparing the dataset, which were image preprocessing and feature extraction based on shape. The image processing procedure involved converting the image to greyscale, edge detection (using Prewitt edge detection), and thinning (to minimise the boundary of the leaf to one pixel). Shape-based leaf feature is one of the most popular approaches for feature extraction as many research have shown that this approach provides not only speed-up image processing but low cost and conveniences (Langner, 2006).

Figure 2

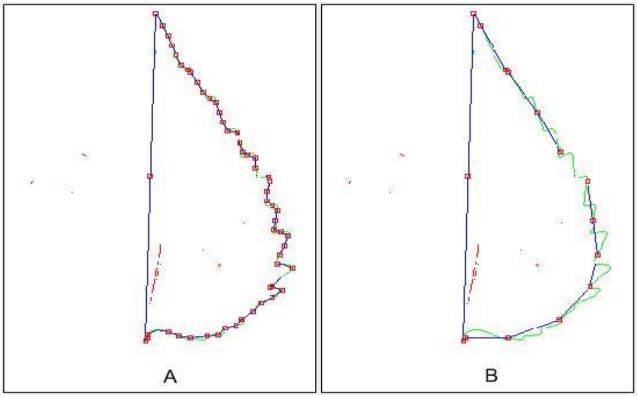
Sample of the Leaf with Points P1 and P2 that Produce the Hypotenuse Angle.



The shape can be constructed using a boundary line, which is the contour coordinate or distance from the leaf centroid. In this study, the coordinate of points from the leaf's shape are used based on angle direction (hypotenuse) of the points concerning the sinus and cosines of the two adjacent points as depicted in Figure 2. The number of points along the leaf shape was determined using a certain distance between the points, where the shorter the distance, the more points were created, and the larger the leaf would also affect the number of features as shown in Figure 3. Finally, the study concluded that 564 was the highest number of points (angles) and therefore became the features. The smaller leaf would fill 0 as the remaining feature value if the shape point was lesser than 564 features.

Figure 3

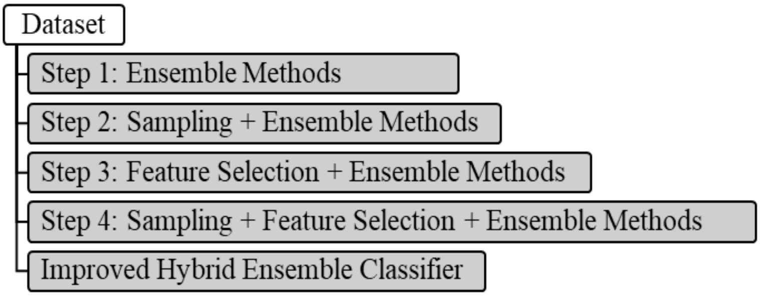
Effect of Distance Between the Points. (A) More Points if the Distance is (E.g. 1), in which more Information is Extracted from the Leaf, and (B) Fewer Points if the Distance Is (E.g. 3).



Generally, the dataset represented a high dimensionality that it might impose the challenge of the possible problems such as small dataset, but high dimensionality and multiclass with an imbalance ratio of approximately 1:3. Although the imbalance ratio was less than 1:19, this sample provided the case of low severity of imbalance but small dataset size and high dimensionality. In further experiments, the findings from this preliminary dataset were applied to the high severity of imbalance (1:19 – 1:4559). Consequently, the dataset could be utilised to explore the impacts of class imbalance to classifier accuracy. In this manner, the created dataset was intended to indicate the performance of the classifiers on the available samples. Furthermore, Phase 3 investigated the possible improved combination of ensemble design, which would be finalised among the designs that produced the best average accuracy on the Malaysian medicinal leaf image dataset. The process workflow of the phase is illustrated in Figure 4.

Figure 4

Workflow for Finalising the Improved Hybrid Ensemble Classifier.


Dataset
Step 1: Ensemble Methods
Step 2: Sampling + Ensemble Methods
Step 3: Feature Selection + Ensemble Methods
Step 4: Sampling + Feature Selection + Ensemble Methods
Improved Hybrid Ensemble Classifier

RESULTS AND FINDING

Ensemble Design Phase (Step 1 – Ensemble Method)

Before the ensemble classifier was designed, it is important to investigate the performance of the data according to the single classifiers. There were five classifiers used (although many other classifiers were tested), which were obtained from WEKA (3.8) and were examined using the data specified in Table 1. The results shown in Table 3 were accuracies and F-measure for the class using default settings in Weka. The percentage accuracy showed that the imbalance problem and high dimensionality greatly affected the performance of the classifiers. The two classifiers, namely SMO and J48, managed to achieve 60 percent accuracy. However, looking at class performance, SMO had better F-measure for every class as compared to J48.

Based on the result, F-measure for class C2 (Kapal Terbang) was 0 for J48 where none of the samples was correctly classified, although the class was among the majority (many samples). J48 tried to fit the minority classes (CL4 and CL5) with high F-measure but performed poorly on other majority samples. In other findings, classifiers such as Naïve Bayes and random forest were examples of algorithms that tried to balance the performance among classes. One common achievement among the classifiers was the F-measure values for class Mengkudu (with only six training samples) that were almost similar (1.0, except for PART).

Table 3

Classification Performance of Single Classifiers

Classifier	% Accuracy	Class F-Measure					$\bar{x}$	ROC	Time
		CL1	CL2	CL3	CL4	CL5			
NaiveBayes	50.00	0.57	0.50	0.29	0.20	1.00	0.51	0.82	0.11
SMO	**60.00**	**0.50**	**0.67**	**0.40**	**0.40**	**1.00**	**0.59**	**0.83**	**0.17**
PART	50.00	0.33	0.00	0.50	0.55	0.80	0.44	0.73	0.06
J48	**60.00**	**0.60**	**0.00**	**0.29**	**0.80**	**1.00**	**0.54**	**0.77**	**0.03**
Random Forest	45.00	0.20	0.57	0.36	0.00	1.00	0.43	0.82	0.19

CL1: Cemumar, CL2: Kapal Terbang, CL3: Kemumur Itik, CL4: Lakom, CL5: Mengkudu

According to the result, SMO performed slightly better than the other classifiers when looking at class F-Measure and in fact, the weighted F-Measure, $\bar{x}$was the highest at 0.59. This indicated that SMO managed to work with a small dataset and high dimensional imbalanced multiclass data. Based on this result, Step 1 in the ensemble design (Figure 4) was investigated through several experiments. In this progression, the investigation utilised seven ensemble design strategies and single classifiers as a base classifier (SMO, decision tree (J48), random tree (RT), NB, and RF) in Weka by applying their best parameter settings. Estimates that were seen in every ensemble used accuracy, receiver operating characteristic (ROC), and F-measure, which were regularly applied in estimating the performance of positive prediction on all classes and their true positive rate. Table 4 shows the results of seven ensemble methods with single base classifiers. Stacking in the result was performed according to combining the single base classifiers using the stacking ensemble method.

Table 4

Ensemble Classifier Performance

Ensemble Method	Base Classifier Performance (%)					Average
	NB	J48	SMO	RT	RF	
DECORATE	50	60	60	60	60	58
Bagging	50	55	40	65	60	54
END	45	65	55	60	60	57
RotationForest	60	55	55	60	65	59
Stacking	65	**70**	55	60	65	63
AdaBoostM1	50	**70**	60	65	**70**	63
MultiBoostAB	55	**70**	60	65	**70**	64

As indicated by the result above, strategies utilising Stacking, MultiBoostAB, and AdaboostM1 yielded comparable performance of 70 percent given the base classifier (RF and J48). With the average percentage of 75 percent, Random Forest was noted as the best base classifier. AdaboostM1 and MultiboostAB outperformed the result obtained by a single classifier in Table 3 (SMO with 60%). This was due to the boosting method on the classifiers, where AdaboostM1 started with one classifier and iteratively added another classifier to the ensemble until some criterion was reached.

The detailed accuracy by class when using AdaboostM1 with J48 and RF as base classifiers is presented in Table 5. Examining the accuracy by class, AdaboostM1 with RF as the base classifier had better performance as compared to using J48, despite having a similar percentage accuracy (70%). AdaboostM1 with J48 had the advantage of better classification on the minority class as shown by F-measure in class leaf Lakom and Mengkudu, yet lower performance on the majority class. This was because the boosting ensemble focused excessively on the minority class. Based on the results in this step, it suggested that AdaboostM1 with random forest as base classifiers could be used to improve the performance of imbalanced multiclass data. The best classifier performance using a single classifier was 60 percent by J48, while ensemble classifier AdaboostM1 combined with base classifier RF had increased the performance to 70 percent.

Table 5

Evaluation Metrics by Class

	Precision	F-Measure	ROC Area	Class
AdaboostM1 and J48	0.333	0.286	0.641	Cemumar
	1	0.4	0.734	Kapal Terbang
	0.5	0.667	0.813	Kemumur Itik
	1	1	1	Lakom
	1	1	1	Mengkudu
AdaboostM1 and RF	0.4	0.444	0.836	Cemumar
	1	0.857	0.938	Kapal Terbang
	0.571	0.727	0.93	Kemumur Itik
	1	0.4	0.914	Lakom
	1	1	1	Mengkudu

Ensemble Design (Step 2 – Sampling + Ensemble Method)

Step 2 of the ensemble design was to investigate the design of sampling and then the ensemble method. There were two sampling approaches applied, namely resample and SMOTE. Specifically, resample was applied with two options, without replacement, and with replacement. The reason for applying with and without replacement in the resample method was to examine the effect of each sample interpolation using available small data to the dataset. However, this paper only showed the results of sampling with replacement as the sampling without replacement had no significant increase in performance. Likewise, minority oversampling was done using SMOTE filter to balance the dataset.

First, the number of ensemble classifiers was identified and applied based on the previous selection, namely RotationForest, MultiBoostAB, Bagging, AdaboostM1, DECORATE, Stacking, and END. In the experiments, the selected base classifiers to examine the accuracy of the ensemble in respect to the sampling method were RandomForest (RF), RandomTree (RT), Naïve Bayes (NB), SMO, and J48. Based on this setting, the results of the percentage of accuracies are listed in Table 6. It was found that DECORATE was the best performer with RF as a base classifier. Additionally, most of the ensemble classifiers agreed that RF was the best base classifier with 75 percent accuracy as compared to the result in Table 4.

Table 6

Resample (with Replacement) Result

	RF	RT	NB	SMO	J48	Average
AdaBoostM1	**75**	45	45	40	60	53
Bagging	**75**	55	45	50	60	57
DECORATE	**75**	**65**	**50**	**55**	**60**	**61**
END	**75**	65	45	45	45	55
MultiBoostAB	**70**	45	45	50	55	53
RotationForest	**75**	65	45	50	60	59
Stacking	**55**	50	55	50	45	51

Further resample method in the sampling was tested using a setting of 150 percent to examine the effect of adding more samples for balancing the dataset. Table 7 is the result of the experiments. Based on the performance of the ensemble classifier for all base classifiers, RF performed averagely better among the other classifiers, except using RT. In terms of the best base classifier, RF was again the best with increased accuracies of 80 percent for DECORATE and Rotation Forest. However, caution should be noted for this method as 150 percent oversampling created many samples during the execution and could impose the problem of overfitting although it was not fully investigated in this study.

Table 7

Sampling Using Resample (150% Rate) and with Replacement

	RF	RT	NB	SMO	J48	Average
AdaBoostM1	**70**	50	50	60	60	58
Bagging	**65**	60	50	40	60	55
DECORATE	**80**	60	45	55	55	59
END	**75**	50	45	55	50	55
MultiBoostAB	**70**	50	50	60	50	56
RotationForest	**80**	**45**	**60**	**50**	**80**	**63**
Stacking	**60**	60	55	40	65	56

The next experiment examined the sampling method using SMOTE with its default settings. The results are presented in Table 8, where it was found that SMOTE might not quite be suitable as a sampling method for the ensemble classifiers. Based on the values, END with RF base classifier was the only ensemble with a higher rate, which was 75 percent.

Table 8

SMOTE Result

	RF	RT	NB	SMO	J48	Average
AdaBoostM1	70	35	40	55	60	52
Bagging	70	40	55	50	55	54
DECORATE	70	60	55	60	55	60
END	75	40	50	55	55	55
MultiBoostAB	65	35	55	55	50	52
RotationForest	60	55	60	50	60	57
Stacking	60	60	65	65	55	61

Summarising the results shown in Tables 6–8, detailed class performance using the combination of the best ensemble classifiers and sampling methods are shown in Table 9. According to the results, the F-measure test accuracies of the methods were almost similar. However, the class accuracy indicated a different point of view, where some methods were good at majority class but performed poorly in minorities (indicated by the F-measures). Performing sampling with replacement at a 150 percent rate and ensemble classifiers using RotationForest and DECORATE (with RF as a base classifier) could produce overall better class accuracy. According to the results, ensemble classifier (RotationForest or DECORATE) combined with sampling (with replacement), 150 percent sample size (oversampling) on minority classes, and using RF as base classifier produced better accuracy at 80 percent. These ensemble methods were worth further investigation in the next step.

Table 9

F-Measure for Classes

Sampling Method	Classifier	Rate %	CL1	CL2	CL3	CL4	CL5
Resample+Single	RF	75	0.50	0.80	0.75	0.67	1.00
Ensemble with Replacement	DECORATE + RF	75	0.33	1.00	0.67	0.67	1.00
	Bagging + RF	75	0.40	0.80	0.73	0.67	1.00
	END + RF	75	0.40	0.80	0.73	0.67	1.00
	RotationForest + RF	75	0.40	0.80	0.73	0.67	1.00
Ensemble + SMOTE	END + RF	75	0.33	0.89	0.73	0.67	1.00
Ensemble with Replacement (150%)	END + RF	75	0.29	0.89	0.80	0.67	1.00
	RotationForest + RF	80	0.57	0.80	0.89	0.67	1.00
	DECORATE + RF	80	0.57	0.80	0.89	0.67	1.00

CL1: Cemumar, CL2: Kapal Terbang, CL3: Kemumur Itik, CL4: Lakom, CL5: Mengkudu

Ensemble Design (Step 3 – Feature Selection with Ensemble Classifier)

In this step, experiments were performed to seven ensemble classifiers using a feature selection method based on FSE+Resample (150% rate), best-first search method, and 20 features. These settings were selected because of their overall better performance in the feature selection experiments. The result of feature selection with the ensemble classifier is presented in Table 10. The improved performance of several classifiers was indicated by average accuracy of 78.90 percent when using RF as base classifiers, as well as other base classifiers showing a similar trend. DECORATE had better average performance across base classifiers, while Bagging produced the highest classification accuracy and RF was still the best base classifier considering significant feature reduction (from 564 to 20 features).

Table 10

Classifier Performance Using FSE+Resample and Bestfirst Search Methods

	NB	SMO	J48	RF	RT	Average
AdaBoostM1	60.00	61.54	58.46	80.00	53.85	62.77
Bagging	64.62	69.23	63.08	**83.08**	58.46	67.69
DECORATE	66.15	63.08	69.23	80.00	66.15	68.92
END	61.53	64.61	56.92	81.53	61.54	65.23
MultiBoostAB	63.08	64.62	58.46	80.00	53.85	64.00
RotationForest	64.62	66.15	69.23	73.85	66.15	68.00
Stacking	52.31	58.46	56.92	73.85	53.85	59.08

The highest accuracy was achieved by the Bagging method at 83.08 percent using an RF base classifier. This result further confirmed that RF was still the best base classifier while ensemble methods varied from experiments. However, based on the results in previous tables, the DECORATE ensemble method showed a common pattern that could be a good design for the multiclass imbalance problem. Table 11 compares the F-Measure score for both Bagging and DECORATE where they had almost similar F-measure values across classes.

Table 11

F-Measure Comparison in Each Class

Search Method	Ensemble	Base	CL1	CL2	CL3	CL4	CL5	Average
FSE+Resample	Bagging	RF	0.77	0.79	0.81	0.89	0.95	**0.83**
FSE+Resample	DECORATE	RF	0.71	0.77	0.75	0.89	0.95	0.80

Ensemble Design (Step 4 – Sampling and Feature Selection with Ensemble Method)

Having the best combination (sampling and feature selection) with the ensemble method identified in previous steps, further experiments were conducted to investigate whether combining these three components in the sequential hybrid ensemble design would improve the classification performance. In Step 2, sampling FSE+Resample (150%) with RotationForest ensemble using RF or DECORATE ensemble using RF produced 80 percent accuracy over the data. Furthermore, in Step 3, feature selection FSE+Resample (150%) with Bagging using RF had the highest accuracy at 83.08 percent, followed by END using RF (81.53%) and DECORATE using RF (80%). Based on this evidence, the next experiments were utilised the settings to combine the three components. Fortunately, FSE+Resample (150%) was already a combination of the three components. The ordering of components (whether sampling or feature selection came first) was also investigated if it could contribute to a performance increase. According to the result in the previous section, Resample (150%) and combined with ensemble classifier RotationForest+RF and DECORATE+RF provided the best method. FSE+Resample combined with Bagging + RF and FSE+Resample with DECORATE+RF gave the best performance.

Although that sampling provided the best results, other sampling techniques for oversampling and down-sampling were experimented to observe the effect of the combinations. Four sampling methods were investigated to be combined with feature selection and ensemble classifier, namely ClassBalancer, Resample, and SMOTE using five-cross validation. Table 12 shows the comparison of the combinations of sampling, feature selection, and ensemble classifier. Based on the result, Resample (with balanced class sampling) combined with the FSE+Resample feature selection approach showed a significant increase in the performance of the ensemble classifier as compared to the result obtained in Table 10. Among the best design was Resample-uniform + (FSE+Resample) + (AdaboostM1+RF), where AdaboostM1 was the ensemble method and RF was the base classifier with 98.46 percent classification accuracy.

Table 12

Performance of the Combined Sampling then Followed by Feature Selection and Ensemble Classifier

Ensemble Classifier	Class Balancer + FSE + Resample (17 Attributes)	Resample (uniform)+ FSE + Resample (18 Attributes)	SMOTE + FSE + Resample (17 Attributes)
AdaboostM1+RF	73.31	**98.46**	72.97
Bagging+RF	68.06	95.38	74.32
DECORATE+RF	71.61	96.92	78.38
END+RF	**74.64**	95.38	**79.73**
MultiBoostAB+RF	73.31	**98.46**	77.03
RotationForest+RF	71.97	93.85	78.38
Stacking+RF	65.58	87.69	68.92

Ensemble Design (Feature Selection then Sampling and Ensemble Classifier)

Four experiments were conducted in this design, i.e. feature selection with (ClassBalancer, Resample with uniform class samples, and SMOTE). The results of this design are summarised in Table 13. A combination of sampling, feature selection, and ensemble classifier was again proven as the best hybrid ensemble design, which consisted of the three components. Although the best ensemble classifier varied in the experiments, almost all ensemble classifiers tested in the experiments were comparable and higher than other designs. The best design was FSE + Resample + Sampling (Resample uniform class) + END + RF, where END was the ensemble method and RF was the base classifier with 94.86 percent classification accuracy.

Table 13

The Performance of Combined Feature Selection then Followed by Sampling and the Ensemble Classifier

Ensemble Classifier	FSE + Resample + Class Balancer (19 Attributes)	FSE + Resample + Resample (uniform) (19 Attributes)	FSE + Resample + SMOTE (19 Attributes)
AdaboostM1+RF	69.47	91.43	70.27
Bagging+RF	69.03	90.00	70.27
DECORATE+RF	70.44	92.86	**74.32**
END+RF	73.31	**94.29**	**74.32**
MultiBoostAB+RF	69.47	91.43	70.27
RotationForest+RF	**75.61**	90.00	71.62
Stacking+RF	62.36	90.00	67.57

Ensemble Design Results Discussion

The results demonstrated that three components of the hybrid ensemble classifier (sampling, followed by feature selection and ensemble classifier) provided the best performance execution as shown in Table 12 in comparison to the ordering in Table 13. Table 14 summarises the results beginning from the original data until the three combinations of the hybrid classifier were designed based on 5-cv to examine the different training and testing effects. Hybrid 1 was (sampling + feature selection + ensemble classifier) while Hybrid 2 was (feature selection + sampling + ensemble classifier). Based on Table 14, the combination of sampling first then followed by feature selection provided the best performance. Precisely, AdaboostM1+RF and MultiBoostAB+RF were the best ensemble classifier designs with 98.46 percent. Surprisingly, sampling with an ensemble classifier performed almost similar to the combination of the three components (sampling + feature selection + ensemble classifier). With the existence of an imbalance problem in multiclass data, sampling was proven to be one of the important methods to solve the problem. RandomForest also performed better although this algorithm was not implemented with different structures of ensemble classifier.

Table 14

Results of 5-cv

Classifier	Original Data	Sampling	Feature Selection	Hybrid 1	Hybrid 2	Average
Random Forest	73.85	96.92	73.85	**93.85**	92.31	86.15
AdaboostM1+RF	70.77	93.85	70.77	**98.46**	91.43	85.05
Bagging+RF	73.85	95.38	73.85	**95.38**	90.00	85.69
DECORATE+RF	75.38	95.38	75.38	**96.92**	92.86	87.19
END+RF	75.38	95.38	73.85	**95.38**	94.29	86.86
MultiBoostAB+RF	70.77	90.77	70.77	**98.46**	91.43	84.44
RotationForest+RF	75.38	95.38	78.46	**93.85**	90.00	86.62
Stacking+RF	64.62	90.77	73.85	**87.69**	90.00	81.38

Benchmark Data (Large and High Imbalance Ratio)

Apart from the experiments on the Malaysian medicinal dataset, comparisons were also performed on publicly available benchmark data. In this investigation, three benchmark datasets were selected, which were highly imbalanced and large data, using the best method as shown in Tables 13 and 14. Table 15 summarises the dataset.

Table 15

Selected Three Benchmark Datasets

Data	#N	#A	Att. Type	#C	#Min	#Max	Ratio	Previous Result
Page Blocks	5473	10	Real, Integer	5	28	4913	1:175	97.28% (Eschrich et al., 2002)
Statlog (Landsat)	6345	36	Integer	6	56	1072	1:19	89.3% (Ghosh et al., 2014)
Statlog (Shuttle)	58000	9	Integer	6	10	45586	1:4559	96.3% (Cohen et al., 2004)

#N: Number of samples, #A: Number of attributes, #C: Number class #Min: Size of minimum class, #Max: Size of maximum class

Using the above data, the results of combining ensemble classifiers with sampling and feature selection are listed in Table 16. In this method, the performance increased as compared to the previous sample results, where Landsat achieved 98.51 percent, 96.83 percent for Shuttle, and 99.09 for PageBlocks. Considering the time for the model build (in Weka), RandomForest showed superiority among other methods where the processing time was less than a second with comparable accuracy.

Table 16

The Performance of the Benchmark Datasets Using the Best Method from Tables 13 and 14

	Landsat	Time	Shuttle	Time	Page Blocks	Time
DECORATE+RF	98.26	126.73	**96.83**	10.74	98.94	126.23
MultiBoostAB+RF	98.51	1.48	96.67	0.38	98.88	4.57
RotationForest+RF	98.12	14.93	95.67	0.42	**99.09**	3.67
Random Forest	98.26	0.46	96.67	0.03	98.99	0.28
AdaboostM1+RF	**98.51**	1.09	96.67	0.37	98.98	4.02
Bagging+RF	98.23	9.95	96.67	0.28	98.92	2.60

Final Design of the Hybrid Ensemble Classifier

Upon completing all experiments and performance investigations using different configurations of the hybrid ensemble design, “one design that fits all” did not exist. However, it was demonstrated that when using ensemble classifiers that were further combined with sampling and feature selection, the performance of the multiclass imbalanced large data could be enhanced as shown in the results. Thus, the proposed hybrid ensemble classifier (using AdaBoostM1+RF) is summarised in the following pseudo-code for future use as in Algorithm 1.

Algorithm 1: Hybrid ensemble method

```
Declaration 1:
D(x) is a distribution of dataset
T(x) is a training dataset (optional)
CV is a cross validation value (0 if no CV)
Input: D(x), T(x), CV
Instances data = D(x)
Instances test = T(x)
Instances new_Train = Resample(data) OR SMOTE(data)
Instances new_Train = FeatureSelection(FSE+Resample, new_Train)
Classifier c = meta(AdaBoostM1) + base(RandomForest)
Model m = c.buildClassifier(new_Train, CV)
evaluate_Model (m, new_Train)
if(test)
      evaluate_Model(m, test)
```

CONCLUSION

As mentioned in the earlier section, multiclass imbalance is still an on-going problem in real-world data mining and machine learning when data are greatly affected by a high imbalance ratio between samples where one or more classes have fewer samples while the other classes have too many samples. In this study, the design of the hybrid classifier was carried out using Weka, a well-known machine learning tool that can be customised to create a new classifier structure. Various design configurations were designed and experimented using machine learning packages to find the best design. Clear evidence from the results presented in the previous section is that the performance of hybrid ensemble classifiers were improved as compared to single classifiers. While hybrid ensemble classifiers tested in this study performed almost similar, there is no hybrid ensemble classifier that best fit to different multiclass imbalanced datasets. If there is no alteration to the data, the ensemble classifier with Random Forest can perform superior to any single classifier.

Moreover, accuracy can be further improved when Random Forest is included as a meta classifier (as a base classifier) such as MultiBoostAB and AdaBoostM1, sampling technique (SMOTE or Resample), and feature selection (FSE+Resample). Based on the results and proposed hybrid ensemble classifier, this study contributes towards the machine learning and data science community by presenting the method to handle multiclass imbalance learning problems. In other contributions, this work also provides a preliminary study on Malaysian medicinal leaf identification and classification using the computational method. While possible methods for handling multiclass imbalance is presented in this paper, the problem for big data is still prominent, where computational power is required to perform the processing of the massive amount of data. Thus, future research recommendations include investigation of an updatable hybrid ensemble classifier (known as incremental learning) and other big data processing for highly imbalanced multiclass data.

ACKNOWLEDGMENT

This work was supported by the Fundamental Research Grant Scheme (FRGS) under Grant No. 13142 (2014).

REFERENCES

Ali, H., Salleh, M. N. M., Saedudin, R., Hussain, K., & Mushtaq, M. F. (2019). Imbalance class problems in data mining: A review. Indonesian Journal of Electrical Engineering and Computer Science, 14(3), 1560–1571. https://doi.org/ 10.11591/ijeecs.v14.i3.pp1552-1563

Álvarez, J. D., Matias-Guiu, J. A., Cabrera-Martín, M. N., Risco-Martín, J. L., & Ayala, J. L. (2019). An application of machine learning with feature selection to improve diagnosis and classification of neurodegenerative disorders. *BMC Bioinformatics*, *20*(491). https://doi.org/10.1186/s12859-019-3027-7

Barati, M., Abdullah, A., Mahmod, R., Mustapha, N., & Udzir, N. I. (2013). Features selection for IDS in encrypted traffic using genetic algorithm. In *Proceedings of the 4th International Conference on Computing and Informatics* (pp. 279–285). http://psasir.upm.edu.my/id/eprint/41307

Basir, M. A., Yusof, Y., & Hussin, M. S. (2018). Optimization of attribute selection model using bio-inspired algorithms. *Journal of Information and Communication Technology*, *18*(1), 35–55.

Bia, J., & Zhang, C. (2017). An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. *Knowledge-Based Systems*, *158*, 81–93. https://doi.org/10.1016/j.knosys.2018.05.037

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1023/a:1018054314350

Cohen, I., Cozman, F. G., Sebe, N., Cirelo, M. C., & Huang, T. S. (2004). Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction. *IEEE Trans. Pattern Anal. Mach. Intell.*, *26*, 1553–1567. https://doi.org/10.1109/TPAMI.2004.127

Demisse, G. B., Tadesse, T., & Bayissa, Y. (2017). Data mining attribute selection approach for drought modelling: A case study for Greater Horn of Africa. *International Journal of Data Mining & Knowledge Management Process*, *7*(4), 1–16. http://doi.org/10.5121/ijdkp.2017.7401

Dong, Q., Gong, S., & Zhu, X. (2019). Imbalanced deep learning by minority class incremental rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(6), 1367–1381. https://doi.org/10.1109/TPAMI.2018.2832629

Eschrich, S., Chawla, N. V., & Hall, L. O. (2002). Generalization methods in bioinformatics. In *2nd International Conference on Data Mining in Bioinformatics (BIOKDD'02)* (pp. 25–32).

Feng, W., Huang, W., & Ren, J. (2018). Class imbalance ensemble learning based on the margin theory. *Applied Science*, *8*(5), 815. https://doi.org/10.3390/app8050815

Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, Bari, Italy (pp. 148–156).

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, *42*(4), 463–484. https://doi.org/10.1109/TSMCC.2011.2161285

Garcia, V., Sanchez, J. S., Mollineda, R. A., & Sotoca, J. M. (2007). The class imbalance problem in pattern classification and learning. In *Tamida 2007, Saragossa, Spain* (pp. 283–291).

Ghosh, S., Biswas, S., Sarkar, D., & Sarkar, P. P. (2014). A tutorial on different classification techniques for remotely sensed imagery datasets. *Smart Computing Review*, *4*(1), 34–43. https://doi.org/10.6029/smartcr.2014.01.004

Gu, S., & Jin, Y. (2014). Generating diverse and accurate classifier ensembles using multi-objective optimization. *IEEE*

Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM), 9–15. https://doi.org/10.1109/MCDM.2014.7007182

Guo, H., Diao, X., & Liu, H. (2019). Improving undersampling-based ensemble with rotation forest for imbalanced problem. *Turkish Journal of Electrical Engineering & Computer Sciences*, *27*, 1371–1386. https://doi.org/10.3906/elk-1805-159

Hall, M. A. (1999). *Correlation-based feature subset selection for machine learning* (Unpublished Doctoral Thesis). The University of Waikato.

Hameed, S. S., Petinrin, O. O., Hashi, A. O., & Saeed, F. (2018). Filter-wrapper combination and embedded feature selection for gene expression data. *Int. J. Advance Soft Compu. Appl*, *10*(1), 90–105.

Jegadeeshwaran, R., & Sugumaran, V. (2015). Health monitoring of a hydraulic brake system using nested dichotomy classifier – A machine learning approach. *International Journal of Prognostics and Health Management*, *6*(1), 1–10.

Jerzy, B., Stefanowski, J., & Idkowiak, Ł. (2013). Extending bagging for imbalanced data. In *Proceedings of the 8th International Conference on Computer Recognition Systems CORES* (pp. 269–278). https://doi.org/10.1007/978-3-319-00969-8_26

Karthikeyan, T., & Thangaraju, P. (2013). Analysis of classification algorithms applied to hepatitis patients. *International Journal of Computer Applications*, *62*(15), 25–30. https://doi.org/10.5120/10157-5032

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324. https://doi.org/http://dx.doi.org/10.1016/S0004-3702(97)00043-X

Krawczyk, B. (2016). Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221–232. https://doi.org/10.1007/s13748-016-0094-0

Kuncheva, L. I., & Rodrıguez, J. J. (2007). An experimental study on rotation forest ensembles. In *Multiple Classifier Systems: 7th International Workshop, MCS 2007* (pp. 459–468). https://doi.org/10.1007/978-3-540-72523-7_46

Ladha, L., & Deepa, T. (2011). Feature selection methods and algorithms. *International Journal on Computer Science and Engineering (IJCSE)*, *3*, 1787–1797.

Langner, J. (2006). *Neuronal network based recognition system of leaf images* (Vol. 2009). http://www.jens-langner.de/lrecog/

Leathart, T., Pfahringer, B., & Frank, E. (2016). Building ensembles of adaptive nested dichotomies with random-pair selection. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 179-194). https://doi.org/10.1007/978-3-319-46227-1_12

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, *5*(1), 42. https://doi.org/10.1186/s40537-018-0151-6

Li K., Zhou G., Zhai, J., Li, F., & Shao M. (2019). Improved PSO_ AdaBoost ensemble algorithm for imbalanced data. *Sensor*, *19*(6), 1476. https://doi.org/10.3390/s19061476

Liu, H., & Setiono, R. (1996). A probabilistic approach to feature selection - A filter solution. In *13th International Conference on Machine Learning* (pp. 319–327).

Mehra, N., & Gupta, S. (2013). Survey on multiclass classification methods. *International Journal of Computer Science and Information Technologies, 4*(4), 572–576.

Melville, P., & Mooney, R. J. (2004). Creating diversity in ensembles using artificial data. *Information Fusion*, *6*, 99–111. https://doi.org/10.1016/j.inffus.2004.04.001

Mohsin, M. F. M., Hamdan, A. R., & Bakar, A. A. (2014). An evaluation of feature selection technique for dendrite cell algorithm. In *Proceedings of the International Conference on IT Convergence and Security (ICITCS)* (pp. 1–5). https://doi.org/10.1109/ICITCS.2014.7021732

Naghibi, S. A., Dolatkordestani, M., Rezaei, A., Amouzegari, P., Heravi, M. T., Kalantar, B., & Pradhan, B. (2019). Application of rotation forest with decision trees as base classifier and a novel ensemble model in spatial modeling of groundwater potential. *Environmental Monitoring and Assessment*, *191*(248). https://doi.org/10.1007/s10661-019-7362-y

Nilashi, M., Ahmadi, H., Shahmoradi, L., Ibrahim, O., & Akbari, E. (2019). A predictive method for hepatitis disease diagnosis using ensembles of neuro-fuzzy technique. *Journal of Infection and Public Health*, *12*(1), 13–20. https://doi.org/10.1016/j.jiph.2018.09.009

Onik, A. R., Haq, N. F., Alam, L., & Mamun, T. I. (2015). An analytical comparison on filter feature eextraction method in data mining using J48 classifier. *International Journal of Computer Applications*, *124*(13), 1–8. https://doi.org/10.5120/ijca2015905706

Rajagopal, S., Kundapur, P. P., & Hareesha, K. S. (2020). A stacking ensemble for network intrusion detection using heterogeneous datasets. *Security and Communication Networks*, *2020*, 4586875. https://doi.org/10.1155/2020/4586875

Ren, Y., Zhang, L., & Suganthan, P. N. (2016). Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational Intelligence Magazine*, *11*(1), 41–53. https://doi.org/10.1109/MCI.2015.2471235

Rodríguez, J. J., & Kuncheva, L. I. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *28*(10), 1619–1621. https://doi.org/10.1109/TPAMI.2006.211

Samsuddin, S., Shah, Z. A., Saedudin, R. D. R., Kasim, S., & Seah, C. Sen. (2019). Analysis of attribute selection and classification algorithm applied to hepatitis patients. *International Journal on Advanced Science, Engineering and Information Technology*, *9*(3), 967–971. http://dx.doi.org/10.18517/ijaseit.8.5.5041

Tasci, E. (2019). A meta-ensemble classifier approach: Random rotation forest. *Balkan Journal of Electrical & Computer Engineering*, *7*(2), 182–187. https://doi.org/10.17694/bajece.502156

Triguero, I., del Río, S., López, V., Bacardit, J., Benítez, J. M., & Herrera, F. (2015). ROSEFW-RF: The winner algorithm for the ECBDL’14 big data competition: An extremely imbalanced big data bioinformatics problem. *Knowledge-Based Systems*, *87*, 69–79. https://doi.org/https://doi.org/10.1016/j.knosys.2015.05.027

Wang, S., & Yao, X. (2012). Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *2*(4), 1119–1130. https://doi.org/10.1109/TSMCB.2012.2187280

Webb, G. I. (2000). MultiBoosting: A technique for combining boosting and wagging. *Machine Learning*, *40*, 159–196. https://doi.org/10.1023/A:1007659514849

Wever, M., Mohr, F., & Hüllermeier, E. (2018). Ensembles of evolved nested dichotomies for classification. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 561–568. https://doi.org/10.1145/3205455.3205562

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, *5*(2), 241–260. https://doi.org/10.1016/S0893-6080(05)80023-1