remote sensing

MDPI

Article

A Novel Double Ensemble Algorithm for the Classification of Multi-Class Imbalanced Hyperspectral Data

Daying Quan [1], Wei Feng [2,*], Gabriel Dauphin [3], Xiaofeng Wang [1], Wenjiang Huang [4] and Mengdao Xing [2]

[1] Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou 310018, China
[2] Hangzhou Institute of Technology, Xidian University, Hangzhou 311200, China
[3] Laboratory of Information Processing and Transmission, L2TI, Institut Galilée, University Paris XIII, 93430 Villetaneuse, France
[4] Key Laboratory of Digital Earth Science, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China
[*] Correspondence: wfeng@xidian.edu.cn; Tel.: +86-138-9575-1095

Abstract: The class imbalance problem has been reported to exist in remote sensing and hinders the classification performance of many machine learning algorithms. Several technologies, such as data sampling methods, feature selection-based methods, and ensemble-based methods, have been proposed to solve the class imbalance problem. However, these methods suffer from the loss of useful information or from artificial noise, or result in overfitting. A novel double ensemble algorithm is proposed to deal with the multi-class imbalance problem of the hyperspectral image in this paper. This method first computes the feature importance values of the hyperspectral data via an ensemble model, then produces several balanced data sets based on oversampling and builds a number of classifiers. Finally, the classification results of these diversity classifiers are combined according to a specific ensemble rule. In the experiment, different data-handling methods and classification methods including random undersampling (RUS), random oversampling (ROS), Adaboost, Bagging, and random forest are compared with the proposed double random forest method. The experimental results on three imbalanced hyperspectral data sets demonstrate the effectiveness of the proposed algorithm.

Keywords: classification; remote sensing; hyperspectral image; imbalance learning; data sampling

check for updates

Citation: Quan, D.; Feng, W.; Dauphin, G.; Wang, X.; Huang, W.; Xing, M. A Novel Double Ensemble Algorithm for the Classification of Multi-Class Imbalanced Hyperspectral Data. *Remote Sens.* **2022**, *14*, 3765. https://doi.org/10.3390/rs14153765

Academic Editor: Junjun Jiang

Received: 22 June 2022
Accepted: 2 August 2022
Published: 5 August 2022

1. Introduction

Hyperspectral image (HSI) is one of the most important data types in the remote sensing field [1]. These images could provide rich information and have obtained lots of applications, including land use surveys, environmental monitoring, mineral development, and so on [2,3]. Machine learning, which automates analytical model building to find hidden insights, is a core sub-area of hyperspectral remote sensing research [4]. Generally speaking, classification approaches can be divided into unsupervised and supervised methods [5]. Unsupervised classification methods, such as K-means, graph-based methods, do not require labeled samples [6]. The relationship between clusters and classes with too little prior knowledge cannot be determined [7]. The supervised classification methods generally present better performance but need the class labels of the training data. However, many supervised classification methods suffer from the class imbalanced problem which exists when some classes (minority classes) are underrepresented as compared to the other classes (majority classes) [8–10]. For example, random forests (RF), which has been employed in many applicative domains, become intractable when an imbalance problem appears [10]. In addition, the "curse of dimensionality" makes imbalanced HSI processing face more challenges concerning other data classification tasks [11,12].

Dealing with imbalanced multi-class tasks is harder than dealing with binary ones [10,13–16]. In binary problems, the roles of classes are easily defined: one is majority and the other one is minority [17]. When dealing with multiple classes, a given class may be at the same time majority when compared to some, minority to others, and even in a relative balance with the rest [4]. This makes designing any balancing algorithms highly challenging, as they must take into account these various roles [17]. The purpose of multi-class imbalance learning is to provide high classification accuracy for the minority classes without heavily compromising the accuracy of the majority classes [10]. A larger number of methods have been proposed to solve the class imbalance problem. They can be divided into three categories: feature-level methods, data-level methods, and algorithm-level methods. Ensemble methods, which combine several base models to produce one strong predictive model [18–20], is one new method to deal with imbalanced data [10,21]. Different from the feature selection, data sampling, and normal algorithms, ensemble-based methods can combine multiple data sampling and classifier optimization algorithms, thus indirectly alleviating the effect of the data imbalance problem. For example, Ribeiro et al. [22] propose a multi-objective optimization design approach for imbalanced learning. This method has four steps: multi-objective ensemble member generation, multi-objective ensemble member selection, multi-objective ensemble member combination, and multi-objective ensemble member selection and combination. Wang et al. [23] developed a multi-matrices entropy discriminant ensemble learning method. This method introduced the multi-matrices approach and nearest entropy into the base classifier. Chen et al. [24] propose a method by combining ensemble learning with the union of a margin-based undersampling and diversity-enhancing oversampling. Wang et al. propose a hybrid ensemble learning strategy named Sample and Feature Selection Hybrid Ensemble Learning [25]. Qin et al. [26] combined the particle swarm optimization-based wrapper method and the dynamic oversampling approach with adaptive boosting to propose a hybrid multi-class imbalanced learning method. The wrapper method is utilized to adaptively select the optimal feature subset, and DySBoost is applied to solve the multi-class class imbalanced problem. Cmv et al. [27] designed a novel sequential ensemble learning framework. The method can divide the majority instances into multiple small and disjoint subsets for training multiple weak learners without compromising accuracy [27]. These researches prove that the problem of class imbalance can be effectively solved by ensemble learning. However, most of these methods do not consider the high dimension problems of imbalance learning.

According to the above presentation, hyperspectral data is commonly characterized by high dimension and multi-class imbalanced nature. This requires dedicated algorithms which use information about relationships among features and classes for multi-class imbalanced data. Most existing state-of-the-art solutions are designed for the binary problem and are not capable of processing multi-class data. This calls for developing efficient algorithms for handling multi-class imbalanced problems. Hence, this paper proposes a double random forest (DRF) method for multi-class imbalance learning by the idea of hybrid ensemble theory. This method firstly computes the feature importance values of the hyperspectral data, then produces several balanced data sets based on oversampling and builds a number of classifiers. Finally, the classification results of these diversity classifiers are combined according to a specific ensemble rule. In the experiment, different data handling methods and classification methods including random undersampling (RUS), random oversampling (ROS), Adaboost, Bagging, and random forest, are compared with the proposed double random forest method.

Thus, the contribution of this paper is to design a hybrid random forest framework that generates an ensemble and is biased toward the minority class(es). The rest of this paper is structured as follows. Section 2 briefly explains and reviews some of the related state-of-the-art approaches developed to tackle the class imbalance problem. Section 3 describes the proposed approach, DRF. Section 4 presents various measures for evaluating the DRF's performance. Section 5 discusses the superior performance of the proposed work. Finally, Section 6 summarizes this paper.

2. Related Works

2.1. Feature-Level Methods

The main goal of the feature-level methods is to select a subset of K features from the original feature space, to increase the distinction between majority and minority classes. Then, redundancy and noisy information can be discarded; thus, this kind of method could reduce the risk of overfitting and lead to better classification performance. Moreover, the feature-level methods can be divided into embedded methods and filter methods. The approach integrates feature selection and classifier learning into a single process.

K-nearest neighborhood (KNN) and Principal Component Analysis (PCA)-based approaches are two well-known feature selection methods. The KNN-based method is a linear algorithm to evaluate the performance of feature subsets. Its shortcoming is that it cannot keep the geometric structure of the data in the originally high-dimensional space, when representing the data in a lower-dimensional space [28]. The PCA-based method is a category of dimensionality reduction approaches in discriminating directions in a data set and finding out the sensitive directions of maximal variance [28]. It is a linear transformation from a high-dimensional space to a lower-dimensional space, and is less effective in nonlinear data with a certain type of topological manifold [28]. However, PCA cannot reflect the true low-dimensional geometry of the manifold [28]. Hence, some nonlinear algorithms are proposed to avoid the above problems. For example, Richhariya et al. [29] propose a reduced Universum twin support vector machine (SVM) for class imbalance learning. This method is applicable for large-scale imbalanced datasets. The filtering methods compute the score or significance of each attribute for the purpose of ranking the input variables. Shahee et al. [30] proposes an effective distance-based feature selection method. This method employs a sophisticated distance measure to tackle the simultaneous occurrence of between and within-class imbalance.

2.2. Data-Level Methods

The random oversampling (ROS) method is a common method to deal with the problem of class imbalance [4]. ROS randomly tries to balance class distribution by randomly replicating minority class instances, thereby increasing the minority samples and creating a more balanced dataset. However, since the samples are randomly selected, over-fitting may be encountered in some cases. The random undersampling (RUS) method is the simplest class-balancing method. While oversampling appends instances to the original dataset, random undersampling eliminates instances from the original dataset. In the majority class data S_{maj}, instances are randomly selected and removed arbitrarily to create a balanced dataset. In this way, the number of total examples in S_{maj} is reduced by E, and the class distribution balance of S is adjusted accordingly. Finally, a balanced data set $S^{'} = S + S_{maj} + S_{min} - E$ is generated. The undersampling method could be used in big data study. However, some rich and important instances of most classes may be discarded [10].

In addition, Chennuru et al. [31] propose a Simulated Annealing-based Under Sampling (SAUS) method. However, the method may remove potentially useful information. Different from undersampling, oversampling achieves rational distribution of samples by increasing the number of minority class samples of the imbalanced training set [4,32]. Chawla et al. [33] proposed the synthetic minority oversampling technique (SMOTE), which is a powerful algorithm and has enjoyed great success in various applications [4,34]. Engelmann et al. present a conditional Wasserstein Generative Adversarial Network-based oversampling method for imbalanced learning [35]. Xu et al. [36] proposed a cluster-based oversampling algorithm by combining SMOTE and k-means. Although oversampling has been proven effective for imbalanced data, most of these methods pay more attention to the binary problem and are very difficult in dealing with the multi class problem.

2.3. Algorithm-Level Methods

The methods of the algorithm level aim to modify the existing classification algorithm model appropriately according to the actual data distribution [4] Cost sensitive learning [37] and active learning [38] are representative algorithms. The core of cost-sensitive learning is to consider the classes with high misclassification costs. However, when dealing with data sets in the real world, the cost matrix is unknown in most cases, so it is difficult to accurately estimate the real error cost [39]. Although the active learning method could improve the classification performance by choosing more valuable instances and discarding those with less information, it is a large computation cost [40]. In addition, because algorithm-level solutions work directly within the training procedure of the considered classifier, they lack the flexibility offered by data-level approaches but compensate with a more direct and powerful way of reducing the bias of the certain learning algorithm [17]. They also require an in-depth understanding of how a given training procedure is conducted and what specific part of it may lead to bias towards the majority class [17].

Decomposition-based approaches transform multi-class problems into a series of binary ones. Then those binary problems are solved separately and later aggregated. The decomposition-based methods were developed from binary imbalanced classification. Those methods are often simple and allow the re-use of well-developed binary imbalanced algorithms [41]. The most popular algorithms rely on one-against-one (OAO) and one-against-all (OAA) ensemble schemes [4,10]. Most decomposition-based approaches are combined with data resampling techniques that modify the original dataset in a pre-processing step before learning the classifier. For example, Abdi et al. propose a Mahalanobis-distance oversampling (MDO) method [42]. SOUP integrates informed undersampling of majority classes and oversampling of minority ones [43]. However, those class decomposition-based schemes are not suitable when a large number of classes is considered [4]. In addition, decomposition methods are sometimes not as effective as some dedicated approaches for multi-class imbalanced data [41].

2.4. Ensemble-Based Imbalance Learning Methods

2.4.1. Adaboost

Boosting is one of the major ensemble learning methods. The term boosting refers to a family of algorithms that can convert weak learners to strong learners. Adaboost [44] is the most influential boosting algorithm, it is summarized in Algorithm 1 [45].

Algorithm 1 Adaboost

```
Input: Data set $S = (x_1, y_1), (x_2, y_2, ), \cdots, (x_m, y_m)$;
Base learning algorithm $\zeta$;
Number of learning rounds $T$;
Initialization: $D_1(x) = 1/m$.
Iterative process:
for $t = 1$ to $T$ do
    1. $h_t = \zeta(S, D_t)$;
    2. $\varepsilon_t = P_{x \sim D_t}(h_t(x) \neq y)$;
    3. if $\varepsilon_t > 0.5$ then break
    4. $\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$;
    5. Update:
       $$D_{t+1}(x) = D_t(x) \cdot \frac{e^{-\alpha_t h_t(x) y}}{Z_t} \quad (1)$$
       $Z_t$ is a normalization factor which enables $D_{t+1}$ to be a distribution
end
Output: $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$
```

Adaboost.NC is the improved version of AdaBoost. Wang and Yao compared the performances of Adaboost.NC and Adaboost combined with random oversampling with or without using class decomposition for multi-class imbalanced data sets [46]. Their results in the case of class decomposition show AdaBoost.NC and Adaboost have similar performance.

2.4.2. Bagging

The Bagging came from the abbreviation of *Bootstrap AGGregatING* [47]. The two key ingredients of Bagging are bootstrap and aggregation [48]. Algorithm 2 summarizes the Bagging procedure. A bootstrap sample is obtained by uniformly subsampling the training data with replacement. To predict a test instance, Bagging feeds the sample to its base classifiers and collects all of their outputs, and then uses the most popular strategies *voting* to aggregate the outputs and takes the winner label as the prediction [48].

Algorithm 2 Bagging

```
    Input: Data set $S = (x_1, y_1), (x_2, y_2, ), \cdots, (x_m, y_m)$;
        Base learning algorithm $\zeta$;
        Number of learning rounds $T$;
  Iterative process:
        for $t = 1$ to $T$ do
        $h_t = \zeta(S, D_{bs})$;
        end
  Output: $H(x) = sign(\sum_{t=1}^{T} h_t(x))$
```

Bagging outperforms boosting over imbalanced data [49]. Moreover, Bagging techniques are not only easy to develop but are also powerful when dealing with class imbalance if they are properly combined [50]. Most related works in the literature indicate good performance of Bagging extensions versus the other ensembles [51,52]. OverBagging [13] is a method for the management of class imbalance that merges Bagging and data preprocessing. It increases the cardinality of the minority class by replication of original examples (random oversampling), while the examples in the majority class can be all considered in each bag or can be resampled to increase the diversity. This method outperforms original Bagging in dealing with binary imbalanced data problems [50].

2.4.3. Random Forest

Illuminated by the Bagging algorithm [47], the random forest (RF) algorithm is proposed by Breiman [53]. A random forest is an ensemble of Classification and Regression Trees (CART) trees. Each tree represents a base classifier and the trainings of the base classifier are independent of each other. The core idea of RF is random sample selection and random feature selection, and the classification process is implemented by taking a majority vote. Let us suppose a dataset S_m with m instances. Firstly, n samples are randomly chosen with replacement from the original data set S_m to build a subset of samples. Secondly, f features are randomly selected from all the features of the selected instances. Then the best features are iteratively chosen based on the criterion of Gini impurity or mean squared error to construct a CART. Finally, the prediction results are obtained by repeating the above operation and employing the majority voting rule.

2.4.4. Ensemble-Based Methods

According to the different ensemble models, imbalance learning algorithms could contain boosting-based methods and Bagging/random forest-based methods [4].

Thanathamathee et al. proposed a class imbalance learning method by combining synthetic boundary data generation and boosting procedures [54]. The method outperforms KNN and AdaBoost.M1 but relies on boundary definition. Random balance boost [55] is proposed by training each classifier with a data set obtained via random balance. The random balance is designed by using both SMOTE and random undersampling to, respectively, increase or reduce the size of the classes to achieve the desired ratios. The combination could lead to better performance when compared with other state-of-the-art combined ensemble methods for binary-class imbalance problems [55]. However, most boosting-based methods face the threat of noise as the original boosting method [55]. In addition, most boosting-based imbalanced learning techniques only focus on two-class imbalance problems and are difficult to extend to multi-class imbalance problems [4].

Bagging significantly outperforms boosting over noisy and imbalanced data [49]. Moreover, Bagging techniques are not only easy to develop but also powerful when dealing with class imbalance if they are properly combined [4,50]. UnderBagging was first proposed by Barandela et al. [56]. In this method, the number of the majority class examples in each bootstrap sample is randomly reduced to the cardinality of the minority class. Simple versions of undersampling combined with Bagging are proved to work better than more complex solutions such as EasyEnsemble and BalanceCascade [51,57]. However, the performance of UnderBagging was not tested for multi-class imbalance learning in their work. OverBagging [13] is a famous class imbalance learning method that merges Bagging and data preprocessing. This method has been demonstrated to be effective in dealing with binary imbalanced data problems [50]. SMOTEBagging is the improved version of OverBagging [13]. The reported results show that SMOTEBagging can obtain better performance than OverBagging for both binary class problems [50,52].

Since the random forest is an improved version of Bagging, in theory, all class imbalance algorithms based on Bagging can be extended to the random forest. In addition, because the random forest has higher accuracy than Bagging, the imbalanced learning algorithm based on random forest should also have better performance in theory.

3. Proposed Method

This section presents the double random forest (DRF) which is a simple and effective method for high-dimensional data. DRF is motivated by the idea of combing the oversampling of the data level and ensemble random forest of the algorithm level. In other words, instead of increasing the minority instances in the data set before the classification, the proposed algorithm aims to oversample for all the ensemble models. In addition, before the class balancing operation, DRF employs the random forest to select the informative features for the imbalance classification. The overall structure of the DRF is shown in Algorithm 3. First, the random forest is adapted to identify the useful features of the data sets. Second, random oversampling, which contains all the majority instances, is operated to increase the number of the minority instances, then the bootstrap constructs several diversity training sets and generates a series of standard random forests. Third, the final results are obtained by combing all the outputs of separate random forests according to the major vote rule.

3.1. Feature Importance Determining Based on Random Forest

The random forest provides the measurement of feature importance. Let us note V as being the number of decision trees to induce. Rows of $\boldsymbol{X}$, figuring the training instances, are sampled with replacement. This is equivalent to the left multiplication of a matrix $\boldsymbol{S}_v$ of size $V \times (\#\mathcal{I})$ where each row contains only one component equal to 1 and its location in the row is evenly drawn from $\{1, \ldots, (\#\mathcal{I})\}$. The assigned labels are also transformed into $\boldsymbol{S}_v\boldsymbol{Y}$. Moreover, a random G-sized subset of the $\boldsymbol{X}$-columns is selected. This is equivalent to the right multiplication of $\boldsymbol{F}_v$ of size $(10D) \times G$ defined as the identity matrix deprived of a random set of $10D - G$ columns. The labels remain unmodified. Finally, $(\boldsymbol{S}_v\boldsymbol{X}\boldsymbol{F}_v, \boldsymbol{S}_v\boldsymbol{Y})$ is used to train a decision tree using the Classification and Regression Trees methodology (CART). The trained decision tree h_v maps any row-vector of size $\#\mathcal{I}$ into an integer $c \in \{1, \ldots, C\}$; C is the number of classes. The above tasks are repeated V times, and the predictions of V decision trees are fused according to the majority rule.

$$h(\boldsymbol{x}) = \underset{c \in \{1,\ldots,C\}}{\operatorname{argmax}} \sum_{v=1}^{V} \mathbf{1}(h_v(\boldsymbol{x}) = c) \tag{2}$$

where $\boldsymbol{x}$ is a row vector, and $\mathbf{1}$(“statement”) is equal to one when “statement” is true and zero if not.

Let us consider the v-indexed decision tree trained with $(\boldsymbol{S}_v\boldsymbol{X}\boldsymbol{F}_v, \boldsymbol{S}_v\boldsymbol{Y})$ and denote N_v the number of the internal nodes of the decision tree. If we assume a node indexed of the tree with n, then there are a set of samples and labels which are rows of $\boldsymbol{S}_v\boldsymbol{X}\boldsymbol{F}_v$ and $\boldsymbol{S}_v\boldsymbol{Y}$. We demote the set of these row indexes by $\mathcal{K}_{v,n}$ and denote the components of $\boldsymbol{S}_v\boldsymbol{X}\boldsymbol{F}_v$ and $\boldsymbol{S}_v\boldsymbol{Y}$ as $\boldsymbol{x}_k^v$ and $\boldsymbol{y}_k^v$. At this node, samples are split with the feature $f_{v,n} \in \{1, \ldots, 10D\}$ and a cutting point s_n.

$$\begin{aligned} \mathcal{K}_{v,n}^- &= \left\{k \in \mathcal{K}_{v,n} \middle| \, \boldsymbol{x}_{k,f_{v,n}}^v \leq s_{v,n}\right\} \\ \mathcal{K}_{v,n}^+ &= \left\{k \in \mathcal{K}_{v,n} \middle| \, \boldsymbol{x}_{k,f_{v,n}}^v > s_n\right\} \end{aligned} \tag{3}$$

Splitting modifies the class distribution $p_{v,n,c}$ into $p_{v,n,c}^-$ and $p_{v,n,c}^+$. These class distributions are associated, respectively, to $K_{v,n}$, $K_{v,n}^-$ and $K_{v,n}^+$.

$$\begin{aligned} p_{v,n,c} &= \tfrac{1}{\#\mathcal{K}_n} \textstyle\sum_{k \in \mathcal{K}_n} \mathbf{1}(y_k^v = c) \\ p_{v,n,c}^- &= \tfrac{1}{\#\mathcal{K}_n^-} \textstyle\sum_{k \in \mathcal{K}_n^-} \mathbf{1}(y_k^v = c) \\ p_{v,n,c}^+ &= \tfrac{1}{\#\mathcal{K}_n^+} \textstyle\sum_{k \in \mathcal{K}_n^-} \mathbf{1}(y_k^v = c) \end{aligned} \tag{4}$$

The importance value of a feature $f_{v,n}$ of the training set in node n is the amount by which splitting reduces entropy.

$$\begin{aligned} \Theta(f) = &\mathbf{1}(f = f_{v,n})(\sum_{c=1}^{C} p_{v,n,c} log_2 \frac{1}{p_{v,n,c}} \\ &- \sum_{c=1}^{C} p_{v,n,c}^- log_2 \frac{1}{p_{v,n,c}^-} - \sum_{c=1}^{C} p_{v,n,c}^+ log_2 \frac{1}{p_{v,n,c}^+}) \end{aligned} \tag{5}$$

When averaging $\Theta_{v,n}$, the relative amount of samples dealt by each node is the weight value.

$$p_{v,n} = \frac{\#\mathcal{K}_{v,n}}{\#\mathcal{K}_v} \tag{6}$$

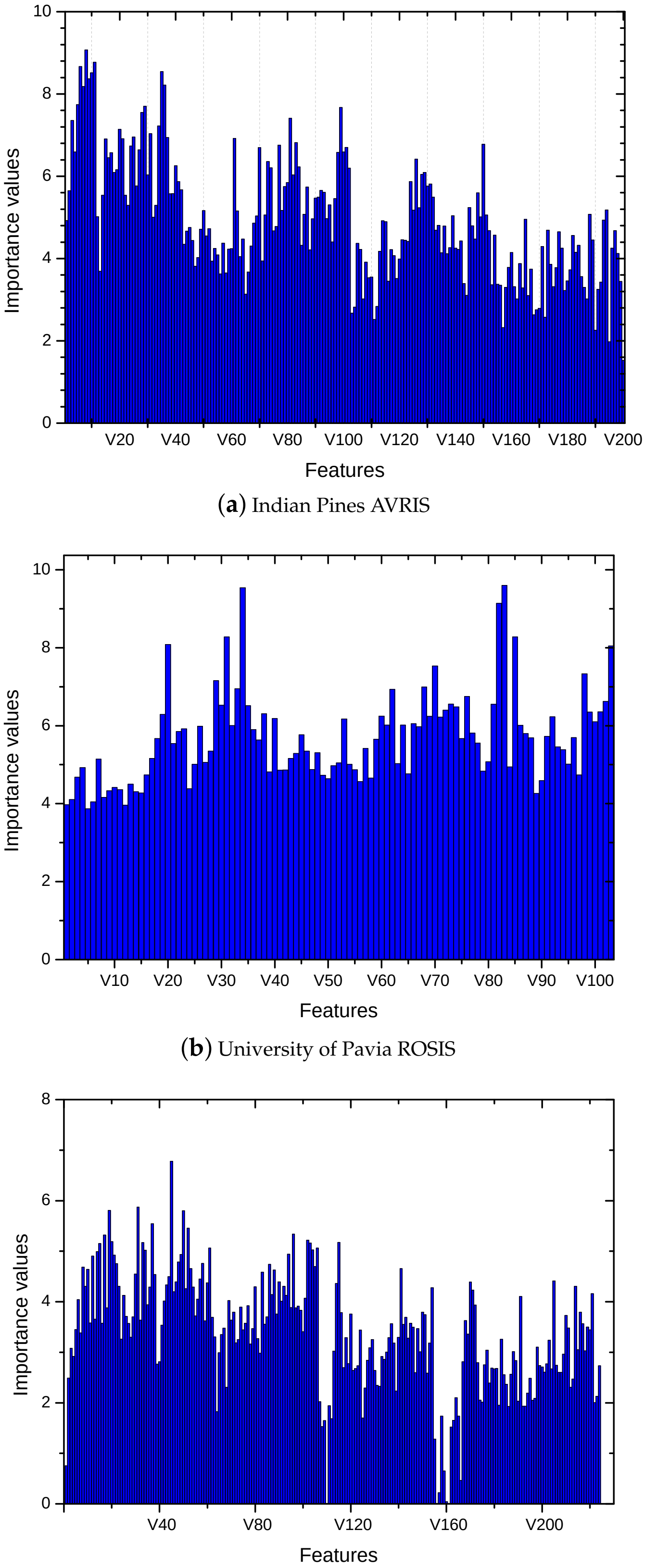
Thus, the measures of feature importance of random forest could be defined as

$$\Theta(f) = \frac{1}{\Theta} \sum_{v=1}^{V} \sum_{n=1}^{N_v} p_{v,n} \Theta_{v,n}(f) \tag{7}$$

where Θ is a normalization factor and can be defined as

$$\Theta = \max_{f \in \{1,\ldots,10D\}} \sum_{v=1}^{V} \sum_{n=1}^{N_v} p_{v,n} \Theta_{v,n}(f) \tag{8}$$

Feature selection is used to avoid the curse of dimensionality and improve the performance of the random forest in the case of the class imbalance. The feature importance analysis results of the experiments are shown in Figure 1. The figure shows that the difference of the feature importance values for each band of each data is very obvious. This phenomenon is particularly evident on *Indian Pines AVRIS* and *Salinas*. Hence, the proposed method could filter out useless features, thereby reducing the impact of noise, which in turn reduces computational complexity of the learning model and improves the final classification accuracy.



(**a**) Indian Pines AVRIS

(**b**) University of Pavia ROSIS

(**c**) Salinas

Figure 1. Feature importance values of the hyperspectral data *Indian*, *University* and *Salinas*.

Algorithm 3 Oversampling-based double random forest methods.

```
    Inputs:
1.  Training set $S = (x_1, y_1), (x_2, y_2, ), \cdots, (x_n, y_n)$;
2.  Number of classes $L$;
3.  $N_i$ is the number of training instances of $ith$ class $N_1 \leqslant N_i \leqslant N_L$ $(1 = smallest\ class, L = largest\ class)$;
4.  $\mathbb{N}$ is the number of feature which will be selected from the original feature set;
5.  Ensemble creation algorithm $\zeta$;
6.  Number of classifiers $T$;
7.  $E = \varnothing$: an ensemble.
    Process:
1.  Calculate the features importance values $\Theta$ of the data sets $S$ via a random forest model.
2.  Order the features of $S$ according to $\Theta$ in descending order.
3.  Select the first best $\mathbb{N}$ features of $S$.
4.  For $t = 1$ to $T$ do
    (a)  Keep all the $N_L$ instances of the majority class $L$.
    (b)  Get a subset $S_t$ of size $N_L$ by performing a boostrap from minority classes samples $N_c$ of the training set $S_c$.
    (c)  Construct a new balanced data set $S_t$ by combining the $N_L$ biggest class training instances with $S_t$ $(c = 1, ..., L - 1)$.
    (d)  Train a random forest classifier $h_t = \zeta(S_t)$.
    (e)  $E \leftarrow E \cup h_t$.
End
Output: The ensemble $E$
```

3.2. Over-Sampling Based Double Ensemble Methods

Let us denote $S = \{X, Y\} = \{x_i, y_i\}_{i=1}^{n}$ as training samples. The first step of the proposed random forest method uses random oversampling to balance the class of the training set. Suppose L is the number of classes, then N_i is the number of training instances of the ith class. N_L is the training size of the biggest class L, and N_1 is the training size of the smallest class 1. For each class c, the resampling operation is used to increase the number of small class instances to contract a balanced data set. All the instances of the biggest class are kept. In the second phase, the bootstrap is employed to generate several diverse balanced training subsets. Random forest is as the base classifier. A series of classifiers are trained by the diverse datasets, and the ensemble results are obtained by a majority vote rule. The description of the proposal is presented in Algorithm 3. The flowchart of the double ensemble-based multi-class imbalanced data learning method is shown in Figure 2.

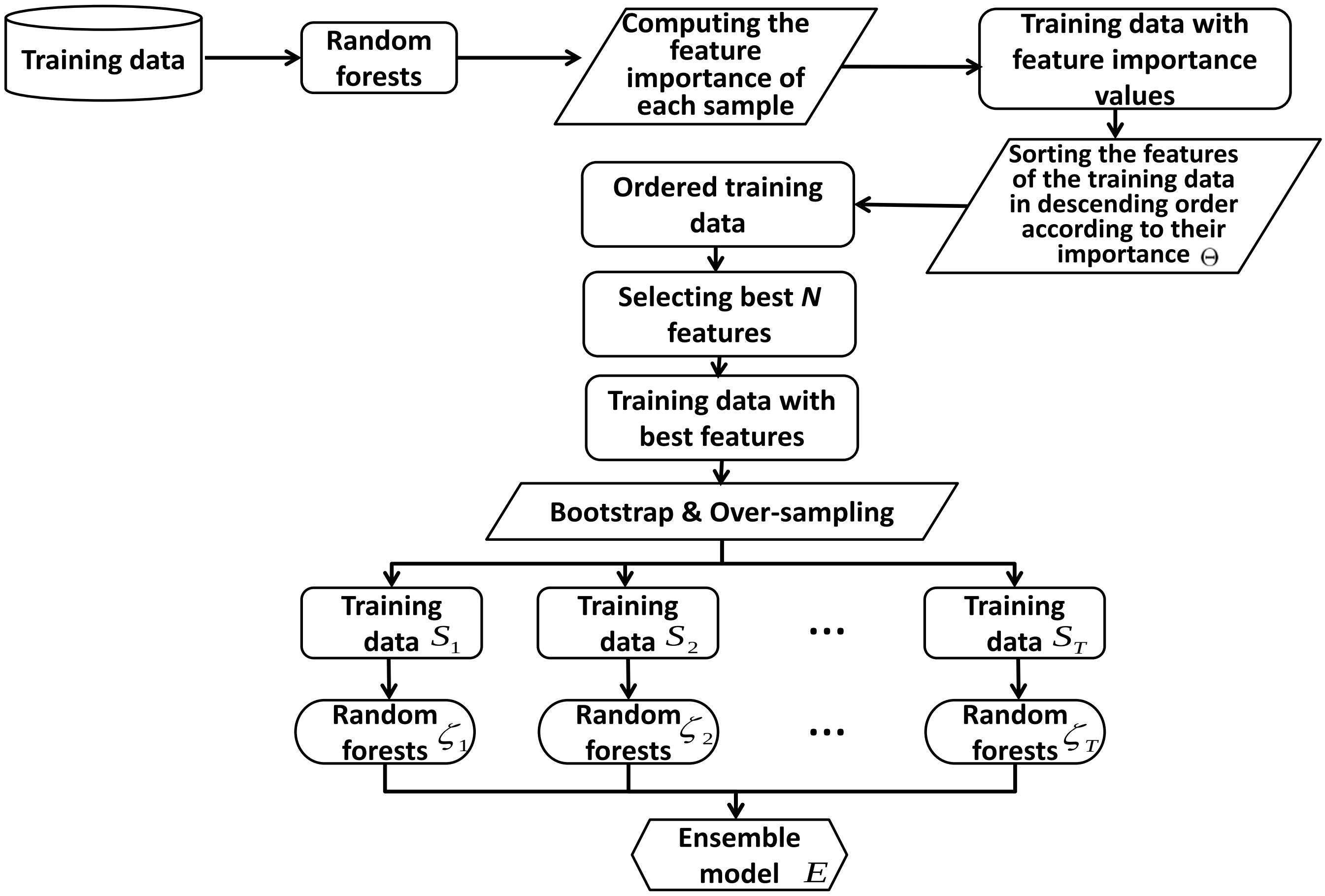

Training data
Random forests
Computing the feature importance of each sample
Training data with feature importance values
Sorting the features of the training data in descending order according to their importance Θ
Ordered training data
Selecting best N features
Training data with best features
Bootstrap & Over-sampling
Training data S1
Training data S2
Training data ST
Random forests ς1
Random forests ς2
Random forests ςT
Ensemble model E

Figure 2. Flowchart of the double ensemble-based multi-class imbalanced data learning method.

4. Experimental Study

To evaluate the performance of DRF, described in the previous section, Adaboost, Bagging, RF, data preprocessing involving random undersampling and random oversampling combined Adaboost, Bagging, and random forest are utilized in the comparative analysis. All ensembles are implemented with 100 trees. Other parameters of Adaboost, Bagging RF are kept to their default values in R-project packages, "randomForest" and "adabag". All the presented results are averaged over 10 independent runs of the algorithm.

4.1. Evaluative Performance Metrics

Accuracy is commonly used as a performance metric for measuring the performance of a classifier model. However, overall accuracy has proven not suitable in class imbalance research. Therefore, we adopt *accuracy per class*, *overall accuracy*, *average accuracy*, *F—measure* and *G—mean* as performance measures in our experiments [4].

- **Overall accuracy(OA)** measures the true prediction rate.

$$\text{OverallAccuracy} = \frac{\sum_{i=1}^{L} Recall_i}{sum(S)} \tag{9}$$

where L stands for the number of classes and $sum(S)$ is the number of the training set. Recall is per class accuracy and can be defined as (10).

$$Recall_i = \frac{n_{ii}}{\sum_{j=1}^{L} n_{ij}} \tag{10}$$

- **Average accuracy(AA)** gives the same weight to each of the classes of the problem. It can be calculated according to the following equation:

$$\text{AverageAccuracy} = \frac{\sum_{i=1}^{L} Recall_i}{L} \tag{11}$$

- **F-Measure** is one of the most popular methods to evaluate the performance of a classifier for imbalance data. It can be calculated according to the following equation.

$$\text{F-measure} = \frac{2}{L}\frac{\sum_{i=1}^{L} Recall_i \sum_{i=1}^{L} Precision_i}{\sum_{i=1}^{L} Recall_i + \sum_{i=1}^{L} Precision_i} \tag{12}$$

where $Precision_i$ can be computed by $\frac{n_{ii}}{\sum_{j=1}^{L} n_{ji}}$.

- **G-mean** is another method to evaluate the performance of a classifier for imbalance data.

$$\text{G-mean} \quad = \prod_{i=1}^{L} Recall_i^{1/L} \tag{13}$$

4.2. Data Information

The proposed DRF is evaluated on three standard hyperspectral images, Indian Pines AVRIS, *University of Pavia ROSIS*, and *Salinas*.

(1) *Indian Pines AVRIS* is highly imbalanced and composed of 145 × 145 pixels, with a spatial resolution of 20 m/pixel and 200 spectral bands. The reference data with 16 classes are composed of 10,249 samples.

(2) *University of Pavia ROSIS* consists of 610 × 340 pixels, and 103 spectral bands. The spatial resolution of this data is 1.3 m/pixel. The reference data is with 9 classes and is composed of 42,776 instances.

(3) *Salinas* consists of 512 × 217 pixels with a spatial resolution of 3.7 m/pixel. The data has 224 spectral bands. The reference data is with 16 classes and is composed of 54,129 instances.

The reference data of all images are presented in Figures 3–5.

The data sampling is used to divide the reference data set into two non-overlapping parts: the training set and the test set. A total of 5% of their instances are selected randomly from original reference data to construct training sets. All the unselected instances are test sets. More details of the data information are presented in Table 1.

Table 1. Data information.

	Indian Pines AVRIS			University of Pavia ROSIS			Salinas		
		Train.	**Test**		**Train.**	**Test**		**Train.**	**Test**
1	Alfalfa	23	23	Asphalt	331	6300	Brocoli_green_weeds_1	100	1909
2	Corn-notill	428	1000	Meadows	932	17,717	Brocoli_green_weeds_2	186	3540
3	Corn-mintill	249	581	Gravel	104	1995	Fallow	98	1878
4	Corn	71	166	Trees	153	2911	Fallow_rough_plow	69	1325
5	Grass-pasture	144	339	Painted metal sheets	67	1278	Fallow_smooth	133	2545
6	Grass-trees	219	511	Bare Soil	251	4778	Stubble	197	3762
7	Grass-pasture-mowed	14	14	Bitumen	66	1264	Celery	178	3401
8	Hay-windrowed	143	335	Self-Blocking Bricks	184	3498	Grapes_untrained	563	10,708
9	Oats	10	10	Shadows	47	900	Soil_vinyard_develop	310	5893
10	Soybean-notill	291	681				Corn_senesced green_weeds	163	3115
11	Soybean-mintill	736	1719				Lettuce_romaine_4wk	53	1015
12	Soybean-clean	177	416				Lettuce_romaine_5wk	96	1831
13	Wheat	61	144				Lettuce_romaine_6wk	45	871
14	Woods	379	886				Lettuce_romaine_7wk	53	1017
15	Buildings-Grass Trees-Drives	115	271				Vinyard untrained	363	6905
16	Stone-Steel-Towers	46	47				Vinyard_vertical_trellis	90	1717
Total		3106	7143		2135	40,641		2697	51,432

4.3. Results and Analysis

This section shows the performance of the proposed DRF method. The objective of this section is as follows:

1 Present the performance of DRF in dealing with the hyperspectral datasets.
2 Compare the performance of DRF with data sampling methods.
3 Analyze the parameter sensitivity of the proposed DRF methods.

Tables 2–4 present the AA, OA, F-measure, and Gmean results of the Adaboost, Bagging, traditional RF, two data sampling combined Adaboost, Bagging, and RF methods, as well as the proposed DRF on the hyperspectral images *Indian Pines AVRIS*, *University of Pavia ROSIS* and *Salinas*, respectively. The best results in all tables are highlighted in bold font. The experimental results in those tables show that Adaboost could obtain better results than the Bagging and the traditional RF. Both Bagging and RF biases in the classification of majority classes. In addition, resampling methods are useful to improve the performances of the ensemble models, especially Adaboost and Bagging, for the hyperspectral remote sensing data classification. Oversampling tends to outperform the undersampling reference strategies, which manifested in the average ranks concerning all of the performance metrics for oversampling-based methods. The undersampling can discard potentially useful data. Furthermore, The proposed DRF algorithm achieved a statistically significantly better performance than all the reference methods. In other means, DRF could obtain better results in dealing with high-dimensional imbalanced data than simple data sampling methods. The feature importance is helpful to improve the performance of the ensemble-based imbalance learning method. On the data set *University of Pavia ROSIS*, with respect to Adaboost, Bagging, RF, RUS-RF and ROS-RF, the best increases in AA are over **6%**, **24%**, **7%**, **10%**, **6%**, and the best increases in OA are over **4%**, **13%** **6%**, **22%**, **4%**. Moreover, the results of the F-measure and Gmean of three reference data also demonstrate the better performance of DRF when compared with other methods.

Table 2. Classification results (%) of the *Indian* image, respectively, obtained by Adaboost, Bagging, RF, RUS + Adaboost, ROS + Adaboost, RUS + Bagging, ROS + Bagging, RUS-RF, ROS-RF, and the proposed DRF.

	Adaboost	**Bagging**	**RF**	**RUS Adaboost**	**ROS Adaboost**	**RUS Bagging**	**ROS Bagging**	**RUS RF**	**ROS RF**	**DRF**
1	0	0	69.57	63.48	58.26	76.52	69.57	73.04	53.91	**76.81**
2	48.42	43.36	75.68	25.98	31.7	30.04	4.24	34.56	75.84	**82.67**
3	30.81	27.57	60.9	32.05	38	32.84	25.78	35.32	66.85	**71.2**
4	0	0	56.87	33.37	48.55	24.58	46.02	38.8	65.9	**86.75**
5	76.52	70.44	90.91	71.86	81.12	70.62	70.86	77.23	92.8	**95.38**
6	97.77	98	96.59	68.49	89.16	71.94	79.65	76.16	96.28	**98.17**
7	0	0	55.71	72.86	68.57	**81.43**	74.29	80	65.71	78.57
8	94.39	94.39	98.57	61.73	78.27	57.97	70.63	72.24	97.55	**100**
9	0	0	52	52	**60**	50	50	**60**	46	56.67
10	29.72	23.79	79.41	36.53	60.68	33.86	23.55	49.84	85.52	**89.38**
11	83.22	83.08	**89.85**	38.55	58.06	48.11	75.64	41.47	82.13	88.37
12	25.82	26.88	68.7	20.96	48.99	16.39	38.12	34.23	76.39	**93.35**
13	92.08	92.22	90.14	91.67	90.83	93.61	90.28	92.5	89.03	**95.14**
14	92.19	92.82	0	80.16	83.93	73.07	81.47	71.74	95.58	**97.07**
15	17.79	12.77	53.65	22.51	35.06	23.1	34.1	29.89	57.42	**75.4**
16	58.72	50.64	91.49	94.04	94.89	92.34	90.64	93.62	96.6	**100**
AA	46.72	44.75	76.65	54.14	64.13	54.78	57.8	60.04	77.72	**86.56**
OA	63.05	61.11	82.72	45.89	59.77	47.34	53.06	50.83	82.65	**88.8**
F-measure	48.47	47.05	81.37	47.09	59	48.44	52.8	53.4	79.32	**88.47**
G-mean	0	0	74.75	46.47	60.69	46.47	10.4	55.27	75.78	**85.63**

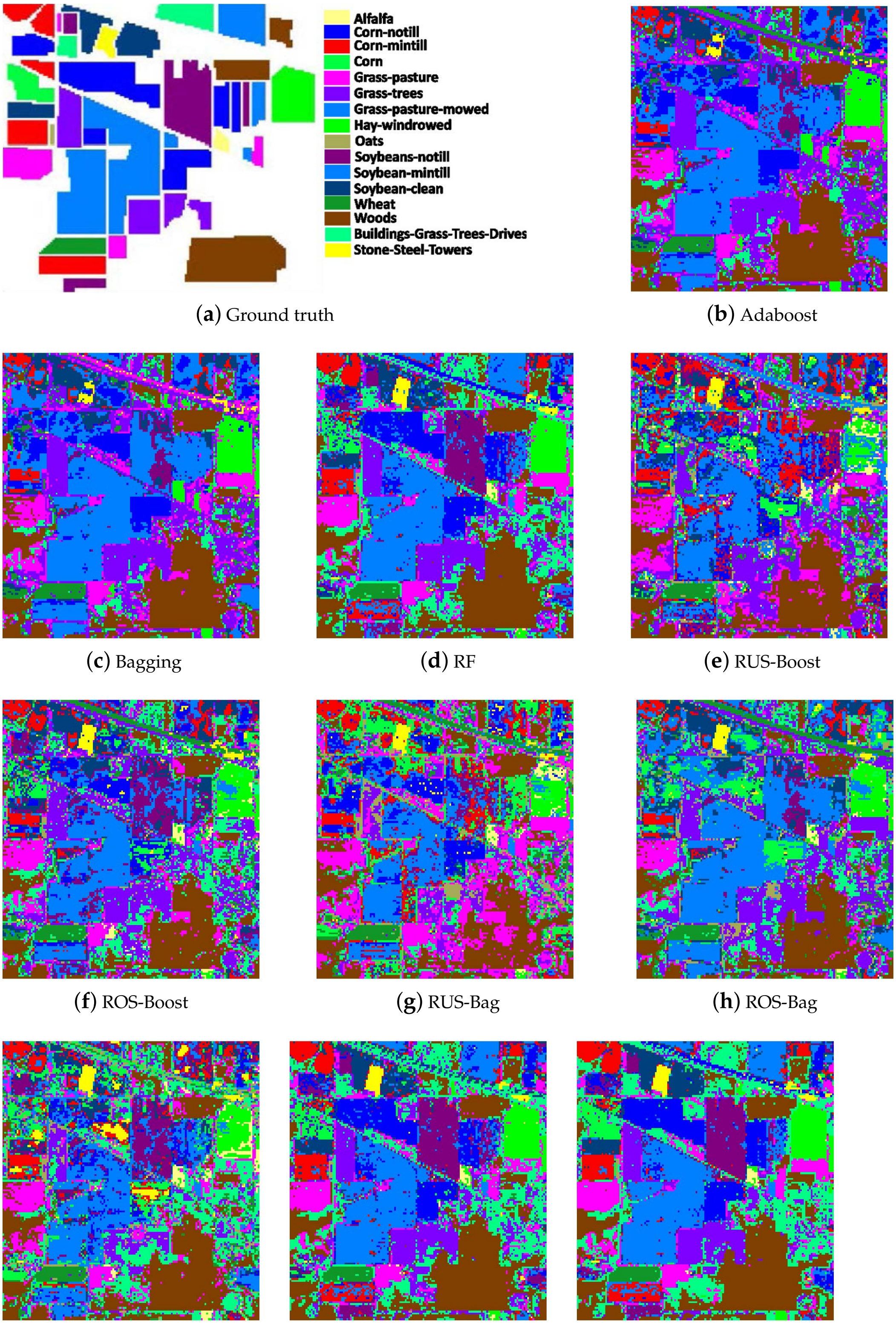
Table 3. Classification results (%) of the *University of Pavia ROSIS* image, respectively, obtained by Adaboost, Bagging, RF, RUS + Adaboost, ROS + Adaboost, RUS + Bagging, ROS + Bagging, RUS-RF, ROS-RF, and the proposed DRF.

	Adaboost	**Bagging**	**RF**	**RUS**	**ROS**	**RUS**	**ROS**	**RUS**	**ROS**	**DRF**
				Adaboost	**Adaboost**	**Bagging**	**Bagging**	**RF**	**RF**	
1	90.27	93.55	92.17	72.44	82.5	66.47	61.09	72.29	89.71	**95.11**
2	96.1	**97.29**	97.09	67.04	82.76	56.64	30.4	61.21	92.1	95.82
3	59.93	20.82	58.64	56.92	63.77	57.26	68.6	70.24	61.83	**75.51**
4	86.42	81.31	86.83	92.09	92.97	87	**96.95**	95.59	91.41	94.53
5	98.37	95.62	98.62	98.72	98.72	98.72	98.44	99.09	99.01	**99.79**
6	71.76	28.66	58.07	68.61	80.56	65.39	88.89	72.36	72.8	**90.25**
7	75.74	0	78.54	**88.67**	86.65	83.61	86.61	87.37	78.89	81.86
8	86.52	**89.08**	85.4	80.22	83.76	73.58	69.19	73.7	85.23	88.16
9	97.62	98.89	99.07	97.27	98.91	99.16	99.56	**100**	**100**	**100**
AA	84.75	67.25	83.82	80.22	85.62	76.42	77.75	81.32	85.66	**91.22**
OA	88.51	79.99	87.63	72.83	83.33	65.96	57.43	71.08	87.32	**93.09**
F-measure	86.2	70.68	86.09	76.29	83.27	71.96	72.96	77.06	85.84	**91.83**
G-mean	83.75	0	82.3	78.88	84.98	74.57	73.68	80.07	84.78	**90.88**

Table 4. Classification results (%) of the *Salinas* image, respectively, obtained by Adaboost, Bagging, RF, RUS + Adaboost, ROS + Adaboost, RUS + Bagging, ROS + Bagging, RUS-RF, ROS-RF, and the proposed DRF.

	Adaboost	**Bagging**	**RF**	**RUS**	**ROS**	**RUS**	**ROS**	**RUS**	**ROS**	**DRF**
				Adaboost	**Adaboost**	**Bagging**	**Bagging**	**RF**	**RF**	
1	99.47	97.48	99.46	98.16	99.58	98.02	97.62	98.96	99.47	**99.76**
2	99.77	98.61	99.79	98.52	99.6	98.31	98.43	97.97	99.84	**99.84**
3	95.08	87.24	95.27	91.95	95.86	85.31	86.42	95.41	97.8	**97.8**
4	97.89	91.56	99.58	95.55	98.61	97.87	99.7	99.14	99.68	**99.7**
5	96.82	94.92	96.86	96.01	96.44	94.92	96.2	96.79	97.38	**98.72**
6	99.82	99.14	99.71	99.69	**99.84**	99.18	99.15	99.43	99.78	99.82
7	99.66	98.94	99.22	98.48	99.65	97.55	98.89	97.92	99.43	**99.72**
8	84.52	77.4	84.49	65.94	78.87	54.19	58.83	65.09	76.36	**91.57**
9	99.21	97.74	99.06	99.02	99.05	97.95	97.25	98.68	99.15	**99.79**
10	89.97	72.35	89.66	85.92	89.84	77.68	77.93	85.16	90.56	**94.85**
11	90.29	86.01	91.9	90.07	91.05	89.6	88.45	89.93	89.64	**95.17**
12	98.44	94.77	98.86	97.33	98.5	95.75	96.69	98.03	98.98	**100**
13	95.59	95.09	96.12	**96.46**	95.2	95.2	94.81	95.27	94.58	95.94
14	97.17	94.69	96.83	96.18	96.79	95.38	94.75	96.62	97.09	**97.97**
15	67.93	53.96	61.96	59.86	**71.09**	68.34	65.47	62.96	68.09	64.63
16	97.48	95.17	97.51	95.55	97.83	96.26	95.57	95.76	97.69	**99.17**
AA	94.32	89.69	94.14	91.54	94.24	90.09	90.39	92.07	94.1	**95.9**
OA	90.85	85.11	90.07	85.08	90.11	82.72	83.43	85.44	89.36	**92.72**
F-measure	94.22	89.24	94	90.25	93.79	88.51	88.84	90.66	93.62	**95.93**
G-mean	93.92	88.72	93.58	90.66	93.86	89	89.42	91.22	93.6	**95.44**

For a more intuitive comparison of the proposed DRF algorithm and the other nine reference methods, the categorization map is adopted in our experiment. Figures 3–5, respectively, exhibit the classification maps obtained by ten classification methods for *Indian Pines AVRIS*, *University of Pavia ROSIS*, and *Salinas* images. Those figures show that DRF could result in more accurate cartography with respect to Adaboost, Bagging, RF, and other improved ensemble methods. According to the map results, we find that the proposed algorithm can improve the classification accuracy of small class samples while maintaining the overall accuracy. This effect is especially pronounced in *Indian* data. This conclusion is consistent with the descriptions for the above tables.



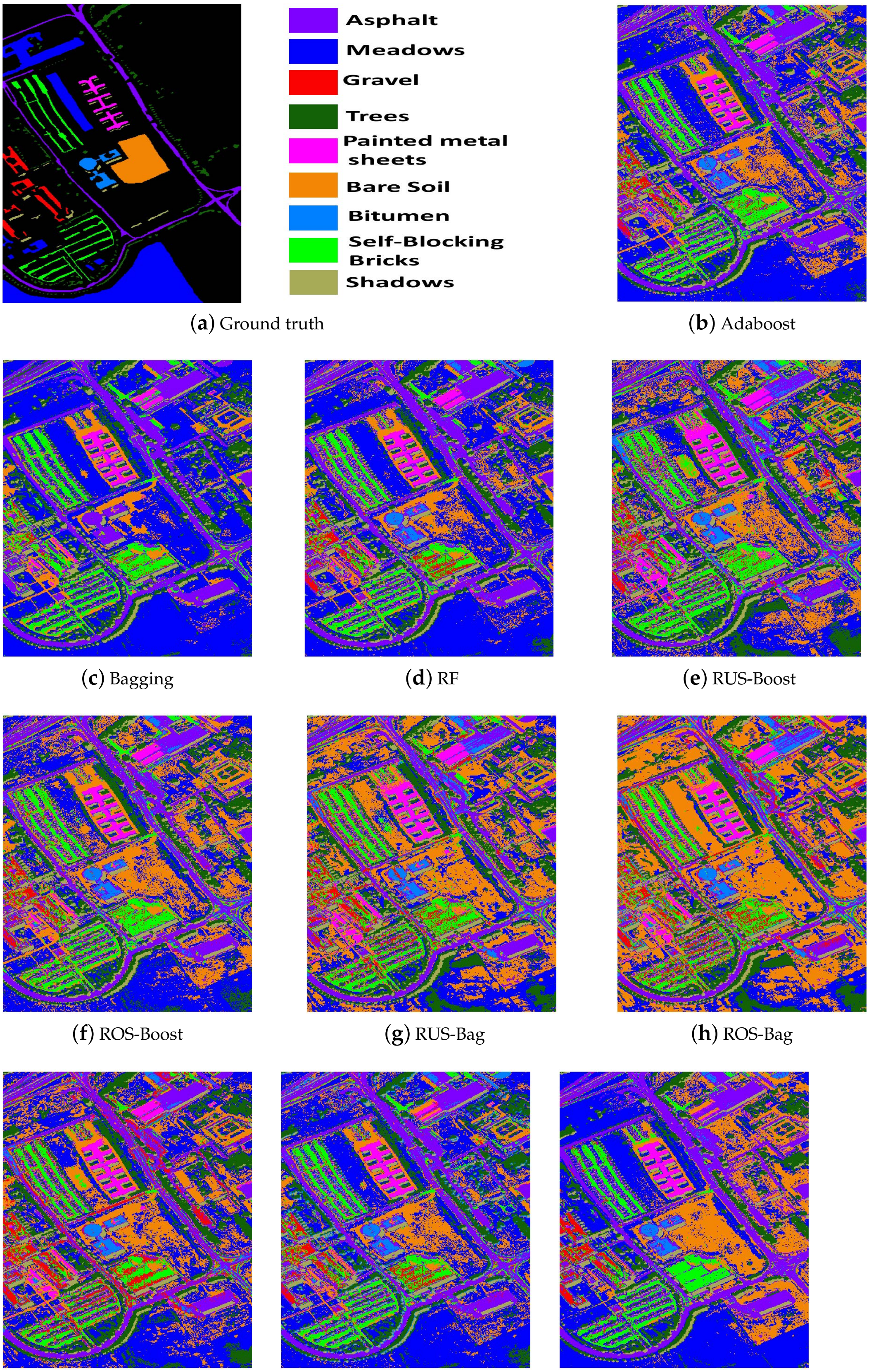
(**a**) Ground truth (**b**) Adaboost

(**c**) Bagging (**d**) RF (**e**) RUS-Boost

(**f**) ROS-Boost (**g**) RUS-Bag (**h**) ROS-Bag

(**i**) RUS-RF (**j**) ROS-RF (**k**) DRF

Figure 3. Ground truth, classification maps of Adaboost, Bagging, random forest, random undersampling combined, random oversampling combined Adaboost, Bagging, and random forest, and the proposed double random forest DRF, on the hyperspectral data *Salinas*.



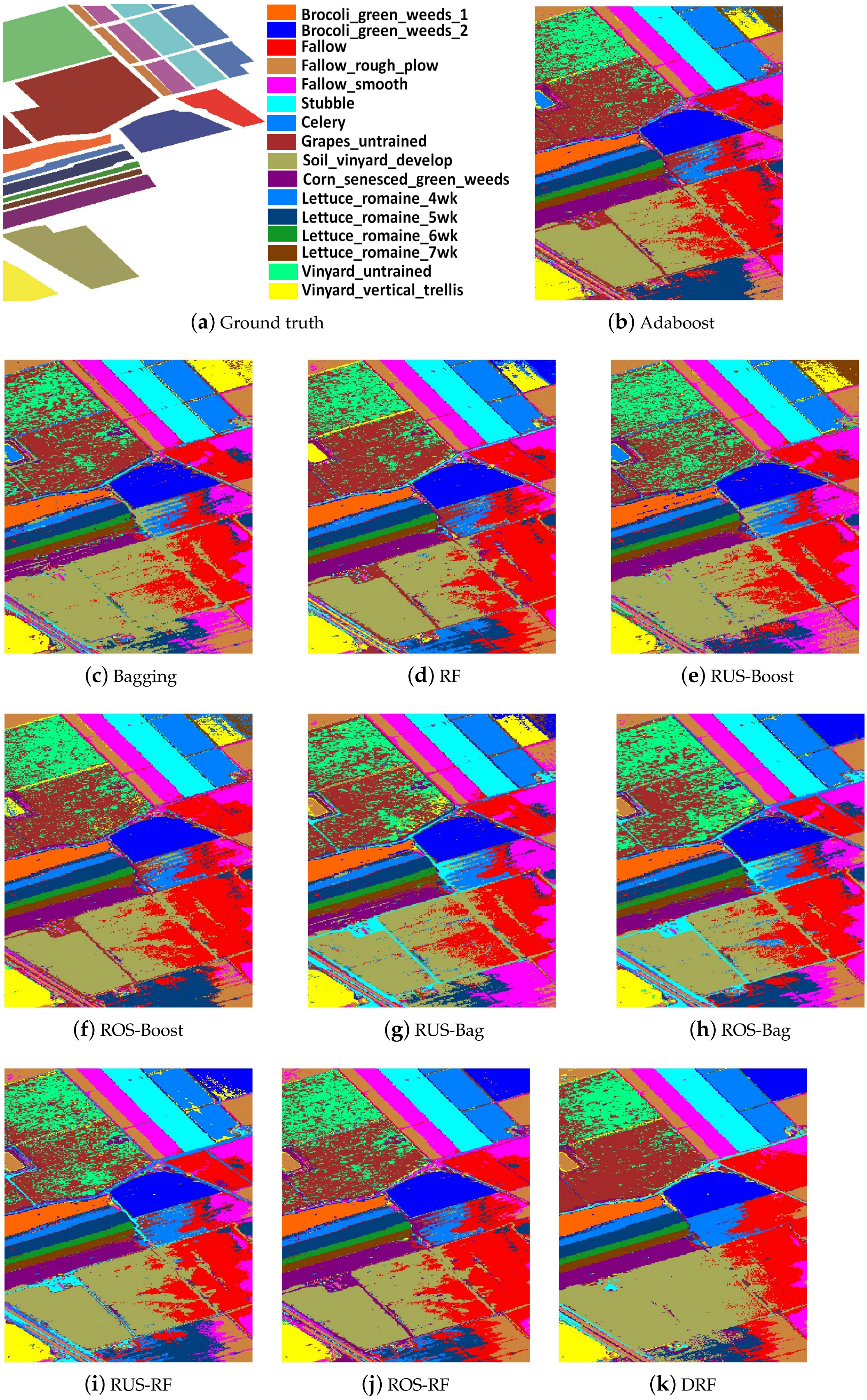
(**a**) Ground truth (**b**) Adaboost

(**c**) Bagging (**d**) RF (**e**) RUS-Boost

(**f**) ROS-Boost (**g**) RUS-Bag (**h**) ROS-Bag

(**i**) RUS-RF (**j**) ROS-RF (**k**) DRF

Figure 4. Ground truth, classification maps of Adaboost, Bagging, random forest, random undersampling combined, random oversampling combined Adaboost, Bagging, and random forest, and the proposed double random forest DRF, on the hyperspectral data *University*.



(**a**) Ground truth

(**b**) Adaboost

(**c**) Bagging

(**d**) RF

(**e**) RUS-Boost

(**f**) ROS-Boost

(**g**) RUS-Bag

(**h**) ROS-Bag

(**i**) RUS-RF

(**j**) ROS-RF

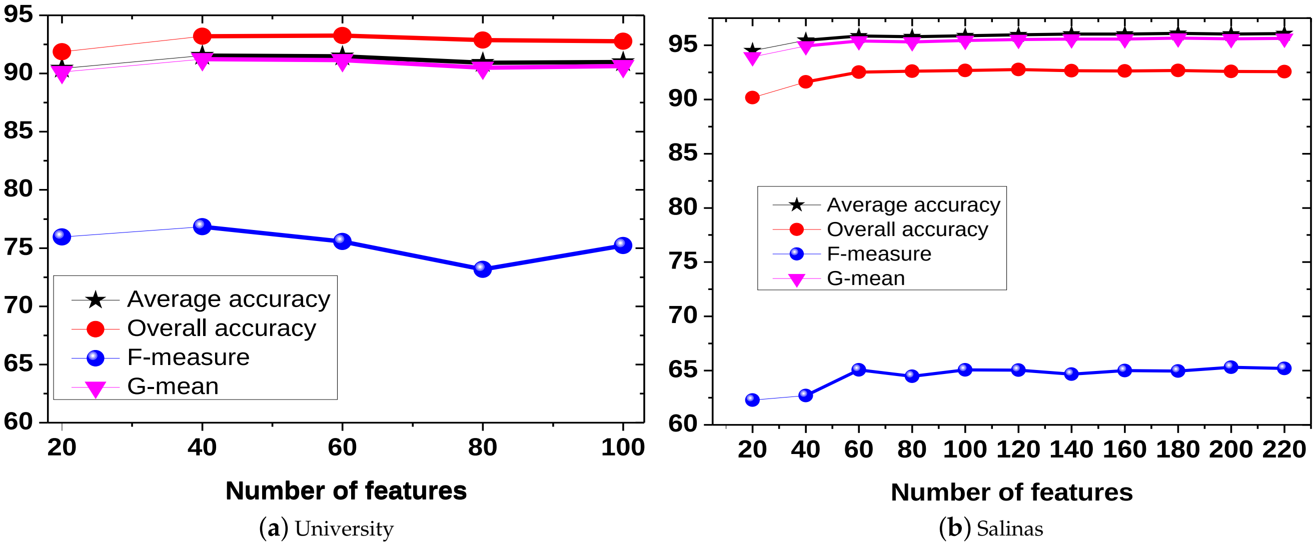
(**k**) DRF

Figure 5. Ground truth, classification maps of Adaboost, Bagging, random forest, random undersampling combined, random oversampling combined Adaboost, Bagging, and random forest, and the proposed double random forest DRF, on the hyperspectral data *Salinas*.

The computational complexity of our proposal arises mainly from two sources: the feature importance calculation and the building of the double ensemble model. The computational cost of building a decision tree is $O(N \times \log(N) \times D)$ where N is the number of the training set and D is the number of the features. The complexity of random forests with T base classifiers is $O(N \times \log(N) \times D \times T)$. In the feature importance calculation process, only one random forest is built. Hence, the complexity of this step is $O(N \times \log(N) \times D \times T)$. In double ensemble model building, because the individual classifiers are trained simultaneously, the computational complexity will be similar to that of a random forest. However, only $\mathbb{N}(\mathbb{N} < N)$ features are used. Then the computational complexity of the second step is $O(N \times \log(N) \times \mathbb{N} \times T^2)$. In our proposal, because all of the positive patterns are used for training in each balanced individual training set, N is approximate twice the size of the positive samples. Therefore, the overall computational complexity of our proposal is $O(N \times \log(N) \times \mathbb{N} \times T \times (T+1) + N)$. Although the training complexity of the framework is a little higher than the random forest, it is normal in dealing with the imbalanced data. In addition, because the produced training set is with low dimension, the proposed method has lower computational complexity than the reference imbalance learning method.

4.4. Parameter Analysis

In order to study the influence of feature size, we present in Figure 1b,c the evaluation of the average accuracy, overall accuracy, G-mean, and F-measure for the proposed method on the *University of Pavia ROSIS* and *Salinas*. In this experiment, the size T of the ensemble is still set to 100 and the tested number of feature set is set as (20, 40, 60, 80, 100) and (20, 40, 60, 80, 100, 120, 140, 160, 180, 200, 220). Figure 6 shows that DRF results in the best result when the feature size is 40 for *University of Pavia ROSIS* and 60 for *Salinas*. In other words, the proposed DRF could show superior performance with a small feature set, thus reducing the computation complexity obviously.



(**a**) University (**b**) Salinas

Figure 6. Performances of the proposed method measured by OA, AA, F—*measure* and G—*mean* with respect to the number of features on hyperspectral data sets *University* and *Salinas.*

5. Discussion

1 The proposed algorithm is effective for multi-class imbalanced hyperspectral remote sensing data. It could increase the robustness of the ensemble method to skewed distributions. Out of the examined methods, the proposed ensemble method, when combined with different information about instance-level difficulties, offers the best performance regardless of the used metric. The standard version of random forest is considered unsuitable for class imbalanced hyperspectral remote sensing data, as it underperformed when compared with random resampling methods.

2 By analyzing the behaviors of single random undersampling or oversampling, we are able to identify the weak spot of the data level methods. As data in each base classifier is selected randomly, resampling performs locally correct oversampling that does not translate to the global characteristics of data. This could lead to increased overlapping among classes. Our proposed method used both feature selection and data sampling. This eliminated the drawbacks of single data resampling in the high dimensional data processing.

3 The analysis of instance-level difficulties in multi-class imbalanced data allowed for a better understanding of each considered classification problem [17]. This paper adopts two popular data resampling methods and focuses on enhancing the presence of the most difficult instances. Although this could lead to improvements for the traditional random forest algorithm, the operation is unsuitable for hyperspectral remote sensing data.

4 In our experiments, we analyzed the effect of feature parameters with the double ensemble algorithm. The results show that the proposed method is insensitive to the number of features. That means the proposed method has a good generalization ability.

6. Conclusions

In this paper, we proposed a novel double ensemble method to deal with multi-class imbalanced datasets. The proposed DRF method consists of two key steps: the feature importance measure of a balanced training dataset, and oversampling-based ensemble random forest model. The proposed DRF method aims to improve the classification performance in the case of high-dimensional multi-class imbalance problems. Experiments are conducted using two popular multi-class imbalanced hyperspectral datasets. The performance obtained by the proposed method is compared with five state-of-the-art techniques Experimental results demonstrated that the proposed DRF outperforms the other techniques in the multi-class imbalanced datasets and could decrease the computation complexity. The proposed method provides a direction for the big class imbalanced data learning.

Compared with random forest, the algorithm in this paper has slightly higher computational complexity. Therefore, in future work, we plan to optimize the structure of the algorithm to reduce the training time of the model.

Author Contributions: D.Q. and W.F. conceived and designed the experiments; W.F. performed the experiments. X.W. and M.X. edited the manuscript. G.D. and W.H. were responsible for validating the experimental conclusions and reviewing the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Zhejiang Province (LQ20F020021).

Acknowledgments: This work is supported by Zhejiang Provincial Natural Science Foundation (LQ20F020021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cao, X.; Yao, J.; Xu, Z.; Meng, D. Hyperspectral Image Classification With Convolutional Neural Network and Active Learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4604–4616. [CrossRef]
2. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J. Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
3. Yang, J.; Wu, C.; Du, B.; Zhang, L. Enhanced Multiscale Feature Fusion Network for HSI Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 10328–10347. [CrossRef]
4. Feng, W.; Huang, W.; Ren, J. Class Imbalance Ensemble Learning Based on the Margin Theory. *Appl. Sci.* **2018**, *8*, 815. [CrossRef]
5. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 279–317. [CrossRef]
6. Tao, C.; Pan, H.; Li, Y.; Zou, Z. Unsupervised Spectral-patial Feature Learning With Stacked Sparse Autoencoder for Hyperspectral Imagery Classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2438–2442. [CrossRef]

7. He, Z.; Liu, H.; Wang, Y.; Hu, J. Generative Adversarial Networks-Based Semi-Supervised Learning for Hyperspectral Image Classification. *Remote Sens.* **2017**, *9*, 1042. [CrossRef]
8. Garcia, S.; Zhang, Z.; Altalhi, A.; Alshomrani, S.; Herrera, F. Dynamic ensemble selection for multi-class imbalanced datasets. *Inf. Sci.* **2018**, *445-446*, 22–37. [CrossRef]
9. Sun, T.; Jiao, L.; Feng, J.; Liu, F.; Zhang, X. Imbalanced Hyperspectral Image Classification Based on Maximum Margin. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 522–526. [CrossRef]
10. Feng, W.; Huang, W.; Bao, W. Imbalanced Hyperspectral Image Classification With an Adaptive Ensemble Method Based on SMOTE and Rotation Forest With Differentiated Sampling Rates. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1879–1883. [CrossRef]
11. Zhu, J.; Fang, L.; Ghamisi, P. Deformable Convolutional Neural Networks for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1254–1258. [CrossRef]
12. Roy, S.K.; Haut, J.M.; Paoletti, M.E.; Dubey, S.R.; Plaza, A. Generative Adversarial Minority Oversampling for Spectral-patial Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1695–1704. [CrossRef]
13. Wang, S.; Yao, X. Diversity analysis on imbalanced data sets by using ensemble models. In Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, Nashville, TN, USA, 30 March–2 April 2009; pp. 324–331.
14. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. In *Progress in Artificial Intelligence*; U.S. Department of Energy: Washington, DC, USA, 2016; pp. 221–232.
15. Saez, J.A.; Krawczyk, B.; Wozniak, M. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognit.* **2016**, *57*, 164–178. [CrossRef]
16. Bi, J.; Zhang, C. An empirical comparison on state-of-the-art multi-class imbalance learning algorithms and a new diversified ensemble learning scheme. *Knowl. Based Syst.* **2018**, *158*, 81–93. [CrossRef]
17. William, I.V.; Krawczyk, B. Multi-class imbalanced big data classification on Spark. *Knowl. Based Syst.* **2020**, *212*, 106598.
18. Dietterich, T. Ensemble Methods in Machine Learning. In Proceedings of the 1st International Workshop on Multiple Classifier Systems, Cagliari, Italy, 21–23 June 2000; pp. 1–15.
19. Feng, W.; Quan, Y.; Dauphin, G.; Li, Q.; Gao, L.; Huang, W.; Xia, J.; Zhu, W.; Xing, M. Semi-supervised rotation forest based on ensemble margin theory for the classification of hyperspectral image with limited training data. *Inf. Sci.* **2021**, *575*, 611–638 [CrossRef]
20. Feng, W.; Quan, Y.; Dauphin, G. Label Noise Cleaning with an Adaptive Ensemble Method Based on Noise Detection Metric. *Sensors* **2020**, *20*, 6718. [CrossRef]
21. Quan, Y.; Zhong, X.; Feng, W.; Chan, C.W.; Xing, M. SMOTE-Based Weighted Deep Rotation Forest for the Imbalanced Hyperspectral Data Classification. *Remote Sens.* **2021**, *13*, 464. [CrossRef]
22. Ribeiro, V.; Reynoso-Meza, G. Ensemble learning by means of a multi-objective optimization design approach for dealing with imbalanced data sets. *Expert Syst. Appl.* **2020**, *147*, 113232. [CrossRef]
23. Wang, Z.; Chen, Z.; Zhu, Y.; Zhang, J.; Li, D. Multi-matrices entropy discriminant ensemble learning for imbalanced problem. *Neural Comput. Appl.* **2020**, *32*, 8245–8264. [CrossRef]
24. Chen, Z.; Duan, J.; Kang, L.; Qiu, G. A Hybrid Data-Level Ensemble to Enable Learning from Highly Imbalanced Dataset. *Inf. Sci.* **2020**, *554*, 157–176. [CrossRef]
25. Wang, Z.; Jia, P.; Xu, X.; Wang, B.; Li, D. Sample and feature selecting based ensemble learning for imbalanced problems. *Appl. Soft Comput.* **2021**, *113*, 107884. [CrossRef]
26. Qin, W.; Zhuang, Z.L.; Guo, L.; Sun, Y. A hybrid multi-class imbalanced learning method for predicting the quality level of diesel engines. *J. Manuf. Syst.* **2021**, 62, 846–856. [CrossRef]
27. Cmv, A.; Jie, D.B. Accurate and efficient sequential ensemble learning for highly imbalanced multi-class data—ScienceDirect. *Neural Netw.* **2020**, *128*, 268–278.
28. Chao, L.; Jiang, D.; Yang, W. Global geometric similarity scheme for feature selection in fault diagnosis. *Expert Syst. Appl.* **2014**, *41*, 3585–3595.
29. Richhariya, B.; Tanveer, M. A reduced universum twin support vector machine for class imbalance learning. *Pattern Recognit.* **2020**, *102*, 107150. [CrossRef]
30. Shahee, S.A.; Ananthakumar, U. An effective distance based feature selection approach for imbalanced data. *Appl. Intell.* **2020**, *50*, 717–745. [CrossRef]
31. Chennuru, V.K.; Timmappareddy, S.R. Simulated annealing based undersampling (SAUS): A hybrid multi-objective optimization method to tackle class imbalance. *Appl. Intell.* **2021**, *52*, 2092–2110. [CrossRef]
32. Lv, Q.; Feng, W.; Quan, Y.; Dauphin, G.; Gao, L.; Xing, M. Enhanced-Random-Feature-Subspace-Based Ensemble CNN for the Imbalanced Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3988–3999. [CrossRef]
33. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority Oversampling Technique. *J. Artif. Intell. Res.* **2004**, *16*, 321–357. [CrossRef]
34. Feng, W.; Boukir, S.; Huang, W. Margin-Based Random Forest for Imbalanced Land Cover Classification. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019.
35. Engelmann, J.; Lessmann, S. Conditional Wasserstein GAN-based Oversampling of Tabular Data for Imbalanced Learning. *Expert Syst. Appl.* **2021**, *174*, 114582. [CrossRef]

36. Xu, Z.; Shen, D.; Nie, T.; Kou, Y.; Han, X. An oversampling algorithm combining SMOTE and k-means for imbalanced medical data. *Inf. Sci.* **2021**, *572*, 574–589. [CrossRef]
37. Sun, Y.; Kamel, M.S.; Wong, A.K.; Wang, Y. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit.* **2007**, *40*, 3358–3378. [CrossRef]
38. Ertekin, S.; Huang, J.; Bottou, L.; Giles, C.L. Learning on the border: Active learning in imbalanced data classification. In *Proceedings of the CIKM (Conference on Information and Knowledge Management)*; Silva, M.J., Laender, A.H.F., Baeza-Yates, R.A., McGuinness, D.L., Olstad, B., Olsen, H., Falcao, A.O., Eds.; ACM: New York, NY, USA, 2007; pp. 127–136.
39. Feng, W.; Dauphin, G.; Huang, W.; Quan, Y.; Li, Q. Dynamic Synthetic Minority Over-Sampling Technique-Based Rotation Forest for the Classification of Imbalanced Hyperspectral Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *PP*, 2159–2169. [CrossRef]
40. Liu, W.; Zhang, H.; Ding, Z.; Liu, Q.; Zhu, C. A comprehensive active learning method for multiclass imbalanced data streams with concept drift—ScienceDirect. *Knowl. Based Syst.* **2021**, *215*, 106778. [CrossRef]
41. Lango, M.; Stefanowski, J. What makes multi-class imbalanced problems difficult? An experimental study. *Expert Syst. Appl.* **2022**, *199*, 116962. [CrossRef]
42. Abdi, L.; Hashemi, S. To combat multi-class imbalanced problems by means of oversampling and boosting techniques. *Soft Comput.* **2015**, *19*, 3369–3385. [CrossRef]
43. Janicka, M.; Lango, M.; Stefanowski, J. Using Information on Class Interrelations to Improve Classification of Multiclass Imbalanced Data: A New Resampling Algorithm. *Int. J. Appl. Math. Comput. Sci.* **2019**, *29*, 769–781. [CrossRef]
44. Freund, Y.; Schapire, R. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
45. Soui, M.; Mansouri, N.; Alhamad, R.; Kessentini, M.; Ghedira, K. NSGA-II as feature selection technique and AdaBoost classifier for COVID-19 prediction using patient's symptoms. *Nonlinear Dyn.* **2021**, *106*, 1453–1475. [CrossRef]
46. Wang, S.; Yao, X. Multiclass Imbalance Problems: Analysis and Potential Solutions. *IEEE Trans. Syst. Man Cybern. Part Cybern.* **2012**, *42*, 1119–1130. [CrossRef] [PubMed]
47. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
48. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2012; p. 236.
49. Khoshgoftaar, T.M.; Hulse, J.V.; Napolitano, A. Comparing Boosting and Bagging Techniques with Noisy and Imbalanced Data. *IEEE Trans. Syst. Man Cybern. Part Syst. Humans* **2011**, *41*, 552–568. [CrossRef]
50. Galar, M.; Fernandez, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Trans. Syst. Man Cybern. Part Appl. Rev.* **2012**, *42*, 463–484. [CrossRef]
51. Liu, X.Y.; Zhou, Z.H. Ensemble Methods for Class Imbalance Learning. In *Imbalanced Learning: Foundations, Algorithms and Applications*; He, H., Ma, Y., Eds.; Wiley-IEEE Press: River Street Hoboken, NJ, USA, 2013; pp. 61–82.
52. Blaszczyński, J.; Stefanowski, J.; Idkowiak, L. Extending Bagging for Imbalanced Data. In *Proceedings of the Eighth CORES (Core Ordering and Reporting Enterprise System)*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 226, pp. 269–278.
53. Breiman, L. Random Forests. *Mach. Learn.* **2001**, 45, 5–32. [CrossRef]
54. Thanathamathee, P.; Lursinsap, C. Handling imbalanced data sets with synthetic boundary data generation using bootstrap re-sampling and AdaBoost techniques. *Pattern Recognit. Lett.* **2013**, *34*, 1339–1347. [CrossRef]
55. Diez-Pastor, J.; Rodriguez, J.; Garcia-Osorio, C.; Kuncheva, L.I. Random Balance: Ensembles of variable priors classifiers for imbalanced data. *Knowl. Based Syst.* **2015**, *85*, 96–111. [CrossRef]
56. Barandela, R.; Sanchez, J.S.; Valdovinos, R.M. New Applications of Ensembles of Classifiers. *Pattern Anal. Appl.* **2003**, *6*, 245–256. [CrossRef]
57. Blaszczynski, J.; Stefanowski, J. Neighbourhood sampling in Bagging for imbalanced data. *Neurocomputing* **2015**, *150*, 529–542. [CrossRef]