DBRF: Random Forest Optimization Algorithm Based on DBSCAN

Wang Zhuo, Azlin Ahmad*
School of Computing Sciences-College of Computing-Informatics and Mathematics,
Universiti Teknologi MARA, Shah Alam, Malaysia

***Abstract*—The correlation and redundancy of features will directly affect the quality of randomly selected features, weakening the convergence of random forests (RF) and reducing the performance of random forest models. This paper introduces an improved random forest algorithm—A Random Forest Algorithm Based on DBSCAN (DBRF). The algorithm utilizes the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm to improve the feature extraction process, to extract a more efficient feature set. The algorithm first uses DBSCAN to group all features based on their relevance and then selects features from each group in proportion to construct a feature subset for each decision tree, repeating this process until the random forest is built. The algorithm ensures the diversity of features in the random forest while eliminating the correlation and redundancy among features to some extent, thereby improving the quality of random feature selection. In the experimental verification, the classification prediction results of CART, RF, and DBRF, three different classifiers, were compared through ten-fold cross-validation on six different-sized datasets using accuracy, precision, recall, F1, and running time as validation indicators. Through experimental verification, it was found that DBRF algorithm outperformed RF, and the prediction performance was improved, especially in terms of time complexity. This algorithm is suitable for various fields and can effectively improve the classification prediction performance at a lower complexity level.**

***Keywords*—Random forest; DBSCAN; feature selection; feature redundancy; classification algorithm**

I. Introduction

The correlation and redundancy of features will directly affect the performance of the random forest model. Especially in high-dimensional features, contain a lot of information, but may also contain a lot of useless, correlated or redundant features, making it difficult to distinguish between important and unimportant features, leading to an increase in the computational complexity of the machine learning model, an increase in the time overhead, a decrease in generalization ability, and a tendency to overfit the model [1-3].

Random forest (RF) is a hybrid classification algorithm that uses random sampling and random selection of features to construct multiple decision trees, making the model highly stable. Compared with other classification algorithms, RF has higher classification accuracy, lower generalization error, and faster training speed, so it has been widely applied in the field of data mining in many aspects. Random forest is a general-purpose algorithm with broad application potential in different fields. It has a large number of application cases in disease gene prediction, soil moisture estimation, industrial robot fault diagnosis, and text classification [1, 4-6]. The RF algorithm has many advantages such as high accuracy and strong generalization, but also has limitations. When dealing with high-dimensional data, its feature random selection mechanism causes poor correlation between the selected features and the category variables. In addition, the randomly selected feature variables may have high redundancy, which directly affects the quality of the feature subset in the random forest and weakens the convergence of the random forest, reducing the accuracy, generalization ability, and performance of the random forest model. Most of the current studies solve this problem by preprocessing and feature selection, but this may lead to new problems such as information loss and a dramatic increase in model complexity. These studies often only focus on the algorithm itself and do not consider its practical value. This study aims to improve the performance and efficiency of random forest model prediction by improving the method of random feature selection. This study designs and implements an improved random forest algorithm based on density clustering algorithm and hierarchical feature extraction mechanism. The significance of this study lies in significantly improving the accuracy and complexity of the random forest model prediction, providing practical and theoretical solutions and foundations for the sustainable development of this technology.

The main contribution of this study is to propose an improved random forest algorithm based on DBSCAN and stratified random sampling. This improved method enhances the quality of randomly selected feature subsets, and also confirms that the algorithm improves the accuracy of random forest models in different scale datasets, showing excellent performance in both cases. This study provides a new solution approach and empirical data for improving the random forest model.

The structure of this paper is as follows: Section II reviews related work, discusses the existing research on improving random forest algorithms and the progress in feature dimensionality reduction, and points out the shortcomings of existing studies. Section III provides a detailed description of the improved random forest algorithm design. Section IV validates the performance of the improved algorithm through experiments, including the evaluation of key performance indicators such as Accuracy, Precision, Recall, F1, and running time. Section V discusses the experimental results. Section VI summarizes the theoretical and practical significance of the research findings and provides a look ahead to future research directions.

*Corresponding Author.

II. Related Work

Many scholars have conducted relevant research on high-dimensional feature problems. Tang, Zhang et al. [7] used Relief F to calculate feature weights and used the Sequential Backward Selection algorithm to remove redundant features and weakly correlated features. The experiment proved that this method can effectively reduce redundant features. Compared with the methods of support vector machines, AdaBoost, and random forests, it has higher classification accuracy and efficiency. Ahmed, Deo et al. [4] proposed a soil moisture estimation model that uses the Boruta algorithm for feature selection. The model determines which features are significant by comparing the importance of the original features with the importance of the randomly generated shadow features. The experiment proved that the model has feature selection ability.

Rani and Baulkani [8] proposed the Lasso with Graph Kernel Feature Selection (LGKFS) algorithm, which combines the sparsity of Lasso regression and the structural information of GK-FS to reduce the feature dimension. When dealing with complex medical imaging data, the feature dimension is often very high, which may lead to the risk of overfitting in classification models and increase computational complexity. Therefore, effective feature selection becomes a key to improving classification performance. LGKFS algorithm combines Lasso regression and GK-FS algorithm to select the most valuable feature subset from high-dimensional features, thereby reducing the feature dimension and improving classification accuracy. Lasso regression is applied to the extracted features for sparse selection, removing most of the non-important features. GK-FS algorithm is used to further select the features after Lasso screening, based on graph kernel functions to calculate the similarity between features and select the most representative feature subset.

Jalal, Mehmood et al. [6] used boosted sampling and random subspace methods to remove unimportant features, dynamically increasing the number of trees to improve text classification performance. Each feature was assigned a weight, which reflected its importance in the classification task. Features were divided into important and unimportant features based on a set threshold. The choice of threshold depends on the distribution of feature weights and the performance requirements of the model. In each iteration, the random forest was updated based on the classified features. Important features were retained and used to construct new decision trees, while unimportant features were excluded. Meanwhile, the optimal number of trees was sought. This was achieved by gradually increasing the number of trees and evaluating the model performance until the optimal classification effect was reached.

Theerthagiri and Ruby [9] proposed a random forest feature selection algorithm based on recursive feature elimination and voting technology. The importance of each feature is evaluated by recursively building a random forest to assess the importance of each feature. The importance of a feature is evaluated by how it affects the prediction result during the decision tree building process.

Wang, Xue et al. [10] proposed a feature selection method based on variable-sized cooperative evolution particle swarm optimization. It includes a spatial division strategy based on feature importance, an adaptive mechanism for adjusting subgroup size, and a feature deletion and generation strategy based on fitness guidance, using the maximum information coefficient (MIC) to evaluate feature importance. Features with larger MIC values are moved to the set U, and the features in U are sorted and clustered based on their MIC values. Redundant features are deleted through the clustering results.

In high-dimensional data scenarios, to improve the accuracy of biomass estimation using a random forest algorithm, Zhang, Shen et al. [11] used an improved Random Forest algorithm by adding two regularization terms to further control the complexity of the model and improve performance. The L1 regularization selects the sum of the absolute values of the model parameters as the penalty term, thus selecting the most important feature at each node. This method helps to select the features that contribute the most to the model's prediction, reducing the influence of irrelevant or redundant features on the model. The average depth regularization term controls the depth of the tree and the number of nodes, thus limiting the complexity of the model. This limitation reduces the risk of overfitting the training data and improves the model's generalization ability. By limiting the depth and number of nodes, the model is more cautious in the feature selection process and avoids introducing noisy features due to the model being too complex.

To improve the performance of speech emotion classification, Xie, Zhu et al. [2] proposed a two-stage feature selection method based on random forest and grey wolf optimization. In the random forest algorithm, the importance of a feature is calculated based on its ability to increase the purity of leaf nodes. Then, the feature subset with the highest classification accuracy and the least number of features is selected through the iterative process of grey wolf optimization, which is used as the final optimal feature subset.

In summary, the quality of features affects the results of classification prediction, and many experts have conducted a series of optimization studies on feature dimensionality reduction. Currently, most studies mainly use feature selection algorithms to reduce the number of features, but ranking-based feature selection and subset-based feature selection both have certain limitations. Ranking-based feature selection algorithms mainly focus on the importance of individual features and ignore the interaction between features and the overall structure, while subset-based feature selection algorithms consider the combination of features, but may face the problem of large computational complexity and overfitting. Based on these problems, this study combines density clustering algorithm and hierarchical extraction to optimize the random forest algorithm. On the basis of establishing the diversity of random feature selection in the RF algorithm, it eliminates the interference of correlated and redundant features and builds a more predictive random forest, thereby improving the comprehensive performance of the model prediction.

III. Methodology

The RF algorithm has many advantages such as high accuracy and strong generalization, and has wide applications. However, when dealing with high-dimensional data sets, its random feature extraction mechanism reduces the correlation between features and category variables. Moreover, the

randomly extracted features may have high redundancy, which lowers the quality of the random feature subset, and weakens the convergence of the random forest, thereby reducing the overall performance of the random forest. Therefore, this paper optimizes the traditional RF algorithm by using the DBSCAN algorithm and hierarchical sampling to change the feature extraction mechanism. By constructing similar feature groups, it reduces the impact of these factors and improves the efficiency of the algorithm.

A. Random Forest

Random Forest is a machine learning algorithm that combines multiple decision trees. RF selects multiple subsets of the original sample set by random sampling with replacements from the set to build decision trees. At each node in the decision tree, a random subset of k attributes is selected from the set of attributes at the node, and then the best attribute is chosen from this subset for splitting. It is generally recommended that $k=log_2 d$, where d is the number of features in the data set [12]. The predictions of each tree are voted on to elect the best result. RF can handle both continuous and categorical variables [13, 14]. It can also rank the importance of features [15].

The architecture of the random forest algorithm is depicted in Fig. 1, and its underlying principles are delineated as follows [16]:

1) Randomly draw n training datasets from the original dataset with replacement.

2) Randomly select K features from each training dataset (where K is less than the total number of features in the original dataset).

3) Employ a specific strategy (e.g., Gini coefficient) to choose 1 feature from the k features as the splitting feature for the node, thereby constructing a decision tree.

4) Iterate through steps 1-3 to construct n decision trees.

5) Utilize each decision tree for result prediction.

6) Aggregate predictions and determine the final prediction result based on majority voting.

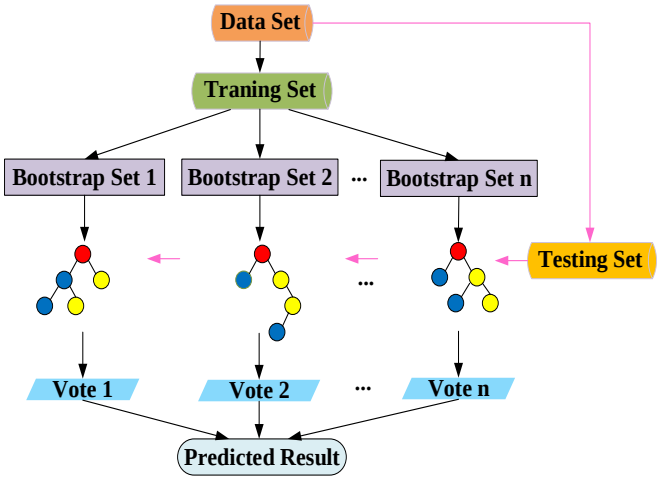


Fig. 1. The architecture of the random forest algorithm.

Random forest uses CART as a single classifier. CART uses the Gini coefficient as a selection criterion for splitting features. The key to building decision trees in random forest is to choose the optimal splitting feature, seeking higher and higher node "purity" in the splitting process [17]. The lower the Gini coefficient of a feature, the lower its impurity, and the feature with the lowest impurity is selected for node splitting. The impurity calculation is repeated for each node. After each split, the overall impurity of the tree decreases, until no features are available or the impurity has reached an optimal level, at which point the decision tree stops growing. The formula for calculating the Gini index is [18]:

$$GI_m = 1 - \sum_{k=1}^{|K|} p_{mk}^2 \quad (1)$$

Where: K denotes the number of categories, and P_{mk} represents the proportion of class column k in node m.

B. DBSCAN Algorithm

DBSCAN is a density-based clustering algorithm based on high-density connected regions. It defines clusters as the largest set of points that are densely connected and can group together regions with sufficiently high density and discover arbitrary-shaped clusters in noisy spatial databases. It uses parameters (*Eps, MinPts*) to describe the tightness of the sample distribution in the neighborhood. Parameter Eps is the maximum radius of the neighborhood. Parameter *MinPts* specifies the density threshold for dense regions. The working principle of DBSCAN is: randomly select a data point p from the dataset and check whether p's Eps neighborhood contains the minimum number of data points *MinPts*. If this condition is met, a new cluster is created and all identified data are added to the new clustering. Then, all data within the cluster will also be checked in the same way based on these two parameters, in order to add as many other data as possible that have not been checked before. This process is repeated until all data in the dataset are accessed [19, 20].

C. Stratified Sampling

Divide the entire sample into distinct strata or categories, and subsequently conduct random sampling from each stratum by selecting a specific number of individuals. Finally, combine the sampled individuals from all strata to form a representative sample. This approach is known as stratified sampling. Through categorization and stratification, it enhances the similarity among individuals within each category, facilitating the selection of a representative survey sample. This method is particularly suitable for complex situations with substantial individual variations and a large population size. The key characteristics of stratified sampling include [21, 22]:

1) Stratification involves the classification of similar individuals into distinct layers, with each layer representing a unique category. This method adheres to the principle of non-overlapping and exhaustive coverage, ensuring that every individual is assigned to one and only one layer.

2) To guarantee an equal opportunity for every individual to be included in the sample, stratified sampling necessitates simple random sampling within each layer. The sample size in each layer is determined proportionally based on the total number of individuals in that layer relative to the overall population size.

D. Design of the DBRF Algorithm

The establishment of a random forest involves two key random processes, one of which is the random selection of features. In high-dimensional datasets, it is highly probable to extract a large portion of irrelevant or redundant features, leading to a decrease in the generalization and accuracy of the RF algorithm. To address this issue, this study proposes an improvement to the traditional random forest algorithm using density clustering. Without removing redundant features and while retaining the original feature information, features are grouped (clustered) based on density to form similar feature groups TG = "TG_1 ,TG_2,…,TG_n". Within these similar feature groups, a certain proportion of features can represent all information for that entire group of features as well as express classification labels C. Features are randomly selected from each similar feature group TG_i in proportion to establish a subset for building individual decision trees. The architecture of the DBRF algorithm is illustrated in Fig. 2.

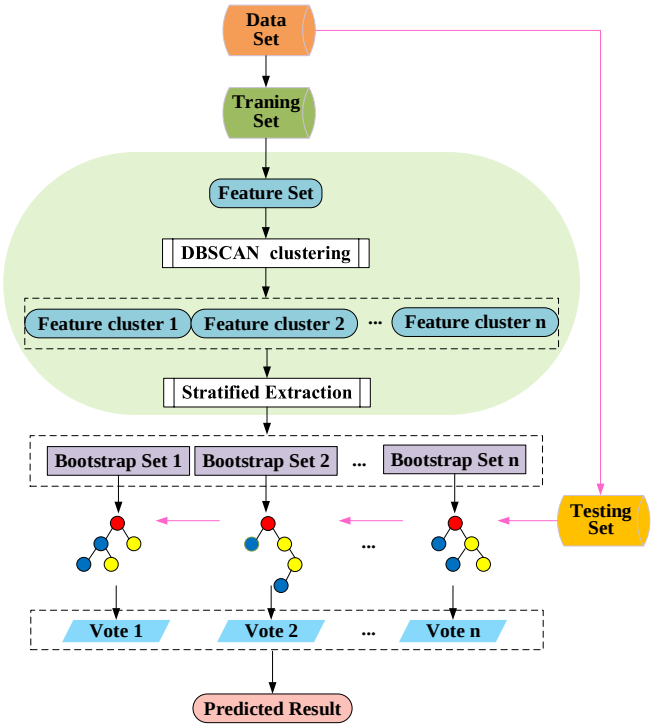


Fig. 2. The architecture of the DBRF algorithm.

The flowchart of the DBRF algorithm is illustrated in Fig. 3. Initially, a training set and a test set are established. Algorithm parameters such as the number of decision trees (*n_estimators*), maximum number of features for splitting (*max_features*), minimum samples in a cluster (*min_samples*), and neighborhood radius (*eps*) are configured. The Gini coefficient (GI_m) for each feature is computed, followed by DBSCAN clustering to form F similar groups. Features are then extracted from each group based on the proportion NF, and a subset of features is selected from other similar feature groups using the same approach to construct individual decision trees. This process is iterated multiple times until reaching the desired scale for constructing the random forest, at which point it terminates.

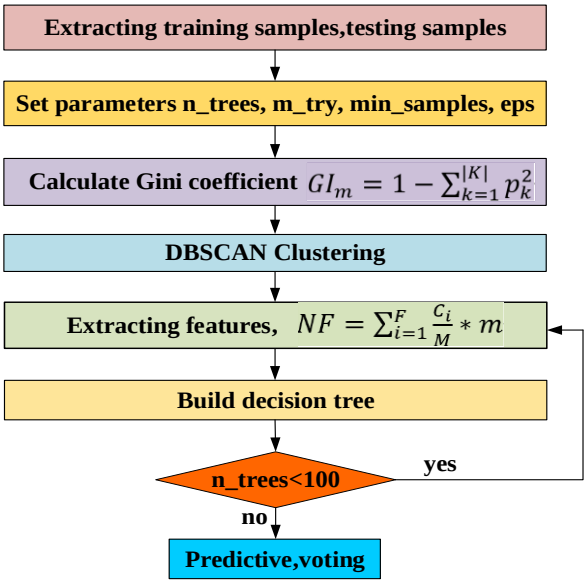


Fig. 3. The flowchart of DBRF.

The flowchart of feature extraction is shown in Fig. 4. After features are clustered, features with similar classification capabilities are grouped into a cluster. Then, features are sampled proportionally from each cluster. This ensures that the extracted features are more representative and do not favor any particular situation. These features are used to build a decision tree. The feature extraction process is repeated until all decision trees have been built.

The formula for proportional sampling is:

$$NF = \sum_{i=1}^{F} \frac{C_i}{M} * m \quad (2)$$

Where: F is the total number of clusters, C_i is the number of features in the *i-th* cluster, M is the total number of features, and m is the number of features to be extracted.

The pseudocode for feature proportional stratified sampling is as follows:

Algorithm 1: stratified sampling algorithm

Input: a set of similar feature clusters

Output: several groups of extracted features.

Method:

(1) According to the total number of features N and the number of features per layer: n_i , Calculate the sampling ratio for each layer $W = \frac{n_i}{N}$.

(2) Calculate the number of features to be extracted from each layer: $NUM = W*n$. And make sure that the total number of features extracted from each layer is n.

(3) Determine the number of features for each layer, then randomly select features from each layer to form a total of n samples.

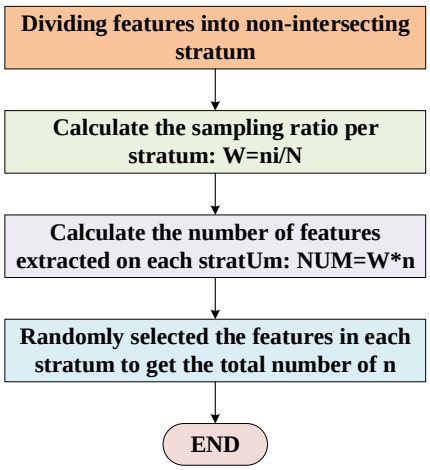


Fig. 4. The flow of feature extraction.

Therefore, the pseudocode for the complete DBRF algorithm is as follows:

Algorithm 2: DBRF algorithm

Input: Dataset: Data, number of decision trees: *n_estimators*, maximum number of features for splitting: *max_features*, minimum number of samples in a cluster: *min_samples*, neighborhood radius: *eps*

Output: An RF classifier

Methods：

The dataset is divided into a training set and a testing set;

for *i*=1 to *Num_Features*

    Compute the Gini coefficient (GI_m) for each feature.

endfor

while(*j*<= *min_samples* and *k*<=*eps*)

    F = DBSCAN(GI_m) // Feature clustering

endwhile

for *t=1* to *n_estimators*

    for *f=1* to *F*

        Extract *NF* features and construct a decision tree;

    endfor

endfor

The time complexity of RF is *O(tfnlog(n))*, where *t* is the number of decision trees built, *f* is the number of features selected at each node, and *n* is the number of samples in the training set [23]. The DBRF algorithm proposed in this paper is divided into two parts: clustering to build similar feature groups and building a random forest model. For the first part, the time is mainly spent on feature Gini coefficient clustering, with a time complexity of *O(mlogm)* [24], where m is the number of features. The second part is the random forest construction. Therefore, the time complexity of the DBRF algorithm proposed in this paper is the sum of the two parts, i.e., *O(tfnlog(n)+m(logm))*.

IV. Experimental Results and Analysis

A. Experimental Data

In order to objectively and comprehensively evaluate the effectiveness and advantages of the DBRF algorithm, the adaptability of the algorithm on different feature dimension datasets was analyzed. Six datasets with different feature dimensions from the UCI were selected, namely SPECT Heart (SPECT), Chess, SCADI, DARWIN, Period Changer (Period), and MicroMass. Table I describes the detailed information of the six datasets. The six datasets were divided into low-dimensional, medium-dimensional, and high-dimensional datasets based on the size of the samples and features [25]. SPECT and Chess belong to low-dimensional feature datasets. SCADI and DARWIN belong to medium-dimensional datasets. Period and Micromass are high-dimensional datasets. At the same time, these datasets include balanced and unbalanced datasets.

B. Experimental Results and Analysis

The experimental environment was set to Windows 11 operating system (64-bit), Intel(R) Core(TM) i7-10510U CPU, 16GB RAM, and Visual Studio Code. To verify the comprehensive performance of the proposed improved random forest, the experimental results of DBRF, RF, and CART classifiers were compared. The experiment used tenfold cross-validation to evaluate the accuracy, precision, recall, F1 score, and running time of the DBRF, RF, and CART models. Their overall performance was compared, highlighting the advantages of the improved algorithm. The experiment parameters were set to *n_estimators=100, max_depth=30, max_features= sqrt(n_features)*. The DBSCAN parameters were set to *min_samples=3, eps=0.02 or eps=0.03.*

TABLE I. The Descriptions of All Datasets

ID	DataSet	Feature Size	Sample Size	Feature Scale	Sample Scale	Balance	DOI
1	SPECT	22	267	Small	Small	unbalance	10.24432/C5P304
2	Chess	36	3196	Small	Large	balance	10.24432/C5DK5C
3	SCADI	205	70	Middle	Small	unbalance	10.24432/C5C89G
4	DARWIN	451	174	Middle	Small	balance	10.24432/C55D0K
5	Period	1177	90	Large	Small	unbalance	10.24432/C5B31D
6	Micromass	1300	571	Large	Middle	balance	10.24432/C5T61S

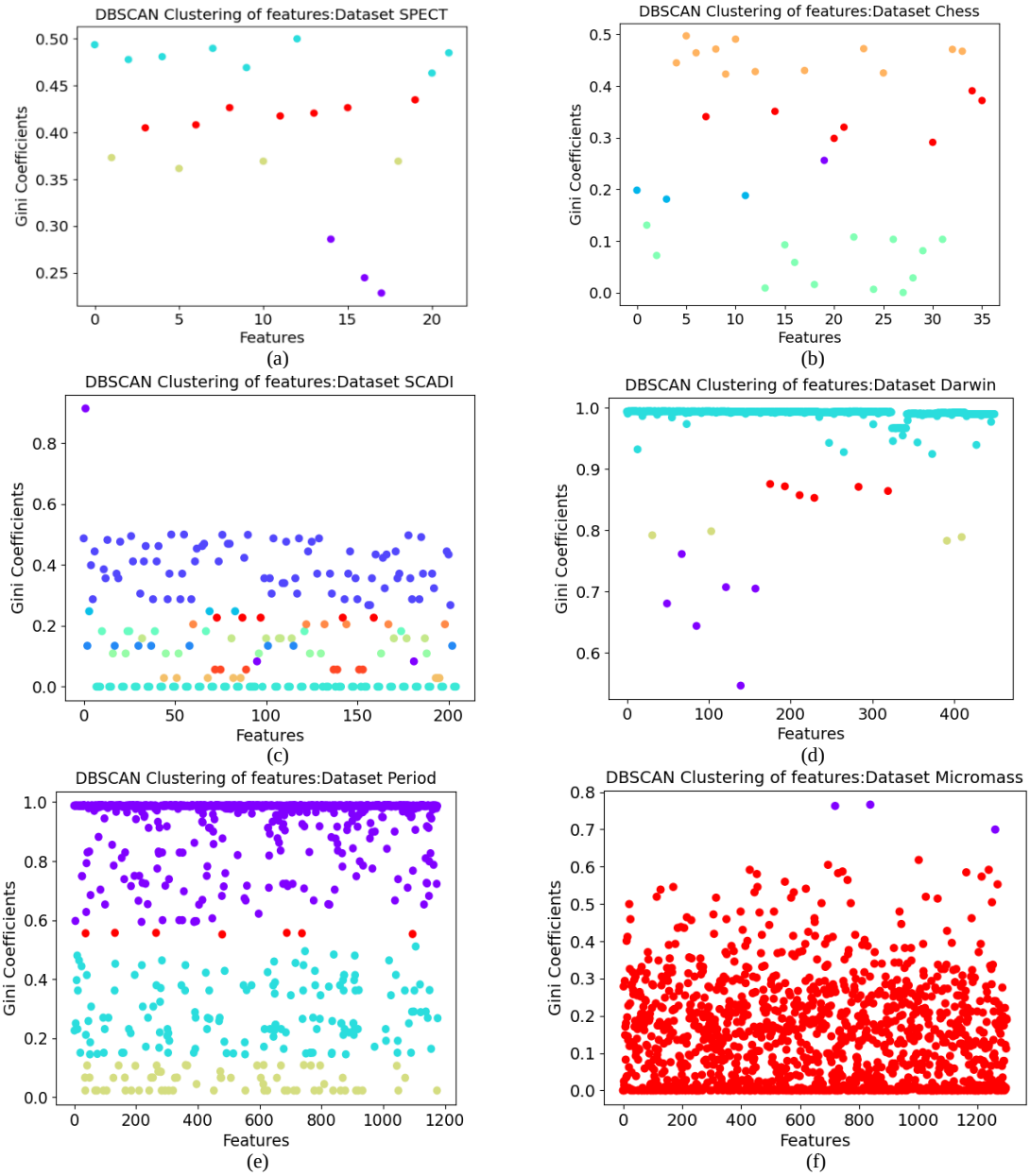


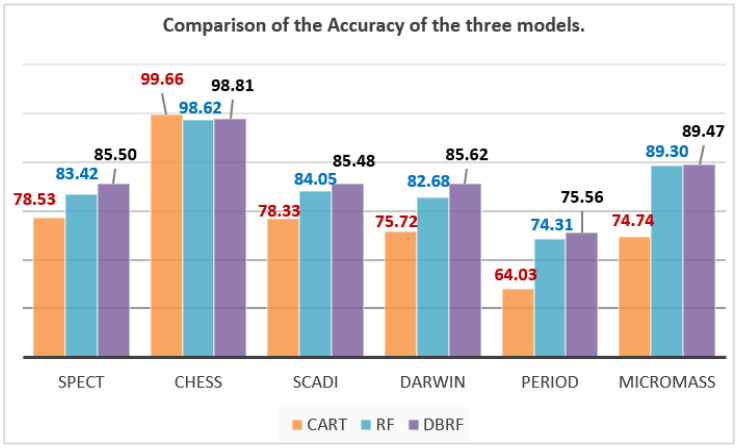
Fig. 5. Clustering results of DBRF on all Datasets (a) SPECT, (b)Chess, (c) SCADI, (d) DARWIN, (e) Period and (f) Micromass.

Fig. 5 illustrates the clustering results of DBRF on all datasets. Features clustered together are represented by points of the same color. Table II presents the number of feature clusters for each dataset, along with the maximum and minimum values of elements within each cluster. SCADI exhibits the highest number of similar feature clusters, with 12 clusters containing a maximum of 77 features and a minimum of three features. SPECT, Darwin, and period are all clustered into four similar feature clusters, with the maximum number of features in a cluster being 958 and the minimum being 3. MicroMass has the fewest feature clusters at only 2, with a maximum of 1297 features in a cluster and a minimum of three features. Chess consists of 36 features clustered into five groups, with each group containing between 1 to 13 features.

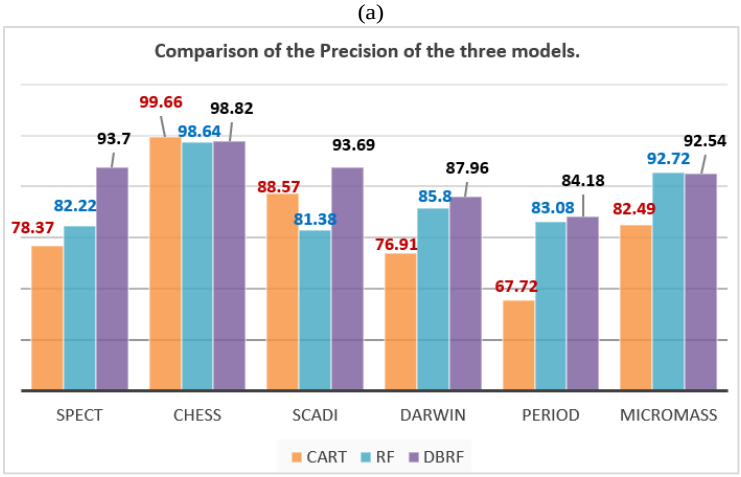
Fig. 6 shows the performance comparison of CART, RF, and DBRF models on all datasets. Fig. 6(a) shows the classification accuracy of CART, RF, and DBRF prediction models on all datasets. On the chess dataset, the accuracy of DBRF is 0.85% lower than that of CART and 0.19% higher than that of RF. For the other five datasets, the accuracy of DBRF is the highest. DBRF is 6.97% to 14.73% higher than CART and 0.17% to 2.94% higher than RF. This data confirms that DBRF as a composite classifier of CART is superior to CART. Furthermore, the higher accuracy of DBRF than RF indicates that the features extracted by DBRF are more representative and have higher accuracy. Importantly, DBRF performs well on both balanced and unbalanced datasets, demonstrating its versatility. Furthermore, DBRF improves performance on low-dimensional and high-dimensional datasets.

TABLE II. THE CLUSTERING RESULTS OF DBRF

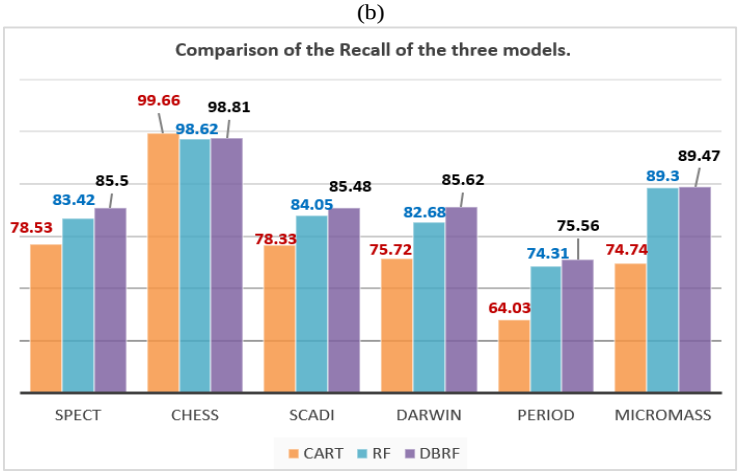
ID	DataSet	Number of clusters	Maximum number of features in a cluster	Minimum number of features in a cluster
1	SPECT	4	8	3
2	chess	5	13	1
3	SCADI	12	77	3
4	darwin	4	434	4
5	period	4	958	7
6	micromass	2	1297	3



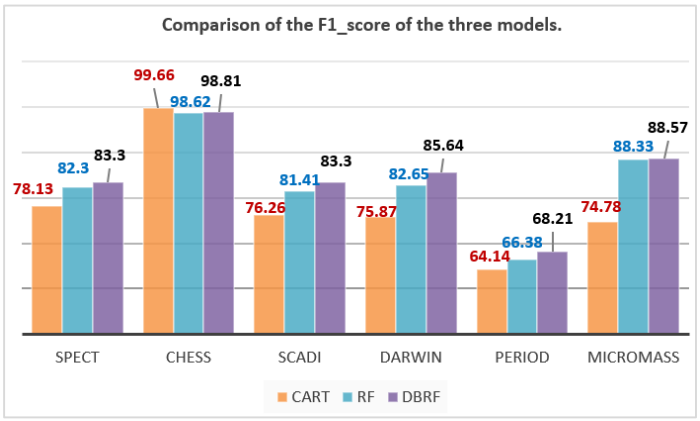
(a)



(b)



(c)



(d)

Fig. 6. Performance comparison of CART, RF, and DBRF models on all datasets.

Fig. 6(b) shows that the accuracy of DBRF on SCADI is 93.69, while the accuracy of RF is 81.38, which has been significantly improved by 12.31%. In Table II, the number of clusters in the dataset is the maximum value of 12, proving the complexity of the feature distribution. The clustering process can extract comprehensive and representative features, thus significantly improving the accuracy. At the same time, the accuracy of DBRF is the highest in all five datasets, with an improvement of 15.33%, 12.31%, 11.05%, 16.46%, and 10.05% compared to the lowest accuracy.

Fig. 6(c) compares the Recall values of the three models. Similarly, DBRF is at the highest level of Recall value. In the five datasets, DBRF's recall value is absolutely dominant, far higher than the RF and CART models. Compared with the lowest value, the increase in DBRF's Recall value is 6.97%, 7.15%, 9.9%, 11.53%, and 14.73%.

Fig. 6(d) shows the combined measure F1. In the low-dimensional datasets SPECT and Chess, DBRF performed 1.0% and 0.19% better in F1 than RF, respectively. However, the improvement was not significant due to fewer features in the low-dimensional datasets. In the medium-dimensional datasets SCADI and Darwin, DBRF's F1 score was 1.89% and 2.99% higher than RF, respectively. In the five datasets, the improvement in DBRF's F1 value compared to the lowest value was 5.17%, 7.04%, 9.77%, 4.07%, and 13.79%, respectively. In the high-dimensional datasets Period and MicroMass, DBRF improved the F1 score by 1.83% and 0.24% compared to RF, respectively. In the medium- and high-dimensional datasets, CART always performed the worst, thereby highlighting the advantages of ensemble learning models.

Table III shows the running times of the three prediction models on all datasets. For the four datasets including the high-dimensional dataset MicroMass, the DBRF model requires less time than the RF model, further emphasizing its efficiency and universality. Although DBRF adds the feature clustering process, it reduces the running time, indicating that balancing the extraction of typical features is more beneficial for the time efficiency of prediction classification. Since both DBRF and RF require the construction of 100 decision trees, their running times are longer than CART, but they achieve higher accuracy. Therefore, from the comprehensive performance indicators, the prediction classification effect of the DBRF model is better than that of the other two models.

TABLE III. RUNNING TIME OF THE THREE MODELS IN SECOND (S)

Dataset	CART	RF	DBRF
SPECT	2.39	8.32	**5.46**
chess	7.07	84.83	145.25
SCADI	2.91	6.36	**5.82**
darwin	101.21	356.80	**276.60**
period	123.59	76.93	153.39
micromass	464.87	572.17	**526.49**

* Bold font is the best results.

V. DISCUSSION

This study used different-sized datasets and conducted comprehensive experimental evaluations to verify the effectiveness of the proposed optimization techniques. The performance of the DBRF algorithm was compared with that of traditional RF and CART algorithms. The experiment demonstrated the technical improvements brought about by density-based feature extraction, and the empirical evidence proved the classification efficiency, scalability, and time complexity. Previous research techniques were only applicable to a single application domain [2, 6, 8, 11], while this study tested the proposed method on datasets with multiple different neighborhoods. In low, medium, and high-dimensional datasets, the DBRF achieved significant improvements in all performance indicators compared with the other two models. The maximum improvement in accuracy indicators was 14.73% in the high-dimensional MicroMass dataset. The highest accuracy value was 98.81% in the low-dimensional Chess dataset. The maximum improvement in precision indicators was 16.46% in the high-dimensional period dataset. In the running time indicator, the DBRF model required less time than the RF model in four datasets, including the high-dimensional MicroMass dataset, further highlighting its superiority and generality. Although the DBRF increased the feature clustering process, it reduced the running time, indicating that balancing the extraction of representative features is more beneficial for the time efficiency of predictive classification. In the five

datasets, the DBRF achieved the highest values for all four evaluation indicators, including accuracy rate. Therefore, by randomly selecting similar feature groups and extracting features from them, it is possible to effectively avoid the formation of redundant feature subsets in traditional RF and improve the accuracy and overall performance of predictive classification.

In summary, the DBRF algorithm proposed in this paper has better experimental effects than the other two algorithms, showing obvious advantages in high-dimensional data sets, low-dimensional data sets, and data sets with highly redundant features. Future research can further study the improvement of other types of clustering algorithms on random forest feature extraction to achieve higher efficiency and performance improvement. At the same time, it can be made more scalable to enable it to have a wider range of applications.

VI. Conclusion

Due to the correlation among features, redundancy, and a large amount of useless information, the overall performance of the machine learning model is affected. This study optimizes the traditional RF algorithm and proposes a DBRF algorithm based on DBSCAN. The experimental results show that the DBRF algorithm has a higher accuracy index improvement of 6.97%-14.73% and an F1 index improvement of 4.07%-13.79% compared with the other two models. In the 5 datasets, the accuracy rate and other four evaluation indicators of DBRF are the highest. For the four datasets including the high-dimensional dataset MicroMass, the DBRF model takes less time than the RF model, which demonstrates its significant advantage in time complexity. Therefore, the DBRF algorithm achieves the research goal of reducing the influence of feature correlation and redundancy on model performance. In future research, further exploration of other types of clustering algorithms for random forest feature extraction will be conducted to achieve higher efficiency and performance improvement, as well as stronger scalability.

References

[1] Ding, H., et al., RGAN-EL: A GAN and ensemble learning-based hybrid approach for imbalanced data classification. Information Processing & Management, 2023. 60(2).

[2] Xie, J., M. Zhu, and K. Hu, Fusion-based speech emotion classification using two-stage feature selection. Speech Communication, 2023. 152.

[3] Zhang, M., et al., Multi-objective optimization algorithm based on clustering guided binary equilibrium optimizer and NSGA-III to solve high-dimensional feature selection problem. Information Sciences, 2023. 648.

[4] Ahmed, A.A.M., et al., LSTM integrated with Boruta-random forest optimiser for soil moisture estimation under RCP4.5 and RCP8.5 global warming scenarios. Stochastic Environmental Research and Risk Assessment, 2021. 35(9): p. 1851-1881.

[5] Wu, Y., et al., Extracting random forest features with improved adaptive particle swarm optimization for industrial robot fault diagnosis. Measurement, 2024. 229: p. 114451.

[6] Jalal, N., et al., A novel improved random forest for text classification using feature ranking and optimal number of trees. Journal of King Saud University - Computer and Information Sciences, 2022. 34(6): p. 2733-2742.

[7] Tang, Q., et al., A Classification Method of Point Clouds of Transmission Line Corridor Based on Improved Random Forest and Multi-Scale Features. Sensors (Basel), 2023. 23(3).

[8] Rani, K.E.E. and S. Baulkani, Multi Variate Feature Extraction and Feature Selection using LGKFS Algorithm for Detecting Alzheimer's Disease. Indian Journal Of Science And Technology, 2023. 16(22): p. 1665-1675.

[9] Theerthagiri, P. and A.U. Ruby, RFFS: Recursive random forest feature selection based ensemble algorithm for chronic kidney disease prediction. Expert Systems, 2022. 39(9).

[10] Wang, P., et al., Feature clustering-Assisted feature selection with differential evolution. Pattern Recognition, 2023. 140.

[11] Zhang, X., et al., Improved random forest algorithms for increasing the accuracy of forest aboveground biomass estimation using Sentinel-2 imagery. Ecological Indicators, 2024. 159.

[12] LEO, B., Random Forests. 2001.

[13] Tyralis, H., G. Papacharalampous, and A. Langousis, A Brief Review of Random Forests for Water Scientists and Practitioners and Their Recent History in Water Resources. Water, 2019. 11(5).

[14] Liang, H., et al., Overflow warning and remote monitoring technology based on improved random forest. Neural Computing and Applications, 2020. 33(9): p. 4027-4040.

[15] Amir Behnamian, K.M., Sarah N. Banks, Lori White, Murray Richardson, and Jon Pasher, A Systematic Approach for Variable Selection With Random Forests: Achieving Stable Variable Importance Values. 2017.

[16] Zhou, J., S. Huang, and Y. Qiu, Optimization of random forest through the use of MVO, GWO and MFO in evaluating the stability of underground entry-type excavations. Tunnelling and Underground Space Technology, 2022. 124.

[17] Zhiqiang Geng , X.D., Jiatong Li , Chong Chu , Yongming Han, Risk prediction model for food safety based on improved random forest integrating virtual sample. 2022.

[18] Urbano, M.N., R.F. Diego, and P. Paulo, A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier. Pattern Recognition Letters, 2017. 99: p. 21-31.

[19] Latifi-Pakdehi, A. and N. Daneshpour, DBHC: A DBSCAN-based hierarchical clustering algorithm. Data & Knowledge Engineering, 2021. 135.

[20] Hanafi, N. and H. Saadatfar, A fast DBSCAN algorithm for big data based on efficient density calculation. Expert Systems with Applications, 2022. 203.

[21] Iliyasu, R. and I. Etikan, Comparison of quota sampling and stratified random sampling. Biometrics & Biostatistics International Journal, 2021. 10(1): p. 24-27.

[22] Latpate, R., et al., Stratified Random Sampling. Advanced Sampling Methods, 2021: p. 37-53.

[23] Akhiat, Y., et al., A New Noisy Random Forest Based Method for Feature Selection. Cybernetics and Information Technologies, 2021. 21(2): p. 10-28.

[24] Ros, F., et al., Detection of natural clusters via S-DBSCAN a Self-tuning version of DBSCAN. Knowledge-Based Systems, 2022. 241.

[25] Wang, Y., S. Krishna Saraswat, and I. Elyasi Komari, Big data analysis using a parallel ensemble clustering architecture and an unsupervised feature selection approach. Journal of King Saud University - Computer and Information Sciences, 2023. 35(1): p. 270-282.