

HCRF: an improved random forest algorithm based on hierarchical clustering

Wang Zhuo, Azlin Ahmad

School of Computing Sciences, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Malaysia

Article Info

Article history:

Received May 14, 2024

Revised Oct 9, 2024

Accepted Oct 30, 2024

Keywords:

Classification algorithm

Feature selection

Hierarchical clustering

Random forest

Redundant feature

ABSTRACT

Random forest (RF) selects feature subsets randomly. Useless and redundant features will lower the quality of the selected features and subsequently affect the overall classification accuracy of the RF. This study proposes an improved RF algorithm based on hierarchical clustering (HCRF). The algorithm uses hierarchical clustering algorithms to optimize the feature selection process, by establishing similar feature groups based on the GINI index, and then selecting features from each group proportionally to construct the feature subset. The feature subset is then used to construct a single classifier. This process increases the filtering of feature subsets, reducing the negative impact of useless and redundant features on the model, and improving the model's generalization ability and overall performance. In the experimental verification, ten datasets of different sizes and domains were selected, and the accuracy, precision, recall, F1 score, and running time of HCRF, support vector machine (SVM), RF, classification and regression tree (CART) were compared using 10-fold cross-validation. Combining all the results, the HCRF algorithm showed significant improvements in all evaluation indicators, proving that its performance is superior to the other three classifiers. Therefore, this algorithm has broad application areas and value, and effectively improves the overall performance of the classifier within a lower complexity range.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Azlin Ahmad

School of Computing Sciences, College of Computing, Informatics and Mathematics

Universiti Teknologi MARA

Shah Alam, Malaysia

Email: azlin@tmsk.uitm.edu.my

1. INTRODUCTION

Random forest (RF) method is an ensemble learning method. It introduces the method of randomly selecting features on the basis of Bagging [1], [2]. Because it has strong classification ability and stability, it has many application cases in the fields of medicine, industry, and agriculture [3]-[5].

However, when the feature dimension of the dataset is high and there are a large number of redundant features, random selection of features may lead to a decrease in the correlation between features and class variables. Moreover, the features randomly selected may have high redundancy, which reduces the quality of the random feature subset, and may result in a decrease in the convergence of the RF model and poor generalization ability, thereby affecting the accuracy of model prediction and overall performance [6], [7].

Many scholars have conducted relevant research on this issue. Motamedi *et al.* [8] used the LASSO algorithm to remove irrelevant features, which had ideal performance in terms of accuracy and time complexity. In the field of recognizing human activities, Thakur and Biswas [4] used regularized RF to select

relevant features in order to achieve the goal of reducing feature dimensions. Disha and Waheed [9] calculated the importance score of features using the GINI coefficient weighting to screen important features and achieve the goal of feature dimension reduction. Jiang *et al.* [5] selected features based on their importance step by step to form the optimal feature variable combination. Mafarja *et al.* [10] used a dimensionality reduction method called binary whale optimization algorithm (BWOA) to eliminate irrelevant and redundant features and improve the accuracy of RF software fault prediction. Sun *et al.* [11] measured the classification accuracy and relevance of decision trees to select the best feature subset. Wu *et al.* [12] proposed an improved adaptive particle swarm optimization algorithm for extracting features from RF for the diagnosis of faults in industrial robots. Thakur and Biswas [13] built a human activity recognition model by randomly permuting each feature and calculating the model performance, obtaining feature importance scores, and selecting features based on those scores. From the above literature, it can be seen that most studies focus on selecting important features to form feature subsets, but this may lead to a reduction in the diversity of RF. Moreover, some technical architectures are too complex, leading to high computational costs. This may limit the feasibility and scalability of the method in practical applications. Furthermore, most of the methods are mainly applicable to a certain field, and their suitability for other application fields has not been fully explored and validated. Therefore, the applicability of these methods may have certain limitations.

In summary, in response to the above problems, this study proposes a hierarchical clustering-based RF optimization algorithm. This improved process optimizes the establishment of feature subsets, reduces the influence of useless and redundant features, increases the correlation between features and class variables, and improves the quality of feature subsets. It also proves the excellent performance of the algorithm in different neighborhoods and the datasets of different scales. Therefore, this study provides new solutions and ideas for improving and optimizing related problems, and also provides a new perspective and direction for the prediction classification framework.

The rest of this paper is structured as follows: the second part provides a detailed description of the improved RF algorithm design. The third part discusses the experimental results. The part four summarizes the research findings and outlines future research directions.

2. METHOD

The problem that this study aims to solve is: how to reduce the influence of useless and redundant features on the model performance without removing redundant features, retaining the original feature information, and maintaining the diversity of RF. The HCRF algorithm proposed in this study attempts to solve this problem during the process of establishing feature subsets. The architecture of the HCRF algorithm is shown in Figure 1. In Figure 1, the HCRF algorithm has two main optimization parts. The first is feature grouping, which is completed by the hierarchical clustering algorithm [14]. The second is that the feature subsets are randomly selected from the feature groups in proportion. The training samples are generated by the random sampling with replacement for the dataset. The pseudocode for the HCRF algorithm is shown in Algorithm 1.

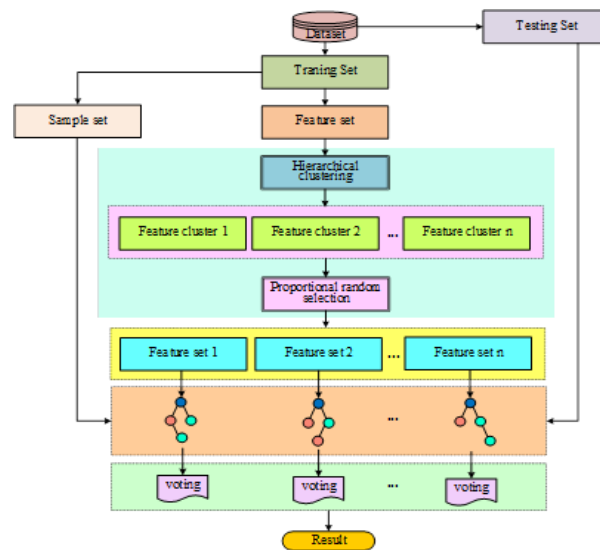


Figure 1. The architectural of the HCRF algorithm

Algorithm 1. An improved RF algorithm based on hierarchical clustering (HCRF)**Input :**

Data: Dataset

n_estimators: The number of decision trees.

max_features: The maximum number of features when a node is split.

max_depth: The maximum depth of each decision tree.

min_samples_split: The minimum number of samples necessary for an internal node to perform a split.

min_samples_leaf: The minimum number of samples necessary for a leaf node.

criterion: Evaluate the quality of each split point (default value: 'GINI index').

K: The number of clustering clusters.

Output : A RF classifier

1. A random sampling method with replacement is employed to construct the training set, while the instances not selected are designated as the test set;

2. **for** i=1 to Num_Features

3. Calculate the Gini coefficient for each feature, the formula for which is:

$$GI_m = 1 - \sum_{k=1}^{|K|} p_k^2 ;$$

4. **endfor**

5. **while** (The number of clusters is less than k)

6. The calculation of the distance metric is as follows:

$$D_{mean}(C_i \boxminus C_j) = \left| \frac{1}{|C_i|} \sum_{p \in C_i} p - \frac{1}{|C_j|} \sum_{q \in C_j} q \right| ;$$

7. Group the features based on the Gini coefficient, and calculate the number of similar feature clusters F.

8. **endwhile**

9. **for** t=1 to n_estimators

10. A random sample is generated to construct the sample set S_i ;

11. **for** f=1 to F

12. Randomly select NF features from the similar features groups in proportion,

$NF = \sum_{i=1}^F \frac{C_i}{M} * m$. Construct a feature subset fg and develop a CART with the sample set S_i ;

13. **endfor**

14. **endfor**

15. Evaluate the model using the test set;

16. Output test metrics values: accuracy, precision, recall, F1 score, and running time;

17. **end**

There are two important improvements in the above algorithm process, the first is to group features, and the second is to extract features to build a feature subset. The following will provide a detailed introduction to the key operations.

2.1. Feature grouping

This study uses hierarchical clustering algorithms to perform feature grouping (clustering) operations [15]. The flowchart of feature clustering is shown in Figure 2. In this flowchart, the parameters that the hierarchical clustering algorithm needs to determine mainly include clustering method, the method of distance measurement, and the clustering number. The agglomerative clustering is selected. The distance measurement method used is the centroid distance. It is determined by the distance between the centroids of the two clusters. Among them, the centroid of the cluster is the average of all sample points (the average of the GINI coefficient of the features). The formula for calculating the GINI index is as [16], [17]:

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

where: K denotes the total number of categories, while P_{mk} represents the proportion of class k within the node m . The formula for calculating the centroid distance is as follows [18], [19]:

$$D_{mean}(C_i \boxminus C_j) = \left| \frac{1}{|C_i|} \sum_{p \in C_i} p - \frac{1}{|C_j|} \sum_{q \in C_j} q \right| \quad (2)$$

where: $|C_i|$ is the number of objects in class C_i , and $|C_j|$ is the number of objects in class C_j .

In hierarchical clustering, the distance matrix DM is represented by the similarity between features (the difference of GINI coefficients), as shown in the following formula [20]:

$$DM = \begin{bmatrix} G_{1,1} & \cdots & G_{1,m} \\ \vdots & \ddots & \vdots \\ G_{m,1} & \cdots & G_{m,m} \end{bmatrix} \tag{3}$$

Where: m represents the number of features.

This matrix represents the difference in GINI coefficient between each pair of features. This matrix is a symmetric matrix. Therefore, only the upper triangle of the matrix needs to be calculated. It reduces the time and space complexity. The minimum value in the matrix is selected and a merging operation is performed. There are three cases:

- G_{ij} is the GINI difference between two individual features. This is the most similar pair of features. Merge the two features into one cluster.
- G_{ij} is the GINI difference between a feature and the centroid of a cluster. The feature is similar to the features in the cluster. Add the feature to the cluster.
- G_{ij} is the GINI difference between the centroids of the two clusters. The features in the two clusters are similar. Merge the two clusters.

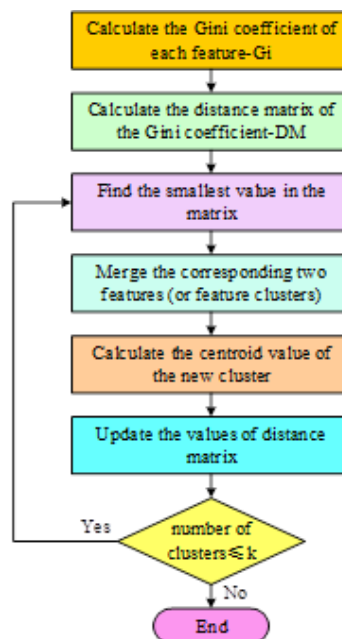


Figure 2. The flowchart of feature clustering

2.2. Feature extraction

After features are clustered, those features with similar classification capabilities are grouped into a cluster. Then, the features are randomly sampled from each cluster in proportion. These features are more representative and unbiased. A decision tree is built using these features. The process is repeated until the RF reaches its predetermined size. The steps for feature proportional sampling are as follows:

- All features are divided into several layers.
- Based on the total number of features N and the number of features per layer n_i , calculate the sampling ratio for each layer: $W = \frac{n_i}{N}$.
- The number of features to be sampled for each layer is calculated as $NUM = W * n$, and the total number of features sampled in all layers should be n .
- Features are randomly sampled from each layer based on the determined number, and the resulting sample set contains a total of n features.

The formula for feature proportional extraction is [21], [22]:

$$NF = \sum_{i=1}^F \frac{C_i}{M} * m \tag{4}$$

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.

3. RESULTS AND DISCUSSION

As shown in Table 1, this experiment selected 7 different neighborhoods and different scale datasets from UCI. The experiment used ten-fold cross-validation. The experiment compared the comprehensive performance of the SVM, CART, RF and HCRF. Accuracy, precision, recall, F1-score, and running time were used as evaluation indicators [13], [23]. The experimental environment is Windows 11 operating system (64-bit), Intel(R) Core(TM) i7-10510U CPU, 16 GB of RAM, and Visual Studio Code. The programming language is Python 3.0. T

His experiment selects two parameters of RF for parameter tuning, namely $n_estimators$ and max_depth . The parameter $n_estimators$ is the number of decision trees that are built by RF, and if the value is too small, it will result in insufficient training. If the value is too large, it will increase the computational complexity and lead to overfitting. The parameter max_depth is the maximum depth of the decision tree. The parameter min_sample is set to 1, and $max_features$ is set to $\sqrt{n_features}$ [24].

Based on the computational power of the experimental environment and the scale of the dataset, the parameter setting standard is the feature size. The initial parameter setting values are as follows: for small-scale datasets, $n_estimators$ is set to 10, 100, 200, and max_depth is set to 5, 20, 30, 35. For medium- and large-scale datasets, $n_estimators$ is set to 10, 50, 100, and max_depth is set to 5, 10, 20, 30 [25]. Table 2 shows the optimal parameter values for each dataset after parameter tuning.

Table 1. The descriptions of all datasets

ID	DataSet	Feature size	Sample size	Class size	Feature scale	Sample scale	Balance	DOI
1	SPECT	22	267	2	Small	Small	unbalance	10.24432/C5P304
2	Sports	59	1000	2	Small	Middle	balance	10.24432/C5801R
3	SCADI	205	70	7	Middle	Small	unbalance	10.24432/C5C89G
4	DARWIN	451	174	2	Middle	Small	balance	10.24432/C55D0K
5	CNAE-9	856	1080	9	Middle	Middle	balance	10.24432/C51G7P
6	Period	1177	90	2	Large	Small	unbalance	10.24432/C5B31D
7	MicroMass	1300	571	20	Large	Middle	balance	10.24432/C5T61S

Table 2. Best parameters of RF on all datasets

ID	DataSet	$n_estimators$	max_depth
1	SPECT	200	5
2	Sports	50	10
3	SCADI	10	30
4	DARWIN	50	20
5	CNAE-9	100	30
6	Period	100	30
7	MicroMass	100	20

Figures 3-6 compares the accuracy, precision, recall, and F1 scores of the four models on seven datasets. As shown in Figure 3, the accuracy of the HCRF is the highest value on six datasets, indicating that the accuracy of the HCRF is higher than the other three models. The accuracy of HCRF is improved by 0.1%-6.32% compared to SVM, and by 0.18%-1.76% compared to RF. In Figures 4-6, the precision, recall, and F1 scores of HCRF have been improved to varying degrees, with a similar trend. Therefore, it can be proved that the optimization method proposed in this study has improved the quality of the feature subset and improved the generalization ability, and improved the accuracy and overall performance of RF. The experimental results have verified that the optimization process proposed is effective and has achieved the research objectives.

The above results include small, medium, and high-dimensional datasets, and HCRF has excellent performance on each dataset. Therefore, it can be proved that the HCRF is suitable for small, medium, and high-dimensional datasets and has broad applicability.

Table 3 shows the running times of the four models on the seven datasets. Figure 7 shows the trend of the model's running time. It is easy to see that the running time of the HCRF is less than that of the RF. Although the running time of HCRF is slightly longer than that of CART, it is because HCRF builds 100 CARTs. But the time does not increase by 100 times. This shows that HCRF has a higher advantage in terms

of time complexity. Therefore, the HCRF algorithm also has higher application value and can be applied in a wider range of fields.



Figure 3. Comparison of accuracy among four models



Figure 4. Comparison of precision among four models

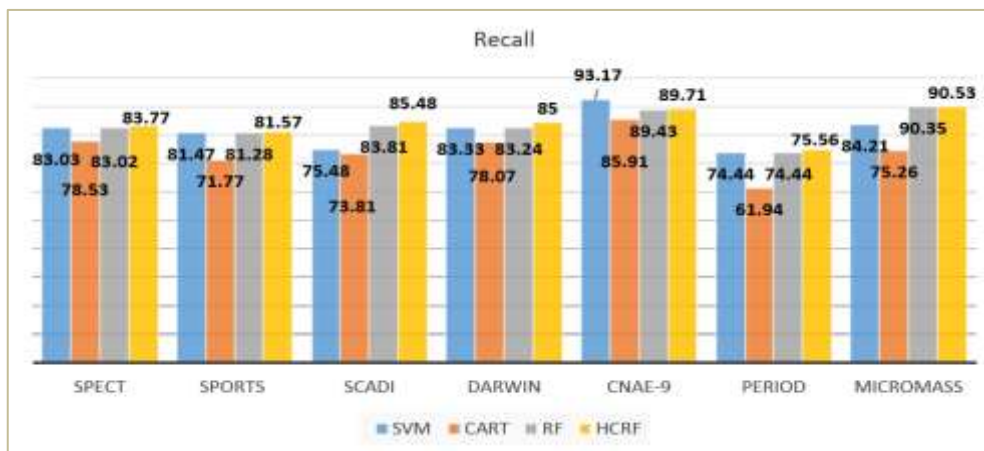


Figure 5. Comparison of recall among four models

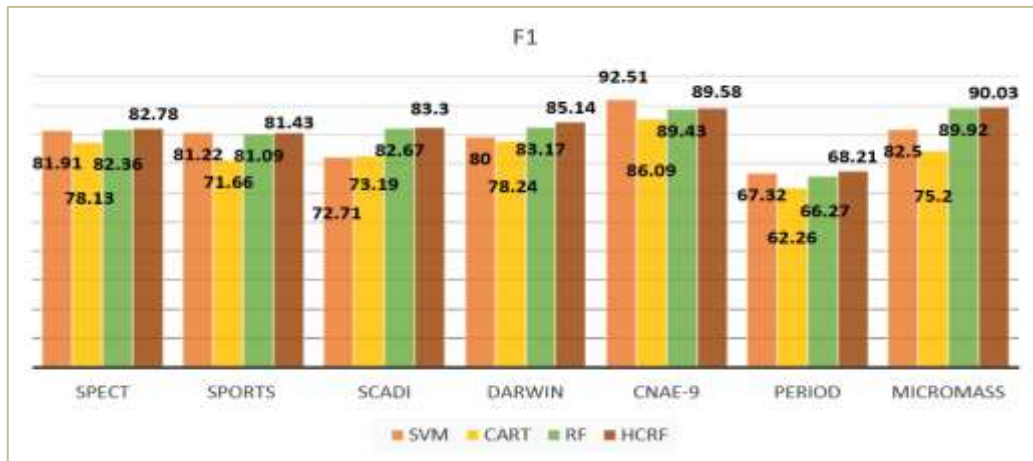


Figure 6. Comparison of F1 scores among four models

Table 3. Running times of four models

	SVM	CART	RF	HCRF
SPECT	2.79	3.11	10.17	10.09
Sports	70.34	116.61	212.14	209.78
SCADI	7.53	3.01	2.18	2.39
DARWIN	16.89	110.39	217.91	130.95
CNAE-9	112.32	173.58	298.56	233.06
period	11.65	121.44	188.61	138.30
MicroMass	61.47	538.28	607.47	485.37

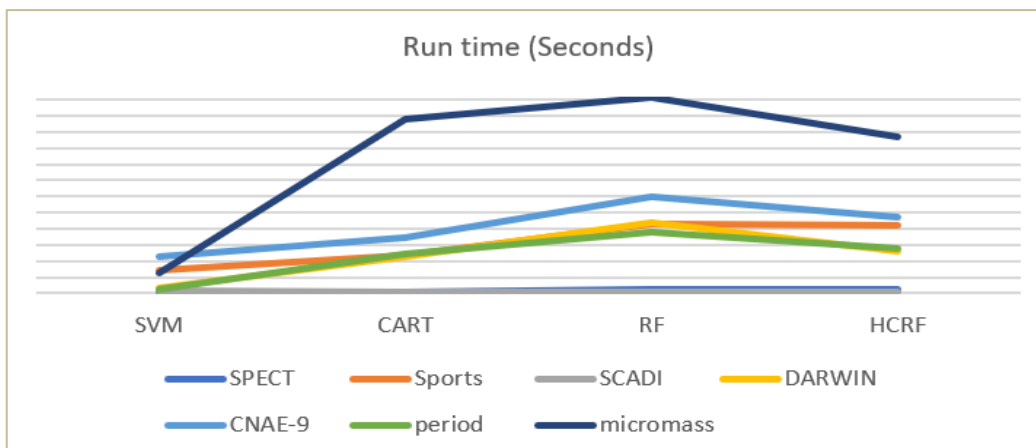


Figure 7. Comparison of running times in four models





4. CONCLUSION

In response to the issues of correlation and redundancy among features, and the quality of randomly selecting features in RF, this study optimizes the RF and proposes an HCRF algorithm based on hierarchical clustering. The experimental results show that the accuracy of the HCRF is improved by 0.1%-15.27% compared to the other three models. Among the 6 datasets, the accuracy, precision, recall, and F1 value of the HCRF are the highest. For the 6 datasets including the high-dimensional dataset period and MicroMass, the HCRF requires less time than the RF, demonstrating its significant advantage in time complexity. The experimental results prove that the HCRF achieves the predefined research objectives: reducing the influence of feature correlation and redundancy on the model, and improving the model's generalization ability and overall performance. In the future, further research will explore refining the details of clustering to achieve higher accuracy and improved time complexity, thereby enhancing its practical value.





REFERENCES

- [1] M. Bader-El-Den, E. Teitei, and T. Perry, "Biased random forest for dealing with the class imbalance problem," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2163-2172, 2019, doi: 10.1109/TNNLS.2018.2878400.
- [2] Y. Manzali and M. Elfar, "Random Forest Pruning Techniques: A Recent Review," *Operations Research Forum*, vol. 4, no. 2, p. 43, 2023, doi: 10.1007/s43069-023-00223-6.
- [3] M. I. Prasetyowati, N. U. Maulidevi, and K. Surendro, "The accuracy of Random Forest performance can be improved by conducting a feature selection with a balancing strategy," *PeerJ Comput Sci*, vol. 8, p. e1041, 2022, doi: 10.7717/peerj-cs.1041.
- [4] D. Thakur and S. Biswas, "An Integration of feature extraction and guided regularized random forest feature selection for smartphone based human activity recognition," *Journal of Network and Computer Applications*, vol. 204, p. 103417, 2022, doi: 10.1016/j.jnca.2022.103417.
- [5] F. Jiang *et al.*, "Estimating the growing stem volume of coniferous plantations based on random forest using an optimized variable selection method," *Sensors*, vol. 20, no. 24, p. 7248, 2020, doi: 10.3390/s20247248.
- [6] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, and Y.-L. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 882-895, 2020, doi: 10.1109/tevc.2020.2968743.
- [7] J. Luo, L. Xiong, and J. Du, "A Random forest optimization algorithm fusion with approximate markov blanket," *Computer Engineering and Applications* vol. 59, no. 20, pp. 77-84, 2023, doi: 10.3778/j.issn.1002-8331.2207-0443.
- [8] F. Motamedi, H. Pérez-Sánchez, A. Mehridehnavi, A. Fassihi, F. Ghasemi, and J. Xu, "Accelerating big data analysis through LASSO-Random forest algorithm in QSAR studies," *Bioinformatics*, vol. 38, no. 2, pp. 469-475, 2022, doi: 10.1093/bioinformatics/btab659.
- [9] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using GINI Impurity-based weighted random forest (GIWRF) feature selection technique," *Cybersecurity*, vol. 5, no. 1, 2022, doi: 10.1186/s42400-021-00103-8.
- [10] M. Mafarja *et al.*, "Classification framework for faulty-software using enhanced exploratory whale optimizer-based feature selection scheme and random forest ensemble learning," *Applied Intelligence*, vol. 53, no. 15, pp. 18715--18757, Feb 9 2023, doi: 10.1007/s10489-022-04427-x.
- [11] Z. Sun, G. Wang, P. Li, H. Wang, M. Zhang, and X. Liang, "An improved random forest based on the classification accuracy and correlation measurement of decision trees," *Expert Systems with Applications*, vol. 237, p. 121549, 2024, doi: 10.1016/j.eswa.2023.121549.
- [12] Y. Wu, Y. Bai, S. Yang, and C. Li, "Extracting random forest features with improved adaptive particle swarm optimization for industrial robot fault diagnosis," *Measurement*, vol. 229, p. 114451, 2024, doi: 10.1016/j.measurement.2024.114451.
- [13] D. Thakur and S. Biswas, "Permutation importance based modified guided regularized random forest in human activity recognition with smartphone," *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107681, 2024, doi: 10.1016/j.engappai.2023.107681.
- [14] P. Shetty and S. Singh, "Hierarchical clustering: a survey," *International Journal of Applied Research*, vol. 7, no. 4, pp. 178-181, 2021, doi: 10.22271/allresearch.2021.v7.i4c.8484.
- [15] L. M. C. Cabezas, R. Izbicki, and R. B. Stern, "Hierarchical clustering: Visualization, feature importance and model selection," *Applied Soft Computing*, vol. 141, p. 110303, 2023, doi: 10.1016/j.asoc.2023.110303.
- [16] M. N. Urbano, 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*, vol. 99, pp. 21-31, 2017, doi: 10.1016/j.patrec.2017.05.004.
- [17] J. Zhu *et al.*, "Coalbed Methane production model based on random forests optimized by a genetic algorithm," *ACS Omega*, vol. 7, no. 15, pp. 13083-13094, 2022, doi: 10.1021/acsoomega.2c00519.
- [18] A. Saxena *et al.*, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664-681, 2017, doi: 10.1016/j.neucom.2017.06.053.
- [19] A. Mantero and H. Ishwaran, "Unsupervised random forests," *Stat Anal Data Min*, vol. 14, no. 2, pp. 144-167, Apr 2021, doi: 10.1002/sam.11498.
- [20] K. Guo, X. Wan, L. Liu, Z. Gao, and M. Yang, "Fault Diagnosis of intelligent production line based on digital twin and improved random forest," *Applied Sciences*, vol. 11, no. 16, p. 7733, 2021, doi: 10.3390/app11167733.
- [21] R. Iiyasu and I. Etikan, "Comparison of quota sampling and stratified random sampling," *Biometrics & Biostatistics International Journal*, vol. 10, no. 1, pp. 24-27, 2021, doi: 10.15406/bbij.2021.10.00326.
- [22] R. Latpate, J. Kshirsagar, V. Kumar Gupta, and G. Chandra, "Stratified random sampling," *Advanced Sampling Methods*, pp. 37-53, 2021, doi: 10.1007/978-981-16-0622-9_3.
- [23] L. Ren, H. Zhang, A. Sekhari Seklouli, T. Wang, and A. Bouras, "Stacking-based multi-objective ensemble framework for prediction of hypertension," *Expert Systems with Applications*, vol. 215, p. 119351, 2023, doi: 10.1016/j.eswa.2022.119351.
- [24] T. Yan, R. Xu, S.-H. Sun, Z.-K. Hou, and J.-Y. Feng, "A real-time intelligent lithology identification method based on a dynamic felling strategy weighted random forest algorithm," *Petroleum Science*, vol. 21, no. 2, pp. 1135-1148, 2024, doi: 10.1016/j.petsci.2023.09.011.
- [25] M. P. Little, P. S. Rosenberg, and A. Arsham, "Alternative stopping rules to limit tree expansion for random forest models," *Sci Rep*, vol. 12, no. 1, p. 15113, Sep 6 2022, doi: 10.1038/s41598-022-19281-7.

BIOGRAPHIES OF AUTHORS

Wang Zhuo     obtained her bachelor's degree from Shanxi Normal University in 2002 and her master's degree from East China Jiaotong University in 2005. She is currently pursuing her PhD at the School of Computing Sciences, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Malaysia. She has written or co-authored more than three international conference and journal publications. Her research interests include ensemble learning and data mining. She can be contacted at email: ncofzw@gmail.com.



Azlin Ahmad     received her Ph.D. from Universiti Teknologi Malaysia (UTM) Kuala Lumpur in 2016. Currently, she is a senior lecturer at the School of Computing Sciences, College of Computing, Informatics and Mathematics, Universiti Teknologi MARA (UiTM). Her current research interests are in data analytics, machine learning, artificial intelligence, and its applications. She can be contacted at email: azlin@tmsk.utm.edu.my.