Available online at http://scik.org
Commun. Math. Biol. Neurosci. 2023, 2023:118
https://doi.org/10.28919/cmbn/8225
ISSN: 2052-2541

BEANS CLASSIFICATION USING DECISION TREE AND RANDOM FOREST WITH RANDOMIZED SEARCH HYPERPARAMETER TUNING

MEIDYA KOESHARDIANTO[1], KURNIAWAN EKA PERMANA[1], DHIAN SATRIA YUDHA KARTIKA[2], WAHYUDI SETIAWAN[3,*]

[1]Department of Informatics, University of Trunojoyo Madura, Bangkalan, Jawa Timur 69192, Indonesia

[2]Department of Information Systems, University of Pembangunan Nasional Veteran, Surabaya, Jawa Timur 60294, Indonesia

[3]Department of Information Systems, University of Trunojoyo Madura, Bangkalan, Jawa Timur 69192, Indonesia

Abstract: Dry-beans are a food with high protein. Dry-beans can be used as processed food products for emergency conditions such as famine, natural disasters, and war. Dry-beans can be used as a long-lasting product. To identify types of beans, manual work certainly requires a lot of time and effort. Therefore, creating a system that can classify beans in a computerized system is necessary. In this study, we classified beans using public data from Koklu. The data consists of sixteen features, seven classes with 13,611 rows. The data for each class of bean is unbalanced, so it is necessary to carry out a balanced dataset using random oversampling. Machine learning for classification using Decision Tree and Random Forest. Apart from that, hyperparameter tuning with randomize search for the number of trees 50, 75, 150, 200, and 300. The test results show that the Random Forest's accuracy, precision, recall, and f1-score reach 0.9658 respectively. The best parameter number of trees is 300.

Keywords: beans; classification; random forest; randomized search; hyperparameter tuning.

2020 AMS Subject Classification: 92B10.

*Corresponding author
E-mail address: wsetiawan@trunojoyo.ac.id

Received September 17, 2023

1. INTRODUCTION

Beans are a plant product that can be used as processed food. The food potential of Beans adds nutrients to the daily menu. Beans contain high protein, vitamins B, minerals, and fiber. Beans can be used for emergency food programs during natural disasters, long dry seasons, fires, and war [1]–[3].

Globally, there are more than 1,300 species of beans, but only about 20 are consumed by humans. Among these beans are dry-beans, which are low in fat, low in sodium, and do not contain cholesterol. Dry-beans are cheaper than animal food products. Also, if stored properly, the product can have a longer lifespan than animal, fruit, and vegetable products. Dry-beans plants can also fix nitrogen in the soil and air [2]. Production and harvest area for dry beans 2020 is 27.5 metric tons and 34.8 hectares. Dry-beans production has increased by 60%, and harvested area has increased by 36% since 1990 [4].

Choosing the type of dry-beans as a processed food ingredient requires precision. Manual processes certainly require physical and visual stability. If the number of types of beans that must be identified is large, a computerized system is necessary. Computer vision is a field that can fulfill this role—research using computer vision on the classification and identification of types of beans using Koklu public data. The total data is 13,611 grains with seven different types of beans. Data was split using 10-fold cross-validation. Classification uses machine learning methods: Multi-layer perceptron (MLP), Support vector Machine (SVM), k-nearest Neighbor (kNN), and Decision Tree (DT). The test results show that the accuracy is 0.9173, 0.9313, 0.8792, and 0.9252, respectively [5].

Other research using the Koklu dataset uses random undersampling. The machine learning classification methods include Logistic Regression, Random Forest, XGBoost, and CatBoost. Test results show the best accuracy using Xboost with 0.938 [6].

Subsequent research used the same beans dataset with machine learning classification methods, including Multinomial naïve Bayes, Support vector Machine, Decision Tree, Random Forest, Voting Classifier, and Artificial neural network. Experimental results show an accuracy between 0.8835 and 0.9361 [7]. Other research using k-nearest neighbor, Decision Tree, SVM, and MLP produces an accuracy of 0.9030, 0.9083, 0.9223, and 0.9249. The study used the same dataset from

Koklu [8].

The results of previous research still need to improve performance. For this reason, this research carried out stages such as balanced data for each class and hyperparameter tuning to optimize classification results.

2. METHODS

This research has stages including Exploratory Data Analysis (EDA), preprocessing by carrying out a balanced dataset, and classification using Decision Tree and Random Forest. Apart from that, carry out optimization using randomized search. The complete steps are shown in Figure 1.

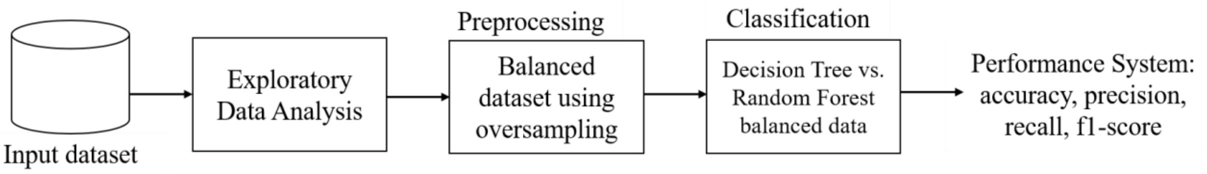

Input dataset
Exploratory Data Analysis
Preprocessing
Balanced dataset using oversampling
Classification
Decision Tree vs. Random Forest balanced data
Performance System: accuracy, precision, recall, f1-score

Figure 1. Proposed System

A. Input Dataset

The Koklu dry-beans data has 13,611 rows, 16 geometric features, and beans species labels. There are seven classes of dry-beans: Barbunya, Bombay, Cali, Dermason, Horoz, Seker, and Sira. Each species has a different amount of data. The amount of data in each class is shown in Table 1 [5].

Table 1. Data rows each class

Class	Data Rows
Dermason	3546
Sira	2636
Seker	2027
Horoz	1928
Cali	1630
Barbunya	1322
Bombay	522

The public data used has imbalanced data for each class. The class with the highest data is Dermason 3,546 and the lowest is Bombay 522.

B. Exploratory Data Analysis (EDA)

EDA aims to determine the characteristics and analysis of data. This stage is carried out before modeling occurs. Generally, EDA give information about [9], [10]:

1. The total amount of data, the number of classes, the amount of data for each class, and the number of features.
2. Data type for each feature. The data type can be numeric or categorical
3. Missing value. In the data, are there any features that have null values?
4. Data duplication. How much data duplication does there exist? Drop duplicated data
5. Correlation between features. What is the degree of correlation between features? A high correlation indicates a close relationship between features.
6. Data outliers. Are there any outlier data? Data that is significantly different in value from other data.

C. Balanced dataset with Oversampling

The amount of data in each class is different in the beans dataset. The smallest category is Bombay, with 522 data, while Dermason has 3546 data. Small amounts of data have the effect of less learning, while large amounts of data can have better learning. This, of course, causes an imbalance in learning between classes. Classes with more data can perform better recognition, while classes with small data do the opposite.

Therefore, it is necessary to balance data between classes so the system can carry out the same learning for each category. Oversampling is a method to overcome class imbalance. Data in small classes is increased by randomly doubling existing data [11] – [13]. Oversampling visualization is shown in Figure 2.

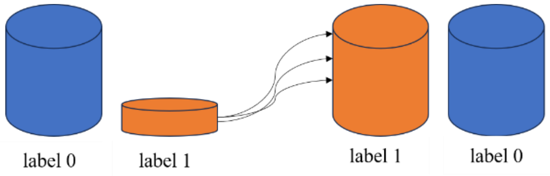


Figure 2. Oversampling

D. Classification using Decision Tree and Random Forest (RF)

Decision Tree is a supervised learning that use for classification and regression. It has hierarchical model that consist of root node, branches, and leaf nodes. The equations used are generally information gain and entropy. This is to determine the features that will become root nodes, branches, and leaf nodes. The commonly used Decision Tree models are ID3, C4.5, and C5.0 [14], [15].

A random forest consists of multiple trees. Random forest is a method that uses ensemble learning techniques. Ensembles combine various models. There are two types of ensemble: bagging and boosting. Bagging performs multiple models in parallel, and the final output is based on majority voting. Random Forest is included in the bagging principle. The Random Forest algorithm can be described as follows [12], [16]–[18]:

1. Select a random sample from the provided dataset.
2. Create a Decision Tree for each selected sample. Then, you will get the prediction results from each Decision Tree created.
3. A voting process is carried out for each prediction result. For classification problems, use the modus (the value that occurs most often).
4. The algorithm will choose the prediction result that has been selected the most (most votes) as the final prediction.

RF has a characteristic: firstly, not all attributes/features/variables are used for each tree. Every tree is different. Second, the feature space is reduced because not all features are used in each tree. Third, work in parallel. Each tree is created with different data and attributes. Fourth, there is no need to split training and testing data in RF because there is always 30% of data not used by the decision tree. Fifth, it has stability because the results are based on majority voting or average [12].

E. Hyperparameter tuning in RF with randomized search

In machine learning, some optimizations occur to improve performance. One thing that can be done is by hyperparameter tuning. In conventional programming, each hyperparameter is tried one by one the existing combinations. The initial hyperparameters were tested with varying values.

The hyperparameters in RF include the number of trees, maximum features/attributes/variables, minimum number of leaves, criterion (entropy/gini impurity/log loss), and maximum leaf node on each tree. Various combinations of hyperparameters were tested one by one. Of course, this requires significant resources if many combinations of hyperparameter values exist.

One solution to overcome this problem is randomized search (RS). The RS technique selects a combination of values for each hyperparameter randomly. So, not all combinations of hyperparameter values are executed, as in Grid Search. Therefore, there is a reduction in the resources required by the system because not all combinations of hyperparameter values are used [19]–[21].

F. Performance system

System performance is measured using a confusion matrix. Because the data has more than two classes, it is included in multiclass classification. The confusion matrix for multiclass is shown in Figure 3 [22], [23].

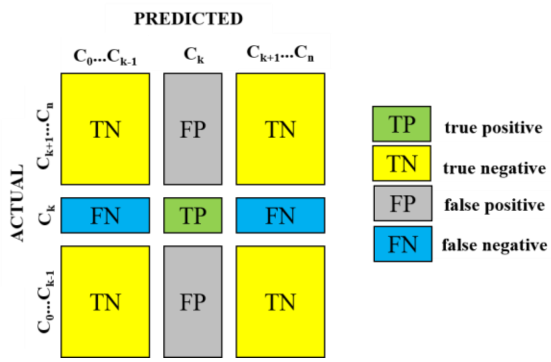


Figure 3. Multiclass Confusion matrix

3. RESULT AND DISCUSSION

A. Exploration Data Analysis

Exploration Data Analysis (EDA) aims to obtain initial information about the data used for experiments. Table 2 are the EDA results.

Table 2. EDA Results

EDA Component	EDA result
Data shape	(13611.17). Total rows 13,611 with 17 columns
Data summary	float64(14), int64(2), object(1)
Duplicate	68
Data after drop duplicate	Dermason 3546, Sira 2636, Seker 2027, Horoz 1860, Cali 1630, Red Mullet 1322, Bombay 522
Missing value	No
Feature correlation	Area, perimeter, MajorAxisLength, MinorAxisLength, ConvexArea, and EquivDiameter has high-value correlation

The initial data component comprises 13,611 rows with 17 columns (16 attributes and one label). For data types, most of the 14 features are float, two features are integer, and one label is object. Meanwhile, when checking duplicated data, there were 68 identical data and no missing values. Next, the feature correlation produces six features with high correlation values between 0.83 and 1.00.

B. Experiment Scenario

This research has four scenarios, as shown in Table 3.

Table 3. Experiment Scenario

No	Data	Classification method
1	Imbalanced Classes	Decision Tree
2	Imbalanced Classes	Random Forest
3	Balanced Classes	Decision Tree
4	Balanced Classes	Random Forest

The scenario consists of four methods: imbalanced and balanced classes with a Decision Tree and imbalanced and balanced classes with a Random Forest. Decision Tree is used as a comparison because the random forest backbone is a tree. For the number of trees (n_estimators used are 50,75,100,150, 200 and 300)

C. Result

The data in the testing scenario consists of two parts, namely training and testing, with a percentage of 70:30. Total data after drop duplicated 13,543. For training data, 9,480, and for testing data, 4,063. The results of testing using a Decision Tree with imbalanced and balanced classes are shown in Tables 4a, 4b and Figures 4a and 4b

Table 4a. Testing Result of Decision Tree with Imbalance Classes

	Barbunya	**Bombay**	**Ali**	**Dermason**	**Horoz**	**Seker**	**Sira**	**Acc.**	**Micro acc.**	**Weighted avg.**
Prec	0.8837	1.0	0.9186	0.8978	0.9278	0.9092	0.8118	0.8915	0.9070	0.8918
Rec.	0.9015	1.0	0.9070	0.8836	0.9295	0.9078	0.8281	0.8915	0.9081	0.8915
f1-score	0.8925	1.0	0.9128	0.8902	0.9286	0.9085	0.8198	0.8915	0.9075	0.8916
support	396	161	473	1065	553	618	797	0.8915	4063	4063

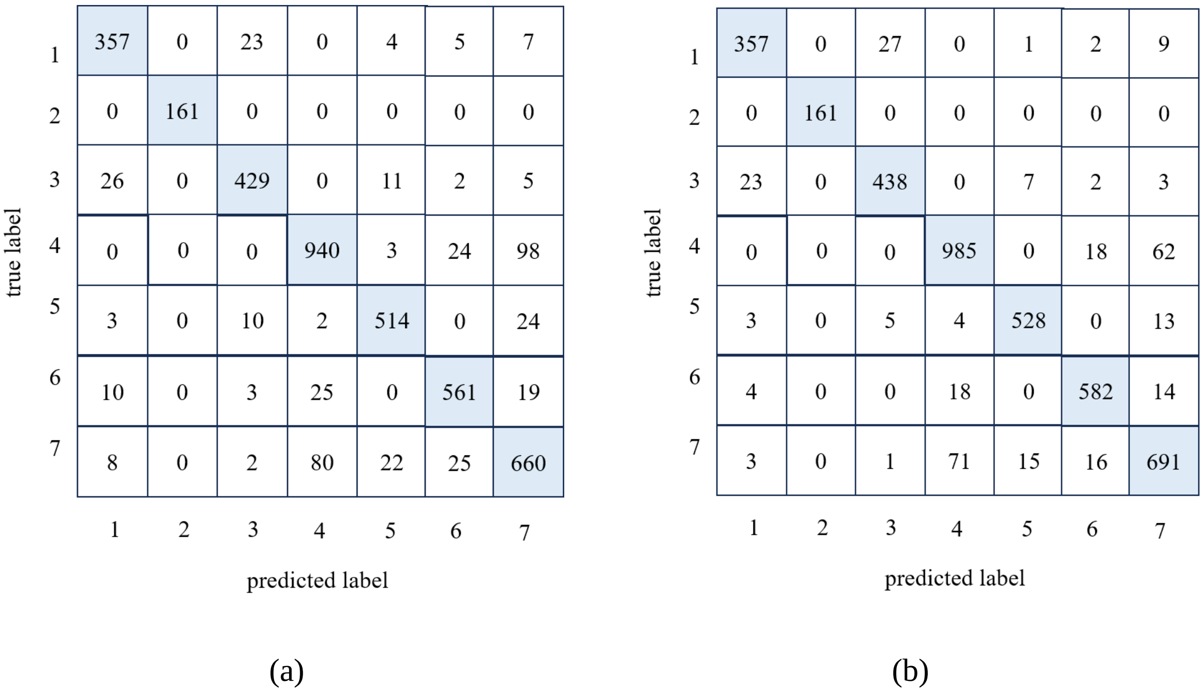
Table 4a shows the results of Decision Tree classification testing with imbalance classes. The test results show an accuracy of 0.8915, an average precision of 0.9070, an average recall of 0.9081, and an average f1-score of 0.9075. Meanwhile, the weighted average is between 0.8915 to 0.8918. The highest classification results were in the Bombay class, while the lowest were in the Sira class.

Table 4b. Testing Result of Random Forest with Imbalance Classes

	Barbunya	**Bombay**	**Ali**	**Dermason**	**Horoz**	**Seker**	**Sira**	**Acc.**	**Micro acc.**	**Weighted avg.**
Prec	0.9154	1.0	0.9299	0.9137	0.9583	0.9387	0.8725	0.9210	0.9326	0.9210
Rec.	0.9015	1.0	0.9060	0.9249	0.9248	0.9417	0.8670	0.9210	0.9308	0.9210
f1-score	0.9084	1.0	0.9280	0.9193	0.9565	0.9402	0.8697	0.9210	0.9317	0.9210
support	396	161	473	1065	553	618	797	0.9210	4063	4063

Table 4b are the results of the Random Forest imbalance classes classification. Testing accuracy up to 0.9210, average precision 0.9326, average recall 0.9308, and average f1-score 0.9317. Meanwhile, the weighted average is between 0.9210. The highest classification results were in the Bombay class, while the lowest were in the Sira class.

Figure 4 shows the confusion matrix from test results using Decision Tree and Random Forest with Imbalance Classes.



(a)

true label \ predicted label	1	2	3	4	5	6	7
1	357	0	23	0	4	5	7
2	0	161	0	0	0	0	0
3	26	0	429	0	11	2	5
4	0	0	0	940	3	24	98
5	3	0	10	2	514	0	24
6	10	0	3	25	0	561	19
7	8	0	2	80	22	25	660

(b)

true label \ predicted label	1	2	3	4	5	6	7
1	357	0	27	0	1	2	9
2	0	161	0	0	0	0	0
3	23	0	438	0	7	2	3
4	0	0	0	985	0	18	62
5	3	0	5	4	528	0	13
6	4	0	0	18	0	582	14
7	3	0	1	71	15	16	691

Figure 4. Confusion matrix of imbalanced classes using (a) Decision Tree (b) Random Forest 1=Barbunya, 2=Bombay, 3=Cali, 4=Dermason, 5=Horoz, 6=Seker, 7=Sira

The confusion matrix in Figure 4 shows that all 161 Bombay class test data were classified correctly. Meanwhile, the class that has the lowest classification level is Sira. Only 660 of the 797 data were classified correctly in testing using a decision tree. Apart from that, eight Sira data are classified as Barbunya, two as Cali, 80 Dermason data, 22 Horoz, and 25 Seker. Meanwhile, Random Forest classification resulted in 691 correctly classified, three barbunya, one cali, 71 dermason, 15 horoz, and 16 seker.

Next for the Table 5 (a) and 5 (b) are the testing result using Decision Tree and Random Forest with balance data between classess.

Table 5a. Testing Result of Decision Tree with Balance Classes

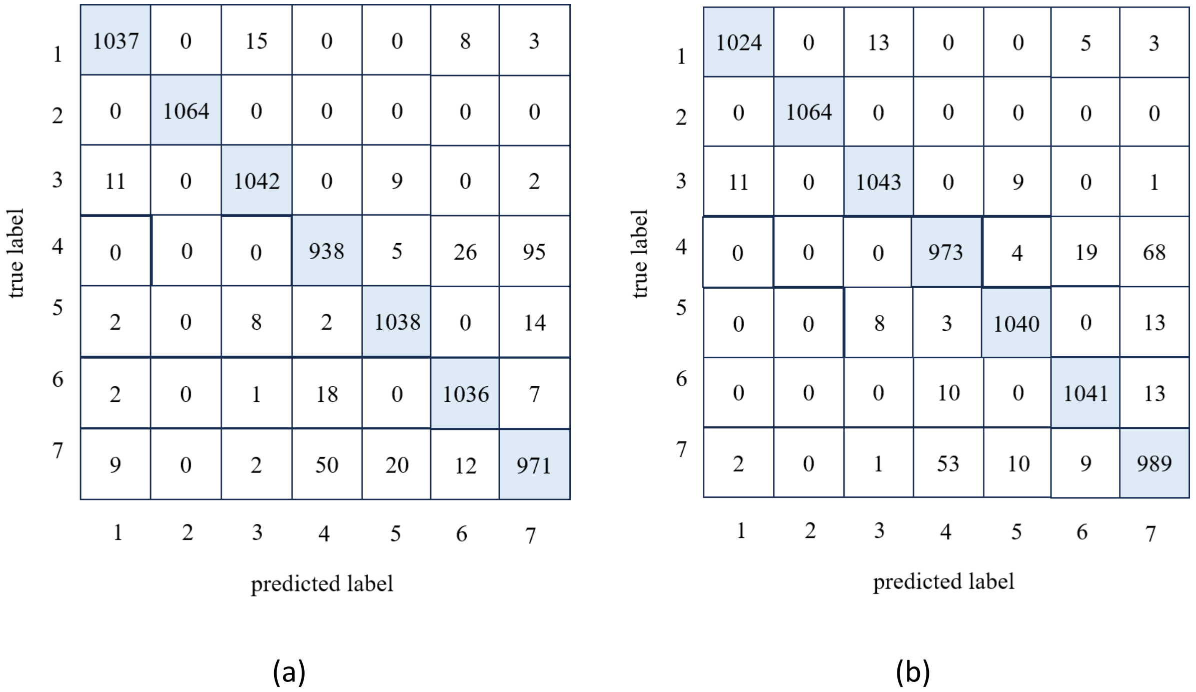
	Barbunya	**Bombay**	**Ali**	**Dermason**	**Horoz**	**Seker**	**Sira**	**Acc.**	**Micro acc.**	**Weighted avg.**
Prec.	0.9774	1.0	0.9766	0.9306	0.9683	0.9575	0.9992	0.9569	0.9569	0.9569
Rec.	0.9755	1.0	0.9793	0.8818	0.9756	0.9737	0.9126	0.9569	0.9569	0.9569
f1-score	0.9765	1.0	0.9775	0.9054	0.9719	0.9655	0.9007	0.9569	0.9568	0.9568
support	1063	1064	1064	1064	1064	1064	1064	0.9569	7447	7447

Table 5a shows the results of Decision Tree classification testing with balanced classes. The test results show an accuracy of 0.9569, an average precision of 0.9569, an average recall of 0.9569, and an average f1-score of 0.9568. Meanwhile, the weighted average is between 0.9568 to 0.9569. The highest classification results were in the Bombay class, while the lowest were in the Sira class.

Table 5b. Testing Result of Random Forest with Balance Classes

	Barbunya	**Bombay**	**Ali**	**Dermason**	**Horoz**	**Seker**	**Sira**	**Acc.**	**Micro acc.**	**Weighted avg.**
Prec	0.9877	1.0	0.9793	0.9365	0.9784	0.9693	0.9098	0.9658	0.9659	0.9659
Rec.	0.9802	1.0	0.9803	0.9145	0.9774	0.9784	0.9295	0.9658	0.9658	0.9658
f1-score	0.9839	1.0	0.9798	0.9253	0.9779	0.9738	0.9196	0.9658	0.9658	0.9658
support	1063	1064	1064	1064	1064	1064	1064	0.9658	7447	7447

Table 5b shows the Random Forest classification testing results with balanced classes. The test results show an accuracy of 0.9658, an average precision of 0.9659, an average recall of 0.9658, and an average f1-score of 0.9658. Meanwhile, the weighted average is between 0.9658 to 0.9659. The highest classification results were in the Bombay class, while the lowest were in the Sira class.



(a)

true label	1	2	3	4	5	6	7
1	1037	0	15	0	0	8	3
2	0	1064	0	0	0	0	0
3	11	0	1042	0	9	0	2
4	0	0	0	938	5	26	95
5	2	0	8	2	1038	0	14
6	2	0	1	18	0	1036	7
7	9	0	2	50	20	12	971

predicted label

(b)

true label	1	2	3	4	5	6	7
1	1024	0	13	0	0	5	3
2	0	1064	0	0	0	0	0
3	11	0	1043	0	9	0	1
4	0	0	0	973	4	19	68
5	0	0	8	3	1040	0	13
6	0	0	0	10	0	1041	13
7	2	0	1	53	10	9	989

predicted label

Figure 5. Confusion matrix of balanced classes using (a) Decision Tree (b) Random Forest 1=Red Beans, 2=Bombay, 3=Cali, 4=Dermason, 5=Rooster, 6=Candy, 7=Sira

The confusion matrix in Figure 5 shows that as many as 1,064 Bombay class test data were classified correctly. Meanwhile, the class that has the lowest classification level is Sira. Only 971 out of 1064 data were classified correctly in testing using a decision tree. Apart from that, nine Sira data are classified as Barbunya, two as Cali, 50 Dermason, 20 Horoz, and 12 Seker. Meanwhile, Random Forest classification resulted in 989 correctly classified, two barbunya, one cali, 53 dermason, ten horoz, and nine seker.

D. Discussion

The proposed method uses balanced data with oversampling, classification using Decision Tree, and Random Forest. The test results show that classification using Random Forest with balanced data achieves better results than Decision Tree. Random Forest classification with oversampling obtained an accuracy of 0.9658, while Decision Tree with oversampling reached 0.9569.

In another part, hyperparameter tuning with Randomized Search uses various values for the number of trees. Tuning allows all variations of the number of trees to be run simultaneously rather than tested individually. The results of the Randomized Search show that the optimal number of trees is 300.

Initialize the number of trees:

```
param_grid = {'n_estimators': [50, 75,100, 150, 200,300]}
```

Output results:

```
Best Parameter: {'n_estimators': 300}
```

In the final section, we compare the proposed method with previous research, which used the same drybeans data from Koklu. The comparison results are shown in Table 6.

Table 6. Comparison with previous research

Method	**Accuracy**	**Reference**
SVM	0.9313	[5]
CatBoost	0.9380	[6]
MLP	0.9457	[23]
ANN	0.9361	[7]
SVM	0.9249	[8]
DT+oversampling	0.9569	(proposed method)
RF+oversampling	0.9658	(proposed method)

4. Conclusion

A classification system for beans has been created using the Decision Tree and Random Forest methods with oversampling balance classes. The performance of the Decision Tree testing results shows accuracy, precision, recall, and f1-score of 0.9569. Meanwhile, the Random Forest test results showed accuracy, precision, recall, and f1-score of 0.9658.

Acknowledgment

This research was funded by the Penelitian Mandiri, University of Trunojoyo Madura, National Collaborative Research Scheme 2023.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interests.

REFERENCES

[1] R. Ekafitri, R. Isworo, Pemanfaatan kacang-kacangan sebagai bahan baku sumber protein untuk pangan darurat, Pangan, 23 (2014), 134-144.

[2] S.K. Sathe, Beans, overview, in: Reference Module in Food Science, Elsevier, 2016. https://doi.org/10.1016/B978-0-08-100596-5.00033-0.

[3] E.B. Nchanji, O.C. Ageyo, Do common beans (Phaseolus vulgaris L.) promote good health in humans? a systematic review and meta-analysis of clinical and randomized controlled trials, Nutrients. 13 (2021), 3701. https://doi.org/10.3390/nu13113701.

[4] M.A. Uebersax, K.A. Cichy, F.E. Gomez, et al. Dry beans (Phaseolus vulgaris L.) as a vital component of sustainable agriculture and food security-A review, Legume Sci. 5 (2022), e155. https://doi.org/10.1002/leg3.155.

[5] M. Koklu, I.A. Ozkan, Multiclass classification of dry beans using computer vision and machine learning techniques, Computers Electron. Agric. 174 (2020), 105507. https://doi.org/10.1016/j.compag.2020.105507.

[6] S. Krishnan, S.K. Aruna, K. Kanagarathinam, et al. Identification of dry bean varieties based on multiple attributes using catboost machine learning algorithm, Sci. Program. 2023 (2023), 2556066. https://doi.org/10.1155/2023/2556066.

[7] G. Słowiński, Dry beans classification using machine learning, in: Proceedings of the 29th International Workshop on Concurrency, Specification and Programming, Vol. 1613, CS & P'21 (2021), p. 0073.

[8] R.M. Dellosa, Determining the classification of dry beans using WEKA, Int. J. Biosci. 23 (2023), 81-92. https://doi.org/10.12692/ijb/23.1.81-92.

[9] M. Komorowski, D.C. Marshall, J.D. Salciccioli, et al. Secondary analysis of electronic health records, Springer, Cham, 2016. https://doi.org/10.1007/978-3-319-43742-2.

[10] A. Unwin, Exploratory data analysis, in: International Encyclopedia of Education, Elsevier, 2010: pp. 156-161. https://doi.org/10.1016/B978-0-08-044894-7.01327-0.

[11] A. Chkifa, M. Dolbeault, Randomized least-squares with minimal oversampling and interpolation in general spaces, preprint, (2023). http://arxiv.org/abs/2306.07435.

[12] T. Wongvorachan, S. He, O. Bulut, A comparison of undersampling, oversampling, and SMOTE methods for

dealing with imbalanced classification in educational data mining, Information. 14 (2023), 54. https://doi.org/10.3390/info14010054.

[13] H.N. Haliduola, F. Bretz, U. Mansmann, Missing data imputation in clinical trials using recurrent neural network facilitated by clustering and oversampling, Biometrical J. 64 (2022), 863-882. https://doi.org/10.1002/bimj.202000393.

[14] B. Charbuty, A. Abdulazeez, Classification based on decision tree algorithm for machine learning, J. Appl. Sci. Technol. Trends. 2 (2021), 20-28. https://doi.org/10.38094/jastt20165.

[15] J. Patalas-Maliszewska, H. Łosyk, M. Rehm, Decision-tree based methodology aid in assessing the sustainable development of a manufacturing company, Sustainability. 14 (2022), 6362. https://doi.org/10.3390/su14106362.

[16] M. Pal, S. Parija, Prediction of heart diseases using random forest, J. Phys.: Conf. Ser. 1817 (2021), 012009. https://doi.org/10.1088/1742-6596/1817/1/012009.

[17] D.M. Raza, D.B. Victor, Data mining and region prediction based on crime using random forest, in: 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), IEEE, Coimbatore, India, 2021: pp. 980–987. https://doi.org/10.1109/ICAIS50930.2021.9395989.

[18] M. Avand, S. Janizadeh, S.A. Naghibi, et al. A comparative assessment of random forest and k-nearest neighbor classifiers for gully erosion susceptibility mapping, Water. 11 (2019), 2076. https://doi.org/10.3390/w11102076.

[19] P. Liashchynskyi, P. Liashchynskyi, Grid search, random search, genetic algorithm: a big comparison for NAS, preprint, (2019). http://arxiv.org/abs/1912.06059.

[20] L. Li, A. Talwalkar, Random search and reproducibility for neural architecture search, in: Proceedings of The 35th Uncertainty in Artificial Intelligence Conference, PMLR 115:367-377, 2020.

[21] M. Andriushchenko, F. Croce, N. Flammarion, et al. Square attack: a query-efficient black-box adversarial attack via random search, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision - ECCV 2020, Springer International Publishing, Cham, 2020: pp. 484–501. https://doi.org/10.1007/978-3-030-58592-1_29.

[22] S. Haghighi, M. Jasemi, S. Hessabi, et al. PyCM: Multiclass confusion matrix library in Python, J. Open Source Softw. 3 (2018), 729. https://doi.org/10.21105/joss.00729.

[23] S. Ruuska, W. Hämäläinen, S. Kajava, et al. Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle, Behav. Processes. 148 (2018), 56-62. https://doi.org/10.1016/j.beproc.2018.01.004.