Recursive Journal of Informatics
Vol. 3, No. 1, March 2025

p-ISSN 2963-5551 https://journal.unnes.ac.id/journals/index.php/rji e-ISSN 2986-6588

Random Forest Algorithm Optimization using K-Nearest Neighbor and SMOTE on Diabetes Disease

Syuja Zhafran Rakha Krishandhie[1], Aji Purwinarko[2]
[1,2]Computer Science Department, Faculty of Mathematics and Natural Sciences,
Universitas Negeri Semarang, Indonesia

Abstract. Diabetes is a chronic disease that can cause long-term damage, dysfunction and failure of various organs in the body. Diabetes occurs due to an increase in blood sugar (glucose) levels exceeding normal values. Early diagnosis of diseases is crucial for addressing them, especially in the case of diabetes, which is one of the chronic illnesses.
Purpose: This study aims to find out how the implementation of the K-Nearest Neighbor algorithm with the Synthetic Minority Oversampling Technique (SMOTE) in optimizing Random Forest algorithm for diabetes disease prediction.
Methods/Study design/approach: This study uses the Pima Indian Diabetes Dataset, the random forest algorithm for the classification, k-nearest neighbor for optimization, and SMOTE for the minority class oversampling.
Result/Findings: The prediction accuracy of the model using SMOTE and k-nearest neighbor is 92,86%. Meanwhile, the model that does not use SMOTE and k-nearest neighbor obtains an accuracy of 83,03%.
Novelty/Originality/Value: This research shows that the use of random forest algorithm with k-nearest neighbor and SMOTE gives better accuracy than without using k-nearest neighbor and SMOTE.

Keywords: Diabetes Disease, Random Forest, K-Nearest Neighbor, SMOTE.
Received February 15, 2024 / **Revised** February 26, 2024 / **Accepted** March 27, 2025

INTRODUCTION

Diabetes is a chronic disease characterized by high blood sugar. It may cause many complicated diseases like stroke, kidney failure, heart attack, etc [1] There are several factors for developing diabetes like genetic susceptibility, body weight, food habits and sedentary lifestyle. Undiagnosed diabetes may result in very high blood sugar levels referred as hyperglycemia which can lead to complication like diabetic retinopathy, nephropathy, neuropathy, cardiac stroke and foot ulcer. Early detection of diabetes is very important to improve the quality of life of patients and enhance their life expectancy [2].

Diabetes is a serious chronic condition associated with diffuse complications and an increased risk of premature death, imposing enormous financial pressure on national health care systems and national economies. In 2017 (8th edition of the IDF Diabetes Atlas) 4.0 million people were estimated to have died from diabetes and its complications. About half (46.1%) of these deaths were in adults under the age of 60 years, the working age group [3]. This figure makes diabetes a disease that must be diagnosed and treated immediately. However, there is diabetes attributed to statistical data that can be analyzed using a machine learning algorithm.

In recent years, there has been a resurgence of attention surrounding machine learning (ML) and artificial intelligence, propelled by the substantial and continually expanding volumes of data, enhanced computational capabilities, and advancements in learning algorithms [4] Exploring a variety of options is necessary to construct an optimal ML model. This involves designing the ideal model architecture and determining the optimal configuration of hyperparameters, a process known as hyperparameter tuning. Tuning hyperparameters is widely regarded as a crucial step in developing an effective ML model [5].

The Random Forest classifier is an ensemble algorithm. Random Forest has emerged as a quite useful algorithm that can handle the feature selection issue even with a higher number of variables [6]. This implies

[1]*Corresponding author.
Email addresses: zjaksu@gmail.com (Krishandie)
DOI: 10.15294/rji.v3i1.1576

that it consists of more than one algorithm. Usually In this case, it consists of several decision tree algorithms. Random forest builds up an entire forest from several uncorrelated and random Decision Trees during training segment [7]. However, datasets used for classification often suffer from imbalanced class distribution, leading to accuracy that tends to favor the majority class. To address this, some methods are needed to overcome data imbalance. One such method is the Synthetic Minority Oversampling Technique (SMOTE).

K-Nearest Neighbor (KNN) algorithm was first described in 1967 as a decision rule for assigning the classification of the nearest of a set of previously classified instances to an unclassified sample instance. This classification method is simple, does not make big assumptions, and is particularly useful in pattern recognition. The algorithm is based on the distance between two instances, which represents their similarity [8].

SMOTE replications and randomly increases the minority class thereby effectively balancing the class distribution. It relies on synthesizing new minority instances from existing ones and uses linear interpolation to generate virtual training records[9]. SMOTE employs an iterative search and selection approach. It generates new artificial minority class samples by selecting a specified number of samples among the k nearest neighbors. The threshold value is contingent upon the desired quantity of synthetic minority samples to be created [10].

METHODS

This research focuses on the application of KNN and SMOTE to optimize random forest classification as an ensemble algorithm. Ensemble learning techniques have achieved state-of-the-art performance in diverse machine learning applications by combining the predictions from two or more base models [11]. Before the classification process using random forest, the data preprocessing stage is carried out. At the data preprocessing stage, data balancing is carried out using SMOTE. Next, both algorithm random forest and KNN are tuned using k-fold cross validation. After that, the data set is divided into two parts, namely training data and testing data. The flowchart of the proposed method is illustrated in Figure 1.

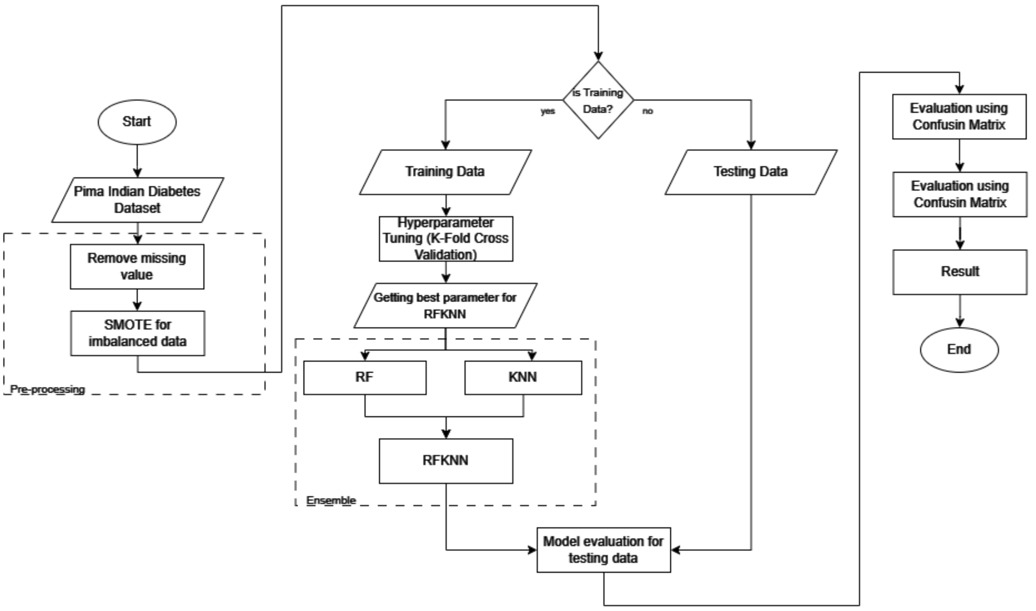
Start
Pima Indian Diabetes Dataset
Remove missing value
SMOTE for imbalanced data
Pre-processing
Is Training Data?
yes
no
Training Data
Testing Data
Hyperparameter Tuning (K-Fold Cross Validation)
Getting best parameter for RFKNN
RF
KNN
RFKNN
Ensemble
Model evaluation for testing data
Evaluation using Confusin Matrix
Evaluation using Confusin Matrix
Result
End

Figure 1. Flowchart of proposed method

Data Research

The dataset is originally from the National Institute of Diabetes and Digestive and Kidney Disease. The data is collected from the website www.kaggle.com. The objective of the dataset is to diagnostically predict wether or not a patient has diabetes, based on certain diagnostic measurement included in the dataset. The total number of datasets is 789. The attributes of match result statistics used in the research are presented in Table 1.

Table 1. The pima indian diabetes dataset attributes used

No	Attribute	Type	Description
1	Pregnancies	Numeric	Number of times pregnant
2	Glucose	Numeric	Glucose
3	BloodPressure	Numeric	Diastolic blood pressure
4	SkinThickness	Numeric	Triceps skin fold thickness
5	Insulin	Numeric	2-Hour serum insulin
6	BMI	Numeric	Body mass index
7	DiabetesPedigreeFunction	Numeric	Diabetes pedigree function
8	Age	Numeric	Age
9	Outcome	Integer	Class Variable 0 or 1

Data Preprocessing

Preprocessing is required so that the property of interest can be predicted correctly [12]. In the data preprocessing stage, there is a process to overcome the missing value in the dataset. The method used to overcome is using mean, mean work by taking the average of all the data contained in a column and then replacing it with the missing column. There is also a method to overcome data imbalance using SMOTE. SMOTE increases the number of data instances by generating random synthetic data of minority class from its nearest neighbours using Euclidean distance. New instances become like the original data because they are generated based on original features [13].

Data Split

Data sets featured in publications are carefully curated and partitioned into training, testing, and validation subsets by domain experts. As a result, machine learning models often exhibit strong performance on these manually prepared hand-crafted datasets [14]. The dataset splitting process divides the data into two parts: training data and testing data, with an 80:20, 75:25, and 70:30 ratio. The data splitting is performed with random state shuffling to ensure consistency in the calculated values [15].

Hyperparameter Tuning

The hyperparameter tuning method used in this research is K-Fold Cross Validation. Cross-validation using randomized subsets of data—known as k-fold cross-validation—is a powerful means of testing the success rate of models used for hyperparameter tuning [16]. This method works by taking specific samples from the data set on which the model is not trained. This method is also used to protect the model from overfitting. In the cross-validation process, the original data sample is randomly divided into subsets of equal or close size, then training data for iteration or repetition and data is performed, so that in each iteration a different fold of data is required for validation, while the remaining K1 folds are used for training [17].

Classification

After balancing the data and tuning both alrogithm, the classification process using random forest is performed. The model is built using ensemble methods. Ensemble method is divided into 3 method it is bagging, boosting, and stacking [18]. The research will use stacking ensemble for random forest and knn. Stacking ensemble is one of the well-known ensemble methods, where there are two levels of classifiers, namely the base classifier (level 0) and the meta-classifier (level 1), which are used to predict the output variable. Stacking ensemble combines the output of the base classifier by using the meta-classifier to learn the relationship between the model output and the actual output [19].

Model Testing and Evaluation

The classification results on the training data are then tested using testing data. The model testing process is carried out using a confusion matrix. A clear and precise evaluation of a machine learning algorithm is essential for classifier design and enhancing performance. In multi-class classification tasks, where each instance is assigned to only one class, the confusion matrix serves as a valuable tool for assessing performance by quantifying classification overlaps [20].

RESULT AND DISCUSSION

Preprocessing

At the preprocessing stage, the class that contains missing value will be filled by means by measured the average of each class. The value of mean in every class can be seen in Table 2.

Table 2. Means of each class

No	Attribute	Mean
1	Glucose	121
2	BloodPressure	72
3	SkinThickness	29
4	Insulin	155
5	BMI	32

At the preprocessing stage, the minority class oversampling process is carried out using SMOTE. This process uses the entire dataset before the classification process. A comparison of the amount of data before and after SMOTE can be seen in Table 3.

Table 3. Total data for each class before and after SMOTE

No	Process	1	0
1	Before SMOTE	268	500
2	After SMOTE	454	500

Data Split

The data split process carried out in this research is divided into the training data and testing data obtained from the dataset with the ratio of training data to testing data being 70:30, 75:25, 80,30. The use of random state is applied to this process to produce random values that remain the same in each execution. The total dataset which previously amounted to 768 data after SMOTE was carried out became 954 data. Comparison of train data and test data can be seen in Table 4.

Table 4. Comparison of train data and test data

No	Precentage (%)	Data Train	Data Test
1	70:20	700	254
2	75:25	750	204
3	80:20	800	154

Classification

The classification is using ensemble classification using random forest and knn (proposed method). In the proposed method, hyperparameter tuning is performed. The tuning involves using a k-fold cross validation method in selecting the parameters, the k will be set to k=10. The k-fold parameter determines the number of decisions to be built in the random forest and knn. Meanwhile, the k parameter determines the number of parameters to be randomly selected at each node and selected the best for the process. In the k-fold cross validation process, the values of the 10-fold and the parameters accuracy used are indicated in Table 5 and Table 6.

Table 5. 10-fold cross validation random forest

fold	accuracy
1	0,7963
2	0,6852
3	0,7222
4	0,7407
5	0,6852
6	0,6852
7	0,7593
8	0,7736
9	0,6792
mean	0,7319

Table 6. 10-fold cross validation knn

fold	accuracy
1	0,7037
2	0,6667
3	0,5370
4	0,6852
5	0,7037
6	0,7222
7	0,6667
8	0,7170
9	0,6415
mean	0,6742

The best results from 10-fold cross validation for random forest is 0,7963 and for knn is 0,7222. The parameter that saved will be used in ensemble classification method. The stacking ensemble method for random forest and knn will have 2 classifiers, the basic classifier (level 0) and meta-classifier (level 1), knn will be the basic classifier and random forest will be the meta classifier. The description of stacking ensemble process is shown in Figure 2.

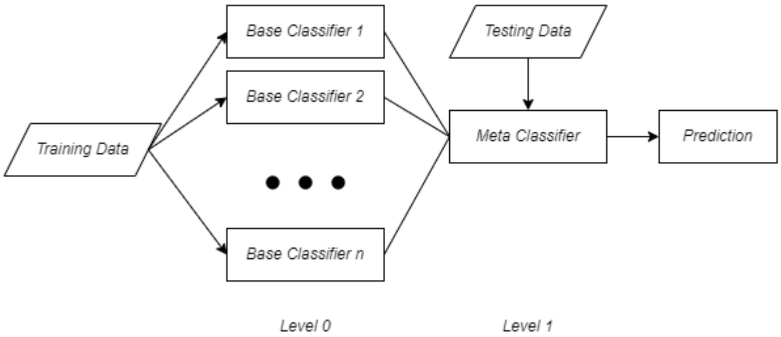

Figure 2. Stacking Ensemble

While the base classifier is already stacked, the ensemble generates a meta classifier Random Forest K-Nearest Neighbor (RFKNN). The RFKNN method will be tested using testing data from the preprocessing stage and evaluated using confusion matrix. The confusion matrix used to evaluate RFKNN algorithm will be shown in the Figure 3 below.

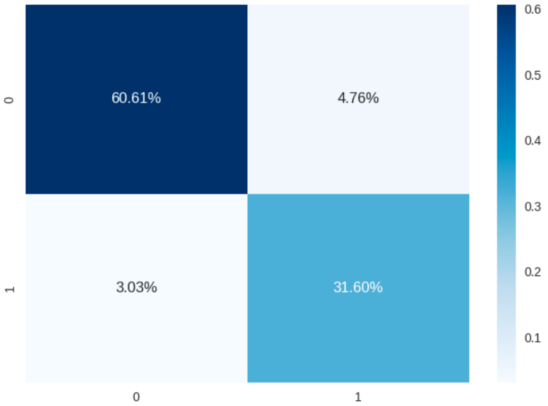

Figure 3. Confusion matrix of proposed method with 70:30 dataset ratio

Based on Figure 3, the accuracy of the proposed method obtained from data split with 70:30 ratio produces a confusion matrix with 0.9221 or 92.21% accuracy.

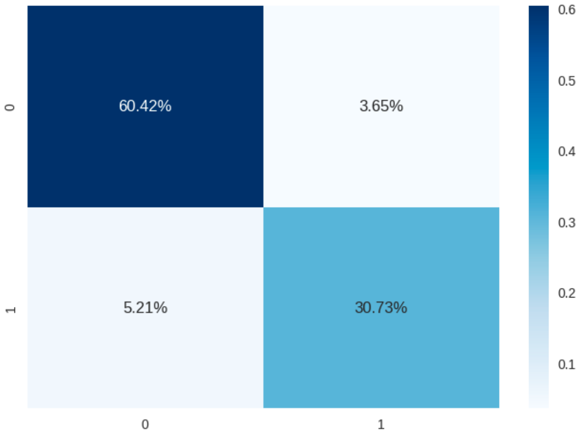

Figure 4. Confusion matrix of proposed method with 75:25 dataset ratio

Based on Figure 4, the accuracy of the proposed method obtained from data split with 75:25 ratio produces a confusion matrix with 0.9215 or 92.15% accuracy.

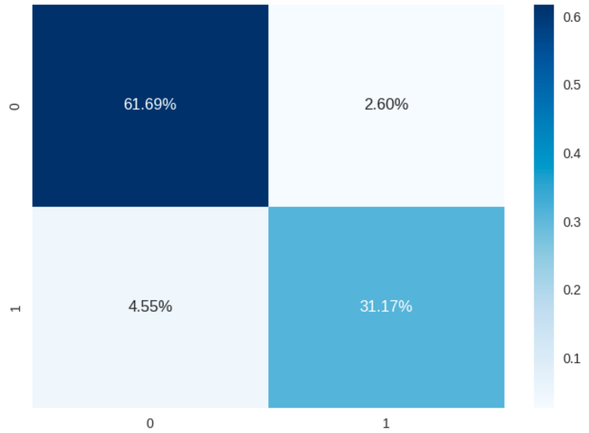

Figure 5. Confusion matrix of proposed method with 80:20 dataset ratio

Based on Figure 5, the accuracy of the proposed method obtained from data split with 80:20 ratio produces a confusion matrix with 0.9286 or 92.86% accuracy.

The following are the respective accuracies of the three classifications using different data ratio carried out, and the best split ratio for the diabetes disease classification is 80:20 ratio which generate 92,68% accuracy. The other results are shown in Table 6.

Table 6. Comparison of the classification's accuracy

Split Ratio	Accuracy
70:20	92,21%
75:25	92,15%
80:20	92,68%

CONCLUSION

Based on the research findings and discussions related to the implementation of KNN and SMOTE on the random forest algorithm using the Pima Indian Diabetes Dataset, several conclusions can be drawn. The application of SMOTE in the preprocessing stage effectively addresses the issue of imbalanced data within the dataset, ensuring that the classification process operates optimally without the presence of an imbalanced class. Additionally, the accuracy results obtained from modeling were tested through three different experiments, each involving a different ratio of training and testing data: 70%:30%, 75%:25%, and 80%:20%. Among these, the highest accuracy was achieved with an 80%:20% data split, where the RFKNN and SMOTE methods yielded an accuracy of 92.82% in predicting diabetes.

REFERENCES

[1] Md. Maniruzzaman, Md. J. Rahman, B. Ahammed, and Md. M. Abedin, "Classification and prediction of diabetes disease using machine learning paradigm," *Health Inf Sci Syst*, vol. 8, no. 1, p. 7, Dec. 2020, doi: 10.1007/s13755-019-0095-z.

[2] H. Kaur and V. Kumari, "Predictive modelling and analytics for diabetes using a machine learning approach," *Applied Computing and Informatics*, vol. 18, no. 1/2, pp. 90–100, Mar. 2022, doi: 10.1016/j.aci.2018.12.004.

[3] P. Saeedi *et al.*, "Mortality attributable to diabetes in 20–79 years old adults, 2019 estimates: Results from the International Diabetes Federation Diabetes Atlas, 9th edition," *Diabetes Res Clin Pract*, vol. 162, p. 108086, Apr. 2020, doi: 10.1016/j.diabres.2020.108086.

[4] S. Badillo *et al.*, "An Introduction to Machine Learning," *Clin Pharmacol Ther*, vol. 107, no. 4, pp. 871–885, Apr. 2020, doi: 10.1002/cpt.1796.

[5] L. Yang and A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice,” *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.

[6] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, “Selecting critical features for data classification based on machine learning methods,” *J Big Data*, vol. 7, no. 1, p. 52, Dec. 2020, doi: 10.1186/s40537-020-00327-4.

[7] P. Ghosh *et al.*, “Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms With Relief and LASSO Feature Selection Techniques,” *IEEE Access*, vol. 9, pp. 19304–19326, 2021, doi: 10.1109/ACCESS.2021.3053759.

[8] R. Garcia-Carretero, L. Vigil-Medina, I. Mora-Jimenez, C. Soguero-Ruiz, O. Barquero-Perez, and J. Ramos-Lopez, “Use of a K-nearest neighbors model to predict the development of type 2 diabetes within 2 years in an obese, hypertensive population,” *Med Biol Eng Comput*, vol. 58, no. 5, pp. 991–1002, May 2020, doi: 10.1007/s11517-020-02132-w.

[9] C. Azad, B. Bhushan, R. Sharma, A. Shankar, K. K. Singh, and A. Khamparia, “Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes mellitus,” *Multimed Syst*, vol. 28, no. 4, pp. 1289–1307, Aug. 2022, doi: 10.1007/s00530-021-00817-2.

[10] B. S. Raghuwanshi and S. Shukla, “SMOTE based class-specific extreme learning machine for imbalanced learning,” *Knowl Based Syst*, vol. 187, p. 104814, Jan. 2020, doi: 10.1016/j.knosys.2019.06.022.

[11] I. D. Mienye and Y. Sun, “A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects,” *IEEE Access*, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[12] P. Mishra, A. Biancolillo, J. M. Roger, F. Marini, and D. N. Rutledge, “New data preprocessing trends based on ensemble of multiple preprocessing techniques,” *TrAC Trends in Analytical Chemistry*, vol. 132, p. 116045, Nov. 2020, doi: 10.1016/j.trac.2020.116045.

[13] A. Ishaq *et al.*, “Improving the Prediction of Heart Failure Patients’ Survival Using SMOTE and Effective Data Mining Techniques,” *IEEE Access*, vol. 9, pp. 39707–39716, 2021, doi: 10.1109/ACCESS.2021.3064084.

[14] K. M. Kahloot and P. Ekler, “Algorithmic Splitting: A Method for Dataset Preparation,” *IEEE Access*, vol. 9, pp. 125229–125237, 2021, doi: 10.1109/ACCESS.2021.3110745.

[15] V. R. Joseph, “Optimal ratio for data splitting,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.

[16] B. G. Marcot and A. M. Hanea, “What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis?,” *Comput Stat*, vol. 36, no. 3, pp. 2009–2031, Sep. 2021, doi: 10.1007/s00180-020-00999-9.

[17] I. K. Nti, O. Nyarko-Boateng, and J. Aning, “Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation,” *International Journal of Information Technology and Computer Science*, vol. 13, no. 6, pp. 61–71, Dec. 2021, doi: 10.5815/ijitcs.2021.06.05.

[18] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, “Fake News Detection Using Machine Learning Ensemble Methods,” *Complexity*, vol. 2020, pp. 1–11, Oct. 2020, doi: 10.1155/2020/8885861.

[19] N. Kardani, A. Zhou, M. Nazem, and S.-L. Shen, “Improved prediction of slope stability using a hybrid stacking ensemble method based on finite element analysis and field data,” *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 13, no. 1, pp. 188–201, Feb. 2021, doi: 10.1016/j.jrmge.2020.05.011.

[20] M. Heydarian, T. E. Doyle, and R. Samavi, “MLCM: Multi-Label Confusion Matrix,” *IEEE Access*, vol. 10, pp. 19083–19095, 2022, doi: 10.1109/ACCESS.2022.3151048.