Implementation of Gentic Algorithm - Random Oversampling on the Random Forest Algorithm to Address Imbalanced Blood Donor Eligibility Data

Al Janatul Ulvivianti [1], Taghfirul Azhima Yoga Siswa [2*], Wawan Joko Pranoto [3]

[1] Universitas Muhammadiyah Kalimantan Timur, Samarinda, 75124, Indonesia

[*] Corresponding author: tay758@umkt.ac.id

Submitted: 29/01/2025 | Accepted: 01/03/2025 | Online: 05/03/2025 | doi: xxxx.xxxx.xxxx.xxxx/xxxxxx

Abstract:

Purpose: This study aims to improve the accuracy of blood donor classification by addressing data imbalance using machine learning techniques. Accurate classification of donor eligibility is crucial for maintaining a reliable blood supply. To achieve this, the research explores the integration of the Random Forest algorithm with the Genetic Algorithm (GA) for feature selection and optimization, alongside Random Oversampling (RO) for data balancing.

Methods: The research employs the Random Forest algorithm combined with GA for feature selection and optimization. Additionally, Random Oversampling is applied to handle the class imbalance in the dataset. The model's performance is evaluated using 10-Fold Cross Validation. The dataset used in this study consists of blood donor records from the Indonesian Red Cross (PMI) in Samarinda City for 2023–2024.

Results: The application of Random Oversampling significantly improved the model's accuracy, achieving 99.94%. However, the use of GA Feature Selection and GA Optimization independently did not result in notable improvements. Furthermore, when both techniques were applied together, the accuracy decreased to 98.78%.

Conclusions: The study confirms that Random Oversampling is highly effective in improving classification accuracy for blood donor eligibility. However, the integration of GA for feature selection and optimization did not yield additional benefits and even reduced accuracy when applied together. Future research could explore alternative feature selection and optimization methods to further enhance classification performance.

Keywords:

Blood Donation, Genetic Algorithm, Imbalanced Data, Random Forest, Random Oversampling

1. Introduction

Technology is rapidly evolving, especially in artificial intelligence and machine learning. Machine learning techniques are designed to enhance automatic detection capabilities. With these systems, the potential for misdiagnosis by medical personnel can be minimized, examinations can be conducted in a shorter time, and the results can be more detailed (Firdaus et al., 2020).

Blood is a vital component required by every living being. Its crucial role is to distribute oxygen and other necessary components to the body's cells (Veronica et al., 2024). Transferred blood can be in the form of whole blood or blood components, which is commonly done among teenagers to adults. The willingness to donate blood should start from adolescence to establish a habit and social responsibility, as blood is obtained from voluntary blood donors or replacement donors (Basri & Rahmita, 2021). In practice, not everyone who wants to donate blood can successfully do so. There are classification criteria used to determine whether a person is eligible to donate blood (Handayani et al., 2021).

One method that can be used to predict the accuracy of a blood donor dataset is the Random Forest method. Random Forest is a part of data mining techniques based on decision trees. This method can also enhance accuracy results (Efendi & Zyen, 2024). In medical classification applications, the Random Forest method has the ability to recognize the importance of each feature and provides ease of interpretation. This method stands out due to its superior performance, allowing users to understand the contribution of each feature to the classification results while also offering an easily interpretable model (Karomi, 2020).

The Genetic Algorithm (GA) is an evolutionary-based optimization algorithm inspired by the principle of natural selection in biology. In the context of Random Forest, GA is used to select the best features, including population initialization, selection, recombination, mutation, evaluation, and iteration. Random Optimization (RO) is an optimization method that utilizes random searches to find good solutions. In the context of Random Forest, RO is used to randomly search for hyperparameter combinations and evaluate their performance (Anjas Aprihartha et al., 2024).

Several previous studies have similarities with this research. First, a study conducted by (Wahono & Riana, 2020) showed that the Decision Tree C4.5 algorithm achieved a higher accuracy of 93.83% compared to the Naïve Bayes algorithm, which had an accuracy of 85.15%, and the K-Nearest Neighbors algorithm, which had an accuracy of 84.10%. Besides these accuracy values, Decision Tree C4.5 also excelled visually, producing a tree model that illustrates attribute relationships and achieving an AUC value of 0.978, while Naïve Bayes had an AUC value of 0.927, and K-Nearest Neighbors had an AUC value of 0.816. Second, a study by (Atmaja et al., 2018) found that private sector employees over the age of 26 were the most frequent blood donors based on decision tree results obtained through data mining using the C4.5 algorithm. Third, a study conducted by (Rivaldo et al., 2024) showed that the Chi-Square method identified four best features: humidity (rh_avg), rainfall (rr), maximum wind direction (ddd_x), and most frequent wind direction (ddd_car). The use of the Naïve Bayes algorithm with the SMOTE technique achieved an accuracy of 71.58%. However, after applying Chi-Square feature selection, accuracy dropped to 60.82%. This decline was due to the reduced number of minority classes after feature selection, indicating that the Chi-Square feature selection method was not effective in improving Naïve Bayes accuracy in high-dimensional datasets.

Based on the explanations provided, this study will implement the GA-RO technique to test blood donor samples using the Random Forest algorithm to address the issue of

imbalanced donor eligibility data. This research case is obtained from the Blood Donor Unit (UDD) of the Indonesian Red Cross (PMI) in Samarinda, located at Jl. Palang Merah Indonesia No.1, Samarinda Ulu District.

2. **Methods**

2.1 Research Object

The research object utilizes data from the Blood Donor Unit (UDD) of the Indonesian Red Cross (PMI), which includes records of both successful and unsuccessful blood donors. The dataset contains variables such as Donor ID, Age, Blood Type, Gender, Status, Hemoglobin Level, Blood Pressure, and Body Weight. The Blood Donor Unit (UDD) of PMI is located at Jl. Palang Merah Indonesia No.1, Samarinda Ulu District.

2.2 Research Procedure

The research procedure is a series of steps applied to collect data and solve problems in the study (Irfan Syahroni, 2022). The steps of the research are as follows:

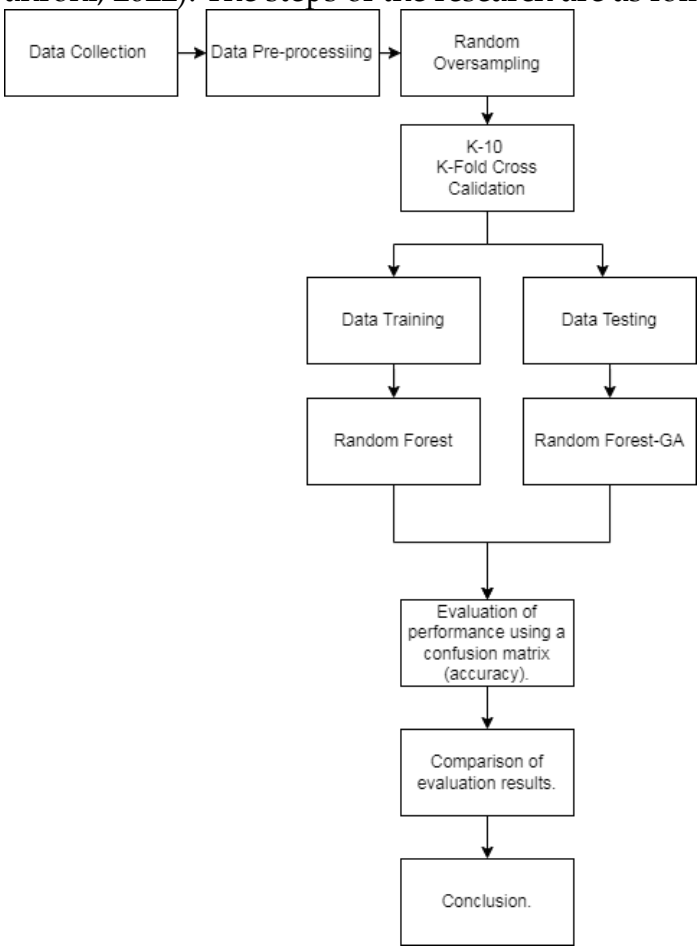

Figure 1 : Research Procedure

2.2.1 Problem Identification

The initial stage involves identifying the problem and determining the best method to address the imbalance in the blood donor data at the Blood Donor Unit (UDD) of the Indonesian Red Cross (PMI). This study applies the GA-RO method to enhance prediction accuracy on imbalanced data.

2.2.2 Data Collection

Data collection is the process of gathering research information from data sources, namely the research subjects or samples (Halim et al., 2023). The data used in this study is obtained from the Blood Donor Unit (UDD) of PMI. This dataset consists of various variables used in the classification process to determine whether a donor is eligible to donate blood.

2.2.3 Data Pre-processing

Data pre-processing is a crucial step since the collected data often contains noise, such as missing values, duplicate data, or other inconsistencies. Therefore, the data cleaning process is necessary to ensure the quality and accuracy of the data used for analysis (Ayuningtyas et al., 2024). The data used in this study is obtained from the Blood Donor Unit (UDD) of the Indonesian Red Cross (PMI) and consists of blood donor records that require further processing using data mining techniques to improve model accuracy. The pre-processing steps include data selection, which involves choosing relevant features from the dataset; data cleaning, which focuses on handling missing values, duplicates, and inconsistencies; and data transformation, which converts data into a suitable format for analysis.

2.2.4 Data Splitting

Before evaluation, the data needs to be divided into two main sets. The Training Set is used to train the machine learning model to recognize patterns and relationships within the data, while the Testing Set is used to test and evaluate the model's performance to ensure it can make accurate predictions on unseen data. In this study, the K-Fold Cross Validation technique with K=10 will be applied. K-Fold Cross Validation with K=10 is a highly effective method for evaluating machine learning models.

2.2.5 Modeling

Model development involves implementing the GA-RO method to handle data imbalance and optimizing parameters using the Random Forest algorithm to enhance the model's performance in classifying blood donor eligibility. The combination of these two methods is expected to produce a more effective model in predicting minority classes and improving overall classification accuracy

- Application Random Forest

Random Forest is an ensemble machine learning algorithm that consists of multiple decision trees. Each decision tree provides a prediction, and the final result is obtained using majority voting (for classification) or averaging (for regression).

The process of forming a decision tree in the Random Forest (RF) method is the same as in the Classification and Regression Tree (CART) method, except that pruning is not performed in RF. The Gini Index is used to select features at each internal node of the decision tree. The Gini Index value can be calculated as follows:

$$\text{Gini}(S_i) = 1 - \sum_{i=0}^{c-1} P_i^2 \quad (1)$$

Where P_i represents the relative frequency of class C_i within the set.
C_i is the class for $i = I, \dots, c-1$ c is the total number of predefined classes.
The quality of the split on feature k into subset S_i is determined by the number of samples belonging to class C_i. It is then calculated as the weighted sum of the Gini

impurity of the resulting subsets. The data can be computed using the following formula:

$$Gini_{split} = \sum_{i=0}^{c-1} \left(\frac{n_i}{n}\right) \text{Gini}(S_i) \quad (2)$$

where n_i represents the number of samples in subset S_i) after the split, and n is the total number of samples in the given node.

- Application of Random Forest with Genetic Algorithm

The genetic algorithm is a problem-solving method adapted from the genetic processes of biological organisms, based on Charles Darwin's theory. The nature of the genetic algorithm is to search for possible solutions to obtain the optimal one.

a. Initial Population

The initial population is formed from chromosomes with a size equal to the population size (UkPop). Each chromosome represents the sequence of offices that the salesman must visit. Therefore, the simplest chromosome representation to express the solution to this problem is described as a permutation of office indices in this problem and can be expressed as the following chromosome v:

$$V_i = [g1, g2, \dots, gN], \quad (3)$$

With 1 ≤ i ≤ UkPop.

b. Evaluation Process

The evaluation process is a process of calculating the fitness value, which represents the quality level of a chromosome as a solution representation. The fitness value indicates whether a solution is good or not. The higher the fitness value, the better the chromosome. The inversion process can be performed using the formula:

$$F_i = \frac{1}{f_i'} \quad (4)$$

where i represents the chromosome.

Description:

F_i: Fitness value of the i-th chromosome.

f_i: Path length of the i-th chromosome.

c. Selection

Selection is the process of choosing chromosomes that will be retained in the next population. This study uses the roulette wheel method, similar to a roulette wheel, where selection is performed randomly using real numbers. the selection process can be carried out with the following steps:

1) Calculate the relative fitness value using the formula:

$$P_i = \frac{F_i}{\sum_{i=1}^{UkPop} F_i} \quad (5)$$

2) Calculate the cumulative fitness value using the formula:

$$P_i \text{ and } q_i = q_{(i-1)} + P_i \text{ , i=2,3,…,UkPop} \quad (6)$$

3) Generate a random number r between 0 and 1 (0 < r < 1).

4) If r < q_i select the first chromosome.

If q_i $r < r$ $q_{(i-1)}$with j=(1,2,…,UkPop) select the i+j-th chromosome. (7)

5) Repeat the process for the number of chromosomes in the population.

2.2.6 Model Evaluation

In the evaluation stage, the accuracy of the algorithm will be measured based on the quality of the training data and tested using the Confusion Matrix technique

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (8)$$

TP : True Positives
TN : True Negatives
FP : False Positives
FN : False Negatives

3. Results and Discussion

3.1 Data Collection Results

Table 1 presents the data used in this study, which was obtained from the Blood Donor Unit (UDD) of PMI Samarinda for the years 2023-2024. The dataset consists of 1,000 rows, covering various variables used for the classification process to determine whether a donor is eligible to donate blood.

This data includes personal information about donors, such as name, age, blood type, gender (JK), donor status (whether the donation was successful or canceled), hemoglobin level (HB), blood pressure (tensi), and body weight. For example, in the first row, a donor named "Name1", aged 36 years, with blood type O+ and male gender, successfully donated blood with a hemoglobin level of 12.07 g/dL, blood pressure of 100/79 mmHg, and body weight of 75 kg. Meanwhile, in the last row, "Name1000", aged 31 years, with blood type A+ and male gender, also successfully donated blood with a hemoglobin level of 14.03 g/dL, blood pressure of 154/98 mmHg, and body weight of 110 kg. This dataset is used to develop a model that can predict a person's eligibility to donate blood based on these variables.

Table 1 : Data UDD PMI Samarinda

No	Nama	Umur	Golongan Darah	JK	Status	HB	Tensi	Berat Badan
1	Name1	36	O+	Pria	Berhasil	12,07	100/79	75
2	Name2	24	O+	Wanita	Berhasil	13,03	131/91	78
…	…	…	…	…	…	…	….	…
999	Name999	24	B+	Wanita	Berhasil	13,05	110/84	60
1000	Name1000	31	A+	Pria	Berhasil	14,03	154/98	110

3.2 Data Pre-Processing Results

After the data collection process is completed, the next step is data pre-processing, which involves cleaning and preparing the data to ensure it is ready for the modeling stage.

3.2.1 Data Selection

In this stage, relevant attributes are selected, while irrelevant ones are removed. During the selection process, two columns were identified as irrelevant for blood donor classification and were eliminated. After the selection process, the dataset consists of seven attributes used as features and one

Table 2 : Result Data Selection

	Umur	Golongan Darah	JK	Status	HB	Tensi	Berat Badan
0	36	O+	Pria	Berhasil	12,07	100/79	75
1	24	O+	Wanita	Berhasil	13,03	131/91	78
…	…	…	…	…	…	….	…
998	24	B+	Wanita	Berhasil	13,05	110/84	60
999	31	A+	Pria	Berhasil	14,03	154/98	110

3.2.2 Data Cleaning

	Umur	Golongan Darah	JK	Status	HB	Tensi	Berat Badan
0	36	O+	Pria	Berhasil	12.07	100/79	75.0
1	24	O+	Wanita	Berhasil	13.03	131/91	78.0
2	21	A+	Pria	Berhasil	15.02	126/76	99.0
3	37	A+	Pria	Berhasil	15.04	150/87	95.0
4	24	B+	Wanita	Berhasil	15.06	140/99	120.0
...	...	...	...	...	...	...	...
995	36	B+	Pria	Berhasil	15.05	117/82	71.0
996	31	O+	Pria	Berhasil	14.08	153/74	74.0
997	55	O+	Pria	Berhasil	14.00	131/85	76.0
998	24	B+	Wanita	Berhasil	13.05	110/84	60.0
999	31	A+	Pria	Berhasil	14.03	154/98	110.0

827 rows × 7 columns

Figure 2 : Result Data Cleaning

The data cleaning process includes removing duplicate data, checking for inconsistencies, and correcting errors such as typos (Ramon et al., 2022). The dataset has undergone a thorough cleaning process to ensure optimal data quality. The first step

involved replacing periods (.) with NaN values to indicate missing data. Rows containing NaN values were then removed using the dropna() function. Additionally, duplicate entries were eliminated using drop_duplicates() to ensure the dataset only contained unique records. After completing this process, the dataset was reduced from 1,000 rows to 827 valid rows, retaining seven columns: Age, Blood Type, Gender (JK), Status (donor eligibility), Hemoglobin Level (HB), Blood Pressure (Tensi), and Body Weight.

3.2.3 Data Transformation

Data transformation involves modifying or adjusting data into a structured format suitable for analysis (Ifongki, 2020). Several data transformation techniques have been applied to prepare the blood donor dataset for machine learning models. These transformation steps aim to improve dataset quality and ensure compatibility with machine learning algorithms. By organizing, cleaning, and refining the dataset, the analysis process becomes more effective, leading to an accurate and reliable predictive model for determining blood donor eligibility.

	Umur	Golongan Darah	JK	Status	HB	Tensi	Berat Badan
0	36	O+	Pria	Berhasil	12.07	100/79	75.0
1	24	O+	Wanita	Berhasil	13.03	131/91	78.0
2	21	A+	Pria	Berhasil	15.02	126/76	99.0
3	37	A+	Pria	Berhasil	15.04	150/87	95.0
4	24	B+	Wanita	Berhasil	15.06	140/99	120.0
...	...	...	...	...	...	...	...
995	36	B+	Pria	Berhasil	15.05	117/82	71.0
996	31	O+	Pria	Berhasil	14.08	153/74	74.0
997	55	O+	Pria	Berhasil	14.00	131/85	76.0
998	24	B+	Wanita	Berhasil	13.05	110/84	60.0
999	31	A+	Pria	Berhasil	14.03	154/98	110.0

Figure 3 : Dataset before the Transformation stage

	Umur	Golongan Darah	JK	Status	HB	Berat Badan	Sistolik	Diastolik
0	36	3	0	1	12.07	75.0	100	79
1	24	3	1	1	13.03	78.0	131	91
2	21	0	0	1	15.02	99.0	126	76
3	37	0	0	1	15.04	95.0	150	87
4	24	2	1	1	15.06	120.0	140	99
..	...	...	..	...	...	...	...	...
995	36	2	0	1	15.05	71.0	117	82
996	31	3	0	1	14.08	74.0	153	74
997	55	3	0	1	14.00	76.0	131	85
998	24	2	1	1	13.05	60.0	110	84
999	31	0	0	1	14.03	110.0	154	98

[827 rows x 8 columns]

Figure 4 : Dataset after the Transformation stage

The displayed dataset has undergone a transformation process to improve data quality and consistency, as well as to prepare it for analysis or machine learning applications. This dataset consists of 827 rows and 8 columns: Age, Blood Type, Gender (JK), Status (donor eligibility), Hemoglobin (HB), Weight, Systolic, and Diastolic.

The data transformation process resulted in a more structured DataFrame, ready for further analysis. Categorical columns such as "Blood Type," "Gender (JK)," and "Status" have been converted into numerical values using label encoding techniques. For example, blood types are represented by specific numbers (e.g., 0 for A+, 1 for B+, etc.), gender is encoded as 0 for male and 1 for female, and donor status is converted into numerical values (1 for "Successful"). The "Blood Pressure" column, which previously contained blood pressure readings in string format (e.g., "120/80"), has been split into two separate numerical columns: "Systolic" (upper blood pressure) and "Diastolic" (lower blood pressure), making it easier to analyze separately. Additionally, columns such as "Hemoglobin (HB)," "Weight," "Systolic," and "Diastolic" have been ensured to contain only numerical values, making them consistent and ready for direct computation or data visualization.

3.2.4 Implementation of Random Oversampling

Distribution Class Before Oversampling Distribution Class After Oversampliung

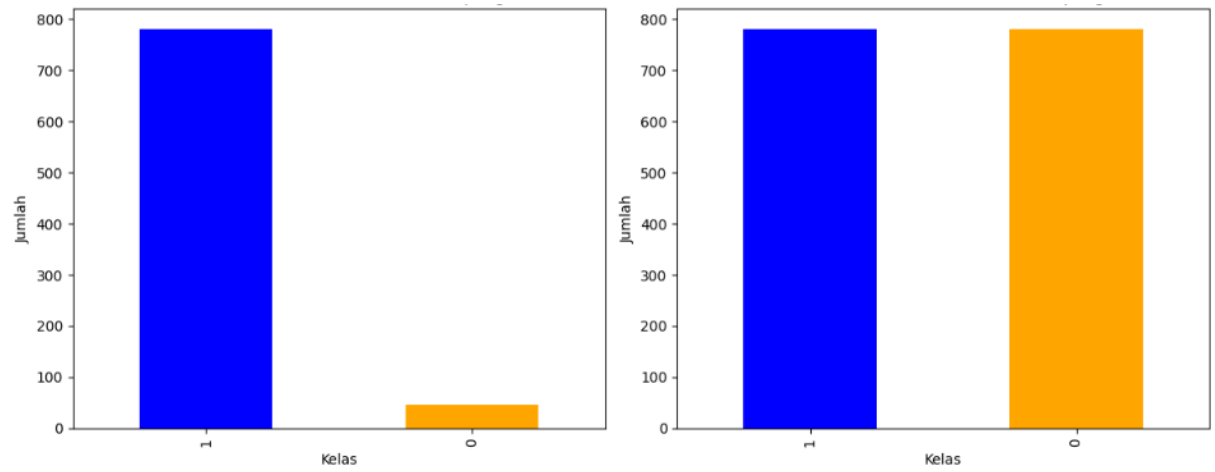

Figure 5 : Visualization of Class Distribution Before and After Oversampling

Table 3 : Class Distribution Before and After Oversampling

Kelas	Before Oversampling	After Oversampling
0	46	781
1	781	781

Table 3 shows the class distribution in the dataset before and after oversampling. Before oversampling, class 0 had significantly fewer samples (46 samples) compared to class 1, which had 781 samples. This condition indicates an imbalanced dataset, where the majority class (class 1) is highly dominant, while the minority class (class 0) has a very small number of samples.

To address this issue, oversampling was applied to the minority class (class 0), increasing its sample size to 781, matching the number of samples in class 1. After oversampling, the class distribution became balanced, with each class having the same number of samples (781 samples). This improvement enhances the representation of both classes during the model training process, helping the algorithm learn patterns more effectively without bias toward the majority class.

3.2 Modeling and Evaluation Results

This stage aims to present the accuracy results of the Random Forest algorithm model, which was applied in combination with the Genetic Algorithm (GA) and Random Oversampling (RO) optimization methods. This approach is designed to address the class imbalance problem in blood donor

eligibility data. The evaluation process involves the 10-Fold Cross Validation technique to systematically split and train the data. Each model's accuracy is evaluated using the confusion matrix to ensure the overall accuracy and performance of the model.

3.3.1 Implementation of Random Forest Without Random Oversampling

Table 4 : Random Forest Without Random Oversampling

Total Value Each Fold	TP	FP	TN	FN	Average Accuracy
	780	4	42	1	99.39%

$$\text{Accuracy} = \frac{780 + 42}{780 + 42 + 4 + 1} = \frac{822}{827} = 0.9939$$

$$Accuracy = 0.9939 \times 100\% = 99.39\%$$

Table 4 shows the average accuracy obtained from the Random Forest model on the data before oversampling. The recorded average accuracy is 99.39%, indicating that the model successfully classifies the data with an overall accuracy of 99.39%. This result suggests that the Random Forest model performs exceptionally well in predicting blood donor eligibility on the dataset before balancing, with most predictions being correct.

3.3.2 Implementation of Random Forest and GA Feature Selection without Oversampling

Table 5 : Average Accuracy of GA Feature Selection without Oversampling

Total Value Each Fold	TP	FP	TN	FN	Average Accuracy
	780	3	43	1	99.52%

$$\text{Accuracy} = \frac{780 + 43}{780 + 43 + 3 + 1} = \frac{823}{827} = 0.9952$$

$$\text{Accuracy} = 0.9952 \times 100\% = 99.52\%$$

Table 5 presents the evaluation results of the Random Forest model optimized using Genetic Algorithm (GA) without applying oversampling to the data. The evaluation was conducted using 10-fold cross-validation, where the metrics True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), and accuracy were recorded for each fold. Overall, this table reflects that the Random Forest model optimized with Genetic Algorithm (GA) is highly capable of capturing patterns in the data, producing accurate predictions, and maintaining a very low error rate. This demonstrates the effectiveness of the optimization and feature selection methods in enhancing model performance, even without applying oversampling to address data imbalance.

3.3.2 Implementation of Random Forest and Optimization without Oversampling

Table 6 : Average Accuracy of GA Optimization without Oversampling

Total Value Each Fold	TP	FP	TN	FN	Average Accuracy
	780	4	42	1	99.39%

$$\text{Accuracy} = \frac{780 + 42}{780 + 42 + 4 + 1} = \frac{822}{827} = 0.9939$$

$$\text{Accuracy} = 0.9939 \times 100\% = 99.39\%$$

Table 3.11 presents the evaluation results of the model with Genetic Algorithm (GA) Optimization without Oversampling, where the values of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) are recorded for each fold. The total values are 780 for TP, 4 for FP, 42 for TN, and 1 for FN, resulting in a model accuracy of 99.39%. This indicates that the model performs exceptionally well, with only a few misclassifications in both positive and negative predictions (4 FP and 1 FN). In other words, despite some classification errors, the model still demonstrates a very high accuracy rate of 99.39%. This result highlights the effectiveness of optimization using Genetic Algorithm, even without applying oversampling to the data.

3.3.4 Implementation of Random Forest + GA Feature Selection + GA Optimization without Oversampling

Table 7 : Average Accuracy of Random Forest + GA Feature Selection + GA Optimization without Oversampling

Total Value Each Fold	TP	FP	TN	FN	Average Accuracy
	782	3	43	1	99.52%

$$\text{Accuracy} = \frac{782 + 43}{782 + 43 + 3 + 1} = \frac{825}{829} = 0.9952$$

$$\text{Accuracy} = 0.9952 \times 100\% = 99.52\%$$

Table 7 presents the evaluation results of the Random Forest model optimized with Genetic Algorithm (GA) without applying oversampling. The evaluation was conducted using 10-fold cross-validation, which divides the data into 10 folds. In each fold, the model is trained on 9 parts and tested on 1 part. The results show that the model produced 782 True Positives (TP), meaning 782 instances that should be positive were correctly classified. However, there were 3 False Positives (FP), indicating that the model incorrectly classified 3 instances that should be negative as positive. The model also correctly classified 43 True Negatives (TN), meaning 43 negative instances were accurately identified, with only 1 False Negative (FN), where the model failed to identify 1 instance that should be positive. Overall, the model achieved an average accuracy of 99.39%, demonstrating excellent performance in correctly classifying the data. Despite a few minor errors (3 FP and 1 FN), the model still exhibits a very high accuracy level across most folds.

3.3.4 Implementation of Random Forest with Random Oversampling

Table 8 : Average Accuracy of Random Forest with Oversampling

Total Value Each Fold	TP	FP	TN	FN	Average Accuracy
	780	0	781	1	99.94%

$$\text{Accuracy} = \frac{780 + 781}{780 + 781 + 4 + 1} = \frac{1561}{1562} = 0.9994$$

$$Accuracy = 0.9994 \times 100\% = 99.94\%$$

Table 8 presents the average accuracy obtained from the Random Forest model after applying oversampling to the data. The accuracy of 99.94% indicates that the model's average accuracy is approximately 99.94%, which is an excellent result. This suggests that after oversampling, the model achieves very high performance with minimal prediction errors. The application of oversampling, which increases the number of minority class samples, appears to have improved the model's ability to handle data imbalance, thereby enhancing overall accuracy. With an almost perfect average accuracy, this model demonstrates the effectiveness of the oversampling technique in improving class representation, providing a more stable and accurate model for predicting blood donor eligibility.

3.3.6 Implementation of Random Forest and GA Feature Selection with Oversampling

Table 9 : Average Accuracy of GA Feature Selection with Oversampling

Total Value Each Fold	TP	FP	TN	FN	Average Accuracy
	783	0	783	1	99,94%

$$\text{Accuracy} = \frac{780 + 781}{780 + 781 + 0 + 1} = \frac{1561}{1562} = 0.9994$$

$$Accuracy = 0.9994 \times 100\% = 99.94\%$$

The provided table presents the average accuracy of the feature selection process using the Genetic Algorithm (GA) with the application of oversampling after undergoing 10-fold cross-validation. Based on the total values for True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), the results are as follows: the model correctly classified 783 positive instances as True Positives, made no errors in classifying negative instances as False Positives, and correctly classified 783 negative instances as True Negatives. However, there was one misclassification of a positive instance, resulting in a False Negative

Despite this minimal error, the model still achieved an average accuracy of 99.94%, reflecting outstanding performance. This indicates that feature selection using GA and the application of oversampling significantly improved the model's accuracy, with

nearly all instances correctly classified, and only a negligible number of errors affecting overall performance.

3.3.7 Implementation of Random Forest and Optimization with Oversampling

Table 10 : Average Accuracy of GA Optimization with Oversampling

Total Value Each Fold	TP	FP	TN	FN	Average Accuracy
	780	0	781	1	99.94%

$$\text{Accuracy} = \frac{780 + 781}{780 + 781 + 0 + 1} = \frac{1561}{1562} = 0.9994$$

$$\text{Accuracy} = 0.9994 \times 100\% = 99.94\%$$

The provided table presents the average accuracy of the model after undergoing optimization using the Genetic Algorithm (GA) with the application of oversampling, following 10-fold cross-validation. Based on the total values—True Positive (TP) of 780, True Negative (TN) of 781, False Positive (FP) of 0, and False Negative (FN) of 1—the model demonstrates excellent performance. The average accuracy achieved is 99.94%, indicating that the model makes almost no classification errors. There is only one misclassification in the positive data (FN = 1), while TP and TN values are nearly perfect. The application of oversampling in this process has helped enhance the model's performance, particularly in handling class imbalance and ensuring highly accurate classification results.

3.3.8 Implementation of Random Forest + GA Feature Selection + GA Optimization with Oversampling

Table 11 : Average Accuracy of Random Forest + GA Feature Selection + GA Optimization with Oversampling

Total Value Each Fold	TP	FP	TN	FN	Average Accuracy
	786	15	786	5	98.78%

$$\text{Accuracy} = \frac{786 + 786}{786 + 786 + 15 + 5} = \frac{1572}{1592} = 0.9878$$

$$\text{Accuracy} = 0.9878 \times 100\% = 98.78\%$$

Table 11 presents the average accuracy of the Random Forest model optimized using GA Feature Selection and GA Parameter Optimization, with the application of oversampling to address data imbalance. Based on the total values—True Positive (TP) of 786, False Positive (FP) of 15, True Negative (TN) of 786, and False Negative (FN) of 5—the model demonstrates excellent performance. With an average accuracy of 98.78%, the model successfully classifies the majority of data correctly, despite some

misclassifications in positive (FN) and negative (FP) instances. This indicates that while minor errors exist, the model still maintains solid and accurate overall performance. The application of oversampling helps the model handle data imbalance more effectively, while GA-based feature selection and parameter optimization allow the model to leverage relevant features and optimal parameters to enhance classification performance.

3.3.9 Comparison of Evaluation Results

Table 12 : Comparison of Average Accuracy Results for Models Without Oversampling

Avarage Accuracy	Random Forest	Random Forest + GA Feature Selection	Random Forest + GA Optimization	Random Forest + GA Feature Selection + GA Optimization	Change from Random Forest to Random Forest + GA Feature Selection	Change from Random Forest to Random Forest + GA Optimization	Change from Random Forest to Random Forest + GA Feature Selection + GA Optimization
	99,39%	99.52%	99.39%	99.52%	+0,13%	+0%	+0,13%

Table 12 presents a comparison of the average accuracy results from various models applied to the dataset without oversampling, highlighting the changes in accuracy between the baseline Random Forest model and the models optimized using GA Feature Selection and GA Optimization. The baseline Random Forest model achieves an average accuracy of 99.39%, serving as the reference point for comparison. After applying GA Feature Selection, the accuracy slightly increases to 99.52%, indicating an improvement. However, when only GA Optimization is applied, the accuracy remains 99.39%, showing no significant change compared to the original Random Forest model. When both GA Feature Selection and GA Optimization are applied together, the accuracy remains at 99.52%, which is the same result obtained with GA Feature Selection alone. These findings indicate that the transition from Random Forest to Random Forest + GA Feature Selection results in a +0.13% increase in accuracy, demonstrating a minor improvement. In contrast, the transition to Random Forest + GA Optimization does not yield any accuracy enhancement, as the value remains unchanged at 99.39%. Furthermore, applying both GA Feature Selection and GA Optimization together leads to the same +0.13% increase, suggesting that the primary contribution to accuracy improvement comes from GA Feature Selection, while GA Optimization does not significantly impact the model's performance on this dataset.

Table 13 : Comparison of Model Average Accuracy Results With Oversampling

Avarage Accuracy	Random Forest	Random Forest + GA Feature Selection	Random Forest + GA Optimization	Random Forest + GA Feature Selection + GA	Change from Random Forest to Random	Change from Random Forest to Random Forest + GA	Change from Random Forest to Random Forest + GA

		n		Optimizatio n	Forest + GA Feature Selectio n	Optimizatio n	Feature Selection + GA Optimizatio n
99,94%	99.94%	99.94%	98.78%	+0%	+0%	+1,16%	

Table 13 presents a comparison of the average accuracy of various models applied to the dataset with oversampling to address data imbalance. The four models compared are Random Forest, Random Forest + GA Feature Selection, Random Forest + GA Optimization, and Random Forest + GA Feature Selection + GA Optimization. The results show that Random Forest, Random Forest + GA Feature Selection, and Random Forest + GA Optimization all achieve the same average accuracy of 99.94%, indicating that the application of GA Feature Selection and GA Optimization does not significantly impact accuracy compared to the standard Random Forest model. However, when both GA Feature Selection and GA Optimization are applied together, the accuracy drops to 98.78%, suggesting that combining these two optimization techniques negatively affects the model's classification performance. The accuracy changes also highlight that transitioning from Random Forest to Random Forest + GA Feature Selection results in no accuracy change (+0%), and similarly, moving from Random Forest to Random Forest + GA Optimization does not yield any improvement (+0%). However, the shift from Random Forest to Random Forest + GA Feature Selection + GA Optimization shows a 1.16% decrease in accuracy, despite initially appearing as an increase compared to the lowest accuracy value. Overall, these findings indicate that while oversampling does not negatively impact the model's performance, the combined application of GA Feature Selection and GA Optimization does not provide a significant improvement and instead leads to a decline in model accuracy.

4. Conclusion

The conclusion of this study is that the application of oversampling to address data imbalance has a positive impact on improving classification model accuracy, particularly for the Random Forest model. Although GA Feature Selection and GA Optimization techniques can enhance accuracy in certain scenarios, combining both techniques (GA Feature Selection + GA Optimization) does not yield significant improvements and may even lead to a decrease in accuracy for the Random Forest model.

Specifically, the Random Forest model with oversampling achieved excellent accuracy, reaching 99.94%, and remained stable after the separate application of GA Feature Selection and GA Optimization. However, when both techniques were combined, the model's accuracy dropped to 98.78%, indicating that combining parameter optimization and feature selection does not always improve model performance for this dataset.

The application of oversampling has been proven highly effective in handling class imbalance, but feature selection and parameter optimization techniques must be carefully evaluated, as they may risk reducing performance if not applied correctly.

Therefore, the use of GA Feature Selection and GA Optimization should be tailored to the characteristics of the data and model used to achieve optimal results.

References

Anjas Aprihartha, M., Zulhan, D., Nurfaizal, A. F., & Nur Alam, T. (2024). Penyelesaian Masalah Ketidakseimbangan Data Melalui Teknik Oversampling dan Undersampling pada Klasifikasi Siswa Tidak Naik Kelas. *Jurnal Teknik Ibnu Sina*, *9*(01), 43–52.

Atmaja, K. J., Anandita, I. B. G., & Dewi, N. K. C. (2018). Penerapan Data Mining Untuk Memprediksi Potensi Pendonor Darah Menjadi Pendonor Tetap Menggunakan Metode Decision Tree C.45. *S@Cies*, *7*(2), 101–108. https://doi.org/10.31598/sacies.v7i2.284

Ayuningtyas, P., Khomsah, S., Informatika, T., Informatika, F., Data, S., & Informatika, F. (2024). *Pelabelan Sentimen Berbasis Semi-Supervised Learning menggunakan Algoritma LSTM dan GRU*. *9*(3), 217–229.

Basri, R. F., & Rahmita. (2021). PENYULUHAN PROSES DONOR DARAH DAN PENTINGNYA DONOR DARAH SEBAGAI EDUKASI PRA-DONASI PADA MASYARAKAT PATTITANGNGANG, KECAMATAN MAPPAKASUNGGU KABUPATEN TAKALAR. *Abdimas Indonesia*, *1*(2), 26–32. https://dmi-journals.org/jai/article/view/226

Efendi, M. S., & Zyen, A. K. (2024). Penerapan Algoritma Random Forest Untuk Prediksi Penjualan Dan Sistem Persediaan Produk. *Resolusi: Rekayasa Teknik Informatika Dan Informasi*, 12–20.

Firdaus, M. R., Latif, A., & Gata, W. (2020). Klasifikasi Kelayakan Calon Pendonor Darah Menggunakan Neura L Network. *Sistemasi*, *9*(2), 362. https://doi.org/10.32520/stmsi.v9i2.840

Halim, T. N., Martin, R., & ... (2023). Klasifikasi Kepuasan Pelanggan Terhadap Platform E-Commerce dengan Metode K–Nearest Neighbor (K-NN). *Jurasik (Jurnal Riset …*, *8*, 512–523. http://ejurnal.tunasbangsa.ac.id/index.php/jurasik/article/view/636%0Ahttps://ejurnal.tunasbangsa.ac.id/index.php/jurasik/article/download/636/609

Handayani, K., Lisnawanty, L., Latif, A., Firdaus, M. R., & Hasan, F. N. (2021). Komparasi Algoritma C4.5 Dan Naïve Bayes Dalam Penentuan Status Kelayakan Donor Darah. *Sistemasi*, *10*(3), 676. https://doi.org/10.32520/stmsi.v10i3.1440

Ifongki, I. (2020). Penerapan Data Mining Menggunakan Algoritma C4.5 Tehadap Pengaruh Penjualan Kopi Pada Pt. Jpw Indonesia. *Jurnal Sistem Informasi Dan Informatika (Simika)*, *3*(1), 40–54. https://doi.org/10.47080/simika.v3i1.836

Irfan Syahroni, M. (2022). Prosedur Penelitian Kuantitatif. *EJurnal Al Musthafa*, *2*(3), 43–56. https://doi.org/10.62552/ejam.v2i3.50

Karomi, M. A. Al. (2020). Optimasi Parameter K Pada Algoritma Knn Untuk Klasifikasi Heregistrasi Mahasiswa. *IC-Tech*, *10*(1), 28–33.

Rivaldo, V. J., Siswa, T. A. Y., & Pranoto, W. J. (2024). Perbaikan Akurasi Naïve Bayes dengan Chi-Square dan SMOTEDalam Mengatasi High Dimensional dan

Imbalanced Data Banjir. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *8*(3), 1656–1664.

Veronica, R., Agustina, Elwindra, Prihatini, F., Vestabilivy, E., & Herlina. (2024). *Kegiatan Donor Darah sebagai Salah Satu Cara Membantu Meningkatkan Kesehatan Diri dan Selamatkan Nyawa Sesama diperoleh dari manusia , dalam keadaan mengalami kecelakaan atau menderita suatu pengganti . Donor darah sukarela merupakan seseorang yang menyum*. *2*(1), 116–125. https://doi.org/10.62354/healthcare.v2i1.21

Wahono, H., & Riana, D. (2020). Prediksi Calon Pendonor Darah Potensial Dengan Algoritma Naïve Bayes, K-Nearest Neighbors dan Decision Tree C4.5. *JURIKOM (Jurnal Riset Komputer)*, *7*(1), 7. https://doi.org/10.30865/jurikom.v7i1.1953