ISSN: 2581-1274

Evaluating the Impact of Data Balancing Techniques on the k-Nearest Neighbors Algorithm for Microarray Data Classification

Febi Nur Salisah [a,1,*], Inggih Permana [a,b,d,2], Sanusi [c,3], Shir Li Wang [d,4]

[a] *Program Studi Sistem Informasi, Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia*
[b] *Puzzle Research Data Technology (Predatech),Universitas Islam Negeri Sultan Syarif Kasim Riau, Indonesia*
[c] *Teknologi Informasi, Fakultas Teknik, Universitas Teuku Umar*
[d] *Faculty of Computing and Meta-Technology, Universiti Pendidikan Sultan Idris, 35900 Tanjong Malim, Perak, Malaysia*
[1]*febinursalisah@uin-suska.ac.id*;* [2]*inggihpermana@uin-suska.ac.id;* [3]*sanusi@utu.ac.id;*
[4]*shirli_wang@meta.upsi.edu.my*
*Corresponding author

ARTICLE INFO

Article history:
Published
July 14, 2025

Keywords:
Microarray
kNN
RUS
ROS
SMOTE

ABSTRACT

Microarray data classification poses significant challenges in bioinformatics due to the nature of the data, which has a very high number of features but a limited number of samples, and an unbalanced class distribution. This condition can cause a decrease in the performance of classification models, including k-Nearest Neighbor (kNN). This study aims to evaluate the performance of the kNN algorithm in classifying unbalanced and balanced data. The balancing techniques used are Random Undersampling (RUS), Random Oversampling (ROS), and Synthetic Minority Over-sampling Technique (SMOTE). The datasets used in this study are three leukemia datasets with different class structures, namely two, three, and four classes. The experimental results show that the ROS and SMOTE techniques consistently improve the performance of kNN, with the best accuracy reaching more than 97%. In the two-class dataset, ROS gave the best performance (99.4%), while in the three-class dataset, SMOTE showed the most optimal results (98.5%). In the four-class dataset, the performance improvement due to balancing was very significant; SMOTE and ROS were able to improve the accuracy from 89.7% (without balancing) to 99.0% and 98.8%, respectively. Although RUS recorded perfect accuracy of 100%, the results were anomalous and inconsistent. RUS showed less stable performance and was often lower than the condition without balancing, especially on datasets with four classes. Overall, the SMOTE technique proved to be the most stable and effective for various class structures. This study shows the importance of balancing strategies in the classification of complex and imbalanced microarray data.

I. Introduction

Microarray data, also known as gene expression data [1], is a type of biological data that records the expression levels of thousands of genes simultaneously from tissue or cell samples. This data has very high-dimensional characteristics with thousands of features but is usually only available in a very limited number of samples, thus creating its own challenges in its analysis [2]. Microarray data is generally presented in a two-dimensional array format, which is very useful in the analysis of diseases such as cancer [3]. One of the problems with microarray data is that it often has an unbalanced class distribution [4]. Class imbalance occurs when the number of samples in one class is much greater than in the other classes [5]. Data imbalance can cause the classification model to be biased towards the majority class and ignore the minority class [6]. This problem can reduce the accuracy of the model in detecting rare or clinically important conditions, such as certain cancer subtypes that have a small number of samples in the data.

To overcome the problem of class imbalance in data, data balancing is carried out. Data balancing aims to equalize the distribution of the majority and minority classes in the dataset. By balancing the number of samples between the majority and minority classes, the model can avoid bias and improve

its ability to detect patterns from the minority class. Data balancing is carried out at the preprocessing stage. Data balancing basically uses resampling of the dataset, which is carried out before the model training process begins. Because this process is carried out separately from the classification process, this method is flexible and simple to implement [7].

Three commonly used data balancing techniques are Random Undersampling (RUS), Random Oversampling (ROS), and Synthetic Minority Over-sampling Technique (SMOTE). RUS is a data balancing technique that is carried out by randomly reducing the number of samples from the majority class to balance it with the number of samples in the minority class. The purpose of reducing the number of samples from the majority class is to overcome the bias that arises due to class imbalance during the learning process, but this approach can result in the loss of some important information [8]. ROS and SMOTE are types of data balancing that apply the oversampling approach. The opposite of undersampling, oversampling is a data balancing technique that is carried out by increasing the number of samples from the minority class to balance the number of samples in the majority class. ROS randomly increases minority class samples, and this approach is often chosen by researchers because it can overcome the problem of undersampling techniques that risk losing important information due to sample reduction. However, this approach can increase the risk of overfitting the model [9]. Meanwhile, the SMOTE technique was created to avoid the weaknesses of conventional sampling techniques, such as overfitting in oversampling and loss of information in undersampling [10]. SMOTE takes a more sophisticated approach than ROS to adding minority data, namely by generating new synthetic samples from the minority class based on interpolation between existing samples.

The k-Nearest Neighbors (kNN) algorithm, introduced in 1951, has developed into an important tool in various fields and functions as an instance-based, non-parametric learning algorithm used in supervised learning for classification and regression tasks [11]. In addition, according to [11], kNN is categorized as a lazy learning algorithm because it does not form an explicit model of the training data but rather makes predictions based directly on the proximity of instances. Previous studies have widely utilized the kNN algorithm in microarray data analysis, especially for feature selection, data imputation, and disease classification tasks. [12] evaluated the effectiveness of a combination of evolutionary algorithms and kNN for classification and feature selection without performing initial dimensionality reduction for cancer microarray data classification. [13] focused on the effect of dimensionality reduction methods on kNN classification performance. Meanwhile, [14] proposed an approach to handle missing values by combining kNN-based feature selection and imputation. [15] explored the application of kNN in disease classification and clinical outcome prediction based on genomic data by highlighting factors that affect the performance of the algorithm in a clinical setting. Further research by [16] discussed the handling of missing values in gene expression data. [17] developed a kNN-based ensemble classification model. Although the contributions of previous studies are significant in improving the classification performance of microarray data, most of them do not explicitly discuss the effect of data balancing techniques on the classification performance of the kNN algorithm, especially in the context of microarray datasets, which are known to be highly imbalanced.

Based on the previous description, this study integrates the kNN algorithm with various data balancing techniques such as ROS, RUS, and SMOTE to improve the accuracy and stability of the model in the task of classifying microarray data. This study aims to evaluate the extent to which data balancing techniques can have a positive impact on classification performance when applied to microarray datasets that have an imbalanced class distribution. In addition, this study also attempts to determine which balancing technique is most effective in producing the best classification performance in the context of microarray data.

The structure of this paper is systematically arranged to provide a comprehensive understanding of the research conducted. Chapter 2 presents the methodology used, including a description of the microarray dataset used, the data balancing techniques applied (RUS, ROS, and SMOTE), the kNN algorithm, and the experimental setup. Chapter 3 contains the results and discussion, which show a comparison of kNN performance on an unbalanced dataset and after balancing, as well as an analysis of the effect of each balancing technique on classification performance. Chapter 4 concludes the main

findings of this study, highlights the effectiveness of balancing methods in improving classification performance, and provides suggestions for further research in the future.

II. Method

This chapter presents in detail the methodology used in evaluating the effect of data balancing techniques on the performance of the kNN algorithm in classifying class-imbalanced microarray datasets. The discussion in this chapter includes the steps of implementing the simulation, a description of the dataset used, the feature selection process, data balancing techniques, data normalization, kNN classification model development, and model performance evaluation strategies.

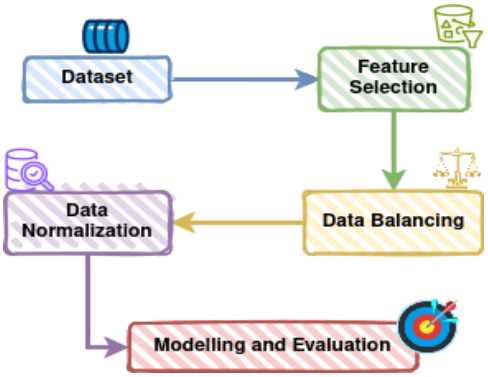

Fig. 1. Simulation steps

As shown in Figure 1, this study begins with a feature selection process for all datasets used. Feature selection is carried out using the Information Gain (IG) method to reduce the number of features and retain the most relevant features to the target class. After the feature selection process, the next stage is data balancing using three techniques, namely the RUS, ROS, and SMOTE techniques. After that, both datasets that have been balanced and those that have not been are normalized using the Min-Max Normalization method to equalize the scale of feature values. Furthermore, a classification model is built using the kNN algorithm on all variations of the dataset. Finally, an evaluation is carried out to assess the effect of using data balancing techniques on classification performance.

A. Datasets

This study utilized three leukemia datasets available for download at https://csse.szu.edu.cn/staff/zhuzx/datasets.html. These datasets were previously used by [18] in their study on feature selection in microarray data. The three leukemia datasets were distinguished based on the number of class labels, consisting of two, three, and four classes. A complete description of each dataset is presented in Table 1.

Table 1. Description of leukemia datasets used

No.	Datasets	Number of Features	Number of Samples	Number of Classes
1	Leukemia with two classes	7129	72	2
2	Leukemia with three classes	7129	72	3
3	Leukemia with four classes	7129	72	4

B. Information Gain (IG)

The dataset used in this study has a fairly large number of features, so a feature selection process is needed to increase the effectiveness of the algorithm. In this case, the feature selection method applied is IG. If it is known that there is an attribute A and a class C, the IG calculation is as follows [19]:

- The first is the calculation of entropy (H) before attribute A is observed. The entropy value is calculated using the following formula:

$$H(C) = -\sum p(c) \log_2 p(c) \quad (1)$$

where H(C) represents the entropy before the variable in class C is observed, p(c) indicates the probability of the occurrence of observation c, and c is one of the elements of class C.

- After that, the entropy calculation is carried out after attribute A is observed, according to the following formula:

$$H(C/A) = \sum p(a) \sum P(c|a) \log_2 p(c) \quad (2)$$

where a is included in set A, and c is included in set C.

- In the final stage, the IG of attribute A is calculated as the difference between the entropy before and after the observation of the attribute. The IG calculation follows the following formula:

$$IG(C, A) = H(C) - H(C|A) \quad (3)$$

where IG(C, A) is the information gain of attribute A based on class C, with H(C) being the initial entropy before attribute A is observed, and H(C/A) being the entropy after the attribute is observed.

C. Random Undersampling (RUS) and Random Oversampling (ROS)

According to [20], in the undersampling technique, some samples from the majority class are randomly removed from the training data. Although this method can help balance the class distribution, the removal risks reducing important information, making it difficult to determine an accurate decision boundary between the majority and minority classes, and potentially reducing classification performance. In addition, [20] also stated that random oversampling works by randomly increasing samples from the minority class through duplication with replacement into the training dataset. This process does increase the proportion of the minority class but can cause the model to overfit, especially if the same instance is duplicated too often during the training process. The distribution of the number of samples per class in each dataset, both before and after resampling, can be seen in Table 2.

Table 2. Distribution of the number of samples per class

No.	Dataset	Class Label	Number of Samples			
			Initial	RUS	ROS	SMOTE
1	Leukemia with two classes	ALL	47	25	47	47
		AML	25	25	47	47
2	Leukemia with three classes	B-Cell	38	9	38	38
		AML	25	9	38	38
		T-Cell	9	9	38	38
3	Leukemia with four classes	B-Cell	38	4	38	38
		BM	21	4	38	38
		T-Cell	9	4	38	38
		PB	4	4	38	38

D. Synthetic Minority Over-sampling Technique (SMOTE)

The SMOTE algorithm is an oversampling technique used to balance the class distribution in training data by generating synthetic instances [21]. Unlike ROS, SMOTE creates new data by interpolating between multiple instances of the minority class in the feature space. This process

considers the feature values and relationships between features, not just the overall position of the data points. The main parameters include the amount of oversampling and the number of nearest neighbors used in random interpolation. Factors such as data dimensionality, variance, correlation, and distribution between training and testing data also affect the effectiveness of this method. The number of samples per class in each dataset after the SMOTE process can be seen in Table 2. The steps of SMOTE can be seen in Algorithm 1.

Algoritma 1. SMOTE Algorithm [21]

```
function SMOTE(T, N, k)
    Input:
        T #number of minority class examples
        N #oversampling rate
        k #number of nearest neighbors
    Output:
        (N / 100) * T #synthetic samples of the minority class
    Variabel:
        Sample[][] #array to store original examples of the minority class
        newindex #counter for the number of synthetic examples generated, initialized with 0
        Synthetic[][] #array to store synthetic examples

    if N < 100 else
        Randomize the order of T minority class samples
        T = (N / 100) * T
        N = 100
    endif

    N = (int) N / 100     #SMOTE count is considered as an integer multiple of 100

    for i = 1 to T do
        Calculate k nearest neighbors for the i-th sample and store their indices in nnarray
        call function POPULATE(N, i, nnarray)
        # The population function can be seen in [21]
    endfor
endfunction
```

E. Min-Max Normalization

Min-max normalization is a normalization method that changes the range of values of all features into a uniform scale, usually between 0 and 1. This technique aims to prevent the dominance of certain features in the modeling and classification process, so that all features have a balanced contribution. This study uses the min-max normalization method as a data normalization approach. Several studies have shown the superiority of min-max normalization over other normalization methods in the context of the kNN algorithm. [22] compared min-max normalization with z-score normalization in the kNN algorithm, and found that min-max normalization provided better performance. Similar results were also obtained by [23] who evaluated three normalization methods, namely min-max normalization, z-score normalization, and decimal scaling in kNN, and concluded that min-max normalization produced the best accuracy. [24] also stated that min-max normalization outperformed z-score normalization in improving kNN performance. Meanwhile, [25] compared various normalization methods, including decimal scaling, sigmoid, softmax, min-max, statistical column, and z-score in the application of the backpropagation algorithm. Although not using the kNN algorithm, the study of [25] showed that min-max normalization also gave the best results in that context. The min-max normalization value can be calculated using the following formula.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4}$$

F. k-Nearest Neighbors (kNN)

kNN is a machine learning algorithm that falls into the lazy learning category and is used for both classification and regression tasks. The basic principle of kNN is to classify a sample based on the majority of labels from its k nearest neighbors. Steps in the kNN algorithm [11]:

- Determine the value of k

  The process begins by selecting the number of nearest neighbors to be used as a reference. Choosing the right k value greatly determines the performance of the algorithm's predictions.

- Calculate the distance

  This is done by measuring the distance between the sample to be classified and all samples in the training set. Choosing the right distance metric greatly affects the performance of the algorithm.

- Determine the k nearest neighbors

  All samples in the training set are sorted by their distance from the sample to be classified, from closest to farthest. Then, a number of k samples are selected that have the closest distance.

- Combine information from the k nearest neighbors

  For classification tasks, predictions are generated based on the labels that appear most frequently among the k nearest neighbors.

G. Experimental Setup

The feature selection and data balancing process were carried out using the Python programming language, while the next analysis stage was carried out using Orange Data Mining software. In this study, the entire dataset was subjected to feature selection by selecting the top 2% of features, which were 142 out of a total of 7129 features based on the highest IG value. For the kNN algorithm, the k values tested were from 1 to 10 in each leukemia microarray dataset. Each k value was applied to four data conditions, namely the original data (without balancing), and data that had been balanced using the RUS, ROS, and SMOTE techniques. The distance measurement metric used by kNN is the Euclidean Distance. Validation was carried out using the hold-out method, where 90% of the data was used as training data and 10% as test data, and repeated 50 times to see the consistency of the results. Model performance was evaluated using two metrics, namely accuracy and F1-score.

III. Results and Discussion

This chapter presents the results and discussion of classification simulations using the kNN algorithm on three different leukemia datasets, namely two-class, three-class, and four-class leukemia datasets. The three datasets are microarray data with characteristics of a high number of features, a small number of samples, and an unbalanced class distribution. The main focus of the analysis is to compare the performance of the model between the unbalanced data conditions and the balanced data using three common techniques, namely RUS, ROS, and SMOTE. Each dataset is tested with various k values to evaluate its impact on the accuracy and F1-score of the model. This simulation aims to examine the extent to which data balancing techniques are able to improve the performance of kNN classification compared to when the model is trained on the original unbalanced data, as well as to identify the most effective balancing approach for each class structure.

Table 3. Simulation results on the leukemia dataset which has two classes

K	Performance (%)							
	Without Balancing		ROS		RUS		SMOTE	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
1	95.50	95.50	98.40	98.40	95.20	95.20	98.60	98.60
2	93.20	93.10	95.80	95.80	95.20	95.20	98.00	98.00
3	97.00	97.00	**99.40**	**99.40**	96.80	96.80	98.60	98.60
4	95.30	95.20	98.60	98.60	96.80	96.80	98.20	98.20
5	95.00	95.00	98.80	98.80	96.80	96.80	98.00	98.00

K	Performance (%)							
	Without Balancing		ROS		RUS		SMOTE	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
6	93.50	93.40	98.80	98.80	95.20	95.20	98.60	98.60
7	94.00	93.90	99.00	99.00	95.20	95.20	98.60	98.60
8	94.30	94.20	97.40	97.40	95.20	95.20	98.60	98.60
9	95.30	95.20	97.60	97.60	95.20	95.20	98.60	98.60
10	95.30	95.20	97.60	97.60	95.20	95.20	98.60	98.60

Table 3 shows the simulation results of the KNN algorithm on the leukemia dataset that has two classes using various K values from KNN. Based on the table, it can be seen that on data that is not balanced, the KNN algorithm shows quite good performance, with an accuracy ranging from 93.2% to 97.0%. The best performance is obtained when the value of K = 3, with an accuracy and F1-score of 97.0%. Table 3 also shows that after the data balancing technique was applied, the increase in KNN performance was clearly seen in the ROS and SMOTE techniques. The ROS technique produced the highest overall performance, with an accuracy and F1-score reaching 99.4% at K = 3. This shows that the addition of data from the minority class randomly effectively helps KNN in detecting patterns better. The SMOTE technique also provides very good and stable performance, with the highest accuracy of 98.6% achieved in most K value variants (1, 3, 6, 7, 8, 9, and 10) and almost matches the results of ROS. Meanwhile, the RUS technique failed to obtain the highest accuracy better than the dataset without balancing, although at several K values (2, 4, 5, 6, 7, and 8), RUS managed to improve the performance of KNN compared to without balancing, although the results were not as high as ROS or SMOTE; the highest accuracy and F1-score values of RUS were 96.8% at K = 3, K = 4, and K = 5.

Overall, the value of K = 3 consistently gave the best results in almost all conditions of the leukemia dataset with two classes, both without balancing and with the application of the balancing technique. However, for data balanced with SMOTE, the value of K = 1 was the best value because it had a lower calculation complexity. In addition, on the leukemia dataset with two classes, the ROS and SMOTE balancing techniques proved effective in overcoming data imbalance, with ROS as the best method, followed by SMOTE. In all simulations, changes in the value of K did not cause drastic performance fluctuations, so all models were relatively stable against small variations in the value of K.

Table 4. Simulation results on the leukemia dataset which has three classes

K	Performance (%)							
	Without Balancing		ROS		RUS		SMOTE	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
1	92.00	92.10	96.70	96.70	94.00	94.00	97.70	97.70
2	94.50	94.60	97.00	97.00	92.00	92.10	**98.50**	**98.50**
3	94.50	94.60	96.80	96.80	96.00	96.00	98.50	98.50
4	94.30	94.30	94.00	94.00	96.00	96.00	98.50	98.50
5	95.50	95.50	94.30	94.40	96.00	96.00	98.50	98.50
6	95.30	95.30	93.80	93.90	96.00	96.00	96.20	96.20
7	96.30	96.30	95.30	95.40	96.00	96.00	96.70	96.70

K	Performance (%)							
	Without Balancing		ROS		RUS		SMOTE	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
8	95.00	95.00	95.00	95.00	96.00	96.00	96.30	96.40
9	94.50	94.50	95.30	95.30	96.00	96.00	97.00	97.00
10	95.30	95.20	94.80	94.80	96.00	96.00	97.70	97.70

Table 4 shows the simulation results of the KNN algorithm on the leukemia dataset that has three classes using various K values from KNN. The table shows that when the data is not balanced, the KNN algorithm shows quite good performance, with an accuracy ranging from 92.0% to 96.3%. The value of K = 7 gives the best results, with an accuracy and F1-score of 96.3%. After balancing with three different approaches, there is a significant increase in performance when using the SMOTE technique and the ROS technique. The SMOTE technique consistently gives higher results than the unbalanced dataset and the one balanced with other methods at all K values. The highest performance when using the SMOTE technique is when K = 2, K = 3, K = 4, and K = 5, with accuracy and F1-score values reaching 98.5% at values K = 2 to K = 5. This shows that the SMOTE technique in balancing the minority class is very effective in improving the ability of KNN to recognize patterns from the three existing classes. The Random Over Sampling (ROS) technique also improves model performance, although not as good as SMOTE. The highest value is achieved at K = 2 with an accuracy value and F1-score of 97.0%. Meanwhile, similar to leukemia data with two classes, in leukemia data with three classes, the RUS technique did not succeed in getting the highest accuracy that was better than the dataset without balancing. However, at several K values (1, 3, 4, 5, 6, 8, 9, and 10), RUS succeeded in improving KNN performance compared to without balancing. In addition, at several K values, the RUS technique succeeded in getting better accuracy values than the ROS technique, namely when the value of K = 4, K = 5, K = 6, K = 7, K = 8, K = 9, and K = 10. The performance of KNN on the RUS technique tends to be stable; the highest accuracy value is obtained at most K values, namely at values of K = 3 to values of K = 10, with accuracy values and F1-scores of 96.00%.

In general, the application of balancing techniques has been proven to improve the performance of the KNN algorithm in the classification of leukemia data with three classes. SMOTE is the most effective method, followed by ROS. The optimal K value in the ROS technique on leukemia data with three classes is in the range of K = 2 to K = 5. The best K value for the ROS technique is 2, because it has a lower classification calculation complexity than a higher K value. Similar to leukemia data with 2 classes, in all simulations, changes in the K value do not cause drastic changes in accuracy and F1-score, so all models are relatively stable against small variations in the K value.

Table 5. Simulation results on the leukemia dataset which has four classes

K	Performance (%)							
	Without Balancing		ROS		RUS		SMOTE	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
1	89.20	89.00	98.80	98.70	95.00	94.90	99.00	99.00
2	89.70	89.40	98.00	98.00	95.00	95.00	99.00	99.00
3	89.70	89.80	96.90	96.90	**100.00**	**100.00**	98.00	98.00
4	83.00	78.50	96.90	96.90	86.00	85.50	97.80	97.70
5	82.50	79.30	95.80	95.70	81.00	81.00	97.00	97.00
6	81.20	76.90	95.80	95.70	66.00	62.80	96.80	96.70

K	Performance (%)							
	Without Balancing		ROS		RUS		SMOTE	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
7	81.50	77.10	95.80	95.70	44.00	46.00	97.00	97.00
8	83.00	78.60	95.80	95.70	33.00	32.30	96.80	96.70
9	83.50	78.90	95.30	95.20	29.00	28.90	97.00	97.00
10	83.50	79.20	95.80	95.70	25.00	23.60	96.60	96.60

Based on the experimental results in Table 5, simulations conducted on the four-class leukemia dataset using the KNN algorithm with various k values, both without and with the application of data balancing techniques, several things can be concluded. On data that is not balanced, the performance of KNN is relatively good although not as good as the highest accuracy obtained when using data balancing. On unbalanced data, the best performance is obtained when the value of k = 3, with an accuracy of 89.70% and an F1 score of 89.80%. However, based on the F1 score value, the classification performance decreases sharply when the value of k is large, equal to 4, decreasing more than 10% when compared to the highest F1 score value. This indicates the sensitivity of the model to the choice of k values in unbalanced data. After the balancing technique is applied, there is a very significant increase in performance. As seen in Table 5, the ROS technique shows high and stable performance, with an accuracy of 98.8% and an F1-score of 98.7% at a value of k = 1. The SMOTE technique is slightly superior to the ROS technique, namely the SMOTE technique achieves an accuracy and F1-score of 99.0% at k = 1 and k = 2. Meanwhile, the RUS technique produces varying performance. Interestingly, at k = 3, this technique records extreme results with perfect accuracy and F1-score of 100%. However, at other k values, the RUS performance decreases drastically and tends to be low, which is less than 70%, as occurs at k = 6 (66%), k = 7 (44%), k = 8 (33%), k = 9 (29%), and k = 10 (25%). This shows the instability of the RUS technique due to the loss of information from the majority class.

Overall, the balancing technique has proven to be very influential in improving the performance of the KNN algorithm in classifying four-class leukemia data. The SMOTE and ROS methods provide superior and stable results compared to without balancing or compared to the RUS technique. The value of k = 1 tends to be the optimal choice for the balancing approach of the ROS technique and the SMOTE technique.

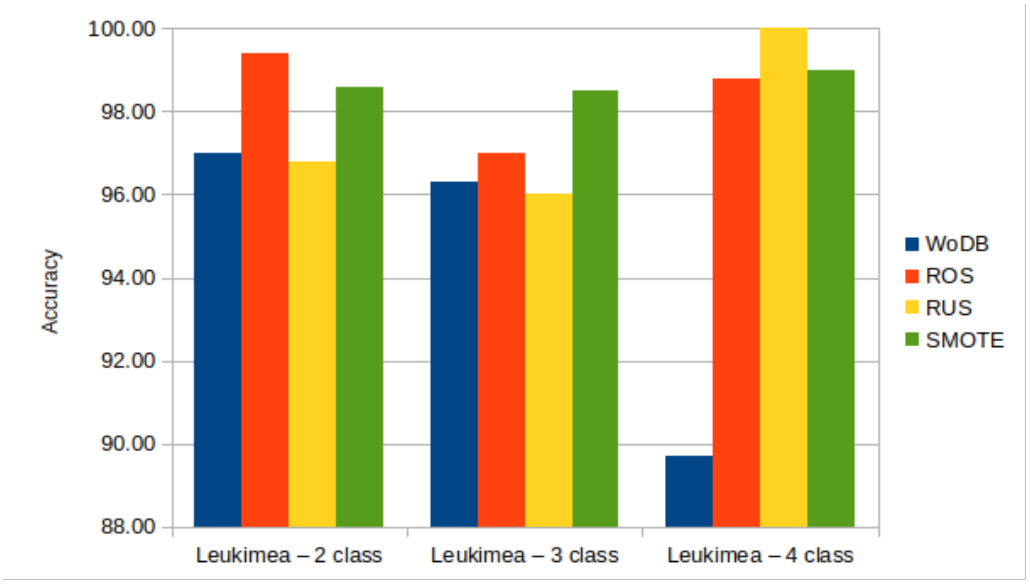

Fig. 2. Comparison of KNN performance on Leukemia microarray datasets between unbalanced and balanced ones

Figure 2 is a comparison graph of the highest accuracy of simulation results using the KNN algorithm between unbalanced data and balanced data using the ROS, RUS, and SMOTE techniques, which are applied to three types of leukemia microarray data, namely leukemia data with two classes, three classes, and four classes. The graph shows that in the leukemia dataset with two classes and leukemia with three classes, both the dataset with balancing and the dataset without balancing have quite good accuracy. However, in the dataset with four classes, the unbalanced data has a much lower accuracy than the balanced data. The difference in accuracy that occurs is significant; in data without balancing, the accuracy only reaches 89.7%, but with balancing it becomes 98.8% with ROS and 99.0% with SMOTE. Interestingly, RUS produces perfect accuracy of 100.0%, although based on previous analysis, this result appears to be an anomaly, because at some K values, RUS actually produces low accuracy. In general, oversampling methods, especially SMOTE, proved to be the most stable and superior across variations in the number of classes, while RUS tended to be less stable despite occasionally recording extreme results.

IV. Conclusion

This study has evaluated the performance of the kNN algorithm in classifying three leukemia microarray datasets that have different class structures (two, three, and four classes), as well as unbalanced data conditions. To overcome the problem of data imbalance, this study has applied three data balancing techniques, namely RUS, ROS, and SMOTE. Simulation results show that oversampling techniques, namely ROS and SMOTE, are able to consistently improve the performance of kNN on all three datasets. On the two-class dataset, ROS gives the best results with accuracy and F1-score reaching 99.4% at k = 3. On the three-class dataset, SMOTE gives the most optimal performance, with the highest accuracy and F1-score of 98.5% at k = 2 to k = 5. For the four-class dataset, the performance improvement is very significant after the balancing technique is applied, where SMOTE and ROS increase the accuracy from 89.7% (without balancing) to 99.0% and 98.8%, respectively. Meanwhile, although RUS has achieved perfect accuracy of 100% under certain conditions, its performance tends to be unstable and even drops drastically at other k values. Overall, SMOTE is proven to be the most stable and effective balancing technique for various class structures. This study shows that balancing strategies play an important role in improving the accuracy and stability of classification models, especially on complex and imbalanced microarray data.

References

[1] Peng, Y. (2006). A novel ensemble machine learning for robust microarray data classification. Computers in Biology and Medicine, 36(6), 553-573.

[2] Bolón-Canedo, V., Sánchez-Marono, N., Alonso-Betanzos, A., Benítez, J. M., & Herrera, F. (2014). A review of microarray datasets and applied feature selection methods. Information sciences, 282, 111-135.

[3] Alrefai, N., & Ibrahim, O. (2022). Optimized feature selection method using particle swarm intelligence with ensemble learning for cancer classification based on microarray datasets. Neural Computing and Applications, 34(16), 13513-13528.

[4] Zeng, Y., Zhang, Y., Xiao, Z., & Sui, H. (2025). A multi-classification deep neural network for cancer type identification from high-dimension, small-sample and imbalanced gene microarray data. Scientific Reports, 15(1), 5239.

[5] Kumar, P., Bhatnagar, R., Gaur, K., & Bhatnagar, A. (2021, March). Classification of imbalanced data: review of methods and applications. In IOP conference series: materials science and engineering (Vol. 1099, No. 1, p. 012077). IOP Publishing.

[6] Wu, G., & Chang, E. Y. (2005). KBA: Kernel boundary alignment considering imbalanced data distribution. IEEE Transactions on knowledge and data engineering, 17(6), 786-795.

[7] Jadhav, A., Mostafa, S. M., Elmannai, H., & Karim, F. K. (2022). An empirical assessment of performance of data balancing techniques in classification task. Applied Sciences, 12(8), 3928.

[8] Dal Pozzolo, A., Caelen, O., & Bontempi, G. (2015, August). When is undersampling effective in unbalanced classification tasks?. In Joint european conference on machine learning and knowledge discovery in databases (pp. 200-215). Cham: Springer International Publishing.

[9] Susan, S., & Kumar, A. (2021). The balancing trick: Optimized sampling of imbalanced datasets—A brief survey of the recent State of the Art. Engineering Reports, 3(4), e12298.

[10] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[11] Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., & Khraisat, A. (2024). Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. Journal of Big Data, 11(1), 113.

[12] Juliusdottir, T., Keedwell, E., Corne, D., & Narayanan, A. (2005, November). Two-phase EA/k-NN for feature selection and classification in cancer microarray datasets. In 2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (pp. 1-8). IEEE.

[13] Deegalla, S., & Boström, H. (2007). Classification of microarrays with knn: comparison of dimensionality reduction methods. In Intelligent Data Engineering and Automated Learning-IDEAL 2007: 8th International Conference, Birmingham, UK, December 16-19, 2007. Proceedings 8 (pp. 800-809). Springer Berlin Heidelberg.

[14] Meesad, P., & Hengpraprohm, K. (2008, June). Combination of knn-based feature selection and knnbased missing-value imputation of microarray data. In 2008 3rd International Conference on Innovative Computing Information and Control (pp. 341-341). IEEE.

[15] Parry, R. M., Jones, W., Stokes, T. H., Phan, J. H., Moffitt, R. A., Fang, H., ... & Wang, M. D. (2010). k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction. The pharmacogenomics journal, 10(4), 292-309.

[16] Keerin, P., & Boongoen, T. (2021). Improved knn imputation for missing values in gene expression data. Computers, Materials and Continua, 70(2), 4009-4025.

[17] Wojtowicz, A., Mrukowicz, M., Gałka, W., Balicki, K., Rzasa, W., & Bentkowska, U. (2024). Binary ensemble kNN based classifier for microarray datasets. Procedia Computer Science, 246, 4411-4420.

[18] Zhu, Z., Ong, Y. S., & Dash, M. (2007). Markov blanket-embedded genetic algorithm for gene selection. Pattern Recognition, 40(11), 3236-3248.

[19] Omuya, E. O., Okeyo, G. O., & Kimwele, M. W. (2021). Feature selection for classification using principal component analysis and information gain. Expert Systems with Applications, 174, 114765.

[20] Kumar, V., Lalotra, G. S., Sasikala, P., Rajput, D. S., Kaluri, R., Lakshmanna, K., ... & Uddin, M. (2022, July). Addressing binary classification over class imbalanced clinical datasets using computationally intelligent techniques. In Healthcare (Vol. 10, No. 7, p. 1293). MDPI.

[21] Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905.

[22] Henderi, H., Wahyuningsih, T., & Rahwanto, E. (2021). Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. International Journal of Informatics and Information Systems, 4(1), 13-20.

[23] Nasution, D. A., Khotimah, H. H., & Chamidah, N. (2019). Perbandingan normalisasi data untuk klasifikasi wine menggunakan algoritma K-NN. Comput. Eng. Sci. Syst. J, 4(1), 78.

[24] Pandey, A., & Jain, A. (2017). Comparative analysis of KNN algorithm using various normalization techniques. International Journal of Computer Network and Information Security, 10(11), 36.

[25] Chamidah, N., & Salamah, U. (2012). Pengaruh normalisasi data pada jaringan syaraf tiruan backpropagasi gradient descent adaptive gain (bpgdag) untuk klasifikasi. ITSMART: Jurnal Teknologi dan Informasi, 1(1), 28-33.