JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1062-1069
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6306

Handling Unbalanced Data Sets Using DBMUTE and NearMiss Methods to Improve Classification Performance of Yeast Data Sets

Bima Mahardika Wirawan*, Mahendra Dwifebri P, Fhira Nhita

Fakultas Informatika, Informatika, Telkom University, Bandung, Indonesia
Email: [1,*]bimamahardika@student.telkomuniversity.ac.id, [2]mahendradp@telkomuniversity.ac.id,
[3]fhiranhita@telkomuniversity.ac.id
Correspondence Author Email: bimamahardika@student.telkomuniversity.ac.id

Abstract−Yeast vacuole biogenesis was chosen as a model system for organelle assembly because most vacuole functions can be used for vegetative cell growth. Therefore it is possible to generate an extensive collection of mutants with defects in unbalanced vacuole assembly. With this in mind, we must find the structural balance of data in yeast. Imbalanced data is when there is an unbalanced distribution of data classes and the number of data classes is either more or lower than the number of other data classes. Our method uses the f1 score performance matrix method and the balanced accuracy on DBMUTE and NearMiss undersampling. Previously, only a few studies explained the results of using a performance matrix and balanced accuracy. Then, find out the performance results of the f1 score and balanced accuracy and get the best score from the yeast data sets. In the study, a comparison between the imbalanced data sets using the undersampling method. Furthermore, to obtain the performance matrix results, use the f1 score and balance accuracy. After testing five yeast data sets, we performed an average f1 score and balance accuracy with the highest average NearMiss f1 score of 62.23% and the highest average balanced accuracy of 78.59%.

Keywords: Imbalance Data; DBMUTE; NearMiss; Support Vector Machine; Undersampling

1. INTRODUCTION

Organelle biogenesis in eukaryotic cells requires the selective assembly of specific proteins, lipids, and other constituents from a shared pool of biosynthetic intermediates. An extensively present eukaryotic microbe is yeast. Yeast vacuole biogenesis was chosen as a model system. It has garnered much interest due to its rapid development and remarkable metabolic efficiency. Therefore, it is possible to generate extensive collections of mutants with defects in unbalanced vacuole assembly. With this in mind, we must find the structural balance of data in yeast[1].

Imbalanced data is when there is an unbalanced distribution of data classes and the number of data classes is either more or lower than the number of other data classes[2]. In this research, unbalanced data is a situation with significantly more observations in one class than in the other. This problem is predominant in cases where anomaly detection is critical, for example, fraud detection in banks, healthcare, insurance, etc.

In 2021, Patel H. et al. [3] discusses determining and evaluating the classification's performance for levels where imbalances have different classes and need to design an experimental setup. The categorization of unbalanced data may be improved by using a variety of crisp and fuzzy closest-neighbor algorithms. Distributions of the source and target projects in these situations reveal the class imbalance. This method uses naive Bayes to learn global information, while the K-nearest neighbor algorithm is utilized to learn local information. High performance is being produced in the prediction of software defects by this hybrid technique. When in the implementation of the use of four methods of data sets yeast, generate value f-measure on K of the NWKNN of 16 %, method of Adpt-NWKNN by 20 %, method of Fuzzy-NWKNN 19 %, and methods of Weighted Fuzzy Adpt KNN of 38 %.

In 2019, Cao L. et al. [4] discusses imbalanced data on Pima, Haberman, Ecoli3, Glass1, Glass16vs2, and Shuttle2vs5 data sets. In this study, the traditional classification of the proposed algorithm based on the symmetry form of the class distribution hypothesis was carried out. This study compared the performance of six algorithms to validate the CUS method's efficacy for unbalanced data classification values. On one of the Pima data sets, this study employed the SVM classification model, which provided an average AUC of 77.5% and a gmean of 68.5%, whereas SMOTE produced an average AUC of 82.1% and a gmean of 75.3%.

In 2020, Untoro Meida C. et al. [5] discuss issues that often occur when classifying undersampling data. The cause of the error is that when classifying is more prone to majority data, so the accuracy obtained is low in the minority data. Here, researchers manipulate data samples and use their algorithms. The classification method provides accuracy scores for all data by eliminating some of its minority classes. Then it deletes them for all data considered to be the majority class. Researchers use Decision Tree classification, KNN classification, Naive Bayes classification, dan SVM classification. In implementing f-measure into the data sets yeast, Decision Tree gets 94.10%, Naive Bayes gets 87.40%, KNN gets 91.20%, and SVM gets 84.40%.

In 2020, Huang B. et al. [6] discusses imbalanced data using oversampling to determine the effectiveness of the given model. The six models ADASYN, SMOTE, RUS, EasyEnsembleClassifier, Fast-CBUS 23, and TomekLinks are all used in this work. This study suggests the use of KU-MSVM, an imbalanced classification system based on clustering and support vector machines. SMOTE and ADASYN sampling are used in this study. The comparison of the positive and negative values of the SMOTE and ADASYN models on the Letter data sets

Submitted: **09/06/2023**; Accepted: **17/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1062-1069
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6306

shows that the positive value of the SMOTE model on the Letter data sets is 16.8%, the positive value of the ADASYN model is 11.8%, and the negative value of the SMOTE model on the Letter data sets is 98.2%.

In 2018, Yu L. et al. [7] discusses the DBN-based SVM resampling ensemble learning technique offered a solution for the problem of data imbalance in credit risk classification. In this study, a competitive ensemble of the DBN model is conducted. Performance in classification can be improved by resampling integration, especially when dealing with the problem of excessively imbalanced data sets. The MAJ-SVM model's accuracy value in the research employing German credit result data sets is 71.68%, while the DBN-SVM model's accuracy value is 69.82%.

This paper uses SVM classification to compare the results of different sampling methods, namely Destiny-Based Majority Sampling and Near Sampling Techniques. We applied undersampling techniques to correct class imbalances in yeast data sets. Previous studies rarely compare the performance of undersampling techniques, particularly in using undersampling DBMUTE and NearMiss on yeast data sets. We use the two performances to share the results of imbalanced data on the data sets. Furthermore, to find out the extent to which the use of f1 score performance and balanced accrued good value against the imbalanced data sets.

In undersampling, several algorithms can be used. In this research, we use DBMUTE and NearMiss undersampling. The DBMUTE undersampling algorithm eliminates majority noise cases that overlap with minority cases. And the NearMiss algorithm reduces information loss during undersampling in the majority class[8]. In this study, we also used a classification method, namely Support Vector Machine. This classification method was chosen because it deals with imbalanced data well. The categorization operates by creating an N-dimensional hyperplane that divides the data into two groups in the best way possible. This research uses references from previous studies relevant to imbalanced data, classification methods, undersampling methods, preprocessing, and evaluation stages. Undersampling is often used to balance data by reducing the size of abundant classes.

2. RESEARCH METHODOLOGY

We built a system for handling unbalanced data sets using DBMUTE and NearMiss methods to improve classification performance of yeast data sets. Here is the system we are going to build in Figure 1:

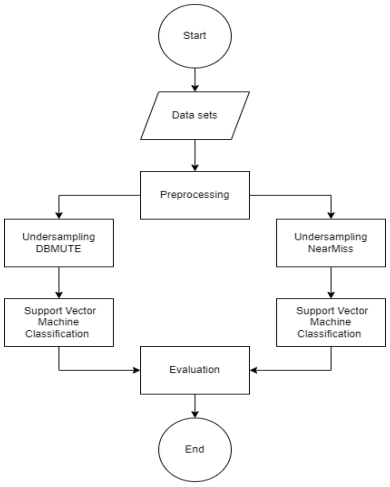

Figure 1. System Design

2.1 Data sets

The data sets collected are imbalanced data from the Keel website. KEEL is open-source software that can be used for various discovery tasks that require valid knowledge data. KEEL has been designed for research and education. The data used in this study only uses yeast data as much as five data sets with unbalanced differences with IR 2.46 - 23.1. Table 1 describes the data yeast used:

Table 1. Research Data Yeast

Data sets	#Feature	#Instances	Positive Instances (%)	Negative Instances (%)	Imbalanced Ratio (IR)	Acronym
Data sets 1	8	1484	28.9 %	71.1 %	2.46	yeast1
Data sets 2	8	1484	10.99 %	89.01 %	8.1	yeast2
Data sets 3	8	514	9.92 %	90.08 %	9.08	yeast3
Data sets 4	8	1004	9.86 %	90.14 %	9.14	yeast4
Data sets 5	8	482	4.15 %	95.85 %	23.1	yeast5

Submitted: **09/06/2023**; Accepted: **17/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1062-1069
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6306

2.2 Preprocessing

Preprocessing methods have been taken into account as a design parameter for deep learning tasks that recognize human action, and they have been proven to be useful[9]. Data processing is the first and most crucial step in the processing of data from data mining. The data will be cleaned and corrected at this stage before being classified. In this research, the preprocessing stage is Missing Values and Duplicate Data. In addition, the preprocessing stage also changes the data type in the label column. In changing the data type, namely changing the label in the positive data sets to 1 and the label in the harmful data sets to 0.

Figure 2. Preprocessing Stage

The missing value is a stage to check whether the data is missing or not readable. Assume that whatever the attribute is, it is not present throughout training. There are several ways to resolve missing values on data sets. First, there are researchers who directly model the dataset by looking at the missing values. Second, by placing missing values to get the complete data sets[10]. The term "duplicate data" refers to entries that users repeatedly contribute to the same database[11]. Label Encoder is a process to convert word labels into numbers in data sets that will be used. When we do classification, we sometimes deal with data in the form of words or something else. The method used is an ordinal encoding which gives an integer to each category[12].

2.3 Split Data

We split the data as 80% train and 20% test. Table 2 describes the results of splitting the data set:

Table 2. Split Data Result

Data set	#Instances		#Positive Instances		#Negative Instances	
	Train	Test	Train	Test	Train	Test
yeast1	1187	297	343	86	844	211
yeast2	1187	297	124	39	1063	258
yeast3	411	103	44	7	367	96
yeast4	803	201	81	18	722	183
yeast5	385	97	15	5	370	92

2.4 Density-Based Majority Undersampling Technique (DBMUTE)

The majority noise cases that overlap with minority cases may be eliminated using DBMUTE's method. The Safe-Level-SMOTE idea determines the safety level for most removal cases. MUTE enhances the prediction rate for minority classes and cuts down on classifier development time by simplifying the data sets. We implement the DBMUTE undersampling approach using a different algorithm. This method is called DBSCAN. One clustering approach that takes advantage of the density of data is the DBSCAN algorithm. The primary benefit of the DBSCAN algorithm in the detection of clusters is its ability to recognize clusters with any forms DBSCAN [13]. DBSCAN is one of the most used algorithms for density-based clustering. DBSCAN can produce clusters in any shape based on density-based clusters. Based on the neighborhood radius Eps and the minimal number of points in the neighborhood MinPts, DBSCAN classifies a collection of points into three categories: the cluster's core points, border points, and outliers. There are MinPts or more points around a core point in the Eps neighborhood[14].

Table 3. DBSCAN Algorithm

No	Description
1.	First, initialize the algorithm by specifying two parameters, eps (epsilon) and min_samples.
2.	Randomly select a data point that has never been visited.
3.	Select and determine whether the selected data point is a point of a core.
4.	Create a new cluster and include it in the collection of clusters if the chosen point is a core point.
5.	Then select each neighbor of the selected point (including the selected point itself).
6.	Then repeat steps 2 to 5 until all data points have been visited.
7.	The algorithm will stop and find the final clustering result of the formed clusters and their noisy points.

2.5 NearMiss

When the majority class is undersampled, NearMiss minimizes information loss. The term "NearMiss" refers to a group of undersampling techniques that choose data depending on how close the minority class data are to the data from the majority class. There are three NearMiss variants: NearMiss 1, NearMiss 2, and NearMiss 3[8]. The

Submitted: **09/06/2023**; Accepted: **17/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1062-1069
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6306

NearMiss itself is the outcome of many approaches to lower the size of the majority class depending on the distance. It operates by arbitrarily eliminating some of the dataset's majority samples. When two classes are relatively near to one another, this is accomplished by deleting instances from their majority classes, which helps the classification process and reduces the classification size. The social class to which most people belong[15].

Table 4. NearMiss Algorithm

No	Description
1.	It begins by determining the separation between each point in the majority class and the minority class.
2	Choose majority class instance that is closest to the minority class.
3	Then, if there is an instance value of the majority class, the algorithm returns the instance value of the minority class.

2.6 Support Vector Machine (SVM)

SVM is a powerful technique for creating a classification. The objective is to establish a decision boundary between two classes that permits the prediction of one or more feature vectors using a table[16]. SVM is to use a surface that optimizes the margin to divide various classes in the training set. The main goal of pattern classification is to create a model that performs as well as possible given the training set of data[17]. SVM seeks to maximize the distance between classes to locate the optimum hyperplane. The distinction between classes is referred to as a hyperplane. SVM is systematically more accurate than other classification approaches and provides a more optimum strategy for classification modeling. SVM maximizes the separation between classes and maximizes the margin to discover the optimum hyperplane[18].

2.7 Evaluation

The evaluation stage, we used a confusion matrix. The purpose of the confusion matrix is to find information factually. A data matrix usually evaluates the performance of a classification system. Table 5 is a table view of confusion matrix to be used:

Table 5. Formula Confusion Matrix

Confusion Matrix		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

Values for True Positive (TP) is a positive state, and the predicted value is actual. Meanwhile, values for True Negative (TN) is a negative state with an actual predicted value. Furthermore, a False Positive (FP) is a negative state, but the predicted value is correct. Meanwhile, a False Negative (FN) is a positive state with a wrong predicted value.

In measuring the performance of the system, we have built. We use the calculations' evaluations such as precision, recall, balanced accuracy, f1 score, specificity, and sensitivity. Following are the formulas for calculating the evaluation:

Precision measures accuracy if a certain class has been predicted[19]. Here is the precision formula:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

The recall is the prediction model's capacity to choose examples from a specific class[20]. Here is the recalled formula:

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

The F1 score is used to compare the average precision and recall values[21]. Here is the f1 score formula:

$$F1\ Score = \frac{(Precision \times Recall)}{(Precision + Recall)} \times 2 \quad (3)$$

Balanced accuracy is the average detection rate obtained in both classes[22]. Here is the balanced accuracy formula:

$$Balanced\ Accuracy = \frac{Specificity+Sensitivity}{2} \quad (4)$$

Specificity is the correct negative proportion identified correctly by the diagnostic test, which shows how well the test identifies normal condition[23]. Here is the specificity formula:

$$Specificity = \frac{TN}{TN+FP} \quad (5)$$

Sensitivity evaluates how well the test detects a positive disease[23]. Here is the sensitivity formula:

Submitted: **09/06/2023**; Accepted: **17/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1062-1069
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6306

$$Sensitivity = \frac{TP}{TP+FN} \quad (6)$$

3. RESULT AND DISCUSSION

In this evaluation, we tested the success of the constructed system by referring to the f1 score and balanced accuracy. We built this system with multiple stages. We used data sets yeast as research material. In the first step, preprocessing consists of Missing, Duplicate Data and Label Encoder values. After Preprocessing, the data is split with a distribution of 80% training and 20% test data. Then enter the next stage, which is undersampling. The data is tested using the NearMiss undersampling and DBMUTE undersampling methods at this stage. The class distribution on the data label is not balanced, so it needs to be balanced. In modeling, we use the Support Vector Machine as its classification. In the first scenario, we compare five yeast datasets without using the undersampling method. In the second scenario, we compare five yeast data sets using DBMUTE sampling methods and the Supported Vector Machines classification. In the third scenario, we compare five data sets using NearMiss sampling methods and the Supported Vector Machines classification.

3.1 Performance result without undersampling method

In scenario one, testing was carried out to compare five yeast data sets without using the undersampling. The results of the scenario one are shown in table 6:

Table 6. Performance Results From Scenario One

Data set	Precision	Recall	F1 Score	Balanced Accuracy	Specificity	Sensitivity
yeast1	82.60%	22.09%	34.86%	60.09%	98.10%	22.09%
yeast2	92.30%	30.76%	46.15%	65.19%	99.61%	30.76%
yeast3	**100%**	42.85%	60%	71.42%	**100%**	42.85%
yeast4	90.90%	55.55%	68.96%	77.50%	99.45%	55.55%
yeast5	**100%**	**60%**	**75%**	**80%**	**100%**	**60%**
Average	93,16%	42,25%	56,99%	70,84%	99,43%	42,25

3.2 Performance result of DBMUTE

In Scenario two, it was performed to compare five sets of yeast data using DBMUTE in undersampling. The results of the scenario two are shown in Table 7:

Table 7. Performance Results From Scenario Two

Data set	Precision	Recall	F1 Score	Balanced Accuracy	Specificity	Sensitivity
yeast1	41.17%	8.13%	13.59%	51.70%	95.26%	8.13%
yeast2	5.55%	2.56%	3.50%	47.98%	93.41%	2.56%
yeast3	**66.66%**	28.57%	40%	63.76%	**98.95%**	28.57%
yeast4	33.33%	16.66%	22.22%	56.69%	96.72%	16.66%
yeast5	50%	**60%**	**54.54%**	**78.36%**	96.73%	**60%**
Average	39,34%	23,18%	26,77%	59,69%	96,21%	23,18%

The following is Table 8 of the data proportion results after training using DBMUTE.

Table 8. The Resulting Proportion of DBMUTE

Data sets	Training Data Before DBMUTE		Training Data After DBMUTE	
	#Positive	#Negative	#Positive	#Negative
yeast1	343	844	171	1016
yeast2	124	1063	171	1016
yeast3	44	367	74	337
yeast4	81	722	149	654
yeast5	15	370	57	328

The results of the tests we conducted on undersampling DBMUTE. As shown in Table 8, the results of the data train on DBMUTE yeast1 and yeast2 are the same. So the results of training data are less than maximal. The causes of yeast1 and yeast2 have similarities in ineffective epsilon and min-sample parameters, resulting in poor classification performance.

3.3 Performance result of NearMiss

In scenario three, tests were conducted to compare five data sets yeast using NearMiss undersampling. The results of the scenario three are shown in table 9:

Submitted: **09/06/2023**; Accepted: **17/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1062-1069
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6306

Table 9. Performance Results From Scenario Three

Data set	Precision	Recall	F1 Score	Balanced Accuracy	Specificity	Sensitivity
yeast1	60.60%	46.51%	52.63%	67.09%	87.67%	46.51%
yeast2	49.15%	74.35%	59.18%	81.36%	88.37%	74.35%
yeast3	80%	57.14%	66.66%	78.05%	98.95%	57.14%
yeast4	44.11%	**83.33%**	57.69%	**86.47%**	89.61%	**83.33%**
yeast5	**100%**	60%	**75%**	80%	**100%**	60%
Average	66,77%	64,26%	62,23%	78,59%	92,92%	64,26%

The following is Table 10 of the data proportion results after training using NearMiss.

Table 10. The Resulting Proportion of NearMiss

Data sets	Training Data Before NearMiss		Training Data After NearMiss	
	#Positive	#Negative	#Positive	#Negative
yeast1	343	844	343	343
yeast2	124	1063	124	124
yeast3	44	367	44	44
yeast4	81	722	81	81
yeast5	15	370	15	15

3.1.4 Analysis of Experiment Result

We compared the performance of imbalanced data without the undersampling method with an undersampling method. We use two matrix performances, namely the f1 score and the balanced accuracy score. Figure 3. F1 score values for the five data sets using undersampling

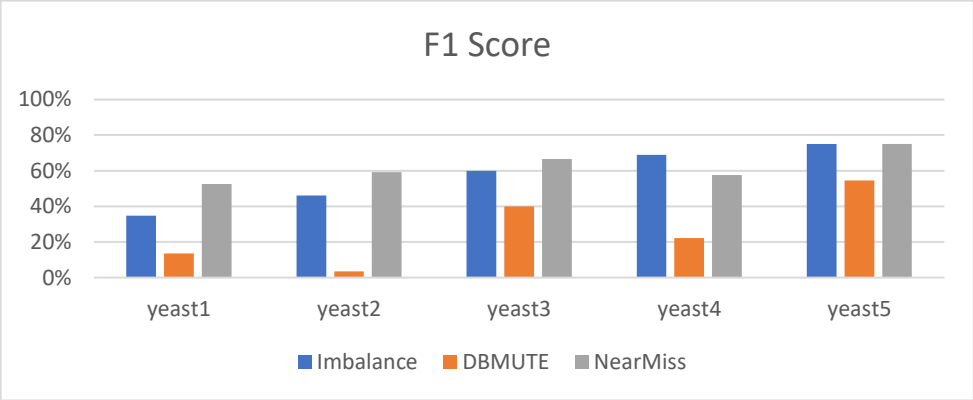

Figure 3. Result F1 Score

Based on the F1 score, imbalanced data sets using DBMUTE yielded less significant results on five data sets, whereas imbalanced data sets using NearMiss achieved prominent results, although there was a decrease in F1 score for the fourth data sets in NearMiss. DBMUTE has less significant results because IR in yeast2, yeast3, and yeast4 have similar data. From the f1 score results, it can be proven that IR values are not biased toward undersampling. The metric only considers trade-offs between precision and recall and needs to account for accuracy for each class individually and calculates the mean across all classes.

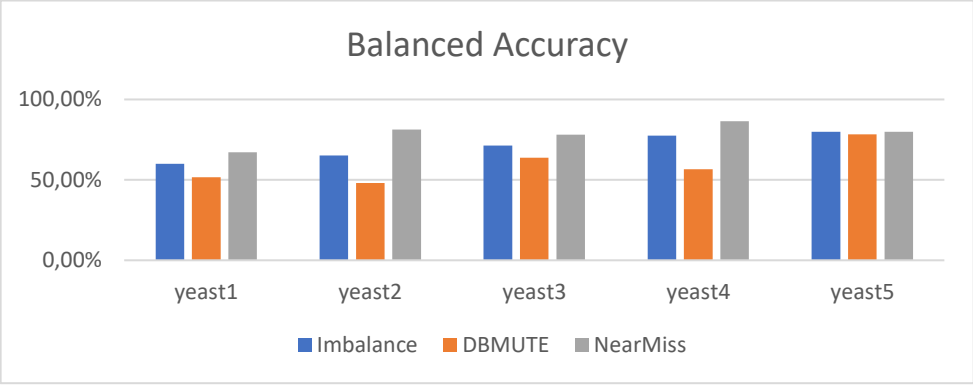

Figure 4. The result from Balanced Accuracy

Submitted: **09/06/2023**; Accepted: **17/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1062-1069
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6306

Based on the results of balanced accuracy and imbalance from five data sets, DBMUTE obtained insignificant results. However, in the fifth yeast based on balanced accuracy results from imbalanced, DBMUTE, and NearMiss, only 1.64%. Then in the fifth yeast, the ratio between DBMUTE and NearMiss was relatively good, with a difference of 1.64%. There is a difference in the f1 score and balance accuracy because it has a difference in the value of the confusion matrix, and there is a difference in the formula for calculating the value. Thus, this also affects the results of undersampling karmically based on the f1 score and balanced accuracy.

In the undersampling test, f1 score on the DBMUTE is less significant because it is not good in the epsilon and min_sample selection. Meanwhile, the comparison of balanced accuracy between DBMUTE and NearMiss in the fifth data sets is exact.

The results of calculating the average value and balanced accuracy of the data sets yeast are in Table 11. It is evident from the outcomes of the f1 score that and balance accuracy of NearMiss gets higher than the results of imbalanced and DBMUTE.

Table 11. Average F1 Score and Balanced Accuracy

Average	Imbalanced	DBMUTE	NearMiss
F1 Score	56,99%	26,77%	**62,23%**
Balanced Accuracy	70,84%	59,69%	**78,59%**

4. CONCLUSION

Based on the test results and analysis, a study was conducted on Handling Unbalanced Data Sets Using DBMUTE and NearMiss Methods to Improve Yeast Data Set Classification Performance. Based on the f1 score, unbalanced using NearMiss is a good result compared to unbalancing using DBMUTE. DBMUTE has less significant results because IR in yeast 2, yeast 3, and yeast 4 have similar data. The training results of yeast1 and yeast2 are similar due to the less effective epsilon and min sample parameters. However, in the fifth yeast based on balance accretion results from reciprocity, DBMUTE, and NearMiss, only a 1.64% difference was found. Furthermore, the ratio of DBMUTE and NearMiss in the fifth yeast also increases by 1.64%. There is a difference in the f1 score and balance accuracy because it has a difference in the value of the confusion matrix, and there is a difference in the formula for calculating the value. Calculation of the average f1 score and balanced accuracy of all five data sets yields the highest f1 score of 62.23% and the highest balance accuracy of 78.59%.

REFERENCES

[1] G. Qadir, "Yeast a magical microorganism in the wastewater treatment," *~ 1498 ~ J. Pharmacogn. Phytochem.*, vol. 8, no. 4, pp. 1498–1500, 2019.
[2] R. Siringoringo, "Klasifikasi data tidak Seimbang menggunakan algoritma SMOTE dan k-nearest neighbor," *J. ISD*, vol. 3, no. 1, pp. 44–49, 2018.
[3] H. Patel, D. S. Rajput, O. P. Stan, and L. C. Miclea, "A new fuzzy adaptive algorithm to classify imbalanced data," *Comput. Mater. Contin.*, vol. 70, no. 1, pp. 73–89, 2021, doi: 10.32604/cmc.2022.017114.
[4] L. Cao and H. Shen, "Imbalanced data classification using improved clustering algorithm and under-sampling method," *Proc. - 2019 20th Int. Conf. Parallel Distrib. Comput. Appl. Technol. PDCAT 2019*, pp. 358–363, 2019, doi: 10.1109/PDCAT46702.2019.00071.
[5] M. C. Untoro, M. Praseptiawan, M. Widianingsih, I. F. Ashari, A. Afriansyah, and Oktafianto, "Evaluation of Decision Tree, K-NN, Naive Bayes and SVM with MWMOTE on UCI Dataset," *J. Phys. Conf. Ser.*, vol. 1477, no. 3, 2020, doi: 10.1088/1742-6596/1477/3/032005.
[6] B. Huang, Y. Zhu, Z. Wang, and Z. Fang, "Imbalanced Data Classification Algorithm Based on Clustering and SVM," *J. Circuits, Syst. Comput.*, vol. 30, no. 2, 2021, doi: 10.1142/S0218126621500365.
[7] L. Yu, R. Zhou, L. Tang, and R. Chen, "A DBN-based resampling SVM ensemble learning paradigm for credit classification with imbalanced data," *Appl. Soft Comput. J.*, vol. 69, pp. 192–202, 2018, doi: 10.1016/j.asoc.2018.04.049.
[8] T. M. Alam *et al.*, "An investigation of credit card default prediction in the imbalanced datasets," *IEEE Access*, vol. 8, pp. 201173–201198, 2020, doi: 10.1109/ACCESS.2020.3033784.
[9] X. Zheng, M. Wang, and J. Ordieres-Meré, "Comparison of data preprocessing approaches for applying deep learning to human activity recognition in the context of industry 4.0," *Sensors (Switzerland)*, vol. 18, no. 7, 2018, doi: 10.3390/s18072146.
[10] Y. Luo, X. Cai, Y. Zhang, J. Xu, and X. Yuan, "Multivariate time series imputation with generative adversarial networks," *Adv. Neural Inf. Process. Syst.*, vol. 2018-December, no. NeurIPS, pp. 1596–1607, 2018.
[11] M. Z. H. Jesmeen *et al.*, "A survey on cleaning dirty data using machine learning paradigm for big data analytics," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 10, no. 3, pp. 1234–1243, 2018, doi: 10.11591/ijeecs.v10.i3.pp1234-1243.
[12] P. Purwono, A. Wirasto, and K. Nisa, "Komparasi Algoritma Machine Learning Untuk Klasifikasi Kelompok Obat," *Sisfotenika*, vol. 11, no. 2, p. 196, 2021.
[13] D. Deng, "DBSCAN Clustering Algorithm Based on Density," *Proc. - 2020 7th Int. Forum Electr. Eng. Autom. IFEEA 2020*, pp. 949–953, 2020, doi: 10.1109/IFEEA51475.2020.00199.
[14] B. Mirzaei, B. Nikpour, and H. Nezamabadi-Pour, "An under-sampling technique for imbalanced data classification based on DBSCAN algorithm," *8th Iran. Jt. Congr. Fuzzy Intell. Syst. CFIS 2020*, pp. 21–26, 2020, doi: 10.1109/CFIS49607.2020.9238718.
[15] F. E. Botchey, Z. Qin, and K. Hughes-Lartey, "Mobile money fraud prediction-A cross-case analysis on the efficiency of support vector machines, gradient boosted decision trees, and Naïve Bayes algorithms," *Inf.*, vol. 11, no. 8, 2020, doi: 10.3390/INFO11080383.
[16] S. Huang, C. A. I. Nianguang, P. Penzuti Pacheco, S. Narandes, Y. Wang, and X. U. Wayne, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics and Proteomics*, vol. 15, no. 1, pp. 41–51, 2018, doi: 10.21873/cgp.20063.
[17] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, no. xxxx, pp. 189–215, 2020, doi: 10.1016/j.neucom.2019.10.118.
[18] A. Fahmi Sabani, Adiwijaya, and W. Astuti, "Analisis Sentimen Review Film pada Website Rotten Tomatoes Menggunakan Metode SVM Dengan Mengimplementasikan Fitur Extraction Word2Vec," *e-Proceeding Eng.*, vol. 9, no. 3, p. 1800, 2022.

Submitted: **09/06/2023**; Accepted: **17/07/2023**; Published: **23/07/2023**

JURNAL MEDIA INFORMATIKA BUDIDARMA
Volume 7, Nomor 3, Juli 2023, Page 1062-1069
ISSN 2614-5278 (media cetak), ISSN 2548-8368 (media online)
Available Online at https://ejurnal.stmik-budidarma.ac.id/index.php/mib
DOI: 10.30865/mib.v7i3.6306

[19] N. Tri Romadloni, I. Santoso, and S. Budilaksono, "Perbandingan Metode Naive Bayes, Knn Dan Decision Tree Terhadap Analisis Sentimen Transportasi Krl Commuter Line," *J. IKRA-ITH Inform.*, vol. 3, no. 2, pp. 1–9, 2019.
[20] F. Ratnawati, "Implementasi Algoritma Naive Bayes Terhadap Analisis Sentimen Opini Film Pada Twitter," *INOVTEK Polbeng - Seri Inform.*, vol. 3, no. 1, p. 50, 2018, doi: 10.35314/isi.v3i1.335.
[21] N. Hendrastuty *et al.*, "Analisis Sentimen Masyarakat Terhadap Program Kartu Prakerja Pada Twitter Dengan Metode Support Vector Machine," *J. Inform. J. Pengemb. IT*, vol. 6, no. 3, pp. 150–155, 2021.
[22] R. Mehmood and A. Selwal, *Fingerprint biometric template security schemes: Attacks and countermeasures*, vol. 597. 2020.
[23] V. S. Spelmen and R. Porkodi, "A Review on Handling Imbalanced Data," *Proc. 2018 Int. Conf. Curr. Trends Towar. Converging Technol. ICCTCT 2018*, pp. 1–11, 2018, doi: 10.1109/ICCTCT.2018.8551020.

Submitted: **09/06/2023**; Accepted: **17/07/2023**; Published: **23/07/2023**