Available at https://jtsiskom.undip.ac.id (16 September 2020)
DOI:10.14710/jtsiskom.2020.13625

Jurnal Teknologi dan Sistem Komputer, 8(4), 2020, 276-283

A proposed method for handling an imbalance data in classification of blood type based on Myers-Briggs type indicator

Ahmad Taufiq Akbar*), Rochmat Husaini, Bagus Muhammad Akbar, Shoffan Saifullah

Department of Informatics, Faculty of Industrial Engineering, Universitas Pembangunan Nasional Veteran Yogyakarta
Jl. Babarsari 2, Kampus Unit 2, Yogyakarta, Indonesia 55281

How to cite: A. T. Akbar, R. Husaini, B. M. Akbar, and S. Saifullah, "A proposed method in handling an imbalanced data in classification of blood type based on Myers-Briggs type indicator," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 4, pp. 276-283, 2020. doi: 10.14710/jtsiskom.2020.13625, [Online].

Abstract – Blood type still leads to an assumption about its relation to some personality aspects. This study observes preprocessing methods for improving the classification accuracy of MBTI data to determine blood type. The training and testing data use 250 data from the MBTI questionnaire answers given by 250 respondents. The classification uses the k-Nearest Neighbor (k-NN) algorithm. Without preprocessing, k-NN results in about 32 % accuracy, so it needs some preprocessing to handle data imbalance before the classification. The proposed preprocessing consists of two-stage, the first stage is the unsupervised resample, and the second is the supervised resample. For the validation, it uses ten cross-validations. The result of k-Nearest Neighbor classification after using these proposed preprocessing stages has finally increased the accuracy, F-score, and recall significantly.

Keywords *– imbalance data; blood type; resample; k-nearest neighbor; MBTI*

I. Introduction

Psychology has become an education that rapidly developed in the world. The wide usage of psychological instruments indicates it in determining a person's character to study its relationship with other aspects [1]. Blood type is the one issue allegedly related to character. The are many people who still believe that a person's character indicates their blood type [2], [3].

Psychology has emerged several instruments to recognize personality types; one of those instruments is MBTI. The Myers-Briggs Type Indicator (MBTI) is a theory based on the personality profile using a questionnaire to identify psychological differences between individuals' perception of the environment and decision-making skills [4]. Personality type in MBTI is examined by identifying an individual preference based on four pairs of opposite preferences (Extrovert/ Introvert, Sensing/iNtuition, Thinking/ Feeling, and Perceiving/Judging), and MBTI describes how the personality type interacts with others and their environment [5]. MBTI contains many attributes that can classify a person in several personality types [6]. Many personality attributes belong to MBTI, so it is quite interesting to observe if MBTI might identify a person's blood type.

Previous studies also lead to the assumption about the relationship between health problems with blood type [7]. For example, the O blood type people are recommended to eat more meat than A, B, and AB blood type due to the research about O blood type commonly produce excessive antibodies that counter the lectin found in wheat and grains. People with O blood type must be less consuming grains but eating more meat [8].

Blood type is also assumed to correlate with certain person characters, as believed by Japanese society [3]. This assumption contradicts several studies that showed almost nothing correlation between character and blood type [8]. Nevertheless, this assumption continues to spread widely in Japan and other East Asian countries whose long tradition of understanding personality traits according to blood type. In theory, according to the Japanese, blood type is related to personality traits that conform to seriousness, enthusiasm level of boredom, friendliness, and individualism [1], [2]. Furthermore, there are still more personality indicators, so in this study, we use indicators based on MBTI features to classify blood type.

Machine learning is a part of artificial intelligence with the ability for reasoning, classification, and prediction. The machine learning calculates the level of closeness (similarity) between training data (in the knowledge base) with testing data. The conclusions or classes for each testing data can be determined based on the knowledge base's closest class. The process by this machine learning is called data mining [9].

Machine learning such as naive Bayes, Support Vector Machines (SVM), artificial neural networks, and k-Nearest Neighbor (k-NN) has been widely used in data mining [10]. The k-NN algorithm is included in lazy learners because it is a practical machine learning that did not require a training cycle [11]. Because of its simplicity, the k-NN algorithm is included in the ten most popular data mining algorithm [12].

However, high accuracy is one of the challenges in the performance of machine learning. One of the obstacles to achieving good accuracy is the unequal distribution of data in each class. Imbalance of class can occur because the number of minority classes is not balanced with the majority class, leading to uneven distribution of data and bias to majority class when

*) Correspondence author (Ahmad Taufiq Akbar)
Email: ahmadtaufiq.akbar@upnyk.ac.id

Submitted: 11 January 2020; Revised: 4 September 2020; Accepted: 11 September 2020; Published: 31 October 2020

classified [13]. This data imbalance is generally found in datasets related to health [10], [14]. Psychology and Blood Type might be related to health seen in some previous studies [7]-[9]; therefore, the data imbalance may occur in both.

The number of blood types in a large data set, for example, based on the world population, has the sequence from the largest to the smallest distribution as follows, O > A > B > AB [15]. That means the data imbalance in MBTI-Blood type data possibly occurs. Hence the imbalance data is our issue to propose some preprocessing method for balancing the class distribution then improve classification performance.

There are two ways to improve the classification accuracy of unbalanced data, both the algorithm and the data [16]. On the algorithm side, it can be reached by adding methods to strengthen the knowledge base that represents the minority class. On the data side, it can be reached through the resampling method. This method balances the data distribution by undersampling or oversampling. Undersampling is reducing the majority class sample while oversampling increasing the number of minority samples. It is reducing the percentage (ratio) of imbalanced classes [10], [16].

The resample method has been used in previous years for some research about the classification of unbalanced data. Chawla et al. [17] overcame class imbalance by SMOTE (Synthetic Minority Oversampling Technique) method, which still has been widely studied and still has some drawbacks due to its overfitting issues and unclear class boundaries between minorities and majority classes [14], [18]. The research was also carried out by developing a borderline-SMOTE with Gradient Boosting as a classifier [14], weighted-SMOTE with SVM as classifier [13], k-means-SMOTE with C.45, SVM, and naive Bayes as classifier [19] supervised resample-SMOTE [9], and SMOTE-SGA with naive Bayes as classifier [18]. Combining these methods has improved accuracy impressively but still limited only for a binary class dataset and based on SMOTE improvement.

Another study without the SMOTE method implemented random oversampling and undersampling. This study analyzed the audio spectrogram data in the music genre classification. However, it gave a lesser improvement in F-score than before resampling [16]. Another oversampling method, namely Sigma Nearest Oversampling based on Convex Combination (SNOCC), had been developed to improve SMOTE performance. However, SNOCC was still limited to handling continuous and ordinal data set [10].

This study aims to test the k-NN algorithm to classify blood types based on MBTI personality attributes, although the correlation between blood type and personality traits is not significant [1]. MBTI indicators are processed by machine learning to recognize blood type. Blood type and MBTI data obtained from respondents in our research had imbalanced distribution in the class of AB blood type. The proposed method in this study is expected to handle imbalanced data and to improve classification accuracy.

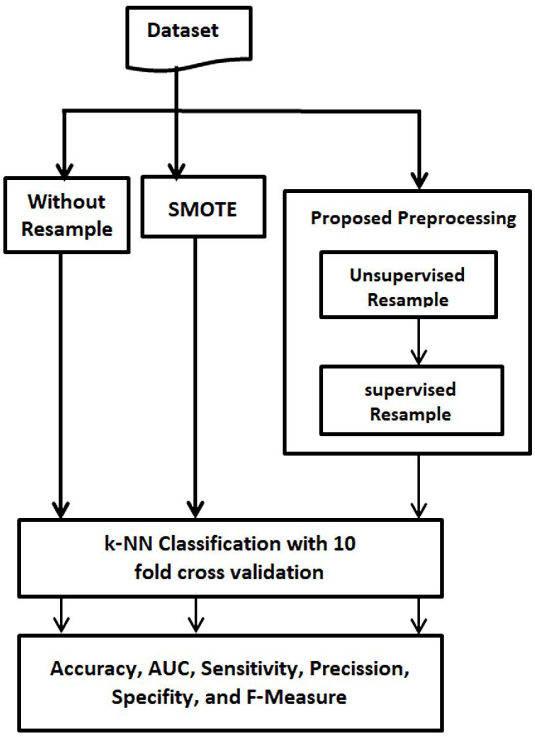

Figure 1. The workflow of this research

This proposed method for preprocessing consists of two stages, respectively, the first is unsupervised resample, and the second is supervised resample. Both of these methods are performed before the classification process. Other datasets from the UCI repository consist of Iris, Segment Test, Segment Challenge, liver disorder, breast cancer Winsconsin, Pima diabetes, yeast, unbalanced, and Soybeans between the proposed method and SMOTE in handling imbalanced data. This research does not concern about feature extraction or feature selection.

II. Research Methods

A. Workflow and dataset

This research is performed using WEKA software according to the workflow of this study. Our workflow for this research is illustrated in Figure 1. We proposed the preprocessing method that contains unsupervised resample (UR) and supervised resample (SR) to improve the classification of k-NN.

The proposed method in this research, as shown in Figure 1, is compared with the SMOTE and without resampling. The datasets in this comparison test are Blood type data with MBTI features, also some datasets from the UCI repository which are Iris, Unbalanced, Yeast, Segment Challenge, Segment Test, liver disorder, breast cancer Winsconsin, Pima-diabetes, and Soybeans. Those UCI datasets are used to analyze that the proposed preprocessing is not only coincidentally able to handle imbalanced data and improved classification performance of blood type-MBTI data. The preprocessing method also involved the value of bias to uniform class, so we can

choose the best bias values for improving the classification performance.

The MBTI- Blood type dataset from all respondent in this study consist of 250 instances. Every instance is loaded by values of 28 MBTI attributes and one blood type attribute (as a class) presented in Table 1. The total of 28 MBTI attributes represents the Introvert or Extrovert (I/E) dimension, Sensing or Intuition dimension (S/N), Thinking or Feeling (T/F) dimension, and Perceiving and Judging (P/J) dimension. Each dimension has seven attributes.

The MBTI personality type is determined by choosing one of the dominant characters from each dimension. The MBTI questionnaire generally has 25 attributes in every dimension. However, in this research, only seven attributes are selected randomly from each MBTI dimension. Numeric data load each attribute with a value 0 or 1. Every feature of *g* is 1 if the answer is the first option and 0 if the answer is the second option (opposed to the first option). Although the number of attributes in this research is not so many as standard MBTI questionnaires, the data seemly has a high dimension, as shown in Table 1.

High-dimensional data is also a challenge in data mining because it can increase calculation complexity in data interpretation and potentially reduces classification performance [20]. Besides that, the data in this research is possibly unbalanced. Therefore, this study aims to test blood types classification based on MBTI attributes using the proposed resample method and the k-NN classification.

Table 1. Data representation

Features (Attributes)	Description
g1 up to g7	Chosen answer from a pair of indicator in the dimension of I/E
g8 up to g14	Chosen answer from a pair of indicator in the dimension S/N
g15 up to g21	Chosen answer from a pair of indicator in the dimension T/F
g22 up to g28	Chosen answer from a pair of indicator in the dimension P/J
Class (Blood Type)	A / AB /O / B

B. K-Nearest neighbor

The k-Nearest Neighbor (k-NN) algorithm is often used in classification and clustering. It is called lazy learning because it directly calculates the closeness between testing data with training data without preceding the training process [11]. This algorithm included practical data mining methods and outperformed the error rate of optimized naive Bayes [12]. The kNN algorithm's performance is done by calculating the similarity (closeness) of the test data to the vector dimension's training data formed from features or attributes. Then this algorithm finds some k nearest samples of the training data. The majority class of the nearest k sample becomes the class for the tested sample [21]. A calculation of similarity in k-NN generally uses Euclidean distance, as expressed in Eq. 1. *Xi* and *Xj* represent two samples (testing samples and training samples) in vector domains. *Xis* and *Xjs* show the value of each attribute in sample *Xi* and sample *Xj*.

$$d(x_i, x_j) = \sqrt{\left(\sum_{s=i}^{m} |x_{is} - x_{js}|^2\right)} \qquad (1)$$

The k-NN algorithm performs better than naive Bayes in error rates, but k-NN still has some flaws requiring some attention. The efficiency of k-NN is influenced by multi-features, where not every feature has a role in supporting classification performance. Some features may be redundant, so it does not support improvements to the classification accuracy. The next weakness, if there is an uneven distribution in the sample of training data, then the difference in the number of the k-neighbors gives different classification results. Thus the determination of k becomes a unique problem in k-NN [22]. Although k-NN has these flaws, handling imbalance data is expected to improve the performance of k-NN classification.

C. Resample

Resample modifies training data with the oversampling or undersampling process. The essence of the oversampling process is to oversize the data distribution of minority classes [14]. Oversampling replicates data from minority classes to match the majority and adds information between instances in minority classes. The undersampling process reduces the majority class's frequency by replacing or deleting some data in the sample, so the data's composition is balanced [20]. However, undersampling has a risk of losing data in the majority class if the data apparently may improve the classification process [23], [24].

To optimize the handling of this imbalanced data, preprocessing by unsupervised and supervised resample is proposed in this study. The testing would be performed using the WEKA platform, and then the result would be compared with the previous study. The supervised resample used for the dataset contains a class with nominal data, and the unsupervised resample is used for data set whose class with numeric or nominal data.

SMOTE is one of the most popular methods used in handling imbalances of data distribution. SMOTE, as shown in Algorithm 1 [25], takes some *k* (number of neighbors) to be formed as additional data for minority classes [17]. Implementation of the SMOTE method can lead to overfitting [25]. This overfitting occurs because the SMOTE creates the same number of new samples between k-nearest neighbor data in the minority class. The boundary between the minority and the majority class became unclear. Thus, it causes the classifier to capture noise data that should be ignored; therefore, this model can lead to low accuracy [18].

New data is added to the minority class, so that is balanced with the number of majority class [17], [25], [26]. Besides the SMOTE algorithm as for comparison, unsupervised resample and supervised resample are chosen as the proposed method in this study for handling imbalance class. Both of those resample algorithms were imported from the WEKA library. The

supervised resample is shown in Algorithm 2. From line 10, the process replicates every the original sample randomly then pushes the replicated sample to the current position in each class.

There is a computation for determining the sample size based on the bias to uniform class, number of all instances, and the number of instances per class in the supervised resample. However, there is no bias value involved in the unsupervised resample algorithm, as shown in Algorithm 3. An unsupervised resample replicates every the original sample randomly. Regardless of what class it belongs to, it then pushes the replicated sample to its current position. The parameter i is slightly different from the supervised resample in the *push* position and without bias.

D. Evaluation

The accuracy is not sufficient as the sole benchmark of classification, so we also analyzed the results through Kappa statistic, AUC, recall (sensitivity), and F-measure values, based on the confusion matrix in Table 2. *TP* is an actual positive that is classified as positive. *FP* is negative but classified as positive. *FN* is an actual positive but classified as negative. *TN* is an actual negative that is classified as negative.

The formulas for recall (sensitivity), precision, and F-score (F-measure) expressed in Eq. 2 through Eq. 4, respectively. F-score is the harmonic mean of precision and recall. The F-score gets better if approaches one and get worse if approaches 0. A good F-score indicates the classification results are not too overfitting and not too underfitting because of the balance between recall and precision. So that a good F-score indicates the method is successful in handling the unbalanced classes. If the recall value increases and precision decreases, or vice versa, the F-score decreases or worsens.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Receiver Operating Characteristic (ROC) curve is a curve lying between the True Positive rate as the y-axis and the False Positive rate as the **x**-axis. The AUC is the area under the ROC curve, which means that the better AUC has a larger area. The better AUC indicates better classification performance. AUC formula is described in Eq. 5. An approximation is in the interval between *a* and *b* in the x-axis. With $f(x_i)$, the rectangle's height Δx is the rectangle's width under the curve. The Δx is getting smaller, whereas *n* is getting larger, *n* is subinterval between *a* and *b*.

$$\int_a^b f(x)\,dx = \lim_{n \to \infty} \sum_{i=0}^{n-1} f(x_i).\Delta x \tag{5}$$

Algorithm 1. SMOTE [17]

```
1: if (N<100) then
2:    randomize(m)
3:    m= (N/100) × m
4:    N=100
   end if
5: Sample[][] : array for minority class samples
6: Cgns ← 0
7: Synthetic[][]: array of synthetic samples
8: for i←1 to m do
9:    Compute( k_nearest_neighbor(i))
10:   nnarray← indices
11:   Collect(N,i,nnarray)
   end for
12: while ( N≠ 0)
13:    nn= random number between 1 to k
14:    for attr← 1 to numattrs
15:       dif= Sample[nnarray[nn]][attr] -Sample[i][attr]
16:       gap= random_number between 0 and 1
17:       Synthetic[Cgns][attr]=Sample[i][attr]+diff×gap
      end for
18:   Cgns ++
19:   N=N-1
   end while
20: return
```

Algorithm 2. Supervised resample (SR)

```
1: numClass[] ← count(numInstancePerClass)
2: for i ←1 to Nclasses do
3:    if (class[] ≠ empty) then c++
4:    numActualClass[]   c
  : end for
5: for i ←1  to numActualClass[] do
6:    SampleSize ←  (m_SampleSize_%   ×
        ((1 -  m_Bias) ×  numInstancesPerClass
        + m_Bias × numInstances / numActualClass[]))
7:    numInstancesToSample[i] ← SampleSize
8:    numEligible ←  numInstancesPerClass
9:    for j←1  to  numInstancesToSample[] do
10:      x ←  randomize(numEligible)
11:      push(instancesPerClass[i][x])
      end  for
     end for
```

Algorithm 3. Unsupervised resample (UR)

```
1: numEligible ←  number_of_Instances
2: SampleSize ← ( numEligible × SampleSize_%)
3: for (i ←0 to sampleSize) do
4:    x← randomize( numEligible )
5:    push( data.instance[x] )
   end for
```

Table 2. Confusion matrix

Actual	Classified as Positive	Classified as Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Kappa Statistic is based on how much agreement of observed accuracy or total accuracy (*Po*) and agreement of expected accuracy or random accuracy (*Pe*), as expressed in Eq. 6 through Eq. 8. The *n* is the number of classes, and *i* as the index of each class. Kappa Statistic value is ranged from 0 to 1. The higher is the better performance of classification [11].

$$P_o = \frac{Corectly_{classified}}{Total_{Instances}} \quad (6)$$

$$P_e = \sum_{i=1}^{n} \left(\frac{\left(TP_i + FN_i\right)\left(TP_i + FP_i\right)}{\left(Total_{Instances}\right)^2} \right) \quad (7)$$

$$Kappa\ Statistic = \frac{\left(Po - Pe\right)}{\left(1 - Pe\right)} \quad (8)$$

In the next testing, the preprocessing method also involved the value of bias to uniform class, so it can choose the best bias values for improving the classification performance.

III. Results and Discussion

The classification of MBTI-Blood type with 250 instances and 28 features using k-NN without resampling only results in an accuracy of 32% with k=1, as shown in Table 3. The accuracy is still relatively low because the classification algorithm commonly performs better when there is no imbalanced data [16]. We do not concern about feature selection because we know from the MBTI-blood typed dataset there is unbalanced data in the class of AB Blood type, so we concern more about instance resample than feature selection.

The blood type classification results using k-NN without resampling show that k=5 gives the highest Kappa, F-score, and accuracy score (Table 3). However, when we perform our proposed resample method (UR-SR), k=5 gives lesser accuracy than k=1. It may occur due to excessive overfitting when k is higher [25]. After that, we try to compare with the same dataset by using UR-SR resample method before classification. Using k=1 is the right choice than a higher k value. Therefore, the k=1 is used in our next testing and analyzing stage.

According to Table 3, when we observe at k=1, the accuracy before using the resample is only about 32 %, the Kappa is 0.0279, and the F-score or F-measure is only 0.314. The low accuracy indicates that misclassification of the MBTI-Blood type dataset still mainly exists, including the high false-positive rate and false-negative rate. Meanwhile, the low F-score also shows the high overfitting and underfitting in the classification results. The Kappa was only around 0.0279 also shows the low agreement rate between the observed accuracy and the expected accuracy. These three values indicate the impact of data imbalance that needs to be handled by the preprocessing method, such as the unsupervised and supervised resample that we propose in this study.

After determining k=1, the classification performance of 10 datasets is also compared between without using any resample, using SMOTE, and using UR-SR resample, as presented in Table 4. The bold values indicate the classification results after preprocessed using the proposed method, namely UR-SR. The UR-SR resample method has improved the classification performance on the entire dataset. The best improvement occurred in the MBTI-Blood type dataset. The F-score increased from 0.314 to 0.852, thus indicates the proportional increase in recall and precision due to overfitting and underfitting decreased.

Table 3. Classification performance of MBTI-blood type data based on k-value of k-NN

k	Resample method	Accuracy	Kappa statistic	F-score
k=1	none	0.32	0.0279	0.314
	UR-SR	**0.852**	**0.7944**	**0.853**
k=3	none	0.34	0.0403	0.318
	UR-SR	0.552	0.3662	0.552
k=5	none	**0.344**	**0.0421**	**0.320**
	UR-SR	0.508	0.2963	0.503
k=7	none	0.296	-0.0301	0.268
	UR-SR	0.46	0.2237	0.442

The accuracy increased from 0.32 to 0.85. Also, the recall increased from 0.32 to 0.852 because of the decrease in underfitting. A decrease in FN indicates it. The increase of recall also shows that the UR-SR method improves k-NN capability in classifying data that corresponds to the actual class. The precision increased from 0.315 to 0.858 due to the decrease of the False Positive (FP). FP's decrease indicates the proposed method successfully improves the classifier to ignore data that does not match with the actual class so that overfitting is reduced. The AUC has increased significantly from 0.496 to 0.925. It indicates that the UR-SR method is successful in increasing the TP rate and reducing the FP rate. The kappa value also highly increased from 0.0279 to 0.794, so it can be stated that the proposed method is capable of handling the data imbalance.

Meanwhile, the SMOTE does not significantly improve classification performance except for the MBTI-Blood type dataset, but still lesser than the proposed method. On the soybean dataset, SMOTE reduces the classification performance on accuracy from 91.2152 % to 91.0275 %. The recall slightly decreased around 0.002, so that SMOTE does not decrease the FN rate but increased. As in [10], [14], [18], adding unique samples by SMOTE does not reasonably fill the distribution space of the original sample so that it can obscure the interclass boundaries due to the increase of FN rate and FP rate. The SMOTE method also decreases the precision, F-score, AUC, and Kappa values with a slight decrease in the soybean dataset.

SMOTE can obscure the information of data interrelations in every class, thereby reducing the k-NN performance. As in previous studies, the information of interrelation data can be obscured due to the addition of unique samples that lead to generalization errors in the classifier [10], [18], [20]. Because of this flaw, SMOTE

Table 4. Comparison of classification performance by k-NN

No.	Dataset	Resample Method	Accuracy (%)	Recall (Sensitivity)	Precision	F-score	AUC	Kappa
1	**Blood type-MBTI** (4 classes)	None	32	0.320	0.315	0.314	0.496	0.0279
		SMOTE	37.5465	0.375	0.359	0.346	0.570	0.1737
		UR-SR	**85.2**	**0.852**	**0.858**	**0.853**	**0.925**	**0.7944**
2	Soybean (19 classes)	None	91.2152	0.912	0.915	0.910	0.975	0.9036
		SMOTE	91.0275	0.910	0.914	0.908	0.977	0.9017
		UR-SR	**97.8038**	**0.978**	**0.978**	**0.978**	**0.993**	**0.9758**
3	Yeast (10 classes)	None	52.2911	0.523	0.524	0.522	0.685	0.3842
		SMOTE	52.5185	0.525	0.526	0.525	0.685	0.3889
		UR-SR	**89.2183**	**0.892**	**0.892**	**0.892**	**0.933**	**0.8619**
4	Segment Test (7Classes)	None	94.6914	0.947	0.948	0.947	0.970	0.938
		SMOTE	95.1327	0.951	0.951	0.951	0.970	0.9428
		UR-SR	**98.7654**	**0.988**	**0.988**	**0.988**	**0.994**	**0.9856**
5	Segment Challenge (7Classes)	None	96.2	0.962	0.962	0.962	0.978	0.9556
		SMOTE	96.9484	0.969	0.970	0.969	0.984	0.9639
		UR-SR	**99**	**0.990**	**0.990**	**0.990**	**0.994**	**0.9883**
6	Iris (3 classes)	None	95.3333	0.953	0.953	0.953	0.966	0.93
		SMOTE	96	0.960	0.960	0.960	0.974	0.936
		UR-SR	**98.6667**	**0.987**	**0.987**	**0.987**	**0.985**	**0.98**
7	Breast cancer-winsconsin (2 classes)	None	95.1359	0.951	0.951	0.951	0.973	0.8919
		SMOTE	97.8723	0.979	0.979	0.979	0.981	0.9574
		UR-SR	**98.9986**	**0.990**	**0.990**	**0.990**	**0.994**	**0.9777**
8	Unbalanced (2 classes)	None	97.6636	0.977	0.977	0.977	0.617	0.1548
		SMOTE	96.659	0.967	0.966	0.966	0.680	0.3658
		UR-SR	**99.6495**	**0.996**	**0.997**	**0.997**	**0.989**	**0.9014**
9	Diabetes (pima) (2 Classes)	None	70.1823	0.702	0.696	0.698	0.650	0.3304
		SMOTE	78.7645	0.788	0.790	0.787	0.772	0.5733
		UR-SR	**92.3177**	**0.923**	**0.923**	**0.923**	**0.905**	**0.8322**
10	Liver disorder (2 classes)	None	62.8986	0.629	0.630	0.629	0.630	0.2401
		SMOTE	74.4898	0.745	0.744	0.738	0.722	0.4538
		UR-SR	**88.9855**	**0.890**	**0.890**	**0.890**	**0.903**	**0.7744**

needs to be developed as in previous studies to improve the interclass boundary [10], [14], [23].

Classification performance with SMOTE preprocessing also decreased on the unbalanced dataset. The accuracy, recall, precision, and F-score decreased slightly, about 0.01, but the AUC increased by 0.07, and Kappa increased by 0.2. However, this increase is not significant because Kappa is still around 0.36, and AUC is below 0.7. It may occur because SMOTE synthesizes new unique samples; therefore, the increase in FN and FP is more represented by precision and recall but uniquely biased on AUC and Kappa. The inconsistency of AUC and Kappa towards F-score, recall, and precision is possibly caused by SMOTE's drawback to fill the distribution space between the original samples in class.

Our proposed preprocessing has improved classification performance and yields higher accuracy and F-Measure than the results in previous research using SMOTE (k-means-SMOTE) [19] and SMOTE Resample [9], [11]. The resample has improved classification performance significantly, even on high-dimensional data without using the feature selection method in line with previous studies. This previous research has shown that the feature selection methods' role, except for the information gain, is still lesser than the resample method to improve the classification performance, including accuracy, sensitivity, and precision [20].

The results of this study's proposed method still quite outperform the F-score on the yeast dataset than the results of the Sigma Nearest Oversampling method based on the Convex Combination (SNOCC), which was still limited to handle features of the continuous and ordinal features [10]. Compared to research that used biased SVM and Weighted SMOTE [13], this proposed method also results in a higher F-score on the Iris dataset. Our study does not resample every fold when classified by ten cross-validation because it can reduce the F-measure due to increased overfitting as in [16]. Meanwhile, the UR-SR preprocessing has resulted in significant improvement in classification performance.

Our proposed method's performance on Pima diabetes, liver-disorder, and breast cancer Winsconsin dataset is also relatively higher than the oversampling by ADASYN, SMOTE, and borderline SMOTE. The accuracy, F-score, precision, and recall of the UR-SR resample with KNN are higher on those three datasets. However, the AUC value is lesser on the liver-disorder and diabetes dataset [14]. The F-score of our study is

also higher in Pima diabetes dataset than the SMOTE-GA oversampling [18].

Our research's UR-SR method does not consider each class's boundary samples as in previous studies, which used oversampling to samples around the hyperplane, then its best performance too dependent on the SVM classifier [24]. We used the UR-SR with the k-NN as a lazy learner to show the better impact of the UR-SR. Hypothetically, if the UR-SR method could handle imbalanced data while using lazy learner k-NN, other advanced classifiers are expected to perform better.

Our proposed method is started by the unsupervised resample (UR), then the supervised resample, and continued by the k-NN classification. The unsupervised resample replicates sample (instances) of the original sample to fill the class's distribution space. So, it does not disperse the boundaries between classes but increases the differentiation between classes. Furthermore, the dataset of unsupervised resampling (UR) results are grouped according to each class, then a supervised resampling (SR) is performed in every class.

This SR randomly replicates each class's original samples and can be optionally controlled by the bias to uniform value. The SR process reinforces the relations between instances in every class. Although replicating the original samples, the UR-SR seems to reinforce the boundaries between classes because it does not synthesize new unique samples as SMOTE does. The SR process is followed by UR to strengthen the integrity of data (instances) in every class. In this way, the performance of the k-NN classification increased, as shown in Table 4.

The bias to uniform values can be assigned to the SR or after the UR stage, and this has been tested on several datasets included segment test (ST), soybean (Sb), iris, and MBTI-Blood type (MB) as presented in Table 5. The bias to the uniform value of 1 yield the highest accuracy of 0.983 on the soybean dataset, 0.993 on the iris dataset, and 0.859 on the MBTI-Blood type dataset. However, the highest average accuracy was 0.95075 on the four datasets that occurs when it applies the bias to the uniform value of 0.7. The effect of this bias to uniform value when combined with other methods also emerges as motivation for further research in the future about the unbalanced handling data.

Table 5. Classification result when used bias value to uniform in UR-SR Preprocessing

Bias value	Accuracy every dataset				Average
	ST	Sb	Iris	MB	
0	0.979	0.957	0.987	0.852	0.94375
0.2	0.975	**0.987**	0.980	0.852	0.94835
0.3	0.983	0.966	0.987	0.826	0.94050
0.5	0.979	0.972	0.987	0.810	0.93700
0.65	0.974	0.963	0.987	0.847	0.94275
0.7	0.981	0.978	0.993	0.851	**0.95075**
1	**0.983**	0.912	**0.993**	**0.859**	0.93675

IV. Conclusion

The UR-SR method has improved the performance of the k-NN significantly to classify Blood type based on MBTI features. It is also proven on other several multiclass and high dimensional dataset, the AUC, Kappa, F-score, accuracy, recall, and precision increase significantly. The UR-SR and kNN yield better results in handling imbalanced data impacts than previous studies' methods. For the future work, combining the UR-SR with the optimization of *k* value and the information gain as the feature selection method is considerable. Therefore the classification performance is expected to increase even with $k>1$.

References

[1] S. Tsuchimine, J. Saruwatari, A. Kaneda, and N. Yasui-Furukori, "ABO blood type and personality traits in healthy Japanese subjects," *PLoS One*, vol. 10, no. 5, pp. 1-10, 2015. doi: 10.1371/journal.pone.0126983

[2] A. Nahida, N. Chatterjee, and C. A. Nahida, "A study on relationship between blood group and personality," *International Journal of Home Sciences*, vol. 2, no. 21, pp. 239–243, 2016.

[3] C. Y. Lee and S. Chin, "Finding EEG correlates of ABO blood types," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 3, pp. 291–300, 2014.

[4] S. Bharadwaj, S. Sridhar, R. Choudhary, and R. Srinath, "persona traits identification based on myers-briggs type indicator (MBTI) - a text classification approach," in *2018 international conference on advances in computing, communications and informatics*, bangalore, india, sept. 2018, pp. 1076–1082. doi: 10.1109/ICACCI.2018.8554828

[5] F. Noori and M. Kazemifard, "Simulation of pair programming using multi-agent and MBTI personality model," in *6th International Conference of Cognitive Science*, Tehran, Iran, Apr. 2015, pp. 29–36. doi: 10.1109/COGSCI.2015.7426665

[6] M. S. Halawa, M. E. Shehab, and E. M. R. Hamed, "Predicting student personality based on a data-driven model from student behavior on LMS and social networks," in *5th International Conference on Digital Information Processing and Communications*, Sierre, Switzerland, Oct. 2015, pp. 294–299. doi: 10.1109/ICDIPC.2015.7323044

[7] S. Selvi, S. Rohini, and C. Velou, "Relation between blood group and mood changes," *Indian Journal of Basic and Applied Medical Research*, vol. 6, no. 3, pp. 118–125, 2017.

[8] J. Patil *et al.*, "Influence of blood group on the character traits - A cross-sectional study on Malaysian student population," *Journal of Chemical and Pharmaceutical Sciences*, vol. 9, no. 2, pp. 865–868, 2016.

[9] L. S. Katore and J. S. Umale, "Comparative study of recommendation algorithms and systems using WEKA," *International Journal of computer Applications*, vol. 110, no. 3, pp. 14–17. doi: 10.5120/19295-0731

[10] Z. Zheng, Y. Cai, and Y. Li, "Oversampling method for imbalanced classification," *Computing and Informatics*, vol. 34, no. 5, pp. 1017–1037, 2015.
[11] G. N. Ramadevi, K. U. Rani, and D. Lavanya, "Evaluation of Classifiers Performance using Resampling on Breast cancer Data," *International Journal of Scientific & Engineering Research*, vol. 6, no. 2, pp. 200–207, 2015.
[12] S. Zhang *et al.*, "Efficient knn classification with different numbers of nearest neighbors," *IEEE Transactions On Neural Networks And Learning Systems*, vol. 29, no. 5, pp. 1–12, 2017. doi: 10.1109/TNNLS.2017.2673241
[13] Hartono, O. S. Sitompul, T. Tulus, and E. B. Nababan, "Biased support vector machine and weighted-SMOTE in handling class imbalance problem," *International Journal of Advances in Intelligent Informatics*, vol. 4, no. 1, pp. 21–27, 2018. doi: 10.26555/ijain.v4i1.146
[14] N. Cahyana, S. Khomsah, and A. S. Aribowo, "Improving imbalanced dataset classification using oversampling and gradient boosting," in *5th International Conference on Science in Information Technology*, Yogyakarta, Indonesia, Oct. 2019, pp. 217–222. doi: 10.1109/ICSITech46713.2019.8987499
[15] M. Tajik, M. Malakpour, and J. G. Bidgoli, "Examine the relationship between blood groups and intercity driving jobs in Iran," *International Journal of Medical Research & Health Science.*, vol. 5, no. 12, pp. 292–301, 2016.
[16] V. D. Valerio, R. M. Pereira, Y. M. G. Costa, and D. Bertolini, "A resampling approach for imbalanceness on music genre classification using spectrograms," in *International Florida Artificial Intelligence Research Society Conference (FLAIRS-31)*, Florida, USA, May 2018, pp. 500–505.
[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minoriy over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321–357, 2002. doi: 10.1613/jair.953
[18] T. E. Tallo and A. Musdholifah, "The implementation of genetic algorithm in SMOTE (synthetic minority oversampling technique) for handling imbalanced dataset problem," in *4th International Conference on Science and Technology*, Yogyakarta, Indonesia, Aug. 2018, pp. 1–4. doi: 10.1109/ICSTC.2018.8528591
[19] H. Hairani, K. E. Saputro, and S. Fadli, "K-means-SMOTE for handling class imbalance in the classification of diabetes with C4.5, SVM, and naive Bayes," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 2, pp. 89–93, 2020. doi: 10.14710/jtsiskom.8.2.2020.89-93
[20] M. Al-Khaldy, "Resampling imbalanced class and the effectiveness of feature selection methods for heart failure dataset," *International Robotics & Automation Journal*, vol. 4, no. 1, pp. 37–45, 2018. doi: 10.15406/iratj.2018.04.00090
[21] J. Huang, Y. Wei, J. Yi, and M. Liu, "An improved knn based on class contribution and feature weighting," in *10th International Conference on Measuring Technology and Mechatronics Automation*, Changsha, China, Feb. 2018, pp. 313–316. doi: 10.1109/ICMTMA.2018.00083
[22] X. Wang, Z. Jiang, and D. Yu, "an improved knn algorithm based on kernel methods and attribute reduction," in *International Conference On Instrumentation And Measurement, Computer, Communication, And Control*, Qinhuangdao, China, Sept. 2015, pp. 567–570. doi: 10.1109/IMCCC.2015.125
[23] A. More, "Survey of resampling techniques for improving classification performance in unbalanced datasets," 2016, arXiv:1608.06048 .
[24] R. Batuwita and V. Palade, "Efficient resampling methods for training support vector machines with imbalanced datasets," in *International Joint Conference on Neural Networks*, Barcelona, Spain, Jul. 2010, pp. 1-8. doi: 10.1109/IJCNN.2010.5596787
[25] A. N. Kasanah, Muladi, and U. Pujianto, "Penerapan teknik SMOTE untuk mengatasi imbalance class dalam klasifikasi objektivitas berita online menggunakan algoritma kNN," *RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 3, no. 10, pp.196-201, 2019. doi: 10.29207/resti.v3i2.945
[26] R. Siringoringo, "K-Nearest Neighbor pada prediksi cacat," *Journal Information System Development (ISD)*, vol. 2, no. 1, pp. 47–58, 2017.