International Journal of Innovative Computing, Information and Control
Volume **15**, Number **2**, April **2019**

ICIC International ©2019 ISSN 1349-4198
pp. 423–444

ENHANCING THE PERFORMANCE OF SMOTE ALGORITHM BY USING ATTRIBUTE WEIGHTING SCHEME AND NEW SELECTIVE SAMPLING METHOD FOR IMBALANCED DATA SET

Tora Fahrudin[1,2], Joko Lianto Buliali[1] and Chastine Fatichah[1]

[1]Department of Informatics
Institut Teknologi Sepuluh Nopember
Jalan Raya ITS, Sukolilo, Surabaya 60111, Indonesia
tora15@mhs.if.its.ac.id; joko@cs.its.ac.id; chastine@if.its.ac.id

[2]School of Applied Science
Telkom University
Jl. Telekomunikasi, No. 1, Terusan Buah Batu, Bandung 40257, Indonesia
torafahrudin@telkomuniversity.ac.id

Received May 2018; revised September 2018

Abstract. *SMOTE is one of the well-known algorithms for balancing train data by adding synthetic data on minor class data. One of the stages in SMOTE is finding the nearest neighbors (kNN) as the basis for creating synthetic data using Euclidean distance. In cases where a small number of attributes having high correlation value than others, finding kNN using Euclidean without considering this correlation may not find representative neighbors. This paper introduces AWH-SMOTE (Attribute Weighted and kNN Hub on SMOTE), which enhances SMOTE in improving neighbors and noise identification using attribute weighting and also improving selective sampling method using occurrence data in the kNN hub. Wojna and Information Gain methods are used for attribute weighting. A small number of occurrences in the kNN hub results in more synthetic data generated so that minority data in dangerous region are more represented. Nine public datasets from Keel repository are used to evaluate AWH-SMOTE. Evaluation shows AWH-SMOTE has better performance on minority precision and minority f-measure for both pruned and unpruned condition than other oversampling algorithms. Information Gain as attribute weighting method in AWH-SMOTE achieves best performance in unpruned condition when compared to other weighting methods for minority recall, minority precision and minority f-measure.*

Keywords: AWH-SMOTE, Attribute weighting, Wojna, Information Gain, *kNN* hub, Noise

1. Introduction.

Imbalanced data set is found in real world cases where the number of one class label is more dominant (major class) than the other class (minor class). In some cases, minor data class has a higher level of importance although its number is lower [1]. Such cases include fraud detection [2], customer credit risk prediction [3], churn detection [4], disease detection to assist medical decisions [5], undesirable news articles detection on stream data [6], keyphrase extraction [7], detection of high school student academic failures [8] and predicting the potential of college student failure [9]. Solutions in such cases generally use data classification techniques using machine learning.

When imbalanced data set is processed (mainly in classification problems) by machine learning techniques, three problems usually arise: small disjunct [10], overlapping [11] and noise [12]. These problems can cause unsatisfactory performance in recognizing minor

DOI: 10.24507/ijicic.15.02.423

class pattern. Existing solutions to the three problems above can be divided into three groups [12].

- Solution at data level
  In data level solution, the appropriate sampling techniques have been developed in order to balance training data (oversampling, undersampling or oversampling with undersampling). Examples in this solution are SMOTE [13] for oversampling, neighborhood cleaning rule [14] for undersampling and one side selection for oversampling with undersampling [15].
- Solution at algorithm level
  In algorithm level solution, the classifier algorithm is modified for handling minor data class. Examples in this solution (which modified artificial neural network (ANN)) are a two-step supervised learning in ANN [16] and MD-SVM [17].
- Hybrid solution
  The combination of data-level solutions and algorithm level solution is called hybrid solutions. Example in this solution is combining cost sensitive SVM with adaptive oversampling using data density [18].

Most papers concentrate on techniques within the data level solution because of the simplicity of the methods and the independence of classifiers. Our proposed method (AWH-SMOTE) falls into data level solution. AWH-SMOTE improves SMOTE on attribute weighting scheme and a new selective sampling method.

In data level solution, one of the balancing techniques is oversampling, which adds minority data to an imbalanced data set. Minority data that are added to an imbalanced data set can be synthetic or original [19]. Techniques for oversampling with additional synthetic data include Synthetic Minority Oversampling TEchnique (SMOTE) [13], Borderline-SMOTE [20], Adaptive Synthetic Sampling Technique (ADASYN) [21]. These techniques aim to enlarge decision area over minor class and reduce the impact of overfitting on oversampling using original data and have been proven as effective techniques to combat the difficulties of classifier algorithm to tackle uneven distribution of testing data.

SMOTE is known as the pioneer of developing oversampling techniques using synthetic data. Many authors have presented works on SMOTE enhancements. We divided those enhancements into two main groups as shown in Table 1.

To the best of our knowledge, all SMOTE enhancements have been done using Euclidean distance for searching k nearest neighbors. Problem arises in the cases where a small number of attributes have high importance or correlation value compared with other attributes, since searching k nearest neighbors using Euclidean distance formula without considering that importance may not find representative neighbors. Therefore, we propose AWH-SMOTE which falls into data level solution in the classification described above. AWH-SMOTE improves SMOTE by adding attribute weighting scheme and introducing a new selective sampling method.

Adding attribute weighting scheme is done by providing four choices of attribute weighting methods to get representative k nearest neighbors: Information Gain [32], Wojna1 & Wojna2 [33] and Scaled Misclassification Ratio Weighting Method (SMR) [34]. After getting representative k nearest neighbors, identifying and removing noise can be conducted more precisely. A new selective sampling method is also proposed in AWH-SMOTE which calculates the occurrence of data in all *kNN* minority class (*kNN* hub) to get safe value of minority data. Minority data with the largest number of occurrences have the highest safe value and become the centroid of minority class. Minority data with small number of occurrences (which means that minority data is far away from the centroid) has low safe

Table 1. Enhancements of SMOTE

Enhancement groups		*Algorithms/Methods*	*Ref*	*Enhancement points*
Enhancing classification process using SMOTE as a part of classification process		SMOTE-Boost	[22]	Combine SMOTE with boosting procedure
		Nested rotation forest SMOTE	[23]	Combine SMOTE with rotation forest method
		ASE-Bagging	[24]	Combine SMOTE with bagging procedure
		COSDF	[25]	Combine SMOTE with co-training method
		SMOTE+TL & SMOTE+ENN	[1]	Combine SMOTE with undersampling methods (Tomek link & Wilson's edited nearest neighbors)
		SMOTE-RSB*	[26]	Combine SMOTE with undersampling method (rough set theory)
		SMOTE-IPF	[27]	Combine SMOTE with additional cleaning method (iterative-partitioning filter)
Enhancing SMOTE algorithm	Selecting particular area of synthetic data generation	Borderline-SMOTE	[20]	Generate synthetic data only on border area
		ADASYN	[21]	Generate synthetic data on safe and border area based on harder level ratio
		Safe-Level-SMOTE	[28]	Generate synthetic data only on safe area
		LN-SMOTE	[29]	Generate synthetic data only on safe area
		MWMOTE	[30]	Generate synthetic data only on border area
	Calculating magnification balance ratio	ADASYN	[21]	Generate synthetic data based on their distribution ratio
		MWMOTE	[30]	Generate synthetic data based on their selection probability
		Weighted-SMOTE	[31]	Generate synthetic data based on their weighted-matrix
	Selecting candidate	Safe-Level-SMOTE	[28]	Use random method with some safe-level constraints
		LN-SMOTE	[29]	Use random method with some safe-level constraints
		MWMOTE	[30]	Use random method with selection probability

value and the areas surrounding these minority data are suitable places for synthetic data because this area contains more majority class data. This idea is inspired by the successful exploration of hubness in determined central point of cluster [35] and classification using hubness aware in the case of imbalanced data set [36].

The remainder of this paper is divided into four sections. In Section 2, we provide related works that are associated with oversampling with SMOTE algorithm, attribute weighting method and hub concepts adopted in imbalanced data cases. Our detailed AWH-SMOTE algorithm is presented in Section 3. The experimental study and simulation results are presented in Section 4. Finally, we give conclusions and some future research directions in Section 5.

2. **Related Works.**

In this section, some literature related to oversampling algorithm with additional synthetic data, attribute weighting and hub concepts adopted in imbalanced data cases is presented.

2.1. **Oversampling with additional synthetic data.**

Oversampling technique using additional synthetic data called SMOTE algorithm was introduced by [13]. The addition of synthetic data aims to extend the minor data class decision area and to avoid overfitting. SMOTE algorithm consists of two stages. In the first stage, k nearest neighbors are found by using Euclidean distance calculation from each minority data with respect to all other minority data and then sorted in ascending order, and then k lowest distance data are taken as the nearest neighbors (*kNN*). Euclidean distance between one minority data (x) and another minority data (y) from the first attribute to n (maximum number of attributes) is defined in Formula (1)

$$d(x,y) = \sqrt{\sum_{a=1}^{n}(x_a - y_a)^2} \tag{1}$$

In the second stage, synthetic data are generated by using the interpolation method between two minority data. One of its *kNN* will be randomized to be candidates in synthetic data generation process. Thereafter, original minor data (x) and one chosen candidate (y) will be used to generate new synthetic data among x and y. Synthetic data formula among x and y for the a-th attribute is defined in Formula (2)

$$\mathit{SyntheticData}_a(x,y) = x_a + r \cdot (x_a - y_a) \quad \text{for } 0 \leq r \leq 1 \tag{2}$$

The above formula is applied for n attributes. The process is repeated until the desired synthetic data amount is reached. Borderline-SMOTE [20] oversamples minority class data which lie near borderline. ADASYN adaptively generates synthetic data based on their distributions, and more synthetic data are generated by using some ratio [21].

2.2. **Attribute weighting methods.**

Finding *kNN* using Formula (1) will treat all attributes with equal level of importance. However, in real data sets, there are many factors that make attributes have unequal importance level because some attributes are strongly correlated with a decision and some attributes are not correlated with the decision, some attributes may contain the same information value (redundant information), and some attributes can have noise which make them less reliable than other attributes [33]. Let a_n be a set of n-attributes. Attribute a_1 can have higher rank correlation than other attributes $a_2, a_3, \ldots, a_n$; therefore, we need an attribute weighting method to get better neighborhood results on *kNN* by treating each attribute differently according to their correlation level.

Table 2 provides an illustration of potential errors in calculation of Euclidean distance in the case of Academic Data. Let us assume that *grade* value is more correlated to academic performance than *presence* and *Internet duration*, and *presence* value is more correlated to academic performance than *Internet duration*. Let us also assume 0.5, 0.3 and 0.2 are the normalized correlation value for grade, presence, Internet duration, respectively. It is seen in the table that the distance $d(x_1, x_2)$ is equal to distance $d(x_1, x_3)$. While in fact, when applying the correlation values above, the distance $d(x_1, x_2)$ should be lower than $d(x_1, x_3)$, since presence attribute is more correlated to academic performance than the Internet duration attribute.

Table 2. Illustration of potential errors in calculation of distance using Euclidean distance

Data	*Grade*	*Presence*	*Internet Duration*	*Academic Performance*
x_1	100	100	100	Success
x_2	50	100	50	Success
x_3	50	50	100	Failed
$d(x_1, x_2) = 70.71$	50	0	50	–
$d(x_1, x_3) = 70.71$	50	50	0	–

This can be corrected by using attribute weighting on Manhattan distance formula. Manhattan distance formula for minority data x and other minority data y with normalized attribute weighted w from the first attribute to n (maximum number of attributes) is defined in Formula (3)

$$d(x, y) = \sum_{a=1}^{n} \boldsymbol{w}_a \cdot |x_a - y_a| \tag{3}$$

When normalized correlation values assumed above (0.5, 0.3, and 0.2) are applied to Formula (3), Manhattan distance calculation results are shown in Table 3 in which $d(x_1, x_2)$ is less than $d(x_1, x_3)$. This corresponds to the fact that the distance $d(x_1, x_2)$ should be lower than $d(x_1, x_3)$, since presence attribute is more correlated to academic performance than the Internet duration attribute. Therefore, attribute weighting gives a more representative k nearest neighbors than regular Euclidean. Some research also shows that Manhattan distance is preferable than Euclidean distance for several reasons.

- Manhattan distance formula can give better different contrast of data [37].
- The *kNN* classification model for numerical attributes achieves the best performance with Manhattan distance [33].
- Manhattan distance formula also gives lower computational cost due to none of square root operations is needed [38], so it can be suitable for applications which need a low computational cost such as wireless sensor network [39].

Table 3. Illustration of Manhattan distance calculation with additional attribute weighting

Data	*Grade*	*Presence*	*Internet Duration*	*Academic Performance*
x_1	100	100	100	Success
x_2	50	100	50	Success
x_3	50	50	100	Failed
$d(x_1, x_2) = 35$	25	0	10	–
$d(x_1, x_3) = 40$	25	15	0	–

- Manhattan distance raises better accuracy than Euclidean distance such as in analysis of shape alignment [40], face recognition [41], classification of hyper spectral images [42] and young and elderly subject classification based on heart rate variability signal [43].

Based on the data used in attribute weighting methods, we can classify attribute weighting into two methods: sampling and population. On sampling method, the attribute weighting values are obtained from an iterative randomization of sampling data. In this research, we use Wojna1 (optimizing distance) method [33], Wojna2 (optimizing classification accuracy) method [33] and Scaled Misclassification Ratio Weighting method (SMR) [34] as sampling attribute weighting methods. Meanwhile, in population method, the attribute weighting values are obtained from all training data. We use Information Gain as population attribute weighting method. Information Gain is the difference between the entropy of the parent and the average of the child's entropy [32]. Information Gain formula ($GAIN_{split}$) for o (the number of parent data), h (attribute partition value) and o_b (amount data on b-partition) is defined in Formula (4)

$$GAIN_{split} = Entropy_{parent} - \left(\sum_{b=1}^{h} \frac{o_b}{o} \cdot Entropy_b \right) \tag{4}$$

2.3. **Hub concepts adopted in imbalanced data cases.** Implementation of the hub concept first appeared on the music retrieval and recommendation systems, where a small number of songs often appeared in *kNN* hub compared to other songs [44]. Let x_i be an i-th minority data, $N_k(x_i)$ be the occurrences of x_i inside *kNN* hub of all minority class data, $D_k(x_i)$ be the *kNN* set defined by the nearest neighbors of minority class data x_i, and D_k be the set of all the *kNN* hub obtained from all minority class data. We use *safe value* assignment concept to select samples in minority class. In a one minority class data (x_i), larger value of $N_k(x_i)$ indicates that the minority class data is closer to the minority class centroid. Meanwhile, smaller value of $N_k(x_i)$ indicates that the minority class data is farther away from the minority class centroid and possibly in overlap, small disjuncts or noise areas which contribute to errors. The errors can be minimized if synthetic data are generated in these areas. Figure 1 illustrates the locations of synthetic data on small number or empty $N_k(x_i)$.

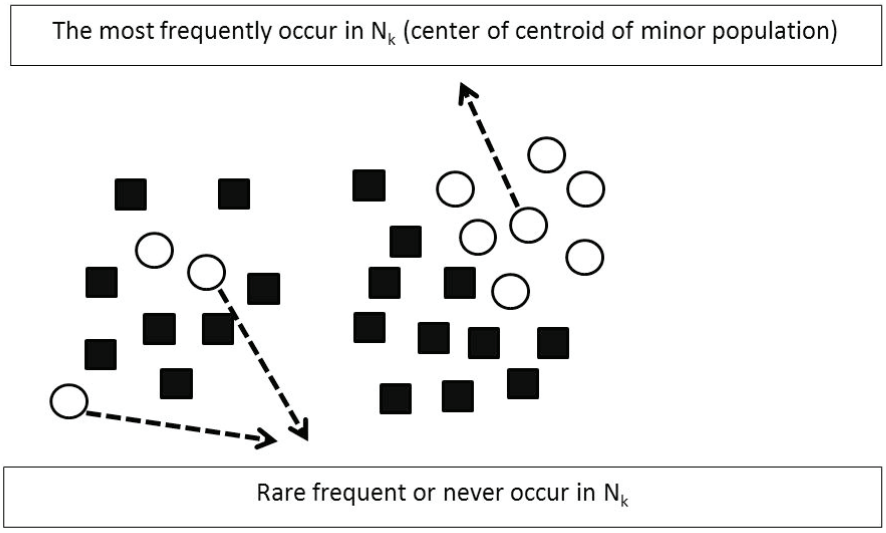
The most frequently occur in N_k (center of centroid of minor population)
Rare frequent or never occur in N_k

FIGURE 1. Illustration of minor data occurrence in *kNN* hub

3. **Proposed Method.**

In this paper we propose the application of Attribute Weighted and *kNN* Hub on SMOTE (AWH-SMOTE) algorithm which enhances SMOTE in two aspects.

- Enhancing k-neighbors and noise identification using attribute weighting

  Based on illustration in Table 3, we can get two benefits of attribute weighting: representative neighbors and representative noise. We explore these two benefits in Section 4.2. We implement Wojna1, Wojna2 and SMR for sampling methods and Information Gain for population method.
- Enhancing selective sampling method using occurrence data in the *kNN* hub

  The idea of selective sampling method using occurrence data in the *kNN* hub method is motivated by the successful exploration of hubness phenomena in determining the cluster's central point [35] and classification using hubness aware classification in the case of imbalanced data set [36]. Smaller occurrence of a minority data in *kNN* hub means that the data are farther away from minority class data cluster; therefore, the *safe value* will be decreased.

There are four main steps in the AWH-SMOTE algorithm (the details of each step will be discussed in Section 3.1 to Section 3.4).

1) Weighting attributes.
2) Finding and removing noise.
3) Selecting samples using hub concepts with 2 processes: magnification balance ratio calculation of each minor class data based on $N_k(x_i)$ in the *kNN* hub and candidate selection using one of their minority neighborhoods which has the smallest *safe value*.
4) Generating synthetic data.

3.1. **Weighting attributes.**

Choose one of four attribute weighting methods (Wojna1, Wojna2, SMR or Information Gain). The choice is arbitrary. The attributes of the data to be processed are weighed using the chosen attribute weighting method and are then normalized. These normalized attributes are subsequently added into Formula (3) to get the *kNN* hub matrix.

3.2. **Finding and removing noise.**

Find noisy minority samples in data set based on Table 4 which defines the number of majority neighbors (m) from k neighbors around the minority data. Based on [28], each minority data can be classified into three areas: safe, border and noise. Minority class data located in safe areas are not altered, and no synthetic data are generated. Synthetic data will be generated by selecting sample (Section 3.3) and generating synthetic data procedure (Section 3.4) in areas where minority class data are located in border. This synthetic data generation can solve problems where small disjunct and overlap exist as shown in Figures 2(a) and 2(b). For minority class data located in noisy area, data removal will be conducted because noisy data or outlier is a random error in labeling examples. This data removal can solve problems where noise exists as shown in Figure 2(c).

Table 4. Three area definitions for minor class data neighborhood

Area	*Definition*
Safe	$0 \leq m \leq \frac{1}{2}k$
Border	$\frac{1}{2}k < m < k$
Noise	$m = k$

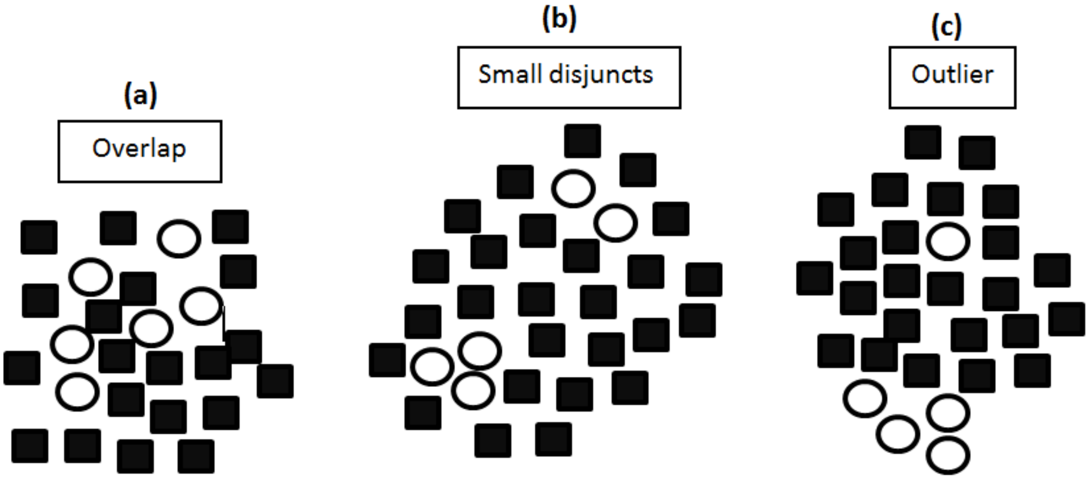

Figure 2. Three main problems on imbalanced data cases: overlap (a), small disjunct (b) and outlier (c)

Noise elimination process cannot guarantee that data will be noise free. This is because after noise elimination process, the order of *kNN* on hub is changed and the remaining data could be potentially identified as noise again. Therefore, the occurrence of noisy data after the first noise removal process is handled by copying the exact minority class data with the magnifications following the formula (described in Section 3.3 and Section 3.4).

3.3. **Selecting sample using hub concepts.** The detailed processes of determining number of magnification and selecting samples using hub concepts on AWH-SMOTE are as follows.

a) Determine the number of magnification for minority group inside the hub area (in hub) and outside the hub (out hub) area. Let D_{inHub} be the set of unique value from minority data located inside the hub D_k (which has $N_k(x_i) > 0$) and D_{outHub} be the set of unique value from minority data located outside the hub D_k (which has $N_k(x_i) = 0$). The larger value of the occurrence of x_i ($N_k(x_i)$) in D_{inHub} means the magnification for minority group is smaller and vice versa. This is because the larger occurrence of minority class data in D_{inHub} means the larger *safe value*. The larger value of normalized ratio of its majority neighbors x_i in D_{outHub} means the smaller safe value. This is because larger normalized ratio of its majority neighbors x_i for *out hub* data means the minority data is near to the majority centroid in D_k and requiring more synthetic data generation. Notation Mag_{inHub} is the magnification for minority data in D_{inHub} and Mag_{outHub} is the magnification for minority data in D_{outHub}. Notation n_{maj} is a number of majority data and $n_{\min}$ is a number of minority data. Final magnification values of minority group inside hub (M_{inHub}) and outside hub (M_{outHub}) are calculated as follows.
   - $M_{inHub} = Mag_{inHub} \cdot (n_{maj} - n_{\min})$.
   - $M_{outHub} = Mag_{outHub} \cdot (n_{maj} - n_{\min})$.
   - $Mag_{inHub} + Mag_{outHub} = 1$.

   In this paper, we use $Mag_{inHub} = 0.5$ and $Mag_{outHub} = 0.5$, which implies that the magnification balance ratios are equal for minority group inside the hub (*in hub*) and outside the hub (*out hub*).

b) Calculate magnification balance ratio for *in hub* minority group and *out hub* minority group. The detailed processes are as follows.

- For *in hub* data $x_i \in D_{inHub}$, magnification balance ratio value of minority class data *in hub* $magin(x_i)$ is calculated using the occurrence of minority data in D_k. The concentration of magnification focuses on minority class data which have small *safe value*. Let $n_{mininHub}$ be the amount of unique values of minority class data inside the hub. Let $Rmagin(x_i)$ be the magnification ratio of minority data x_i. The $Norm_{Rmagin(x_i)}$ notation is the normalized form of $Rmagin(x_i)$. The $magin(x_i)$ value is calculated using Formulas (5), (6) and (7) as follows:

$$Rmagin(x_i) = \frac{1}{\left(\frac{N_k(x_i)}{\sum_{j=1}^{n_{mininHub}} N_k(x_j)}\right)} \tag{5}$$

$$Norm_{Rmagin(x_i)} = \frac{Rmagin(x_i)}{\sum_{j=1}^{n_{mininHub}} Rmagin(x_j)} \tag{6}$$

$$magin(x_i) = Norm_{Rmagin(x_i)} \cdot M_{inHub} \tag{7}$$

- For *out hub* data $x_i \in D_{outHub}$, magnification balance ratio value of minority class data outside $magout(x_i)$ is calculated by using normalized ratio of majority neighbors. Let $n_{minoutHub}$ be the amount of the unique values of minority class data outside the hub, z_j be the j-th majority neighbor of minority data x_i ($z_j \in D_k(x_i)$), $n_{majinHub}$ be the amount of unique values of majority data inside D_k and $n_{majinHub_{x_i}}$ be the amount of majority neighbors of minority data x_i. The notation $ratio(z_j)$ is defined as the ratio of the occurrence of majority neighbor data z_j for minority data $x_i(N_{kz_j})$ to all occurrences of majority neighbors inside D_k ($\sum_{c=1}^{n_{majinHub}} N_k z_c$). The sum of $ratio(z_j)$ for all j in x_i is denoted as $Rmagout(x_i)$ and are then normalized to $Norm_{Rmagout(x_i)}$. The $magout(x_i)$ value is calculated using Formulas (8), (9), (10) and (11).

$$ratio(z_j) = \frac{N_{kz_j}}{\sum_{c=1}^{n_{majinHub}} N_k z_c} \tag{8}$$

$$Rmagout(x_i) = \sum_{j=1}^{n_{majinHub_{x_i}}} ratio(z_j) \tag{9}$$

$$Norm_{Rmagout(x_i)} = \frac{Rmagout(x_i)}{\sum_{c=1}^{n_{minoutHub}} Rmagout(x_c)} \tag{10}$$

$$magout(x_i) = Norm_{Rmagout(x_i)} \cdot M_{outHub} \tag{11}$$

c) Select candidate

Based on SMOTE algorithm, the synthetic data are generated by putting a point anywhere between a minor data observation x_i with one randomly chosen neighbor from k-minority neighbors ($x_i pair$). In AWH-SMOTE, the process of choosing one neighbors from k-neighbors is not done randomly but by selecting one of the minority class neighbors in $D_k(x_i)$ which has the smallest *safe value* ($x_i candidate$). Let $x_{i,j}$ be the j-th minority neighbor data of k-neighbors x_i and $sv_{i,j}$ be the *safe value* for $x_{i,j}$. The notation $n_{maj_{x_{i,j}}}$ is defined as the number of majority neighbors of $x_{i,j}$. *Safe value* is a value which represented region of minority data $x_{i,j}$ in D_k. The greater *safe value* means data $x_{i,j}$ closer to the centroid of minority class, and vice versa. *Safe value* is obtained by using Formula (12)

$$sv_{i,j} = \begin{cases} magin(x_{i,j}), & \text{if } x_{i,j} \in DinHub \\ 0, & \text{if } x_{i,j} \in DoutHub \end{cases} + \sum_{l=1}^{n_{maj_{x_{i,j}}}} \left(\frac{1}{ratio\left(z_{x_{i_{j_l}}}\right)} \right) \tag{12}$$

Figure 3 shows how magnification of balance ratio is calculated and how $x_i candidate$ is chosen for x_i in AWH-SMOTE in a case where $k = 5$ which consists of two minority neighbors ($x_{i,2}$ and $x_{i,4}$) data and three majority neighbors data ($x_{i,1}$, $x_{i,3}$ and $x_{i,5}$). First, we calculate magnification for x_i. If x_i is *in hub*, we calculate $magin(x_i)$ by using Formulas (5) to (7). If x_i is *out hub*, we calculate $magout(x_i)$ by using Formulas (8) to (11). Second, we calculate *safe value* of each minority data in k neighbors of x_i ($x_{i,2}$ and $x_{i,4}$) by using Formula (12). The smallest *safe value* will be $x_i candidate$.

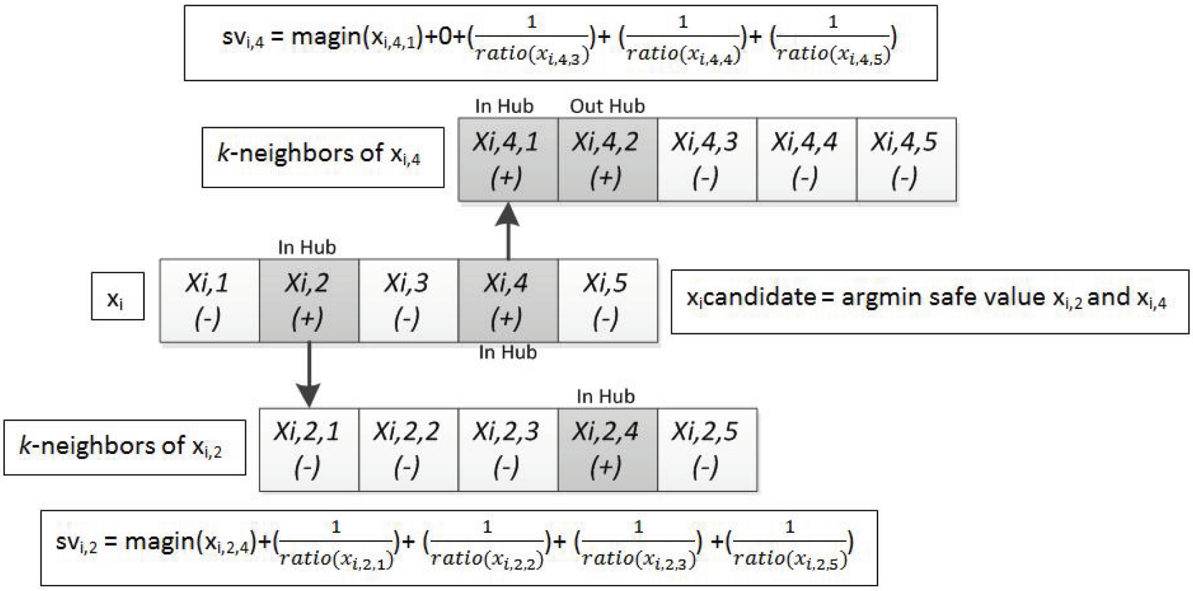
$sv_{i,4}$ = magin($x_{i,4,1}$)+0+($\frac{1}{ratio(x_{i,4,3})}$)+ ($\frac{1}{ratio(x_{i,4,4})}$)+ ($\frac{1}{ratio(x_{i,4,5})}$)
In Hub
Out Hub
k-neighbors of $x_{i,4}$
Xi,4,1 (+)
Xi,4,2 (+)
Xi,4,3 (-)
Xi,4,4 (-)
Xi,4,5 (-)
In Hub
x_i
Xi,1 (-)
Xi,2 (+)
Xi,3 (-)
Xi,4 (+)
Xi,5 (-)
$x_i$candidate = argmin safe value $x_{i,2}$ and $x_{i,4}$
In Hub
In Hub
k-neighbors of $x_{i,2}$
Xi,2,1 (-)
Xi,2,2 (-)
Xi,2,3 (-)
Xi,2,4 (+)
Xi,2,5 (-)
$sv_{i,2}$ = magin($x_{i,2,4}$)+($\frac{1}{ratio(x_{i,2,1})}$)+ ($\frac{1}{ratio(x_{i,2,2})}$)+ ($\frac{1}{ratio(x_{i,2,3})}$) +($\frac{1}{ratio(x_{i,2,5})}$)

FIGURE 3. Illustration of selecting sample process based on smallest *safe value*

In SMOTE, Borderline-SMOTE and ADASYN algorithms, local neighbors are used to determine the number of magnifications and selection of candidates to produce synthetic data. While in AWH-SMOTE, a set of local neighbors is used to determine the number of magnifications and selection of candidates to produce synthetic data.

3.4. Generating synthetic data.

Formula (2) is used to generate synthetic data between x_i and $x_i candidate$. The number of synthetic data generated is equal to $magin(x_i)$ or $magout(x_i)$, depending on whether the observed minority data x_i is inside hub or outside hub. In the case that minority data x_i do not have minority neighbors inside $D_k(x_i)$, we do not generate synthetic data, but replicate exact u minority data x_i, where u equals $magin(x_i)$ or $magout(x_i)$, depending on whether the minority data inside hub or outside hub.

4. Experimental Setting & Results.

In this section, we discuss how to design experiments, evaluate attribute-weighted effects and evaluate the performance of AWH-SMOTE algorithm.

4.1. Design experiments.

The data sets used in this experiment were two classes of quantitative imbalanced data set from Keel data repository[1] which have different imbalance ratios ranging from 0.11 to 0.55. Table 5 shows the description of this data set. Imbalance ratios are obtained from dividing the frequencies of the majority class by the minority class.

[1] http://sci2s.ugr.es/keel/imbalanced.php

Table 5. Dataset description

Data	*Attribute*	*Majority*	*Minority*	*Imbalance Ratio*
ecoli-0_vs_1	7	143	77	0.54
ecoli2	7	284	52	0.18
ecoli3	7	301	35	0.12
glass0	9	144	70	0.49
glass1	9	138	76	0.55
Haberman	3	225	81	0.36
Pima	8	500	268	0.54
yeast1	8	1055	429	0.41
yeast-2_vs_4	8	463	51	0.11

Table 6. Confusion matrix

Actual	*Predicted P (+)/Minor Class*	*Predicted N (−)/Major Class*
P (+)/Minor Class	TP	FN
N (−)/Major Class	FP	TN

The experiment was done by using Weka API [45] with C4.5 (default setting) as a base classifier in two conditions: pruned and unpruned. Magnification balance ratio for major and minor data was set to 50:50 which according to [46] gave the best results on C4.5 algorithm. The nearest neighbor parameter k is set to 5, which is the most common value used in imbalanced data cases.

In the case of binary classification for imbalanced data cases, the accuracy assessment measures are precision, recall and f-measure [47]. We use (+) symbol for minor class and (−) symbol for major class in confusion matrix as shown in Table 6. *Precision* is defined as the number of correctly predicted instances divided by total predicted instances. *Recall* is defined as the number of correctly predicted instances divided by total actual instances. *F-Measure* is defined as harmonic mean of precision and recall. Formulas (13) to (15) show the calculation of precision, recall and f-measure value, respectively

$$Precision = \frac{TP}{(TP + FP)} \tag{13}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{14}$$

$$F\text{-}Measure = \frac{(1+\beta)^2 \cdot Recall \cdot Precision}{\beta^2 \cdot (Recall + Precision)} \tag{15}$$

Parameter β is set to 1 which means that precision and recall have equal importance. The experiment was conducted by using 5 × 10 fold cross-validation evaluation.

4.2. **Attribute weighting effect on identifying neighbors and noise against improvement of classification accuracy.** Implementation of the Wojna1, Wojna2 and Scaled Misclassification Ratio (SMR) algorithms in this experiment is done by taking the 10% number of samples to produce $\mathrm{S}_{\mathrm{train}}$ (70%) and $\mathrm{S}_{\mathrm{test}}$ (30%) from all training data ($\mathrm{U}_{\mathrm{train}}$). We use 20 iterations ($l = 20$) in this experiment according to the main reference of the Wojna algorithm. The effect of attribute weighting on neighborhood and noise identification can be seen from improvement of recall, precision and f-measure value.

Figure 4 illustrates how to find and remove noisy data. The identification of noise is done by looking for noise from the first fold until the tenth. The noisy data are identified

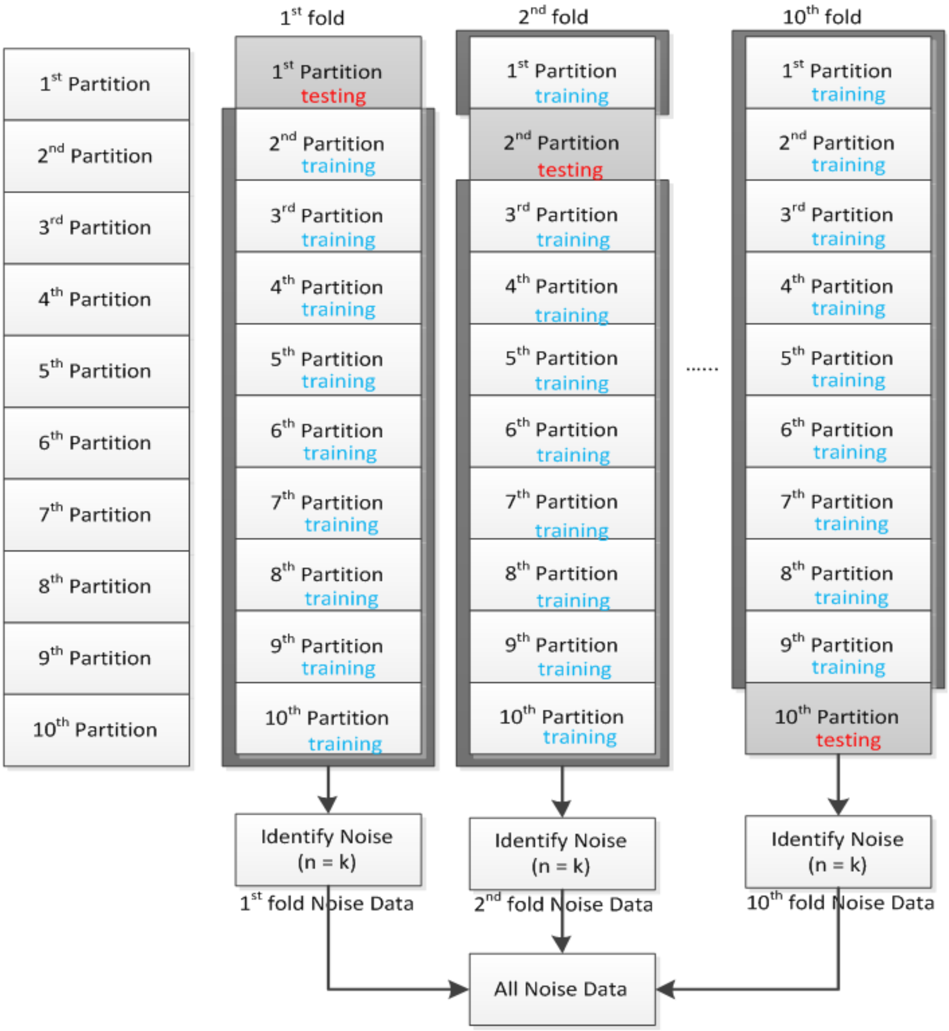

FIGURE 4. Illustration of noise identification and removal

TABLE 7. Number of identified noise with and without attribute weighting scheme

Data	*Without Attribute Weighting Scheme*	*With Attribute Weighting Scheme*			
	Euclidean	*Information Gain*	*Wojna1*	*Wojna2*	*SMR*
ecoli-0_vs_1	2	2	8	2	6
ecoli2	4	8	5	4	5
ecoli3	4	6	5	6	6
glass0	4	6	2	4	4
glass1	14	12	13	15	10
Haberman	34	29	35	31	33
Pima	56	54	62	54	54
yeast1	95	122	105	107	107
yeast-2_vs_4	11	9	13	12	17

by searching for minor data that has the number of major neighbors as many as k based on noise definition from Table 4. Noisy data in all partitions are then removed.

Table 7 shows the result of noise identification process (without attribute weighting scheme or with attribute weighting scheme). Generally, the number of identified noise using attribute weighting scheme is larger than that without attribute weighting scheme. The larger number in the table indicates that more data is identified as noise and will be removed.

Two testing scenarios in this section are developed to evaluate the attribute weighting scheme performance compared to without weighting scheme. Those two scenarios are Scenario 1 (to explore weighting effect without noise removal process) and Scenario 2 (to explore weighting effect with additional noise removal process). Both scenarios will be executed on three different data conditions: original data without oversampling, oversampling using original data only (ROS: Random Oversampling) and oversampling using additional synthetic data (SMOTE, Borderline-SMOTE, and ADASYN). Figure 5 shows how the two testing scenarios are conducted to ascertain attribute weighting effect on identifying neighbors and noise.

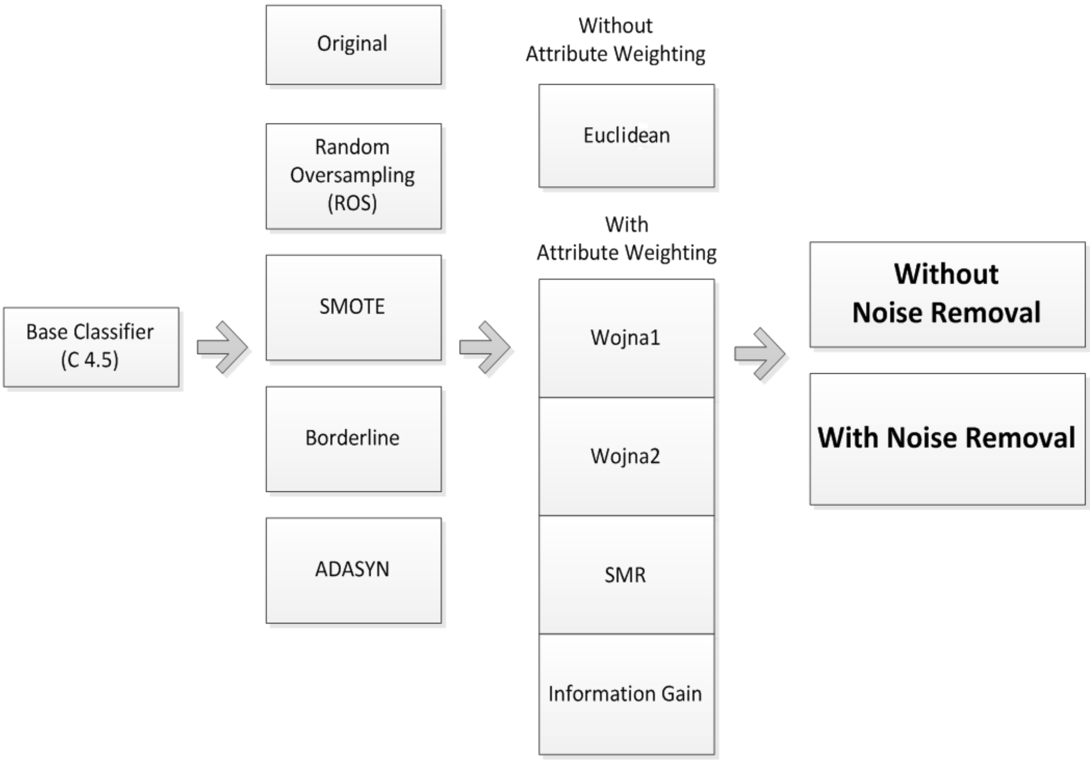

Figure 5. Two testing scenarios to ascertain attribute weighting effect

Table 8 shows the average results of Scenario 1 (attribute weighting without noise removal process) over all data. Table 9 shows comparison between non-weighting scheme and average result on four weighting schemes. Three best results are highlighted in bold and the rank is shown in superscripts. Figure 6 and Figure 7 show the number of the first ranks of all data from three different methods: original, random oversampling and oversampling with additional synthetic data (with attribute weighting and without attribute weighting).

From the result of Scenario 1, it is seen that the weighting scheme can improve accuracy when compared to a non-weighting scheme. Based on comparison results in Table 9 and the number of first ranks in Figure 6 and Figure 7, all weighting schemes give promising results in the minority recall (both pruned and unpruned conditions) compared to a non-weighting scheme. Figure 6 and Figure 7 show the number of first ranks on weighting scheme is larger in all metrics except on majority recall unpruned metric.

Table 8. Average results over all data on the Scenario 1

Method	*Pruned*			*Unpruned*		
	Recall (+/−)	*Precision (+/−)*	*F-Measure (+/−)*	*Recall (+/−)*	*Precision (+/−)*	*F-Measure (+/−)*
Original	63.10/**89.43**^{1}	**67.29**^{1}/87.77	63.74/**88.28**^{1}	64.71/**88.66**^{1}	**66.93**^{1}/88.12	64.47/**88.10**^{1}
ROS	70.31/**83.94**^{2}	**63.98**^{2}/88.96	65.31/**85.90**^{2}	69.75/**83.96**^{2}	**63.94**^{2}/88.87	64.96/**85.84**^{2}
SMOTE	74.72/**82.96**^{3}	**63.63**^{3}/89.73	67.20/**85.71**^{3}	74.63/**82.82**^{3}	63.50/89.76	67.10/85.62
SMOTE Infogain	75.72/82.30	62.71/89.88	67.08/85.39	75.38/82.10	62.39/89.82	66.81/85.22
SMOTE Wojna1	75.92/82.45	63.39/90.00	**67.58**^{3}/85.54	75.75/82.25	63.07/90.00	**67.28**^{3}/85.40
SMOTE Wojna2	75.30/82.70	63.15/89.72	67.16/85.55	75.43/82.19	62.71/89.73	66.90/85.23
SMOTE SMR	76.30/82.67	63.53/90.26	**67.92**^{1}/**85.90**^{2}	76.07/82.59	63.07/**90.23**^{3}	**67.58**^{1}/**85.71**^{3}
Borderline	73.50/81.82	63.39/89.38	66.23/84.80	74.46/81.23	62.54/89.60	66.24/84.54
Borderline Infogain	76.20/81.11	63.56/90.11	**67.72**^{2}/84.75	75.91/81.25	**63.51**^{3}/90.14	**67.54**^{2}/84.87
Borderline Wojna1	74.56/79.88	60.63/89.75	64.92/83.80	74.59/79.65	60.50/89.69	64.83/83.65
Borderline Wojna2	75.06/80.53	62.67/89.67	66.38/84.17	75.01/80.45	62.47/89.60	66.33/84.08
Borderline SMR	75.52/80.61	62.85/89.82	66.76/84.26	75.56/80.53	62.70/89.92	66.64/84.25
ADASYN	77.30/79.54	60.22/90.11	65.75/83.77	77.49/79.00	59.83/90.06	65.59/83.39
ADASYN Infogain	**79.18**^{1}/78.22	60.06/**90.82**^{1}	66.66/83.17	**78.94**^{1}/78.29	60.26/**90.67**^{1}	66.66/83.12
ADASYN Wojna1	77.00/79.75	60.17/**90.31**^{3}	65.84/84.05	77.01/79.30	59.70/90.21	65.51/83.70
ADASYN Wojna2	**78.04**^{2}/79.05	60.28/**90.55**^{2}	66.12/83.61	**77.89**^{3}/78.72	59.93/**90.46**^{2}	65.81/83.32
ADASYN SMR	**77.82**^{3}/78.79	59.33/90.24	65.74/83.35	**78.13**^{2}/78.50	59.31/90.22	65.84/83.16

Table 9. Non-weighting scheme versus weighting scheme results on the Scenario 1

Method	*Pruned*			*Unpruned*		
	Recall (+/−)	*Precision (+/−)*	*F-Measure (+/−)*	*Recall (+/−)*	*Precision (+/−)*	*F-Measure (+/−)*
SMOTE	74.72/**82.96**	**63.63**/89.73	67.20/**85.71**	74.63/**82.82**	**63.50**/89.76	67.10/**85.62**
Avg SMOTEWeighting	**75.81**/82.53	63.19/**89.96**	**67.44**/85.59	**75.66**/82.28	62.81/**89.94**	**67.14**/85.39
Borderline	73.50/**81.82**	**63.39**/89.38	66.23/**84.80**	74.46/**81.23**	**62.54**/89.60	66.24/**84.54**
Avg BorderWeighting	**75.33**/80.53	62.43/**89.84**	**66.44**/84.25	**75.27**/80.47	62.29/**89.84**	**66.33**/84.21
ADASYN	77.30/**79.54**	**60.22**/90.11	65.75/83.77	77.49/**79.00**	**59.83**/90.06	65.59/**83.39**
Avg ADASYNWeighting	**78.01**/78.95	59.96/**90.48**	**66.09**/**83.54**	**78.00**/78.70	59.80/**90.39**	**65.96**/83.32

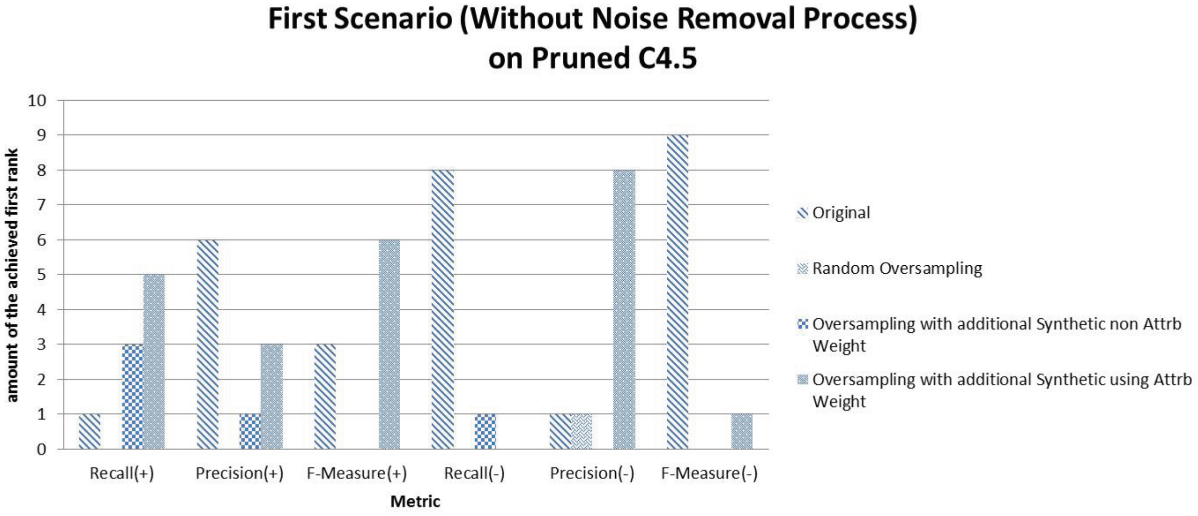

Figure 6. The number of first ranks of all data with pruned C4.5 on the Scenario 1

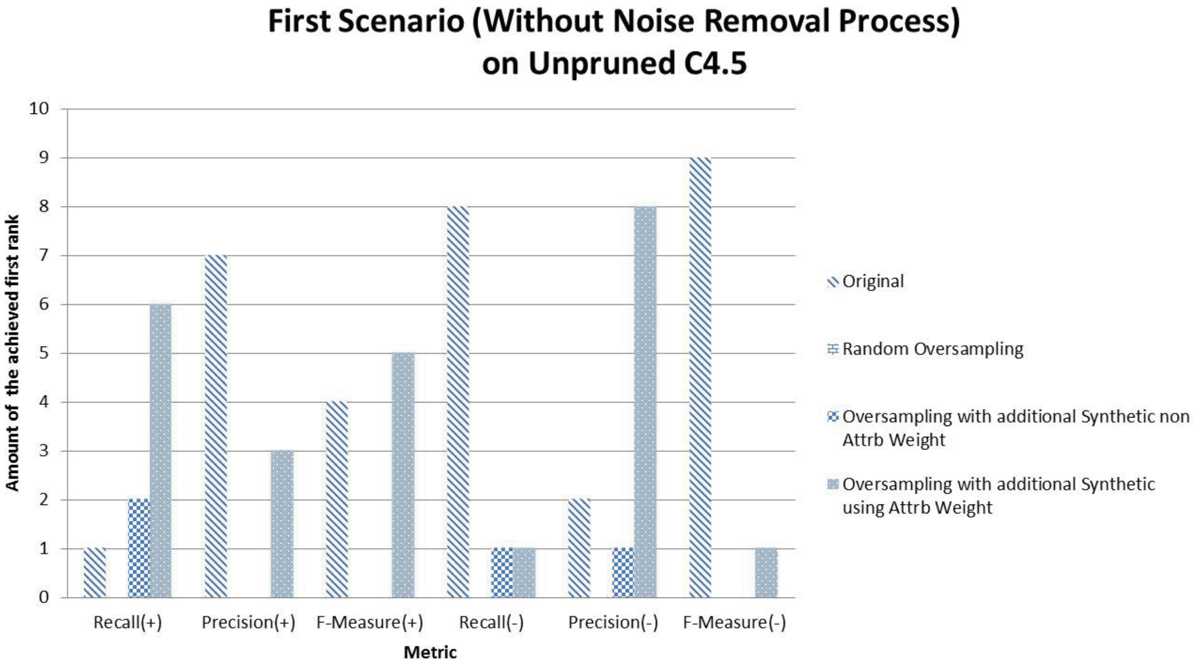

Figure 7. The number of first ranks of all data with unpruned C4.5 on the Scenario 1

Comparison between non-weighting scheme and weighting scheme shows that weighting scheme achieves better results on both minority recall and minority f-measure for SMOTE, Borderline-SMOTE and ADASYN on both pruned and unpruned conditions. The average improvement of weighting schemes compared to non-weighting scheme are as follows: 1.62% for minority recall in pruned condition, 1.04% for minority recall in unpruned condition, 0.397% for minority f-measure in pruned condition and also 0.253% for minority f-measure in unpruned condition.

The results of Scenario 2 are shown in Table 10 (average result over all data) and Table 11 (comparison between non-weighting scheme and average results on four weighting schemes). Three best results of accuracy for each metricare are highlighted in bold and superscripts font type in Table 10. Bold typeface numbers in Table 11 indicate the best results of the comparison. Figure 8 and Figure 9 show the number of the first ranks over all data from three different data conditions: original, random oversampling and oversampling with additional synthetic data. All data conditions are observed in two schemes (weighting scheme and non-weighting scheme).

From the result of Scenario 2, it can be seen that weighting scheme dominates the best three results in the minority recall, minority precision and minority f-measure. Weighting scheme also achieves the first best rank in minority recall compared to non-weighting scheme. Borderline-SMOTE and ADASYN methods provide better result on average of four weighting schemes both minority recall and minority precision for pruned and unpruned condition. Average results on four weighting schemes also show better performance in minority f-measure for SMOTE, Borderline-SMOTE and ADASYN methods for pruned and unpruned condition. Based on Figure 8 and Figure 9, on original data condition, weighting scheme can achieve better accuracy of all almost majority and minority metrics such as majority recall in pruned condition, majority precision in pruned and unpruned condition, and also minority and majority f-measure in pruned and unpruned condition. Similar results are seen on ROS method, and weighting scheme can achieve

Table 10. Average results over all data on the Scenario 2

Method	*Pruned*			*Unpruned*		
	Recall (+/−)	*Precision (+/−)*	*F-Measure (+/−)*	*Recall (+/−)*	*Precision (+/−)*	*F-Measure (+/−)*
Original *NR Euclidean	67.28/**91.55**^{3}	70.57/91.38	66.88/91.13	73.29/85.65	70.19/91.39	66.91/90.70
Original *NR Infogain	69.49/**91.78**^{2}	**71.20**^{3}/91.77	68.21/**91.52**^{2}	70.35/**91.34**^{1}	**70.42**^{3}/91.90	68.28/**91.36**^{1}
Original *NR SMR	69.85/**91.78**^{2}	**73.68**^{2}/91.40	69.08/**91.31**^{3}	70.07/**91.03**^{3}	**72.18**^{2}/91.29	68.92/**90.85**^{3}
Original *NR Wojna1	69.39/**92.18**^{1}	**74.31**^{1}/91.65	69.33/**91.61**^{1}	70.59/**91.14**^{2}	**73.40**^{1}/91.67	69.55/**91.07**^{2}
Original *NR Wojna2	67.24/90.99	70.48/91.23	67.40/90.88	67.92/90.14	69.26/91.26	66.77/90.53
ROS *NR Euclidean	77.37/88.14	69.16/93.14	71.21/90.19	76.57/88.63	69.81/92.98	71.28/90.38
ROS *NR Infogain	74.73/88.52	69.08/92.12	69.85/89.95	72.45/88.69	69.02/91.71	68.52/89.82
ROS *NR SMR	78.39/88.24	70.64/93.13	72.81/89.94	75.55/88.62	69.28/92.73	70.57/90.22
ROS *NR Wojna1	76.89/88.81	69.70/93.10	71.17/90.55	75.85/89.01	70.18/92.95	70.86/90.56
ROS *NR Wojna2	77.56/88.01	68.96/93.19	71.28/90.16	76.36/88.19	69.75/93.01	70.92/90.15
SMOTE *NR Euclidean	81.35/86.91	68.44/94.15	72.32/89.83	80.85/86.82	68.09/94.08	71.89/89.74
SMOTE *NR Infogain	80.49/88.49	70.92/93.68	**73.32**^{1}/90.63	80.40/87.70	69.86/93.57	**72.70**^{2}/90.14
SMOTE *NR SMR	81.22/86.58	68.58/93.59	72.69/89.42	80.78/86.44	68.35/93.54	72.23/89.37
SMOTE *NR Wojna1	81.41/87.01	67.80/94.10	72.09/89.97	80.90/86.98	67.88/94.03	71.63/89.88
SMOTE *NR Wojna2	81.67/86.51	68.14/94.06	72.78/89.66	81.16/86.51	68.14/93.92	**72.44**^{3}/89.52
Borderline *NR Euclidean	80.74/85.84	66.36/94.11	70.65/89.19	80.75/85.74	66.24/94.06	70.53/89.12
Borderline *NR Infogain	81.68/86.72	69.49/94.16	**72.98**^{3}/89.90	80.97/86.49	69.06/94.03	72.32/89.68
Borderline *NR SMR	81.56/85.71	67.60/93.68	71.54/88.93	81.36/85.43	67.51/93.65	71.36/88.73
Borderline *NR Wojna1	82.95/85.61	66.97/**94.47**^{3}	71.89/89.16	82.74/85.60	67.03/**94.40**^{3}	71.79/89.11
Borderline *NR Wojna2	80.49/85.51	66.98/94.11	70.73/88.88	80.41/85.54	67.32/94.08	70.90/88.90
ADASYN *NR Euclidean	82.13/85.50	66.03/94.20	71.13/89.00	81.98/85.50	66.05/94.15	71.08/88.98
ADASYN *NR Infogain	**84.63**^{1}/84.42	65.18/**94.55**^{2}	72.08/88.72	**84.44**^{1}/84.38	65.26/**94.47**^{2}	71.99/88.65
ADASYN *NR SMR	**83.35**^{3}/84.84	66.87/93.97	72.06/88.52	**83.32**^{3}/84.62	66.72/93.99	71.97/88.37
ADASYN *NR Wojna1	82.80/85.73	67.05/94.22	71.99/89.17	82.69/85.56	66.88/94.19	71.76/89.06
ADASYN *NR Wojna2	**84.43**^{2}/85.65	67.60/**94.80**^{1}	**73.22**^{2}/89.42	**84.43**^{2}/85.71	67.86/**94.82**^{1}	**73.33**^{1}/89.48

*NR = Noise Removal

Table 11. Non-weighting scheme versus weighting scheme results on the Scenario 2

Method	*Pruned*			*Unpruned*		
	Recall (+/−)	*Precision (+/−)*	*F-Measure (+/−)*	*Recall (+/−)*	*Precision (+/−)*	*F-Measure (+/−)*
Original *NR Euclidean	67.28/91.55	70.57/91.38	66.88/91.13	**73.29**/85.65	70.19/91.39	66.91/90.70
Avg of Ori*NR Weighting	**68.99/91.68**	**72.42/91.51**	**68.51/91.33**	69.73/**90.91**	**71.32/91.53**	**68.38/90.95**
ROS *NR Euclidean	**77.37**/88.14	69.16/**93.14**	71.21/**90.19**	**76.57**/88.63	**69.81/92.98**	**71.28/90.38**
Avg of ROS *NR Weighting	76.89/**88.40**	**69.60**/92.89	**71.28**/90.15	75.05/88.63	69.56/92.60	70.22/90.19
SMOTE *NR Euclidean	**81.35**/86.91	68.44/**94.15**	72.32/89.83	80.85/86.82	68.09/**94.08**	71.89/**89.74**
Avg of SMOTE *NR Weighting	81.20/**87.15**	**68.86**/93.86	**72.72/89.92**	**80.81/86.91**	**68.56**/93.77	**72.25**/89.73
Borderline *NR Euclidean	80.74/85.84	66.36/**94.11**	70.65/89.19	80.75/85.74	66.24/**94.06**	70.53/**89.12**
Avg of Border *NR Weighting	**81.67/85.89**	**67.76**/94.10	**71.78/89.22**	**81.37/85.77**	**67.73**/94.04	**71.59**/89.10
ADASYN *NR Euclidean	82.13/**85.50**	66.03/94.20	71.13/**89.00**	81.98/**85.50**	66.05/94.15	71.08/**88.98**
Avg of ADASYN *NR Weighting	**83.80**/85.16	**66.68/94.38**	**72.34**/88.96	**83.72**/85.07	**66.68/94.37**	**72.26**/88.89

*NR = Noise Removal

better accuracy of all almost majority and minority metrics such as minority and majority recall in pruned condition, majority precision in pruned condition, and also minority and majority f-measure prune condition. Lastly, in oversampling with additional synthetic data method, weighting scheme also shows promising results because it achieves the best result in all metrics both majority and minority in pruned and unpruned condition compared to oversampling with additional synthetic without weighting scheme.

The accuracy of classification using weighting scheme in the Scenario 2 is mostly (thirty eight of sixty for non-weighting scheme) better than the accuracy of classification using

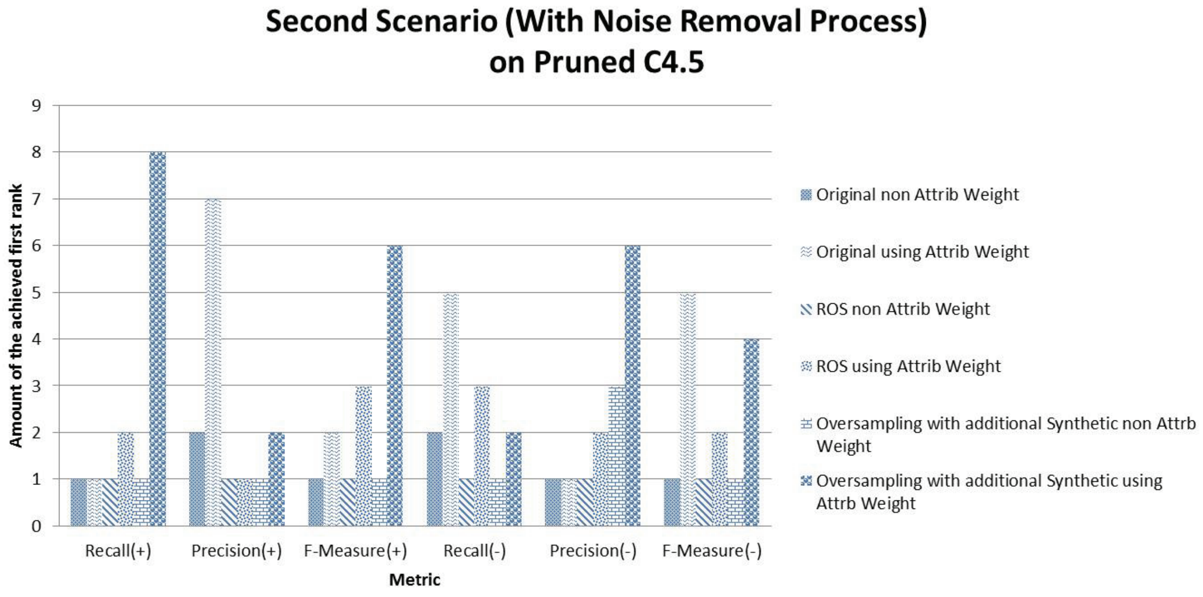

FIGURE 8. The number of first ranks of all data with pruned C4.5 on the Scenario 2

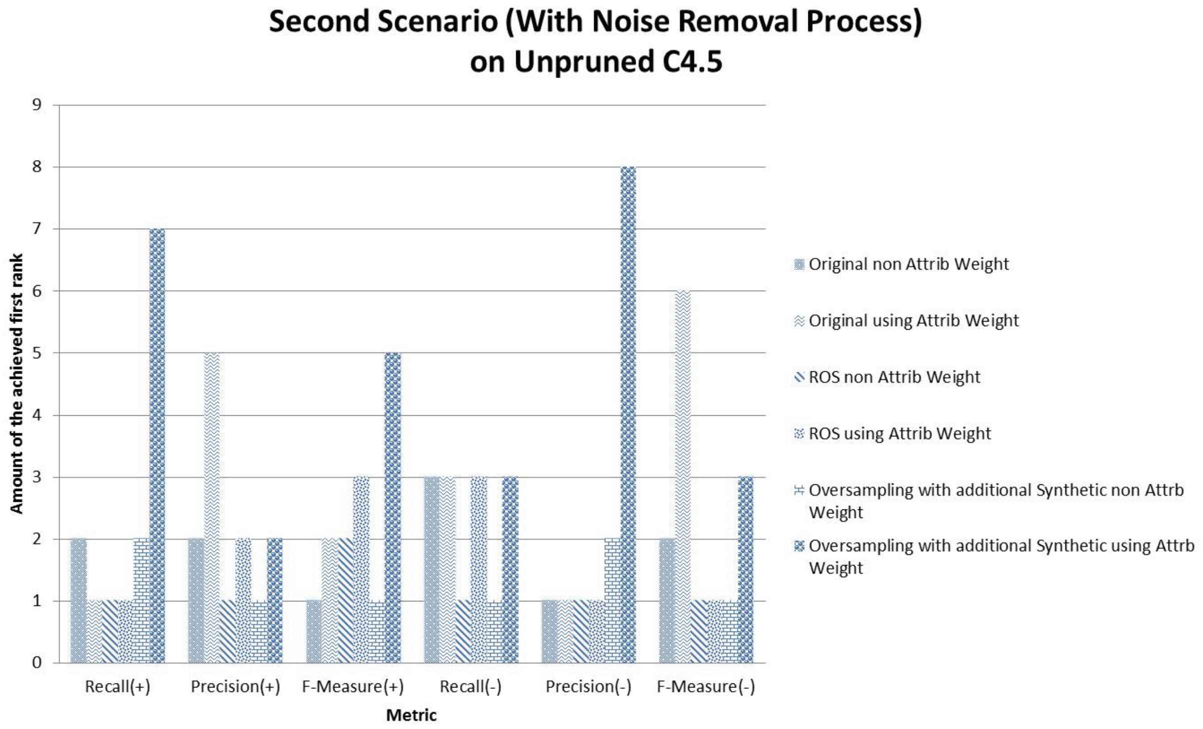

FIGURE 9. The number of first ranks of all data with unpruned C4.5 on the Scenario 2

non-weighting scheme. This is in line with the outcome of the Scenario 1 in which accuracy improvement in the Scenario 1 is caused by weighting scheme which gives more representative neighbors. Accuracy results in Scenario 2 are better than Scenario 1 because the noisy data were deleted more precisely by using more representative neighbors. It can be seen that on the same method, accuracy in Table 10 is higher than accuracy in Table 8. The average improvement of the Scenario 2 compared to the Scenario 1 can be

Table 12. Average improvement of the Scenario 2 compared to the Scenario 1

Metric	*Condition*	*Average Improvement in Percentage*
Minority recall	Pruned	9.17%
Minority recall	Unpruned	8.38%
Minority precision	Pruned	8.15%
Minority precision	Unpruned	8.39%
Minority f-measure	Pruned	8.37%
Minority f-measure	Unpruned	7.84%

Table 13. AWH-SMOTE performance average results on all data

Method	*Condition*	*Recall (+/−)*	*Precision (+/−)*	*F-Measure (+/−)*
AWH-SMOTE Euclidean	Pruned	78.82/86.20	66.08/93.49	69.87/88.993
AWH-SMOTE Infogain		**81.03**^{1}/86.30	**68.36**^{3}/**94.10**^{1}	**72.20**^{3}/**89.56**^{3}
AWH-SMOTE SMR		**80.76**^{3}/**86.64**^{3}	**68.55**^{2}/93.60	**72.46**^{2}/89.55
AWH-SMOTE Wojna1		80.75/**87.06**^{1}	68.07/**93.70**^{3}	72.08/**89.78**^{2}
AWH-SMOTE Wojna2		**80.99**^{2}/**86.89**^{2}	**68.86**^{1}/**93.97**^{2}	**72.49**^{1}/**89.79**^{1}
AWH-SMOTE Euclidean	Unpruned	78.63/86.21	65.68/93.41	69.71/88.98
AWH-SMOTE Infogain		**80.85**^{1}/86.36	**68.66**^{1}/**94.07**^{1}	**72.33**^{1}/**89.58**^{3}
AWH-SMOTE SMR		**80.57**^{3}/**86.74**^{2}	**68.51**^{2}/93.48	**72.31**^{2}/89.56
AWH-SMOTE Wojna1		80.53/**87.17**^{1}	67.98/**93.65**^{3}	71.84/**89.69**^{1}
AWH-SMOTE Wojna2		**80.67**^{2}/**86.69**^{3}	**68.26**^{3}/**93.86**^{2}	**72.05**^{3}/**89.65**^{2}

seen in Table 12. Overall it can be concluded that adding attribute weighting scheme can improve the accuracy of classification in various methods (other than AWH-SMOTE).

In the next section, we observe non-weighting scheme in AWH-SMOTE using Euclidean distance as k-neighbors and noise identification method. Euclidean distance method is tested in AWH-SMOTE to see the effects of neighbor identification on AWH-SMOTE. We also observe the performance of AWH-SMOTE with weighting scheme is compared to all other methods on both minority and majority class (with noise removal process and without noise removal process) to see the effects of our proposed new sampling method.

4.3. **Summaries of overall AWH-SMOTE performance.** In this section, the performance of AWH-SMOTE is observed in two circumstances. First, the performance of AWH-SMOTE using non-attribute weighting method (i.e., Euclidean distance) is compared with the performance of AWH-SMOTE using attribute weighting method. The results are shown in Table 13. Second, the performance of AWH-SMOTE is compared with the performance of all other methods on both minority and majority class (with noise removal process and without noise removal process). The results are shown in Figure 10 and Figure 11. The performance of AWH-SMOTE is measured in terms of the accuracy of classification results.

Table 13 shows that AWH-SMOTE with attribute weighting methods achieves three best ranks on all cases. For unpruned condition, AWH-SMOTE using Information Gain as attribute weighting method gives the best results on minority recall, minority precision and also minority f-measure. AWH-SMOTE also shows better performance on minority

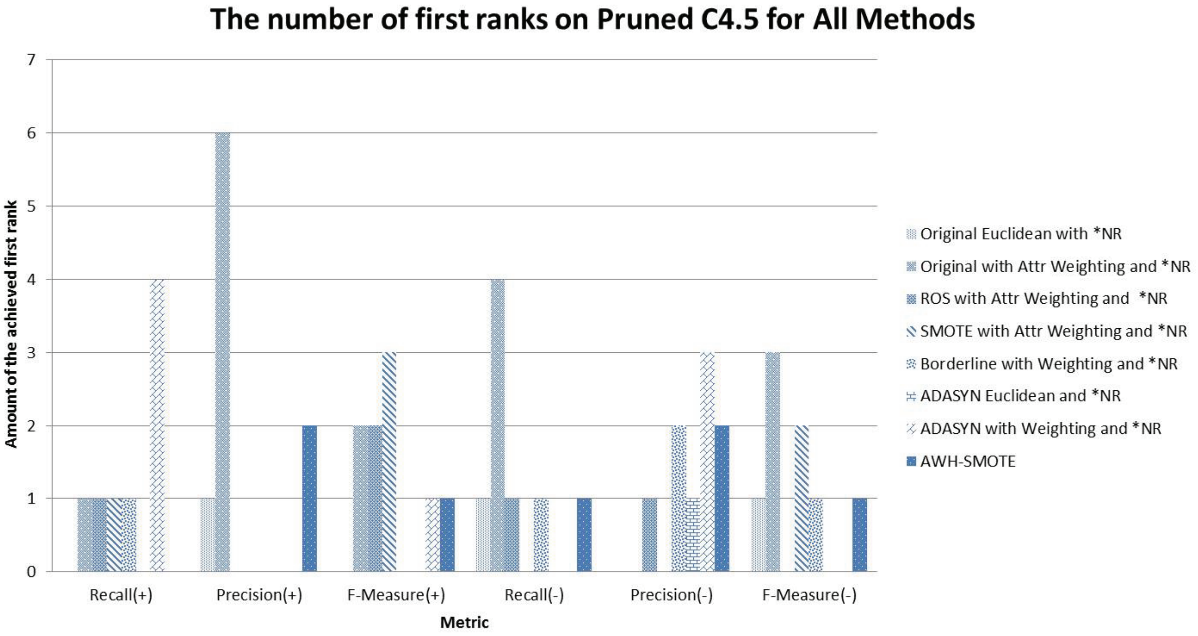

FIGURE 10. The number of first ranks of all data with pruned C4.5 on all methods

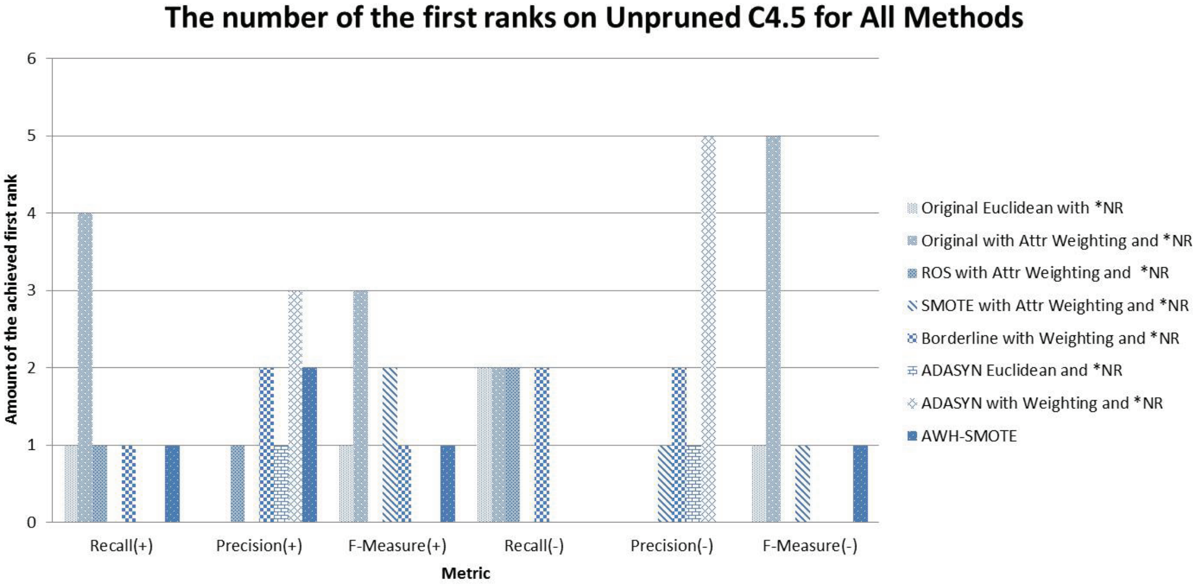

FIGURE 11. The number of first ranks of all data with unpruned C4.5 on all methods

precision and minority f-measure for both pruned and unpruned condition compared to other oversampling with additional synthetic methods. Figure 10 summarizes the number of the first ranks of all data in all circumstances and all methods for pruned condition while Figure 11 summarizes those for unpruned condition. AWH-SMOTE achieves the first rank on minority precision metric for glass0 and yeast-2_vs_4 data and also the first rank on minority f-measure metric for Haberman data. AWH-SMOTE also achieves the first rank on majority recall metric for yeast-2_vs_4 data, majority precision metric for ecoli3 data and also majority f-measure metric for ecoli3 and yeast-2_vs_4 data. Other weighting schemes with noise removal process also achieve better performance than non-weighting scheme on both minority and majority classes such as ROS *NR Infogain for ecoli2 data, ADASYN *NR Infogain for ecoli3 data, Borderline-SMOTE *NR Infogain for pima data and SMOTE *NR SMR for yeast-2_vs_4 data.

From the test results in this section, we can see that the weighting scheme method gives a significant accuracy improvement both for identifying neighbors and noise. The three best accuracy results are dominated by weighting scheme method in both noise removal process and without noise removal process. This is in line with the finding that attribute weighting scheme can improve the accuracy of classification (in Section 4.2).

The proposed new selective sampling method improves minority precision and minority f-measure proven by the performance on minority precision and minority f-measure for both pruned and unpruned condition of AWH-SMOTE is better compared to other oversampling with additional synthetic methods.

5. Conclusions and Future Work.

One method to combat the problem of imbalanced data set is oversampling with additional synthetic data. SMOTE algorithm is one of the state of the art of the oversampling methods with the additional synthetic data. We have presented AWH-SMOTE, a development of the SMOTE algorithm which introduces 1) more representative *kNN* using attribute weighting scheme, 2) a new concept for selecting sample methods using occurrence data in the *kNN* hub.

In our experiments, nine numerical binary data classes with varying degrees of imbalanced dataset from Keel data repository were used to compare the performance of AWH-SMOTE with the performance of four algorithms (Random Oversampling, SMOTE, Borderline-SMOTE and ADASYN). Testing has been conducted on two scenarios to evaluate the attribute weighting effect on identifying neighbors and noise and also on two circumstances to evaluate AWH-SMOTE performance. Three assessment metrics (recall, precision and f-measure) on both minor and major classes have been used as measurement of accuracy of classification results.

Some conclusions of our experiments results are as follows.

1) Weighting attribute in *kNN* provides more representative neighbors and noise, so accuracy of the existing algorithms can be increased as many 1% for weighting schemes without noise removal process and 9% for weighting schemes with noise removal process.
2) AWH-SMOTE also shows better performance on minority precision and minority f-measure for both pruned and unpruned condition compared to other oversampling with additional synthetic methods. AWH-SMOTE using Information Gain as attribute weighting method raises best performance on minority recall, minority precision and minority f-measure.
3) Various kinds of data have been used in the testing, and AWH-SMOTE algorithm in general achieves good performance on other numerical binary data which contains imbalanced data set.

This paper still retains some research topics that we are focusing on: applying another attribute weighting methods, applying sophisticated noise reduction techniques and applying AWH-SMOTE in multiclass imbalanced data set.

REFERENCES

[1] G. E. A. P. A. Batista, R. C. Prati and M. C. Monard, A study of the behavior of several methods for balancing machine learning training data, *SIGKDD Explor. Newsl.*, vol.6, no.1, pp.20-29, 2004.

[2] T. Fawcett and F. Provost, Adaptive fraud detection, *Data Min. Knowl. Discov.*, vol.316, no.1, pp.291-316, 1997.

[3] G. Wang, D-self-SMOTE: New method for customer credit risk prediction based on self-training and smote, *ICIC Express Letters, Part B: Applications*, vol.9, no.3, pp.241-246, 2018.

[4] J. Burez and D. Van den Poel, Handling class imbalance in customer churn prediction, *Expert Syst. Appl.*, vol.36, no.3, pp.4626-4636, 2009.

[5] X. Wan, J. Liu, W. K. Cheung and T. Tong, Learning to improve medical decision making from imbalanced data without a priori cost, *BMC Med. Inform. Decis. Mak.*, vol.14, no.1, p.111, 2014.

[6] H. M. Nguyen, E. W. Cooper and K. Kamei, Adaptive data reuse for classifying imbalanced and concept-drifting data streams, *International Journal of Innovative Computing, Information and Control*, vol.8, no.7(B), pp.4995-5010, 2012.

[7] G. Wang, S. Yang and J. Ma, COS-training: A new semi-supervised learning method for keyphrase extraction based on co-training and SMOTE, *ICIC Express Letters, Part B: Applications*, vol.6, no.1, pp.233-238, 2015.

[8] C. Márquez-Vera, A. Cano, C. Romero and S. Ventura, Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data, *Appl. Intell.*, vol.38, no.3, pp.315-330, 2013.

[9] T. Fahrudin, J. L. Buliali and C. Fatichah, Predictive modeling of the first year evaluation based on demographics data: Case study students of Telkom University, Indonesia, *International Conference on Data and Software Engineering (ICoDSE)*, 2016.

[10] D. T. Jo and N. Japkowicz, Class imbalances versus small disjuncts, *ACM SIGKDD Explor. Newsl. – Spec. Issue Learn. from Imbalanced Datasets*, vol.6, no.1, pp.40-49, 2004.

[11] J. Stefanowski, Dealing with data difficulty factors while learning from imbalanced data, in *Challenges in Computational Statistics and Data Mining. Studies in Computational Intelligence*, S. Matwin and J. Mielniczuk (eds.), Springer, Cham, 2016.

[12] B. Krawczyk, Learning from imbalanced data: Open challenges and future directions, *Prog. Artif. Intell.*, vol.5, no.4, pp.221-232, 2016.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, *J. Artif. Intell. Res.*, vol.16, pp.321-357, 2002.

[14] J. Laurikkala, Improving identification of difficult small classes by balancing class distribution, *Artificial Intelligence in Medicine*, Springer Berlin Heidelberg, pp.63-66, 2001.

[15] M. Kubat and S. Matwin, Addressing the curse of imbalanced training sets: One sided selection, *Proc. of the 14th Int. Conf. Mach. Learn.*, vol.4, no.1, pp.179-186, 1997.

[16] A. Adam, Z. Ibrahim, M. I. Shapiai, L. C. Chew, L. W. Jau, M. Khalid and J. Watada, A two-step supervised learning artificial neural network for imbalanced dataset problems, *International Journal of Innovative Computing, Information and Control*, vol.8, no.5(A), pp.3163-3172, 2012.

[17] L.-S. Chen, C.-C. Hsu and Y.-S. Chang, Developing a novel two-phase learning scheme for the class imbalance problem, *International Journal of Innovative Computing, Information and Control*, vol.6, no.11, pp.4979-4994, 2010.

[18] Q. Cao and S. Wang, Applying over-sampling technique based on data density and cost-sensitive SVM to imbalanced learning, *Proc. of the 4th International Conference on Information Management, Innovation Management and Industrial Engineering*, vol.2, pp.543-548, 2011.

[19] T. Fahrudin, J. L. Buliali and C. Fatichah, RANDSHUFF: An algorithm to handle imbalance class for qualitative data, *Int. Rev. Comput. Softw.*, vol.11, no.12, 2016.

[20] H. Han, W. Wang and B. Mao, Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning, *Proc. of International Conference on Intelligent Computing, Part I*, Hefei, China, pp.878-887, 2005.

[21] H. He, Y. Bai, E. A. Garcia and S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, *IEEE International Joint Conference on Neural Networks, (IEEE World Congress on Computational Intelligence)*, no.3, pp.1322-1328, 2008.

[22] N. V. Chawla, A. Lazarevic, L. O. Hall and K. Bowyer, SMOTEBoost: Improving prediction of the minority class in boosting, *Principles of Knowledge Discovery in Databases*, pp.107-119, 2003.

[23] X. Fang, H. Zhang, S. Gao and Y. Tan, Imbalanced web spam classification based on nested rotation forest, *ICIC Express Letters*, vol.9, no.3, pp.937-944, 2015.

[24] G. Wang, J. Ma and L. Huang, ASE-bagging: An embedded approach for imbalanced customer credit risk assessment, *ICIC Express Letters, Part B: Applications*, vol.2, no.4, pp.787-791, 2011.

[25] F. Yue and G. Wang, COSDF: A novel method for software defect detection based on co-training and SMOTE with density based noise filtering strategy, *ICIC Express Letters*, vol.11, no.10, pp.1507-1513, 2017.

[26] E. Ramentol, Y. Caballero, R. Bello and F. Herrera, SMOTE-RS B*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory, *Knowl. Inf. Syst.*, vol.33, no.2, pp.245-265, 2012.

[27] J. A. Sáez, J. Luengo, J. Stefanowski and F. Herrera, SMOTE-IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, *Inf. Sci. (Ny).*, vol.291, pp.184-203, 2015.

[28] C. Bunkhumpornpat, K. Sinapiromsaran and C. Lursinsap, Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, *Pacific-Asia Conference on Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, vol.5476, pp.475-482, 2009.

[29] T. Maciejewski and J. Stefanowski, Local neighbourhood extension of SMOTE for mining imbalanced data, *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pp.104-111, 2011.

[30] S. Barua, M. M. Islam, X. Yao and K. Murase, MWMOTE – Majority weighted minority oversampling technique for imbalanced data set learning, *IEEE Trans. Knowl. Data Eng.*, vol.26, no.2, pp.405-425, 2014.

[31] M. R. Prusty, T. Jayanthi and K. Velusamy, Weighted-SMOTE: A modification to SMOTE for event classification in sodium cooled fast reactors, *Prog. Nucl. Energy*, vol.100, pp.355-364, 2017.

[32] P.-N. Tan, M. Steinbach and V. Kumar, *Introduction to Data Mining*, 2006.

[33] A. Wojna, Analogy-based reasoning in classifier construction, *Lecture Notes in Computer Science*, pp.277-374, 2004.

[34] A. C. Frazee, M. A. Hathcock and S. C. B. Prins, Distance functions and attribute weighting in a k-nearest neighbors classifier, *Electronic Proceedings of Undergraduate Mathematics Day*, no.3, pp.1-13, 2010.

[35] N. Tomašev, M. Radovanović, D. Mladenić and M. Ivanović, The role of hubness in clustering high-dimensional data, *IEEE Trans. Knowl. Data Eng.*, vol.26, no.3, pp.739-751, 2014.

[36] N. Tomašev and D. Mladenić, Class imbalance and the curse of minority hubs, *Knowledge-Based Syst.*, vol.53, pp.157-172, 2013.

[37] C. C. Aggarwal, A. Hinneburg and D. A. Keim, On the surprising behavior of distance metrics in high dimensional space, *International Conference on Database Theory*, pp.420-434, 2001.

[38] M. S. Elgamel and A. Dandoush, A modified Manhattan distance with application for localization algorithms in ad-hoc WSNs, *Ad Hoc Networks*, vol.33, pp.168-189, 2015.

[39] S. Gupta and V. Bhatia, A manhattan distance approach for energy optimization in wireless sensor network, *Proc. of the 1st Int. Conf. Next Gener. Comput. Technol.*, 2015.

[40] R. Kumar, Analysis of shape alignment using Euclidean and Manhattan distance metrics, *International Conference on Recent Innovations in Signal Processing and Embedded Systems (RISE)*, pp.326-331, 2017.

[41] M. D. Malkauthekar, Analysis of Euclidean distance and Manhattan distance measure in face recognition, *The 3rd International Conference of Computational Intelligence and Information Technology (CIIT)*, vol.1, no.4, pp.3-7, 2013.

[42] F. X. Arias, H. Sierra, L. O. Jimenez-Rodriguez and E. Arzuaga, Supervised sparse-representation classification on hyperspectral images using the city-block distance to improve performance, *The 8th International Conference of Pattern Recognition Systems (ICPRS 2017)*, no.3, 2017.

[43] S. Vlachothanasi and G. Manis, Using distances to classify recordings of young and elderly subjects, *Computing in Cardiology*, vol.44, pp.1-4, 2017.

[44] J.-J. Aucouturier and F. Pachet, Improving timbre similarity: How high is the sky?, *J. Negat. Results Speech Audio Sci.*, vol.1, no.1, pp.1-13, 2004.

[45] *Weka 3: Data Mining Software in Java*, Machine Learning Group at the University of Waikato, 2016, http://www.cs.waikato.ac.nz/ml/weka/, [Accessed: 01-Sep-2017].

[46] S. Visa and A. Ralescu, Issues in mining imbalanced data sets – A review paper, *Proc. of the 16th Midwest Artificial Intelligence and Cognitive Science Conference*, pp.67-73, 2005.

[47] H. He and E. A. Garcia, Learning from imbalanced data, *IEEE Trans. Knowl. Data Eng.*, vol.21, no.9, pp.1263-1284, 2009.