Journal of King Saud University – Computer and Information Sciences 34 (2022) 3413–3423

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

SMOTE-LOF for noise identification in imbalanced data classification

Asniar [a,c,*], Nur Ulfa Maulidevi [a,b], Kridanto Surendro [a]

[a] *School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha 10, Bandung, Indonesia*
[b] *PUI-PT AI-VLB (Artificial Intelligence for Vision, Natural Language Processing & Big Data Analytics), Indonesia*
[c] *School of Applied Science, Telkom University, Jalan Telekomunikasi, Terusan Buah Batu, Bandung, Indonesia*

ARTICLE INFO

Article history:
Received 4 August 2020
Revised 27 December 2020
Accepted 20 January 2021
Available online 9 February 2021

Keywords:
Imbalanced data
SMOTE
Noisy data
Outliers
Predictive accuracy

ABSTRACT

Imbalanced data typically refers to a condition in which several data samples in a certain problem is not equally distributed, thereby leading to the underrepresentation of one or more classes in the dataset. These underrepresented classes are referred to as a minority, while the overrepresented ones are called the majority. The unequal distribution of data leads to the machine's inability to carry out predictive accuracy in determining the minority classes, thereby causing various costs of classification errors. Currently, the standard framework used to solve the unequal distribution of imbalanced data learning is the Synthetic Minority Oversampling Technique (SMOTE). However, SMOTE can produce synthetic minority data samples considered as noise, which is also part of the majority classes. Therefore, this study aims to improve SMOTE to identify the noise from synthetic minority data produced in handling imbalanced data by adding the Local Outlier Factor (LOF). The proposed method is called SMOTE-LOF, and the experiment was carried out using imbalanced datasets with the results compared with the performance of the SMOTE. The results showed that SMOTE-LOF produces better accuracy and f-measure than the SMOTE. In a dataset with a large number of data examples and a smaller imbalance ratio, the SMOTE-LOF approach also produced a better AUC than the SMOTE. However, for a dataset with a smaller number of data samples, the SMOTE's AUC result is arguably better at handling imbalanced data. Therefore, future research needs to be carried out using different datasets with combinations varying from the number of data samples and the imbalanced ratio.

1. Introduction

Imbalanced data typically refers to a condition in which a number of data samples in a certain problem is not equally distributed (Fernándezet al., 2018a). This disproportion in distribution occurs when the number of data samples representing a class is significantly lower than those in other classes, thereby leading to the underrepresentation of one or more classes in the dataset (Chawla et al., 2002; Durán-rosal et al., 2018; Fernándezet al., 2018a; Gutiérrez et al., 2017). These underrepresented classes are referred to as the minority or positive class, while others are called the majority or negative class (Chen et al., 2019; Fahrudin et al., 2019; Fernándezet al., 2018a; Galar et al., 2012; Mohamad et al., 2019). Imbalanced data is a type of problematic data different from the big, uncertain and inconsistent data set (Mohamad et al., 2019).

Many real world applications utilize highly imbalanced data, with the target variable located in the minority class (Branco et al., 2016; Bunkhumpornpat et al., 2009; Chawla et al., 2004; Chen et al., 2019). This rare or uncommon value of the target variable is usually related to a highly relevant and important event for end users (Branco et al., 2016; Chen et al., 2019). Therefore, despite its scarcity, data samples of minority classes have a higher level of importance in some cases, such as diagnosis of rare diseases, fraud detection, churn detection, identifying students' academic failures, unusual returns on the stock market, disaster anticipation, etc. (Branco et al., 2016; Fahrudin et al., 2019). In many practical applications, it is often more important to correctly classify samples of the minority class than the majority (Sanz et al., 2015). This is because the results of predictive accuracy of machine learning are inaccurate in predicting the minority class, thereby leading to various costs of classification errors (Chawla et al., 2002;

* Corresponding author.
E-mail addresses: asniar@telkomuniversity.ac.id (Asniar), ulfa@informatika.org (N.U. Maulidevi), endro@informatika.org (K. Surendro).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.01.014
1319-1578/© 2021 The Authors. Published by Elsevier B.V. on behalf of King Saud University.

Fernándezet al., 2018a). Furthermore, this is also due to habitual biasness of the classifier algorithms of machine learning toward the majority class.

Consequently, the overall accuracy of the majority and minority classes are very high and low, in accordance with the frequently classified data samples. Ideally, the expected accuracy of both classes is 100% (Chen et al., 2019; Fernándezet al., 2018a). According to (Barandela et al., 2003; Fernándezet al., 2018a), the overall accuracy no longer holds as the proper measurement in the scenario of imbalanced data due to its inability to show the number of data samples that are correctly classified from different classes. Therefore, the imbalanced data need to be properly handled to improve the accuracy of machine learning in predicting the minority class without damaging their accuracy in predicting the majority class.

Undersampling and oversampling approaches are the standard techniques used in handling imbalanced data, however, both have their own limitations. For instance, undersampling causes more erasure of data samples which ultimately leads to problems in data shortage, with an increase in the probability of losing important data (Fernándezet al., 2018b; Pak et al., 2018; Wasikowski and Chen, 2010), while oversampling leads to the duplication of original data, thereby causing overfitting of minority classes (Chawla, 2009; Fernándezet al., 2018a). The issue of oversampling has been controlled with the Synthetic Minority Oversampling Technique (SMOTE), which is a technique that increases synthetic data based on the closest kNN of each instance of the minority class (Chawla et al., 2002). With SMOTE, the original data are used to synthesize new minority data that are different from the original, thereby alleviating the impact of overfitting (Fahrudin et al., 2019; Pak et al., 2018). According to (Fernándezet al., 2018b), SMOTE has been the standard in the learning frame work of imbalanced data, however, this technique is known to produce noise, thereby risking synthetic data samples of the minority class to be recognized as part of the majority (Barua et al., 2014; Ramentol et al., 2012; Sáez et al., 2015).

Several studies have been carried out to improve the errors associated with the SMOTE, such as a study that combined the SMOTE method with Tomek Links and Wilson's Edited Nearest Neighbor Rule data cleansing method (Batista et al., 2004). Another study is the FRIPS-SMOTE-FRBPS that used fuzzy as the prototype selection approach for cleaning data noise before and after the SMOTE process (Verbiest et al., 2014). Meanwhile, the study that combined the SMOTE method with the data selection method was the RST (Rough Set Theory) approach to reselect each synthetic data example produced by the SMOTE based on similarity relations in attempt to eliminate noise (Ramentol et al., 2012). Furthermore, SMOTE-IPF that attempted to combine a SMOTE-based oversampling technique with an iterative filtering method called IPF (Iterative-Partitioning Filter) to eliminate synthetic data examples that are considered as noise (Sáez et al., 2015). The other studies are LN-SMOTE, which exploits more precisely information about the local neighbourhood of the considered examples (MacIejewski and Stefanowski, 2011) and selective oversampling for empowering SMOTE (Nnamoko and Korkontzelos, 2020). However, those studies still exhibit a number of shortcomings such as inability of users to determine the distribution of the class produced, the complex rules and repeated use of erasure, which are considered as noise. Therefore, this study aims to determine a new noise identification method without having to commit repeated erasure of data samples considered as noise.

The local outlier factor (LOF) (Breuniq et al., 2000) originally used to detect samples of outlier, have been well researched (Tu et al., 2018). Therefore, this study aims to develop the SMOTE-LOF to improve the traditional SMOTE by adding Local Outlier Factor (LOF) to identify noise from synthetic minority data and improve the performance of predictive accuracy in handling imbalanced data. The proposed method is composed of five main steps (You et al., 2020). Firstly, determine the k-nearest neighbors (kNN) of each minority class data sample from SMOTE. Secondly and thirdly, calculate the k-distance and reachability distance, respectively. Fourthly, calculate the local reachability density and the equivalent LOF value. Finally, analyze and identify the noises based on the LOF value. The experiment was carried out using imbalanced datasets followed by comparing the accuracy results with those achieved by SMOTE.

This research is organized as follows. Section 2 presents the problem of imbalanced data classification and the impact of noise. Section 3 discusses the method proposed, while Section 4 describes the experimental framework. Furthermore, Section 5 presents the analysis of experimental results and evaluation, while Section 6 contains the discussion on the compatibility of the proposed method as well as future research possibilities. Lastly, the research is concluded in Section 7.

2. Imbalanced data classification

This section introduces the imbalanced data problems in Section 2.1, with the impact of noise described in Section 2.2.

2.1. The imbalanced data problems

According to (Krawczyk, 2016), imbalanced classification problems is classified into binary and multi-class. In binary imbalanced classification, the relationship between classes is well-defined in which one of them is the majority, while the other is the minority (Krawczyk, 2016). Imbalanced class distribution occurs when instances in the majority class exceed the those in the minority (Chen et al., 2019; Galar et al., 2012).

Fig. 1 represents a classification problem of two imbalanced data classes with a ratio of 1:100, which means that for every positive class (minority samples) there were 100 negative classes (majority samples). The positive class samples were denoted by the blue star '*', while those in the negative class were represented by the red dot '.' (Fernándezet al., 2018a). Based on this figure, it can be observed that the positive classes were underrepresented, with difficulty in determining the boundary decision that separates the two classes.

A dataset can be categorized as imbalanced data in accordance with the imbalance ratio (IR), which is defined as a proportion of the number of data samples in the majority class of those in the minority (Fernández et al., 2010). The equations of IR measure are indicated using the following formula (Sáez et al., 2015):

$$IR = \frac{N^-}{N^+} \tag{1}$$

where N^- and N^+ are the number of samples in the majority and minority classes, respectively. Therefore, a dataset is imbalanced when IR > 1 (Sáez et al., 2015).

The problem arising from this imbalanced data classification is that the classifier algorithm of machine learning is typically known to be biased toward the majority class. Consequently, the overall accuracy is high, while those of the majority and minority classes are high and low, respectively (Chen et al., 2019; Fernándezet al., 2018a). Therefore, imbalanced data handling is needed to improve the accuracy of predictive performance, without damaging the accuracy in predicting the minority class.

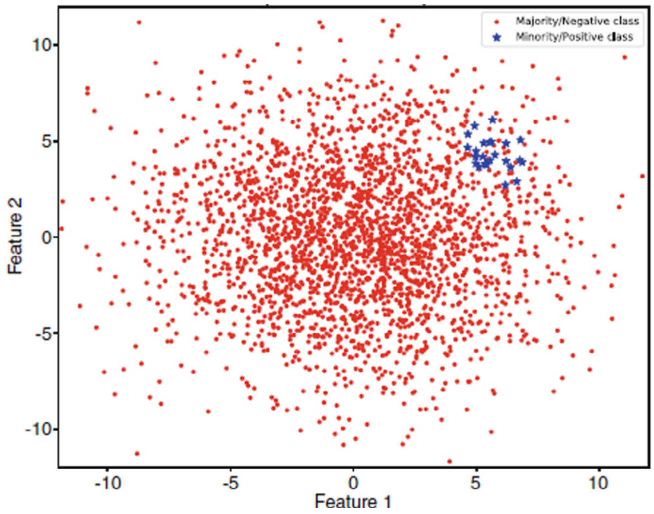

Fig. 1. Imbalanced data between two classes with a ratio of 1:100 (Fernándezet al., 2018a).

2.2. The impact of noise in imbalanced data classification

Noise is defined as a wrong label (class noise) or errors in the attribute value (attribute noise) (Salgado et al., 2016). Fig. 2 shows the data samples comprising of noise, such as borderline and safe samples. Where n, b, and s denotes noise, borderline, and safe samples (Sáez et al., 2015).

According to (Napierała et al., 2010), noise complicates the task of the resampling method even when the data have already been balanced with the SMOTE. The samples of the minority class's synthetic data sometimes becomes part of the majority class (Barua et al., 2014; Chawla et al., 2002; Ramentol et al., 2012; Sáez et al., 2015).

3. The combination of SMOTE and local outlier factor to overcome noise problem in handling imbalanced data

This section discusses the proposed method which is the combination of the SMOTE and Local Outlier Factor (LOF) to overcome noise problem in handling imbalanced data. Section 3.1 focuses on the SMOTE (Chawla et al., 2002) as the foundation of the proposed method, while Section 3.2 focuses on the proposal of using LOF to identify noise produced by the SMOTE.

3.1. Synthetic Minority Over-sampling Technique

The Synthetic Minority Oversampling Technique (SMOTE) is an oversampling process achieved using additional synthetic data (Chawla et al., 2002). According to (Fahrudin et al., 2019; Pak et al., 2018), the original data obtained using SMOTE are used to synthesize new minority data that are different from the original ones, thereby alleviating the impact of overfitting on the minority class.

Algorithm 1 *SMOTE*(T, N, k)

```
Input: Number of minority class samples $T$, Amount of SMOTE $N$%, Number of nearest neighbors $k$
Output: $(N/100)$ * $T$ synthetic minority class samples
(*If N is less than 100%, randomize the minority class samples because only a random percent of them is going to be SMOTEd. * )
if N less than 100
  then Randomize the T minority class samples
       $T = (N/100) * T$
       $N = 100$
endif
$N = (int)(N/100)$ (*The amount of SMOTE is assumed to be in integral multiples of 100. * )
$k$ = Number of nearest neighbors
numattrs = Number of attributes
Sample[ ][ ]: array for original minority class samples
newindex: keeps a count of number of synthetic samples generated, initialized to 0
Synthetic[ ][ ]: array for synthetic samples
 (*Compute k nearest neighbors for each minority class sample only. * )
for $i \leftarrow 1$ to $T$
    Compute k nearest neighbors for i, and save the indices in the nnarray
    Populate(N, i, nnarray)
endfor
 Populate(N, i, nnarray) (*Function to generate the synthetic samples. * )
while $N \neq 0$
    Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
    for $attr \leftarrow 1$ to numattrs
      Compute: dif = Sample[nnarray[nn]][attr] – Sample[i][attr]
      Compute: gap = random number between 0 and 1
      Synthetic[newindex][attr] = Sample[i][attr] + gap * dif
    endfor
    newindex++
    $N = N - 1$
endwhile
return (*End of Populate. * )
End of Pseudo-Code.
```

The SMOTE is based on the idea of the nearest neighbor algorithm (kNN) and assumes that a synthetic data sample can be interpolated between an original and one of the closest neighbors. The SMOTE algorithm calculates the neighbor environment of each data sample from the minority class, randomly selects one of its neighbors and makes synthetic data through the interpolation of data between each sample and the nearest neighbor selected. When the number of synthetic data samples to be made is smaller than the size of the original dataset, the algorithm is randomly selected and an original data sample is used to create synthetic data samples. Conversely, when the number of synthetic data samples to be made is greater than the size of the original dataset, the algorithm iteratively create synthetic data samples using predetermined oversampling ratio (Chawla et al., 2002; Gutiérrez et al., 2017).

The algorithm SMOTE (Chawla et al., 2002) needs inputs in the form of the number of minority data samples (T), oversampling ratio (N), and the nearest neighbors (k). The main process is searching and determining the nearest neighbors, followed by synthetic generation through data interpolation between each minority instance and the nearest neighbors (Chawla et al., 2002).

3.2. Local Outlier Factor for identifying noise generated by the SMOTE

According to Foreman (2014) an outlier is the odd point in a dataset. It is also known as abnormalities, discordances, deviations or anomalies, while noise can be defined as the wrong label (class noise) or errors in the attribute value (attribute noise) (Salgado et al., 2016). Typically, most of outliers are noise, however, sometimes they correct data. Therefore, out of a number of outliers detected from this study, the ones identified as noise were derived from synthetic minority class samples generated by the SMOTE.

This study aims to identify the noise generated by SMOTE using Local Outlier Factor (LOF), which can be utilized to identify outliers in a more meaningful way, with each object given a degree (Breuniq et al., 2000). Other methods used to detect outliers, are LiCS that classifies the samples using K-Nearest Neighbors (kNN) of each node (Benjelloun et al., 2019) and outlier detection using kNN graphs with k-distance calculation (Asniar and Surendro, 2014). Similar to the LOF, this k-distance calculation assigns an outlier degree score to each object and provides meaningful data because the calculation takes into account local factors in the neighboring environment of each object (Foreman, 2014).

The LOF algorithm, which is an unsupervised data mining technique, was first proposed for outlier detection based on the density and completely free of distribution assumptions (Breuniq et al., 2000). Unlike the traditional outlier detection methods, which consider an outlier as a binary property, the LOF assigns a degree of being an outlier to all data records (Tu et al., 2018).

Algorithm 2 *SMOTE-LOF (x,o,k)*

```
Input: Dataset containing minority class data samples from SMOTE $o$, An object of minority class data sample × in dataset $o$, Number of nearest neighbors $k$
Output: Local Outlier Factor lof$(x,o,k)$
mean$(x)$
stdev$(x)$
normalize$(x)$
dist$(x,y)$
knn $(k)$
k_dist$(x)$
reach_dist$_k(x,o)$
lrd$(x)$
lof$(x,o,k)$ = lof$(x)$
return lof$(x,o,k)$
End of Pseudo-Code.
```

Fig. 3 and SMOTE-LOF Algorithm illustrate the technical steps to detect outliers with the LOF from minor class data samples generated by the SMOTE, as follows:

(1). Datasets processed for the noise identification were in the minority class. This is because what was identified as noise was only data derived from synthetic samples, which are only processed by the minority class.

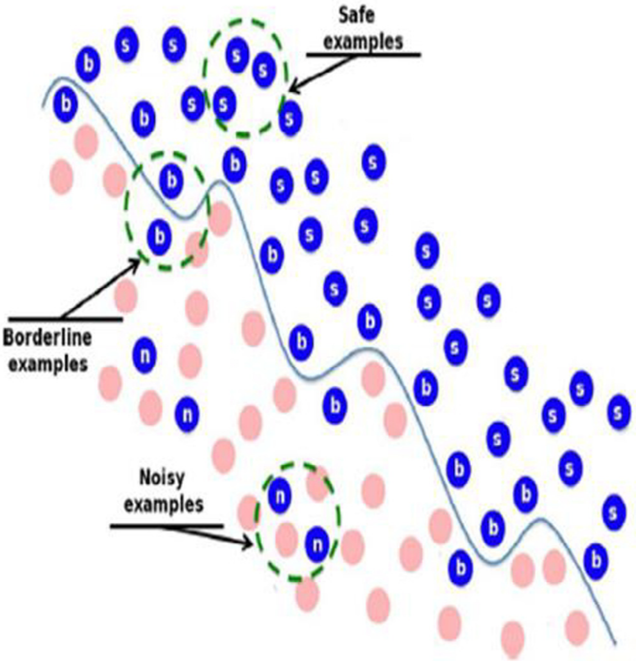

Fig. 2. Noisy Samples (Sáez et al., 2015).

(2). Determining the mean and standard deviation values of each dataset attribute.

$$\bar{X} = \frac{\sum(Xi)}{N} \tag{2}$$

$$S = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2}{n-1}} \tag{3}$$

where,

$\bar{X}$	:	Mean (The mean distribution value of each attribute)
S	:	Standard Deviation (The distribution standard deviation of each attribute)
X_i	:	The attribute value measured from each minority class data sample
$\sum(Xi)$	:	The sum of values for each attribute in the minority class data sample
N	:	The sum of minor class data samples

(3). Standardizing the value of each attribute measured from each minority data sample.

The standard method used to standardize the value of each attribute is by subtracting the mean value of the attribute, which is further divided by the standard deviation of the attribute.

$$Z = \frac{X - \mu}{\sigma} \tag{4}$$

where,

Z	:	Z-score or standard normal value
X	:	The value of each attribute
μ	:	The distribution mean value of each attribute
σ	:	The distribution deviation standard of each attribute

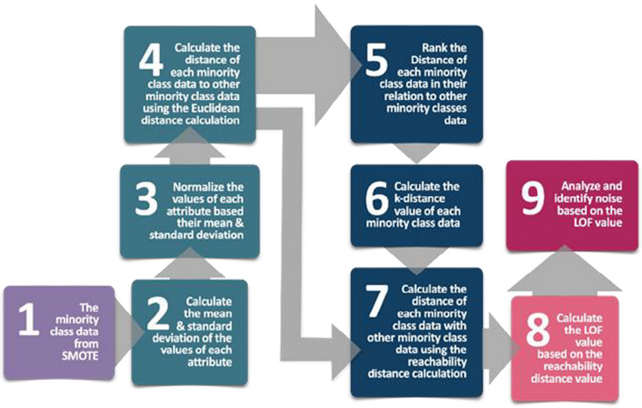
1 The minority class data from SMOTE
2 Calculate the mean & standard deviation of the values of each attribute
3 Normalize the values of each attribute based their mean & standard deviation
4 Calculate the distance of each minority class data to other minority class data using the Euclidean distance calculation
5 Rank the Distance of each minority class data in their relation to other minority classes data
6 Calculate the k-distance value of each minority class data
7 Calculate the distance of each minority class data with other minority class data using the reachability distance calculation
8 Calculate the LOF value based on the reachability distance value
9 Analyze and identify noise based on the LOF value

Fig. 3. Noise Identification with Local Outlier Factor (LOF).

(4). Determining the distance between minority data samples using the Euclidean distance calculation.

The Euclidean distance is the distance between two nodes (minority class data samples), i.e. the square root of the sum of the square of difference of each column value for the two nodes (minority class data samples).

$$dist(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5}$$

where,

$dist(x,y)$	:	The Euclidean distance between object x and y (minority class data samples).
x_i	:	A minority class data sample
y_i	:	Another minority class data sample
n	:	The number of attributes

(5). Ranking the distances between data samples based on the Euclidean distances previously calculated. In this stage, a distance ranking of each minority sample was made against the minority starting from the 0th rank, which is diagonal to the graph.

Then determine the k-nearest neighbors (kNN) of *x:*
kNN returns the set $N_k^{(x)} \subseteq O$ of size k such that:

$$\forall o \in N_k(x), \forall_y \in O, y \notin N_k(x) \Rightarrow dist(x,o) \leq dist(x,y) \tag{6}$$

where,

o	:	Dataset containing minority class data samples from SMOTE
x	:	An object of minority class data sample
k	:	Number of nearest neighbors
dist (x,y)	:	The Euclidean distance between objects *x* and *y*.

(6). Calculating k-distance value

This stage first determined the distance rating for other minority samples nearest to the neighbors with k based on the calculated distance rating. K-distance is the distance from a minority sample to the nearest neighbors. For k = 5, it means that k-distance returns the distance of each minority data sample to the 5th nearest neighbor.

$$k\_dist(x) = \max\{dist(x,o) | o \in N_k(x)\} \tag{7}$$

where,

k_dist(x)	:	The k-distance of object *x* to another in dataset *o*, (ie the nearest neighbor).
dist(x,o)	:	The Euclidean distance between object *x* and another in dataset *o* (i.e. nearest neighbor).

(7). Determining the distance between minority class data examples using the reachability distance calculation.

$$reach\_dist_k(x,o) = Max\{k\_distance(o), dist(x,o)\} \tag{8}$$

where,

$reach\_dist_k(x,o)$	:	The reachability distance between object *x* and another in dataset *o* (nearest neighbor).
$k\_distance(o)$	:	The k-distance of another object in dataset *o* to another in dataset *o* (nearest neighbor).
$dist(x,o)$	:	The Euclidean distance between *x* minority sample and another in dataset *o* (nearest neighbor).

(8). Calculating the LOF value based on the reachability distance calculation.

The LOF value is the mean ratio of mean reachability distance of each minority sample divided by the mean reachability distance of each k-neighbor.

$$lrd(x) = \frac{k}{\sum_{o \in N_k(x)} reach\_dist(x,o)} \tag{9}$$

$$lof(x) = \frac{\sum_{o \in N_k(x)} \frac{lrd(o)}{lrd(x)}}{k} \tag{10}$$

where,

$lof(x)$	:	Local Outlier Factor of object *x* in dataset *o*
$lrd(o)$	:	Local Reachability Density of objects in dataset *o*
$lrd(x)$	:	Local Reachability Density of object *x*

(9). Noise analysis and identification based on the LOF value

Outliers identified as noise were only those derived from the synthetic minority class samples generated by the SMOTE, and not from the original.

The data identified as noise is erased, followed by the recombination of the minority and majority class samples for conducting classification using the machine learning classifier as shown in Fig. 4.

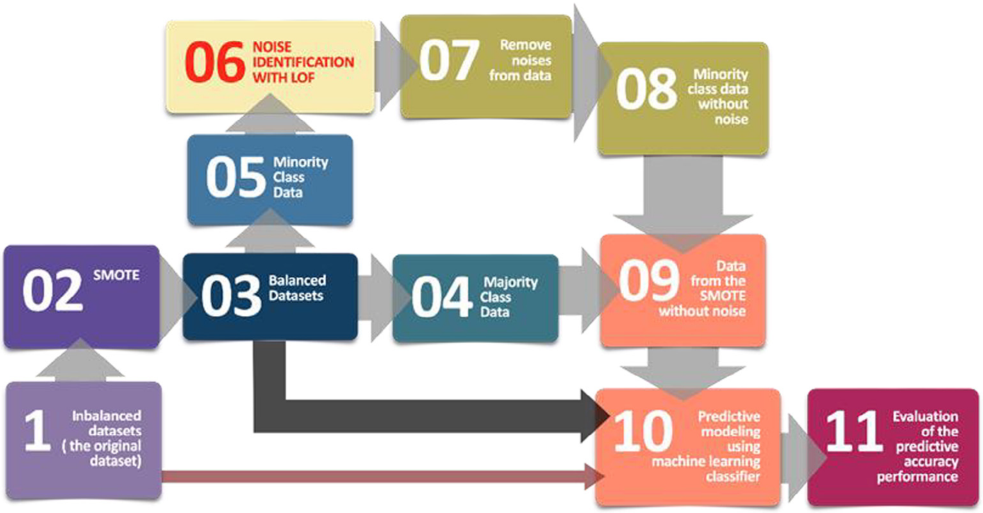
06 NOISE IDENTIFICATION WITH LOF
07 Remove noises from data
08 Minority class data without noise
05 Minority Class Data
02 SMOTE
03 Balanced Datasets
04 Majority Class Data
09 Data from the SMOTE without noise
1 Inbalanced datasets (the original dataset)
10 Predictive modeling using machine learning classifier
11 Evaluation of the predictive accuracy performance

Fig. 4. Experimental Framework.

Table 1
Characteristics of datasets used.

Dataset	#Examples	#Attributes	Minority Class	#Minority Classes	#Majority Classes	IR
Pima	768	8	Positive	268	500	1.87
Haberman	306	3	Died	81	225	2.78
Glass	214	9	Positive	76	138	1.82

4. Experimental framework

This section describes the experiment carried out using the combination of the SMOTE and LOF to overcome noise problem in handling imbalanced data. The experiment used 3 imbalanced datasets, namely Pima[1], Haberman[2], and Glass[3].

The pima dataset is used to determine the medical records for Pima Indians and whether or not each patient have an onset of diabetes within five years. Fields description comprises of number of times pregnant, plasma glucose concentration, 2-hour oral glucose tolerance test, diastolic blood pressure (mm Hg), triceps skin fold thickness (mm), 2-hour serum insulin (mu U/ml), body mass index (weight in kg/(height in m)^2), diabetes pedigree function, age (years) and class variable, where 1 and 0 are used to denote those tested positive and negative for diabetes, respectively.

The Haberman dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago's Billings Hospital on the survival of patients that had undergone surgery for breast cancer. The attributes consists of, patients age at time of operation, year of operation, number of positive axillary nodes detected and survival status (class attribute, 1: the patient survived 5 years or longer 2: the patient died within 5 years).

The Glass dataset is an unbalanced version of the Glass Identification Data Set, where positive and negative samples belong to class 1 and the rest, respectively. The sample identification number, consists of nine input variables that summarize the properties of the glass dataset namely RI (refractive index), Na (Sodium), Mg (Magnesium), Al (Aluminum), Si (Silicon), K (Potassium), Ca (Calcium), Ba (Barium), and Fe (Iron).

The characteristics of each dataset were shown in Table 1. For each dataset, the number of data samples (#Examples), attributes (#Attributes), minority class attributes (Minority Class), minority class attributes (#Minority Classes), majority class attributes (#Majority Classes) and the imbalance ratio between majority and minority classes (IR) are shown.

The study uses the default parameter settings of the WEKA tool (http://www.cs.waikato.ac.nz/ml/weka/) for simulation. The experiment was carried out using C4.5, Naïve Bayes, and SVM as a base classifier with 5-fold cross-validation. Furthermore, for the SMOTE processing, the major and minor data's magnification balance ratio was set to 50:50. The nearest neighbor parameter k is set to 5, which is the most common value used in imbalanced data cases. For identifying noise generated by the SMOTE with LOF, the nearest neighbor parameter k is set to 3 and 5. The experiment was evaluated to determine the accuracy, precision, recall, f-measure, and Area under the Receiver Operating Characteristic Curve (AUC), which are the most often used binary classification measures based on the values of the confusion matrix (Sokolova and Lapalme, 2009).

The experiment steps are shown in Fig. 4. The approach taken included (i) without SMOTE preprocessing indicated by red lines, (ii) an approach by applying the SMOTE for handling imbalanced data (Chawla et al., 2002) shown by black lines, and (iii) a proposed approach by applying the combination of SMOTE and LOF to identify noise in handling imbalanced data shown by grey lines. Each approach is separately compared.

All three approaches started with an unbalanced dataset with the performance of modeling in the absence of SMOTE preprocessing using machine learning classifier as shown by red lines. For the other approaches, imbalanced datasets needed to be first balanced using the SMOTE as shown by black lines, while for the SMOTE approach, prediction modelling was performed after balance data was obtained using machine learning classifier. While for the

[1] https://www.kaggle.com/kumargh/pimaindiansdiabetescsv.
[2] https://www.kaggle.com/saguneshgrover/haberman.
[3] https://sci2s.ugr.es/keel/imbalanced.php#sub2A

proposed approach, after the SMOTE process, noise identification on balanced datasets was performed using the Local Outlier Factor (LOF) as shown by grey lines. Furthermore, balanced datasets are separated into the majority and minority classes, with the minority processed for noise identification. This is because what was identified as noise was only data samples derived from synthetic data, while those from the SMOTE process, were possessed by the minority class. All minority class examples generated by the SMOTE are processed during the noise identification using LOF with the detailed steps shown in Fig. 3. Next, data samples identified as noise are erased, while the minority class data generated by the SMOTE whose noises have been erased are recombined with the majority data to obtain SMOTE-produced datasets without noise. This is following by the use of predictive modeling as a machine learning classifier as shown in Fig. 4. The experiment evaluation used the accuracy, precision, recall, f-measure and AUC from the classification modeling built with C4.5, Naïve Bayes, and SVM with 5-fold cross-validation.

5. Results and evaluation

This section discusses the results obtained from the noise identification process using the LOF as previously shown in Fig. 3. It also evaluates the results of the experimental framework as shown in Fig. 4.

5.1. Visualization of Noise Identification using Local Outlier Factor (LOF)

After carrying out detailed noise identification steps as shown in Fig. 3, the noise was identified from the minority class data with the samples generated by the SMOTE as shown in Fig. 5.

For experiments using Pima dataset with k = 3, the outlier identification results obtained were the 6th, 63rd, 244th, 274th, 307th, 451st, and 483rd minority samples. Since the 1st to 268th are the original minority samples, then a number of identified outliers that were between them were not identified as noise. Therefore, those that are not between this interval with a LOF value of 1.974 are identified as noise. Examples are the 307th, 451st, and 483rd minority samples with LOF values of 2.123, 1.881 and 2.195, respectively.

For experiments using Pima dataset with k = 5, the outlier identification results obtained were the 6th, 63rd, 207th, 244th, 307th, 330th, 371st, 451st, and 483rd minority samples. Since the 1st to the 268th are the original minority examples, then a number of identified outliers that were between them were not identified as noise. Therefore, the data sample identified as noise are those outside this interval, namely the 307th, 330th, 371st, 451st, and 483rd, with LOF values of 1.941, 1.513, 1.522, 1.673 and 1.817.

For experiments using Haberman dataset with k = 3, the outlier identification results obtained were the 6rd, 93rd, and 105th minority samples. Since the 1st to the 81st were the original minority samples, then outliers that were between them were not identified as noise. Therefore, those identified as noise were the samples outside this interval, such as the 93rd, and 105th with LOF values of 5.933 and 29.245.

For experiments using Haberman dataset with k = 5, the outlier identification results obtained were the 2nd, 3rd, 10th, 81st, 199th and 224th minority samples. Since the 1st to the 81st were the original minority samples, then the outliers between this interval were not identified as noise. Therefore, the data samples identified as noise were the outliers outside this interval, namely the 199th, and 224th with LOF values of 1.679 and 1.675.

For experiments using Glass dataset with k = 3, the outlier identification results obtained were the 19th, 71st, 72nd, 111st and 117th minority samples. Since the 1st to the 76th were the original minority samples, then the outliers outside this interval were not identified as noise. Therefore, the data examples identified as noise are the 111st and 117th with LOF values of 1.781 and 1.876.

For experiments using Glass dataset with k = 5, the outlier identification results obtained were the 19th, 27th, 49th, 71st, 113th, 117th and 133th minority examples. Since the 1st to the 76th were the original minority samples, then the outliers between this interval was not identified as noise. Therefore, the data samples identified as noise were those outside this interval, namely the 113th, 117th and 133rd with LOF values of 1.441, 1.668 and 1.438.

This is followed by erasing all the data samples identified as noise, followed by the process of recombining the major class data examples used for classification using the machine learning classifier as shown in Fig. 4. Furthermore, the predictive accuracy of the experimental results is evaluated.

5.2. Experimental result evaluation

Each dataset was evaluated by comparing the approaches (i) without SMOTE preprocessing (None), (ii) using SMOTE for handling imbalanced data (Chawla et al., 2002), and (iii) the proposed approach by combining SMOTE and LOF for noise identification in handling imbalanced data, called the SMOTE-LOF approach. The evaluation utilized accuracy, precision, recall, f-measure and Area under the Receiver Operating Characteristic Curve (AUC) from the classifier modelling built with C4.5, Naïve bayes and SVM with 5-fold cross-validation each are shown in Tables 2, 3 and 4.

Tables 2 presents the approach without SMOTE processing (None), the SMOTE approach, and the SMOTE-LOF approach using the Pima dataset. From this table, the results show that the SMOTE and the SMOTE-LOF have better precision (3–25%), recall (9–44%), f-measure (10–31%), and AUC (0.2–9%) than the approach without SMOTE processing (None) of all classifiers. Obtained by C4.5, the SMOTE and the SMOTE-LOF also have 5.6% better accuracy than the approach without SMOTE processing (None). However, obtained by Naïve Bayes and SVM techniques, the approach without SMOTE processing (None) has 3–5% better accuracy than the SMOTE and SMOTE-LOF.

Table 3 compares three approaches using Haberman dataset. The results show that the SMOTE and the SMOTE-LOF have better precision (35–74%), recall (68–2750%), f-measure (29–1892%), and AUC (3–32%) than the approaches without SMOTE processing (None), except those obtained by SVM, the approach without SMOTE processing (None) has better precision (26–29%) than the SMOTE and the SMOTE-LOF. The approach without SMOTE processing (None) also has 1–23% better accuracy than the SMOTE and the SMOTE-LOF of all classifiers, except those obtained by C4.5, the SMOTE-LOF (k = 3) has 1% better accuracy than the approach without SMOTE processing (None).

Tables 4 compares three approaches using Glass dataset. The results show that the SMOTE and the SMOTE-LOF have better accuracy (3–18%), precision (23–37%), recall (0.1–43%), f-measure (21–34%), and AUC (2–22%) than the approaches without SMOTE processing (None), except those obtained by SVM, the approach without SMOTE processing (None) has 6–15% better accuracy than the SMOTE and SMOTE-LOF. It also obtained Naïve Bayes, the approach without SMOTE processing (None) has 0.1% better recall than the SMOTE-LOF (k = 5). Obtained by C4.5, the approach without SMOTE processing (None) has 1.22% better accuracy than the SMOTE and the SMOTE-LOF (k = 5).

Overall all three data sets were successfully classified in the three approaches, with table 5 indicating their average performance. Compared to the approach without SMOTE processing (None), the result indicates that SMOTE and SMOTE-LOF have better precision, recall, f-measure, and AUC results for all datasets. The

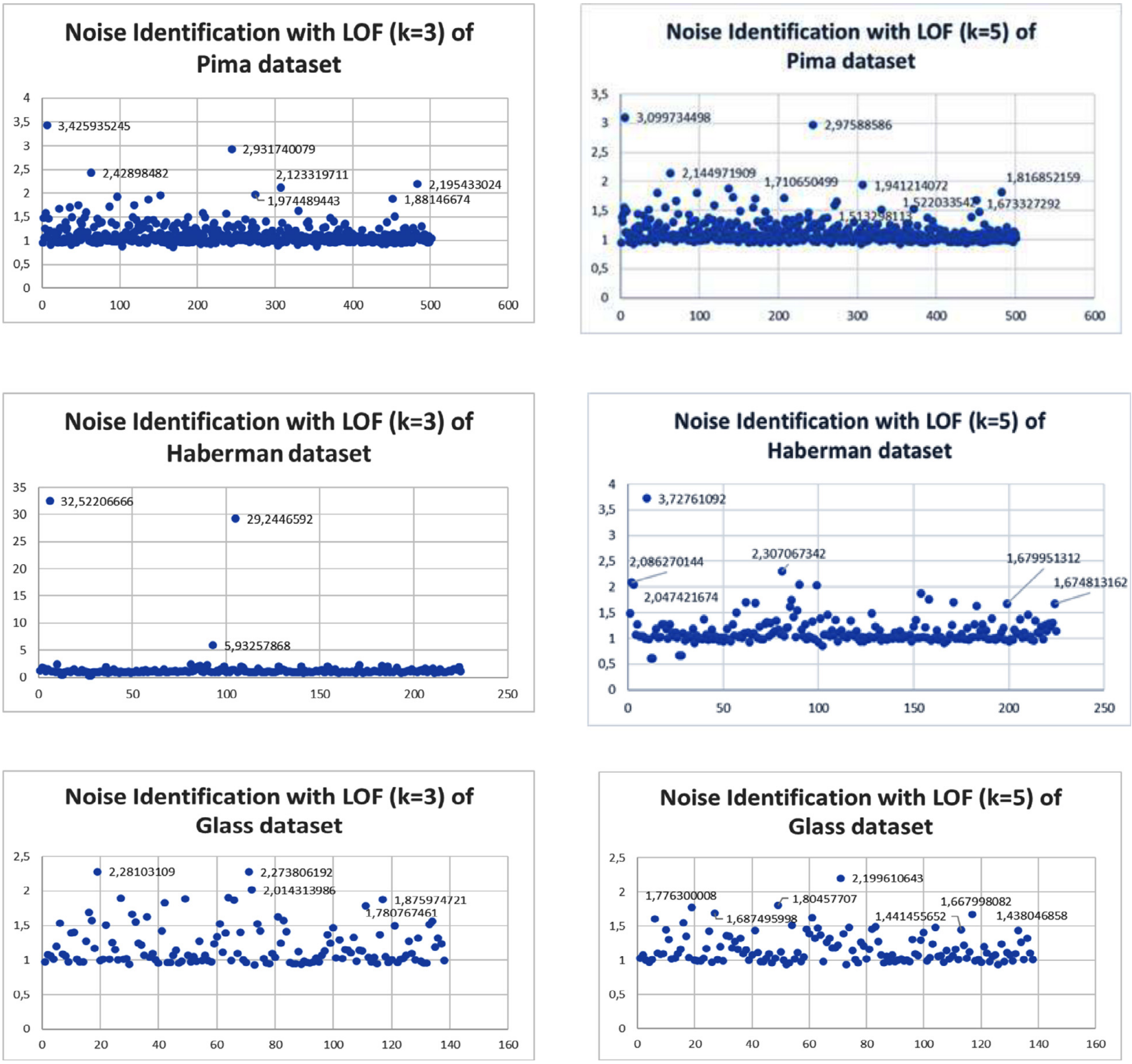

Fig. 5. Visualization of Noise Identification with Local Outlier Factor (LOF).

Table 2
Results (%) of Pima Dataset.

Classifier		None	SMOTE	SMOTE-LOF	
				k = 3	k = 5
C4.5	Accuracy	71.09	**73.03**	**75.13**	**75.10**
	Precision	58.30	**72.90**	**70.40**	**71.60**
	Recall	60.10	**73.50**	**86.30**	**82.90**
	F-Measure	59.20	**73.20**	**77.60**	**76.80**
	AUC	72.62	**76.53**	**77.03**	**79.26**
Naïve Bayes	Accuracy	**76.04**	73.53	72.42	72.69
	Precision	67.40	**75.70**	**75.30**	**75.90**
	Recall	60.80	**69.50**	**66.40**	**66.10**
	F-Measure	63.90	**72.40**	**70.60**	**70.70**
	AUC	81.69	**81.87**	**81.23**	**81.36**
SVM	Accuracy	**77.21**	74.83	74.22	74.10
	Precision	73.40	**76.10**	**75.80**	**75.40**
	Recall	54.50	**72.50**	**71.00**	**71.20**
	F-Measure	62.50	**74.20**	**73.30**	**73.20**
	AUC	71.94	**74.83**	**74.24**	**74.08**

The best case for each classifier is highlighted in bold.

Table 3
Results (%) of Haberman Dataset.

Classifier		None	SMOTE	SMOTE-LOF	
				k = 3	k = 5
C4.5	Accuracy	70.92	68.89	**71.88**	70.54
	Precision	42.30	**69.80**	**73.70**	**72.60**
	Recall	27.20	**66.70**	**67.70**	**65.50**
	F-Measure	33.10	**68.20**	**70.60**	**68.90**
	AUC	56.44	**74.61**	**73.28**	**73.05**
Naïve Bayes	Accuracy	**74.51**	62.22	61.16	62.05
	Precision	55.20	**76.70**	**74.70**	**77.30**
	Recall	19.80	**35.10**	**33.20**	**33.60**
	F-Measure	29.10	**48.20**	**46.00**	**46.90**
	AUC	63.98	**65.97**	**65.62**	**67.13**
SVM	Accuracy	**73.86**	62.67	61.61	60.04
	Precision	**100.00**	79.40	78.00	77.50
	Recall	1.20	**34.20**	**31.80**	**27.80**
	F-Measure	2.40	**47.80**	**45.20**	**40.90**
	AUC	50.62	**62.67**	**61.47**	**59.90**

The best case for each classifier is highlighted in bold.

Table 4
Results (%) of Glass Dataset.

Classifier		None	SMOTE	SMOTE-LOF	
				k = 3	k = 5
C4.5	Accuracy	68.22	70.29	**71.53**	67.40
	Precision	55.60	**70.00**	**69.90**	**68.50**
	Recall	52.60	**71.00**	**75.00**	**63.00**
	F-Measure	54.10	**70.50**	**72.30**	**65.60**
	AUC	67.93	**74.09**	**70.95**	**69.34**
Naïve Bayes	Accuracy	50.93	59.78	**62.04**	60.07
	Precision	41.50	**55.80**	**57.00**	**55.80**
	Recall	93.40	**93.50**	**95.60**	93.30
	F-Measure	57.50	**69.90**	**71.40**	**69.80**
	AUC	63.05	**65.92**	**66.07**	**64.41**
SVM	Accuracy	**64.49**	60.14	60.95	56.04
	Precision	0.00	**57.50**	**58.80**	**54.30**
	Recall	0.00	**77.50**	**71.30**	**69.60**
	F-Measure	0.00	**66.00**	**64.50**	**61.00**
	AUC	50.00	**60.14**	**61.02**	**56.19**

The best case for each classifier is highlighted in bold.

Table 5
Average Results of Performance.

Dataset		None	SMOTE	SMOTE-LOF	
				k = 3	k = 5
Pima	Accuracy	**74.78**	73.80	73.92	**73.96**
	Precision	66.37	**74.90**	73.83	**74.30**
	Recall	58.47	**71.83**	**74.57**	73.40
	F-Measure	61.87	**73.27**	**73.83**	73.57
	AUC	75.42	**77.74**	77.50	**78.23**
Haberman	Accuracy	**73.10**	64.59	**64.88**	64.21
	Precision	65.83	**75.30**	75.47	**75.80**
	Recall	16.07	**45.33**	**44.23**	42.30
	F-Measure	21.53	**54.73**	**53.93**	52.23
	AUC	57.01	**67.75**	**66.79**	66.69
Glass	Accuracy	61.21	**63.40**	**64.84**	61.17
	Precision	32.37	**61.10**	**61.90**	59.53
	Recall	48.67	**80.67**	**80.63**	75.30
	F-Measure	37.20	**68.80**	**69.40**	65.47
	AUC	60.33	**66.72**	**66.01**	63.31

The best case for each dataset is highlighted in bold.

Table 6
Comparative Analysis of Accuracy Results (%) obtained by C4.5.

Dataset	#Examples	IR	SMOTE Accuracy	SMOTE-LOF Accuracy	
				k = 3	k = 5
pima	768	1.87	73.03	**75.13**	75.10
haberman	306	2.78	68.89	**71.88**	70.54
glass	218	1.82	70.29	**71.53**	67.40

The best case for each dataset is highlighted in bold.

Table 7
Comparative Analysis of F-Measure Results (%) obtained by C4.5.

Dataset	#Examples	IR	SMOTE F-Measure	SMOTE-LOF F-Measure	
				k = 3	k = 5
Pima	768	1.87	73.20	**77.60**	76.80
haberman	306	2.78	68.20	**70.60**	68.90
glass	218	1.82	70.50	**72.30**	65.60

The best case for each dataset is highlighted in bold.

Table 8
Comparative Analysis of AUC Results (%) obtained by C4.5.

Dataset	#Examples	IR	SMOTE AUC	SMOTE-LOF AUC	
				k = 3	k = 5
pima	768	1.87	76.53	77.03	**79.26**
haberman	306	2.78	**74.61**	73.28	73.05
glass	218	1.82	**74.09**	70.95	69.34

The best case for each dataset is highlighted in bold.

accuracy values of the approach without SMOTE preprocessing (None) are better than the SMOTE and the SMOTE-LOF in Pima and Haberman data sets. However, accuracy is not an effective indicator of performance evaluation, specifically in the case of imbalanced data sets. Furthermore, accuracy does not distinguish between the numbers of correctly classified examples of different classes. This is discussed in a number of literature studies (Fernándezet al., 2018a; He and Garcia, 2009; Malhotra and Khanna, 2017).

Table 5 also shows that SMOTE-LOF with a parameter k values of 3 has a better average recall and f-measure than the approach with a parameter k values of 5 for all datasets. SMOTE-LOF with a parameter k value of 3 also has better average accuracy and AUC in two data sets, namely Haberman and Glass. However, SMOTE-LOF with a parameter k value of 5 has better average precision in two data sets namely Pima and Habermen. SMOTE-LOF with a parameter k value of 5 also has better average accuracy in Pima dataset.

6. Analysis and discussion

After carrying out experimental works for all selected data sets, the results show that the proposed SMOTE-LOF approach can be implemented for imbalanced data classification. Based on the experimental result previously evaluated, C4.5 has a better average performance than Naïve Bayes and SVM. Furthermore, it was also used in many works related to imbalanced classification in particular those concerning SMOTE (Sáez et al., 2015; Stefanowski and Wilk, 2008; Su and Hsiao, 2007). Therefore, this analysis and discussion are focused on comparing the accuracy, f-measure, and AUC results of the SMOTE and the proposed approach (SMOTE-LOF) obtained by C4.5 as shown in tables 6, 7, and 8.

Tables 6, 7, and 8 compare the accuracy, f-measure, and AUC results obtained by C4.5 in Pima, Haberman, and Glass dataset. Tables 6 and 7 show that the SMOTE-LOF has 2–4% better accuracy and 1–6% better f-measure than SMOTE, except in the Glass dataset, the SMOTE has 4% better accuracy and 7% better f-measure the SMOTE-LOF with a parameter k value of 5. However, table 8 shows that the SMOTE has 2–7% better AUC than the SMOTE-LOF, except in the Pima dataset, the SMOTE-LOF with parameter k values of 3 and 5 have 0.6% and 3.6% better AUC than SMOTE.

Furthermore, these tables show that the SMOTE-LOF has better accuracy and f-measure in all datasets used. In those with a large number of data examples and a smaller imbalance ratio like the Pima dataset, the SMOTE-LOF approach produces a better AUC than the SMOTE. However, in a dataset with a smaller number of data examples like Haberman and Glass, the SMOTE's AUC result is more capable of handling imbalanced data. This is undoubtedly a challenge for further research to test the proposed method using another dataset with different characteristics combinations between the number of data examples and the imbalance ratio.

7. Conclusion

In conclusion, this study proposed the SMOTE-LOF approach to improve the SMOTE technique by adding Local Outlier Factor (LOF), which is used to identify noise from synthetic minority data generated by the SMOTE. The experiment was carried out using imbalanced datasets while comparing the results of the accuracy of SMOTE-LOF and SMOTE. The results showed that the SMOTE-LOF has better accuracy and f-measure in all datasets used. In a dataset with a large number of data examples and a smaller imbalance ratio, The SMOTE-LOF also produced a better AUC than the SMOTE. However, in a dataset with a smaller number of data examples, the AUC produced by the SMOTE is better at handling imbalanced data. Therefore, future research needs to be carried out using different datasets with varying combinations of characteristics of the number of data samples and the imbalance ratio.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Asniar, Surendro, K., 2014. Using data science for detecting outliers with k Nearest Neighbors graph. In: Proceedings - 2014 International Conference on ICT for Smart Society: "Smart System Platform Development for City and Society, GoeSmart 2014. ICISS, pp. 300–304. https://doi.org/10.1109/ICTSS.2014.7013191.

Barandela, R., Sánchez, J., García, V., Rangel, E., 2003. Strategies for learning in class imbalance problems. Pattern Recogn. 36, 849–851. https://doi.org/10.1016/S0031-3203(02)00257-1.

Barua, S., Islam, M.M., Yao, X., Murase, K., 2014. MWMOTE – Majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans. Knowl. Data Eng. 26, 405–425. https://doi.org/10.1109/TKDE.2012.232.

Batista, G.E.A.P.A., Prati, R.C., Monard, M.C., 2004. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6, 20–29. https://doi.org/10.1145/1007730.1007735.

Benjelloun, F.Z., Oussous, A., Bennani, A., Belfkih, S., Ait Lahcen, A., 2019. Improving outliers detection in data streams using LiCS and voting. J. King Saud Univ. – Comput. Inform. Sci. https://doi.org/10.1016/j.jksuci.2019.08.003.

Branco, P., Torgo, L., Ribeiro, R.P., 2016. A survey of predictive modeling on imbalanced domains. ACM Comput. Surv. 49, 1–50. https://doi.org/10.1145/2907070.

Breuniq, M.M., Kriegel, H.P., Ng, R.T., Sander, J., 2000. LOF: Identifying density-based local outliers. SIGMOD Record (ACM Special Interest Group on Management of Data) 29, 93–104. https://doi.org/10.1145/335191.335388.

Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2009. Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 475–482. https://doi.org/10.1007/978-3-642-01307-2_43

Chawla, N.V., 2009. Data Mining for Imbalanced Datasets: An Overview, in: Data Mining and Knowledge Discovery Handbook. Springer US, Boston, MA, pp. 875–886. https://doi.org/10.1007/978-0-387-09823-4_45

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357. https://doi.org/10.1613/jair.953.

Chawla, N.V., Japkowicz, N., Kotcz, A., 2004. Editorial: special issue on learning from imbalanced data sets. ACM SIGKDD Explorations Newsletter 6, 1–6. https://doi.org/10.1145/1007730.1007733.

Chen, H., Li, T., Fan, X., Luo, C., 2019. Feature selection for imbalanced data based on neighborhood rough sets. Inf. Sci. 483, 1–20. https://doi.org/10.1016/j.ins.2019.01.041.

Durán-rosal, A.M., Gutiérrez, P.A., Salcedo-sanz, S., 2018. A statistically-driven Coral Reef Optimization algorithm for optimal size reduction of time series. Appl. Soft Comput. J. 63, 139–153. https://doi.org/10.1016/j.asoc.2017.11.037.

Fahrudin, T., Lianto Buliali, J., Fatichah, C., 2019. Enhancing the performance of SMOTE algorithm by using attribute weighting scheme and new selective sampling method for imbalance data set. Int. J. Innov. Comput., Inform. Control 15, 423–444 https://doi.org/10.24507/ijicic.15.02.423.

Fernández, A., del Jesus, M.J., Herrera, F., 2010. On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced datasets. Inf. Sci. 180, 1268–1291. https://doi.org/10.1016/j.ins.2009.12.014.

Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F., 2018a. Learning from Imbalanced Data Sets, Learning from Imbalanced Data Sets. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-98074-4

Fernández, A., García, S., Herrera, F., Chawla, N.V., 2018b. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J. Artif. Intell. Res. 61, 863–905. https://doi.org/10.1613/jair.1.11192.

Foreman, J.W., 2014. Data Smart: Using Data Science to Transform Information into Insight. John Wiley & Sons Inc, Indianapolis, IN, p. 46256.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F., 2012. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst., Man, Cybern. Part C (Applications and Reviews) 42, 463–484. https://doi.org/10.1109/TSMCC.2011.2161285.

Gutiérrez, P.D.D., Lastra, M., Benítez, J.M.M., Herrera, F., 2017. SMOTE-GPU: Big Data preprocessing on commodity hardware for imbalanced classification. Progr. Artif. Intell. 6, 347–354. https://doi.org/10.1007/s13748-017-0128-2.

He, H., Garcia, E.A., 2009. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21, 1263–1284. https://doi.org/10.1109/TKDE.2008.239.

Krawczyk, B., 2016. Learning from imbalanced data: open challenges and future directions. Progr. Artif. Intell. 5, 221–232. https://doi.org/10.1007/s13748-016-0094-0.

MacIejewski, T., Stefanowski, J., 2011. Local neighbourhood extension of SMOTE for mining imbalanced data. IEEE SSCI 2011: Symposium Series on Computational Intelligence – CIDM 2011: 2011 IEEE Symposium on Computational Intelligence and Data Mining 104–111. https://doi.org/10.1109/CIDM.2011.5949434

Malhotra, R., Khanna, M., 2017. An empirical study for software change prediction using imbalanced data. Empir. Software Eng. 22, 2806–2851. https://doi.org/10.1007/s10664-016-9488-7.

Mohamad, M., Selamat, A., Subroto, I.M., Krejcar, O., 2019. Improving the classification performance on imbalanced data sets via new hybrid parameterisation model. J. King Saud Univ. – Comput. Inform. Sci. https://doi.org/10.1016/j.jksuci.2019.04.009.

Napierała, K., Stefanowski, J., Wilk, S., 2010. Learning from imbalanced data in presence of noisy and borderline examples, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 158–167. https://doi.org/10.1007/978-3-642-13529-3_18

Nnamoko, N., Korkontzelos, I., 2020. Efficient treatment of outliers and class imbalance for diabetes prediction. Artif. Intell. Med. 104. https://doi.org/10.1016/j.artmed.2020.101815.

Pak, C., Wang, T.T., Su, X.H., 2018. An empirical study on software defect prediction using over-sampling by SMOTE. Int. J. Software Eng. Knowl. Eng. 28, 811–830. https://doi.org/10.1142/S0218194018500237.

Ramentol, E., Caballero, Y., Bello, R., Herrera, F., 2012. SMOTE-RSB *: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. Knowl. Inf. Syst. 33, 245–265. https://doi.org/10.1007/s10115-011-0465-6.

Sáez, J.A.A., Luengo, J., Stefanowski, J., Herrera, F., 2015. SMOTE-IPF: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Inf. Sci. 291, 184–203. https://doi.org/10.1016/j.ins.2014.08.051.

Salgado, C.M., Azevedo, C., Proença, H., Vieira, S.M., 2016. Noise versus outliers. Secondary Analysis of Electronic Health Records. https://doi.org/10.1007/978-3-319-43742-2_14.

Sanz, J.A.A., Bernardo, D., Herrera, F., Bustince, H., Hagras, H., 2015. A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data. IEEE Trans. Fuzzy Syst. 23, 973–990. https://doi.org/10.1109/TFUZZ.2014.2336263.

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Inf. Process. Manage. 45, 427–437. https://doi.org/10.1016/j.ipm.2009.03.002.

Stefanowski, J., Wilk, S., 2008. Selective pre-processing of imbalanced data for improving classification performance, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 283–292. https://doi.org/10.1007/978-3-540-85836-2_27

Su, C.T., Hsiao, Y.H., 2007. An evaluation of the robustness of MTS for imbalanced data. IEEE Trans. Knowl. Data Eng. 19, 1321–1332. https://doi.org/10.1109/TKDE.2007.190623.

Tu, B., Zhou, C., Kuang, W., Guo, L., Ou, X., 2018. Hyperspectral imagery noisy label detection by spectral angle local outlier factor. IEEE Geosci. Remote Sens. Lett. 15, 1417–1421. https://doi.org/10.1109/LGRS.2018.2842792.

Verbiest, N., Ramentol, E., Cornelis, C., Herrera, F., 2014. Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection. Appl. Soft Comput. J. 22, 511–517. https://doi.org/10.1016/j.asoc.2014.05.023.

Wasikowski, M., Chen, X.W., 2010. Combating the small sample class imbalance problem using feature selection. IEEE Trans. Knowl. Data Eng. 22, 1388–1400. https://doi.org/10.1109/TKDE.2009.187.

You, L., Peng, Q., Xiong, Z., He, D., Qiu, M., Zhang, X., 2020. Integrating aspect analysis and local outlier factor for intelligent review spam detection. Future Gener. Comput. Syst. 102, 163–172. https://doi.org/10.1016/j.future.2019.07.044.