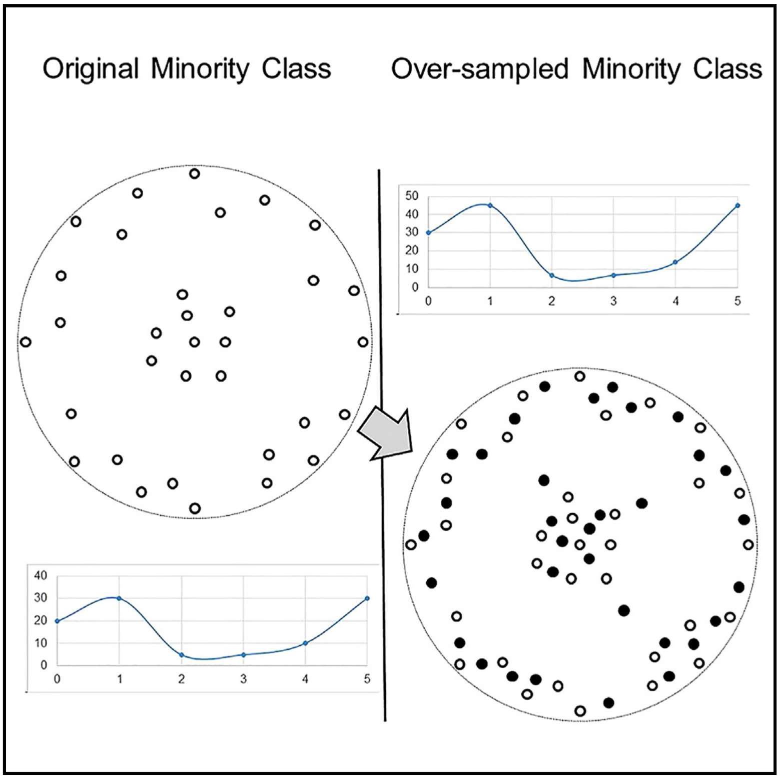
Patterns

Article

FLEX-SMOTE: Synthetic over-sampling technique that flexibly adjusts to different minority class distributions

Graphical abstract



Highlights

- FLEX-SMOTE is an over-sampling technique for handling the class imbalance problem
- The method generates synthetic instances by applying a density-based concept
- The method can be used for various distributions of minority classes
- After over-sampling, the prediction rate on a minority class is improved

Authors

Chumphol Bunkhumpornpat, Ekkarat Boonchieng, Varin Chouvatut, David Lipsky

Correspondence

ekkarat.boonchieng@cmu.ac.th

In brief

In this work, the authors present an over-sampling technique for upsizing a minority class before training on an imbalanced dataset. Synthetic instances are generated based on the density of a minority class region. The method, called FLEX-SMOTE, can be applied to minority classes with any distribution and still provide satisfactory performance.

Bunkhumpornpat et al., 2024, Patterns *5*, 101073
November 8, 2024 © 2024 The Author(s). Published by Elsevier Inc.
https://doi.org/10.1016/j.patter.2024.101073

CellPress
OPEN ACCESS

Article

FLEX-SMOTE: Synthetic over-sampling technique that flexibly adjusts to different minority class distributions

Chumphol Bunkhumpornpat,[1,2] Ekkarat Boonchieng,[1,2,3,*] Varin Chouvatut,[1,2] and David Lipsky[2]
[1]Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai 50200, Thailand
[2]Center of Excellence in Community Health Informatics, Chiang Mai University, Chiang Mai 50200, Thailand
[3]Lead contact
*Correspondence: ekkarat.boonchieng@cmu.ac.th

https://doi.org/10.1016/j.patter.2024.101073

THE BIGGER PICTURE Machine learning methods often encounter imbalanced classification problems, particularly when dealing with binary classification. This occurs when the distribution of classes in the training dataset is uneven, which can lead to bias in the trained model. Fraud detection, claim prediction, default prediction, spam filtering, disease screening, churn prediction, anomaly detection, and outlier identification tasks are a few examples of imbalanced classification issues. To enhance the performance and ensure the accuracy of a model, it is crucial to address the issue of class imbalance. Predictive modeling is complicated by imbalanced datasets, but this is to be expected, as the real world consists of biased cases. By preventing the dataset from becoming biased toward one class, balancing it makes it easier to train a model. To put it another way, the model will not continue to favor the majority class solely based on having more data.

SUMMARY

Class imbalance is a challenge that affects the prediction rate on a minority class. To remedy this problem, various SMOTEs (synthetic minority over-sampling techniques) have been designed to populate synthetic minority instances. Some SMOTEs operate on the border of a minority class, while others concentrate on the class core. Unfortunately, it is difficult to put the right SMOTE to the right dataset because distributions of classes are varied and might not be obvious. This paper proposes a new technique, called FLEX-SMOTE, that is flexible enough to be used with all sorts of datasets. The key idea is that an over-sampled region is selected based on the characteristics of minority classes. This approach is based on a density function that is used to describe the distributions of minority classes. Herein, we have included experimental results showing that FLEX-SMOTE can significantly improve the predictive performance of a minority class.

INTRODUCTION

Class imbalance[1–5] has been identified as one of the ten most difficult problems in data mining research.[6,7] An imbalanced dataset has one or more majority classes that are significantly larger than the minority class. Classifiers can struggle to recognize minority instances in such cases. A number of empirical studies[8–10] have provided evidence of this weak performance. In imbalanced applications, such as network intrusion detection, incorrectly classified minority instances can have serious consequences.[11] An intrusion is defined as unauthorized access to a computer network that circumvents security measures. This type of attack is not only uncommon, but it is also critically important to detect. Misclassified intrusion can have serious consequences in the real world.

In unbalanced domains, over-sampling is a well-known and widely used approach. Before feeding a dataset to a classification algorithm, this preprocessing step enlarges a minority class to generate a more balanced dataset. Random over-sampling is considered the simplest procedure to accomplish this task.[12] In this method, minority cases are randomly duplicated; however, duplicate instances can lack new information, resulting in overfitting.[13] Complex approaches such as SMOTE (synthetic minority over-sampling technique)[14] populate a minority class with

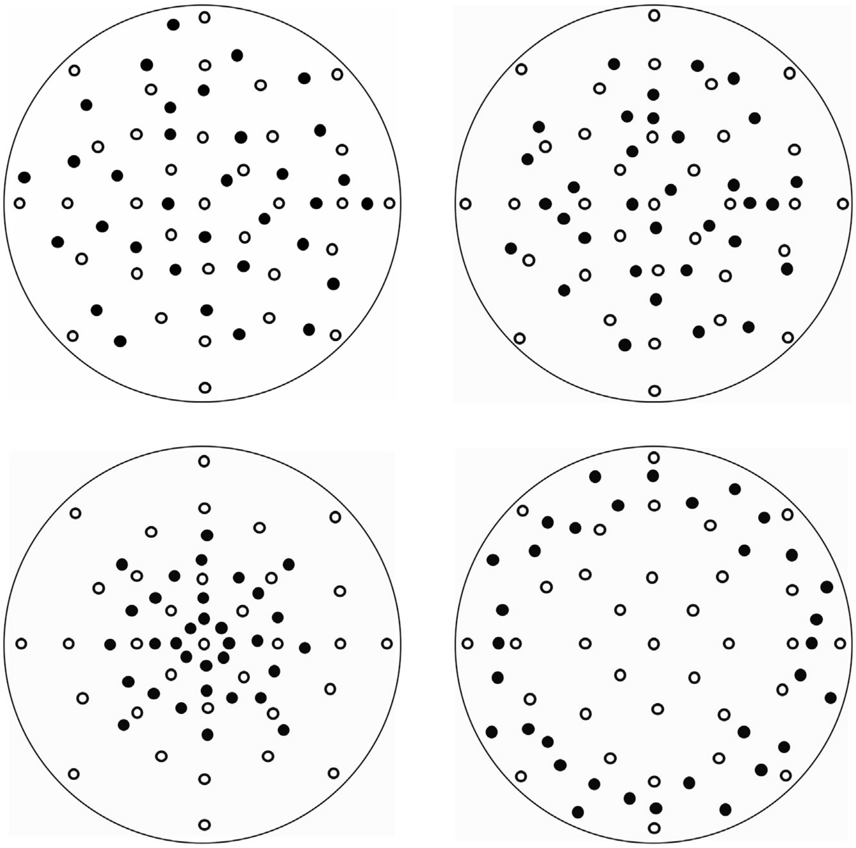


Figure 1. A summary of the various SMOTE approaches

White and black dots represent training and synthetic minority class examples, respectively. SMOTE (top left) blindly generalizes throughout a minority class. Safe-Level-SMOTE (top right) locates synthetic instances closer to a minority class than a majority class. DBSMOTE (bottom left) concentrates on the core of a minority cluster. Borderline-SMOTE (bottom right) operates only on borderline instances in the over-lapping region.

synthetic instances. The decision boundary of a minority class is then broadened and better recognized by classifiers. Due to the widespread popularity of SMOTE, other variants have been produced. While the original version of SMOTE operates throughout a minority cluster, versions such as Borderline-SMOTE[15] and Safe-Level-SMOTE[16] target the class's edge and center, respectively. In this paper, we define the term "SMOTE" as any technique that synthesizes instances into a minority class. Accordingly, we can use the term "SMOTEs" as the plural form.

A safe level (*SL*) graph[17,18] is a visualization tool for the distribution of minority instances. Based on the structure of this graph, unbalanced datasets can be loosely categorized into three categories, in which minority occurrences are dense within the core of the minority class, congested around the class's boundary, or evenly distributed throughout. *SL* graphs are useful for determining the most suitable SMOTE version for a specific dataset. For instance, Borderline-SMOTE is recommended if instances of the minority are concentrated along the border.

Unfortunately, no single SMOTE can deal with all distributions of imbalanced datasets. Some techniques are designed to concentrate on the border of minority class clusters, while some operate on the core of the clusters. However, the original method equally prioritizes all regions of minority classes. The main challenge involves how to put the right SMOTE to the right dataset without going through a trial-and-error process.

The following are the primary contributions of this paper. Firstly, the outcome of this study can provide a flexible over-sampling technique for incorporating minority instances into imbalanced datasets. The second contribution is related to the provision of the theorems and proofs of an algorithm that can accurately demonstrate its properties. Lastly, our research establishes an experiment to test whether our hypothesis of adopting unique over-sampling approaches for class distribution would be appropriate.

This work aims to boost the F-measure[19] and area under the curve (AUC)[20] of various types of classifiers to more accurately forecast minority cases in imbalanced datasets. This research has focused on the over-sampling schema because we intend to investigate the behavior of classifiers after synthesizing minority cases from various minority areas. This raises the primary research question of which area should be over-sampled. Since under-sampling[21] is used to reduce instances, it will not be considered.

The remainder of this work is divided into five sections: related work examines known approaches to class imbalance learning. FLEX-SMOTE describes and theoretically examines our novel technique. Experiment presents an experiment that validates our hypothesis. Conclusion concludes this paper with a summary and recommendations for further research.

RELATED WORK

SMOTEs

N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer introduced the first developed technique, namely SMOTE,[14] for synthesizing new minority instances as opposed to random over-sampling or replicating existing instances. A synthetic instance is generated at a random place along a line segment between two specified instances: a minority instance under consideration and one of its k closest neighbors from the same minority class chosen at random. Since the values of synthetic instances are distinct from those of training instances, the decision region of an enlarged minority class is enlarged and overfit. A shortcoming of SMOTE is that it operates blindly on minority instances without taking proximity to majority instances into account. When two classes are highly intertwined, classifiers can face an overlapping data problem.[22] Several SMOTE successors have been developed in response to this circumstance.

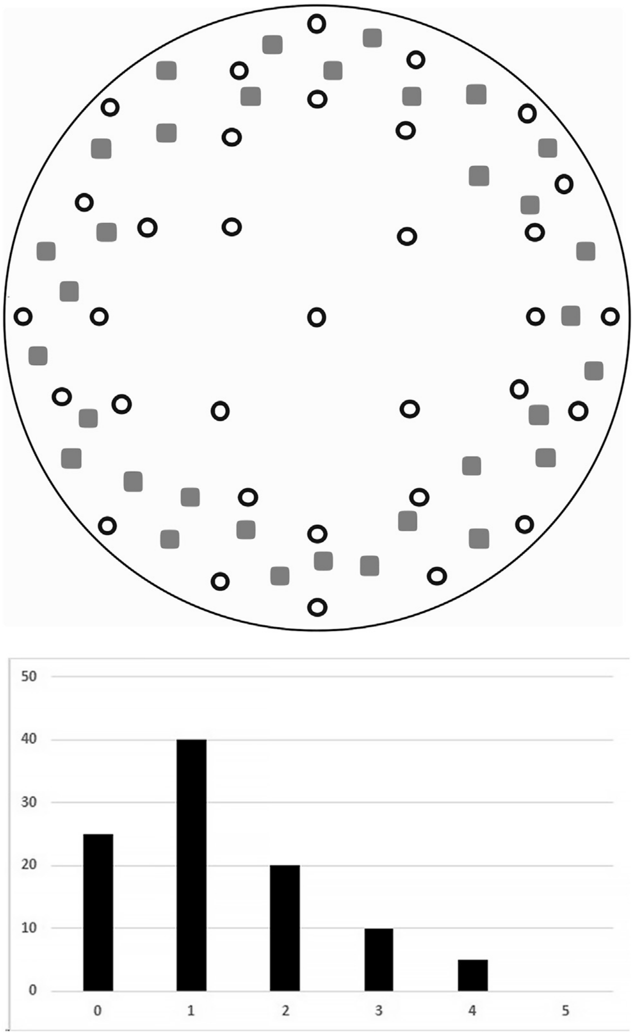


Figure 2. Examples of safe level graphs for each of the distribution shapes

This graph (top) has skewed-right property. The most minority instances (bottom) are located in an over-lapping region.

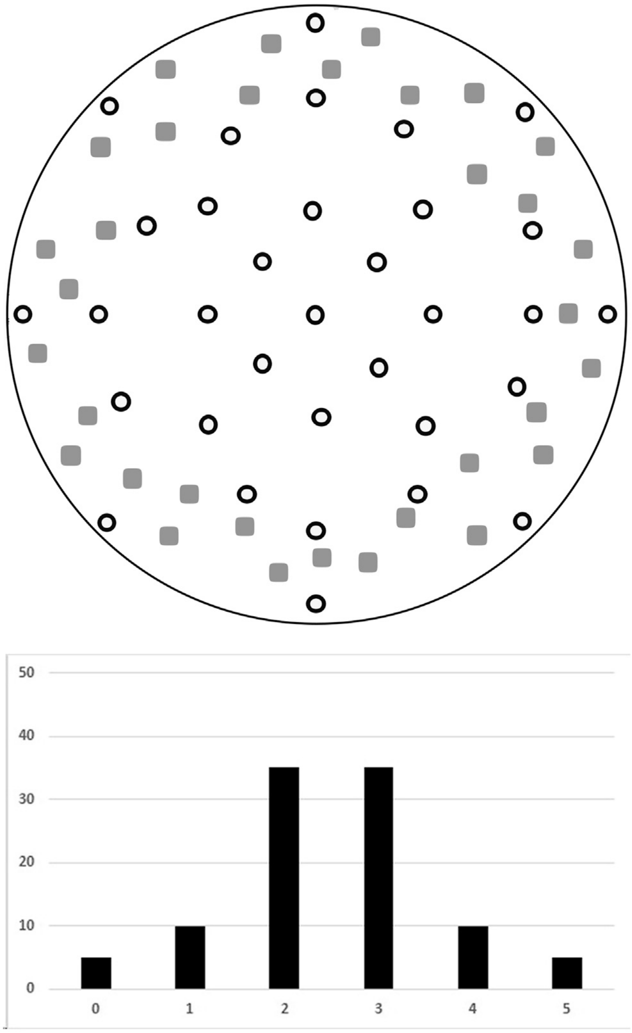


Figure 3. Examples of safe level graphs for each of the distribution shapes (more)

This graph (top) has bell curve property. All minority instances (bottom) are spread throughout a minority cluster.

C.B., K. Sinapiromsaran, and C. Lursinsap proposed Safe-Level-SMOTE[16] to address the problem of classes that overlap. The basic concept is to assign a number known as the *SL* to each instance of a minority. Counting the number of minority cases among the *k*-nearest neighbors of a given instance yields a *SL*. A higher *SL* indicates that an instance is positioned in a safer position inside a minority territory, whereas a lower *SL* indicates that the instance is primarily viewed as noise. Safe-Level-SMOTE generates new instances adjacent to minority instances with a high level of safety. In other words, this version of SMOTE would prevent the overlap of minority and majority instances by strategically positioning synthetic instances near a safe location.

C.B., K. Sinapiromsaran, and C. Lursinsap presented DBSMOTE,[23] which combines the density-based clustering algorithm DBSCAN[24] with an advanced over-sampling version of SMOTE. In this context, DBSCAN is executed to identify a minority cluster, while SMOTE is initiated to intensify the cluster's core. To accomplish this, a graph with directly accessible density will be defined. It is a DBSACN cluster that has been turned into an underlying linked graph. A synthetic instance is generated at a random position along the shortest path between two particular instances: a minority instance and the pseudo-centroid of the minority cluster. DBSMOTE is compatible with the majority of cluster shapes and is noise resistant. If the centroid is not positioned within a dense minority region (for example, if the cluster is in the shape of a donut), the cluster core will not be strengthened, and the classifier will continue to struggle to learn the minority class. Figure 1 depicts a summary of the various SMOTE approaches, with white and black dots representing training and synthetic minority class examples, respectively (Figure 1).

In contrast to Safe-Level-SMOTE and DBSMOTE, H. Han, W.-Y. Wang, and B.-H. Mao designed Borderline-SMOTE,[15] which focuses on borderline cases occurring near a minority region's boundary. Specifically, a minority instance is considered borderline if at least half of its *k*-closest neighbors belong to a

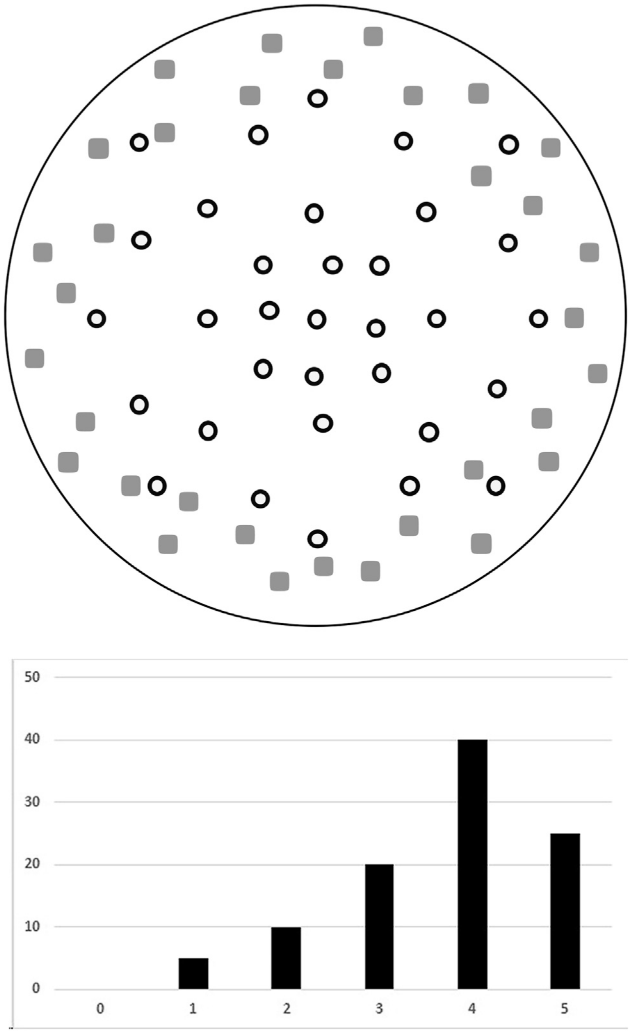


Figure 4. Examples of safe level graphs for each of the distribution shapes (more)

This graph (top) has skewed-left property. The most minority instances (bottom) are located at the core of a minority cluster.

different class. Borderline cases are traditionally susceptible to misclassification due to their location in an overlapping zone. This method is used exclusively for questionable cases. After over-sampling, the border will have a greater proportion of minority examples and a greater degree of mixture between minority and majority instances. The experiment described herein[15] reveals that Borderline-SMOTE achieves a greater F-value and TP rate on circular datasets. Nonetheless, the geometries of minority clusters can vary; therefore, borderline over-sampling may not be appropriate for all types of imbalanced datasets.

DDSC-SMOTE, an imbalanced data over-sampling algorithm based on data distribution and spectral clustering,[25] proposes a novel over-sampling method for addressing imbalanced datasets. It introduces DDSC-SMOTE, which combines data distribution analysis and spectral clustering techniques to generate synthetic samples for the minority class. By leveraging spectral clustering, the method aims to preserve local and global data characteristics, thereby potentially improving the representation of minority class samples. This paper includes experimental validation to demonstrate the effectiveness of DDSC-SMOTE compared to existing over-sampling techniques.

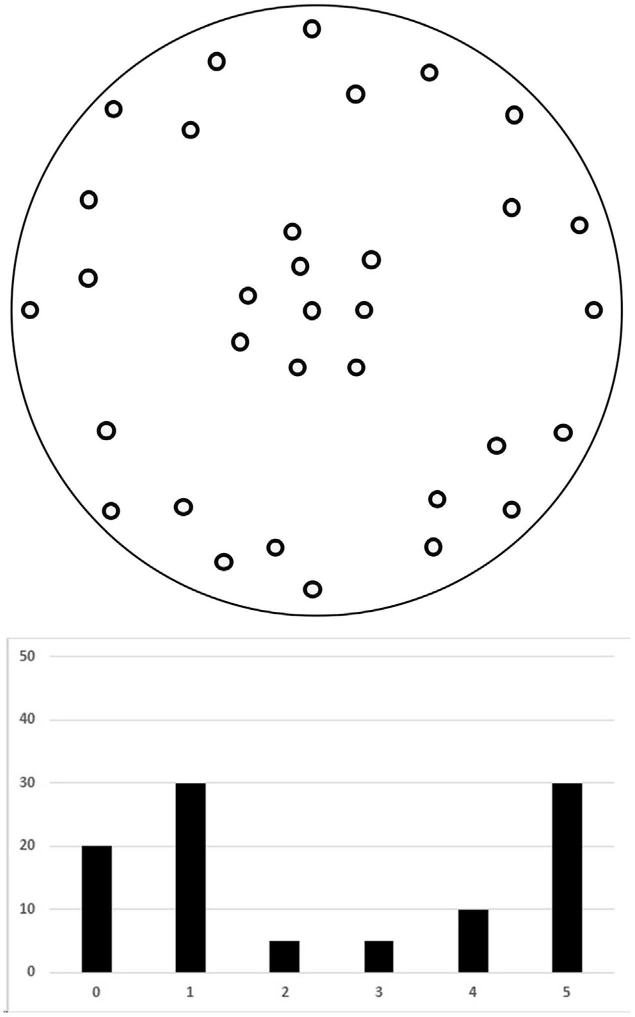


Figure 5. Examples of safe level graphs for each of the distribution shapes

Most datasets in real-world are unidentified graphs (top). The minority instances (bottom) associated with this distribution cannot be classified as skewed right, skewed left, or bell curve.

An improved SMOTE based on the center offset factor and synthesis strategy for imbalanced data classification[26] introduces enhancements to the SMOTE algorithm to better address

Table 1. Density levels of dataset *D*

SL	N_{SL}	DL_{SL}
0	$N_0 = 20$	$DL_0 = 20$
1	$N_1 = 30$	$DL_1 = 30$
2	$N_2 = 05$	$DL_2 = 05$
3	$N_3 = 05$	$DL_3 = 05$
4	$N_4 = 10$	$DL_4 = 10$
5	$N_5 = 30$	$DL_5 = 30$

Table 2. Interpretation of density level and safe level

Density level	Safe level	Dense or sparse regions
High	high	dense minority core
High	low	dense minority border
Low	high	sparse minority core
Low	low	sparse minority border

imbalanced datasets. It proposes using a center offset factor and a novel synthesis strategy to improve the quality of synthetic samples generated by SMOTE. These enhancements aim to preserve the intrinsic characteristics of the minority class while reducing the risk of overfitting. This paper includes experimental results to validate the effectiveness of these improvements compared to traditional SMOTE and other over-sampling methods.

An empirical assessment of SMOTE variant techniques and interpretation methods in improving the accuracy and the interpretability of student performance models[27] conducts a study to evaluate various SMOTE variants and interpretation methods for enhancing the accuracy and interpretability of models predicting student performance. It empirically compares different SMOTE variants to determine their effectiveness in handling imbalanced data, particularly in educational data, where predicting student outcomes is crucial. The study also assesses how these over-sampling techniques impact the interpretability of the resulting predictive models. Experimental results and findings from the paper provide insights into which SMOTE variants and interpretation methods are most beneficial for improving both prediction accuracy and the interpretability of models in educational contexts.

DDSC-SMOTE[25] combines data distribution analysis and spectral clustering techniques to generate synthetic samples for imbalanced datasets. This approach considers both local data distribution and global data structure, potentially leading to more representative synthetic samples. The disadvantage of DDSC-SMOTE is that the use of spectral clustering can be computationally intensive, especially for large datasets. This may limit the scalability of DDSC-SMOTE in real-time or large-scale applications.

The improved SMOTE method based on the center offset factor and synthesis strategy[26] aims to generate synthetic samples that better represent the minority class. By incorporating a center offset factor and refining the synthesis strategy, the method can potentially produce synthetic samples that are more realistic and closer to the actual minority class distribution. However, implementing an enhanced SMOTE method may introduce additional complexity and require careful parameter tuning. Using a center offset factor and a refined synthesis strategy could increase the number of parameters that need to be optimized, which might require more computational resources and expertise in determining the optimal settings for different datasets.

For the empirical assessment of SMOTE variants and interpretation methods,[27] this study evaluates interpretation methods alongside SMOTE variants, which can improve the transparency and understanding of how the predictive models work. This can be crucial in educational contexts where stakeholders need to interpret and trust the predictions made by the models. Unfortunately, the effectiveness of SMOTE variants and interpretation methods can vary depending on the specific characteristics of the educational datasets used. Some methods may perform differently depending on factors such as the size of the dataset, the degree of class imbalance, and the distribution of features, which could limit generalizability across different educational contexts.

Safe level graph

C.B., K. Sinapiromsaran, and S. Subpaiboonkit create *SL* graphs.[17,18] These data structures are based on the *SL*s that have been defined in the Safe-Level-SMOTE. The formula for the *SL* is exhibited in Equation 1. *SL* graphs, illustrated in Figures 2, 3, and 4 (left side), are two-dimensional graphs used to visualize the number of minority instances for each *SL*. The *SL* is on the x axis, and the y axis gives the percentage of minority instances with the given *SL*. *SL* graphs can be roughly classified into three shapes as follows.

$$SL = \text{the number of minority instances among k} - \text{nearest neighbors.} \qquad \text{(Equation 1)}$$

Skewed right: distributions in which most of the instances have a low *SL*, with relatively few having a high *SL*. In this case, minority instances are badly intermixed with majority instances, as a low *SL* implies the presence of several nearby members of the majority class. In other words, in this case, there is a very high degree of overlap between the two classes.

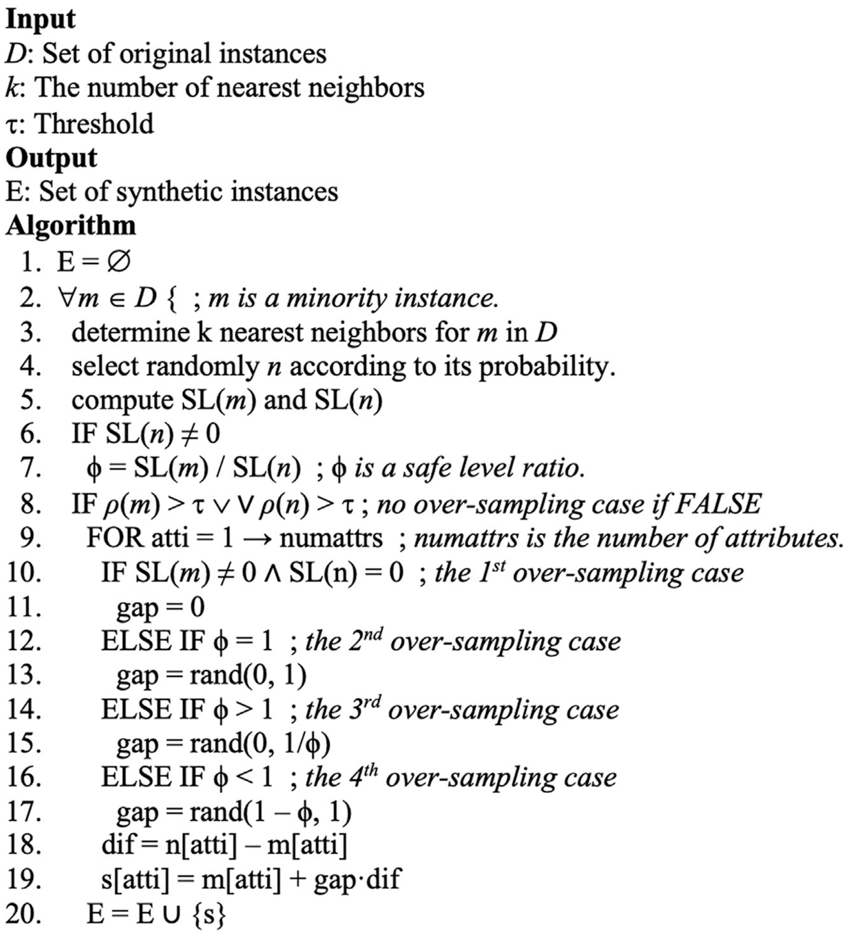
Skewed left: Distributions in which most of the instances have a high *SL*, with relatively few having a low *SL*. In contrast to the previous case, here the minority instances form a cluster with relatively little overlap.

Bell curve: A more symmetrical distribution that appears to be shaped like a normal distribution. In this case, minority instances tend to be spread throughout the minority region without clustering densely in any one location.

Examples of *SL* graphs for each of the distribution shapes above are illustrated in Figures 2, 3, 4, and 5. In these figures, white circles and gray squares represent minority and majority instances, respectively. To summarize, the overlapping data problem mostly affects datasets that are skewed right because,

Table 3. Differences among safe level, density level, and density function

Terminology	Symbol	Description
Safe level	SL	a property of each minority instance
Density level w.r.t. the *SL*	DL_{SL}	a property of the whole dataset, depending on a given safe level
Density function w.r.t. the majority instance *m*	$\rho(m)$	a property of each minority instance, based on its safe level



```
Input
$D$: Set of original instances
$k$: The number of nearest neighbors
$\tau$: Threshold
Output
E: Set of synthetic instances
Algorithm
$E = \varnothing$
$\forall m \in D$ {  ; m is a minority instance.
  determine k nearest neighbors for $m$ in $D$
  select randomly $n$ according to its probability.
  compute SL($m$) and SL($n$)
  IF $SL(n) \neq 0$
    $\phi = SL(m) / SL(n)$  ; $\phi$ is a safe level ratio.
  IF $\rho(m) > \tau \vee \vee \rho(n) > \tau$ ; no over-sampling case if FALSE
    FOR atti = 1 → numattrs  ; numattrs is the number of attributes.
      IF $SL(m) \neq 0 \wedge SL(n) = 0$  ; the 1st over-sampling case
        gap = 0
      ELSE IF $\phi = 1$  ; the 2nd over-sampling case
        gap = rand(0, 1)
      ELSE IF $\phi > 1$  ; the 3rd over-sampling case
        gap = rand(0, 1/$\phi$)
      ELSE IF $\phi < 1$  ; the 4th over-sampling case
        gap = rand(1 – $\phi$, 1)
      dif = n[atti] – m[atti]
      s[atti] = m[atti] + gap·dif
    E = E ∪ {s}
```

Figure 6. FLEX-SMOTE algorithm needs original instances and 2 parameters as inputs, while the output is a set of synthetic instances

in such cases, most minority instances are located at the border of a cluster. In a skewed-left dataset, minority instances are concentrated in the core of a cluster so that most minority instances are safe for learning by classifiers. For the bell curve dataset, there was some overlap, but the minority instances were not obviously formed as a core.

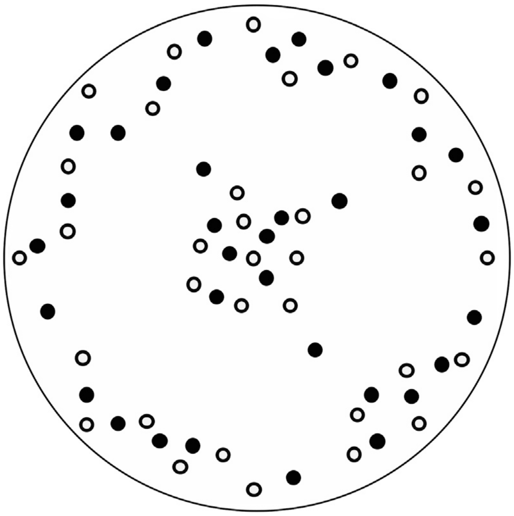


Figure 7. This dataset is generated by FLEX-SMOTE

The white dots represent original minority instances, while black dots represent synthetic minority instances.

Other re-sampling techniques

Over-sampling is a technique that creates instances in a minority class. The simplest one is random over-sampling,[21] which randomly copies minority instances. In this case, the output involves a bigger minority class. The duplicate instances are not new information in a minority class but are part of the cost of computing.

Opposite to the over-sampling approach, such as with the SMOTEs that increase the size of a minority class, under-sampling is a technique that decreases the size of the majority class. The simplest one is characterized as random under-sampling,[21] wherein majority instances are randomly deleted. This method does not concentrate on any region, core, or border, while the classification results can affect randomness.

A study[28] that only took into account numerical attributes revealed that negative instances can be loosely classified into four categories: noisy, borderline, safe, and redundant. A minority class's decision regions are covered by noise, while positive and negative regions are separated by the borderline. This is because even a modest quantity of noise might cause borderline cases to move to the incorrect side of the decision surface and result in an untrustworthy borderline area. The secure area is retained for upcoming classification activities. Although the redundant instances raise classification costs, they do not hinder accurate classifications. One-sided selection, a heuristic under-sampling strategy, has also been implemented by the authors. By removing negative instances from noise and borderline regions, it under-samples a majority class. The idea of Tomek linkages makes it simple to find these harmful cases.[29] The trials, however, showed that the selection of the eliminated negative cases had little impact on the induced classifier's performance.

FLEX-SMOTE

The motivation for this paper comes from an understanding that *SL* graphs[17,18] are only able to suggest one SMOTE for a given imbalanced dataset. Unfortunately, minority instances in most real-world datasets do not conform to the simple distributions discussed above; see Figure 5 as an example. This makes it difficult to use an *SL* graph to choose the right SMOTE for the

Table 4. Example of the symmetrical property between the 4th and 5th cases

$SL(m)$	$SL(n)$	ϕ	Gap	Gap length
2	4	0.5	[0.75, 1]	0.25
4	2	2	[0, 0.25]	0.25

right dataset. Moreover, applying only a single SMOTE might not be the right strategy for handling the class imbalance problem because some datasets have a combination of distributions. In this paper, we have designed a new, flexible over-sampling technique called FLEX-SMOTE. It integrates the various strengths of the other SMOTE techniques by using the density function as defined below to guide the generation of synthetic instances.

We have introduced a new term called the density level (DL_{SL}) w.r.t. as representative of a specific SL. As is shown in Equation 2, the density level is equal to the number of minority instances within a given SL. For example, if there are 20 minority instances where the SL is equal to 0, then the corresponding density level will be 20 (see Table 1). In addition, we let N_{SL} be the number of minority instances with respect to the SL, and we can then conclude that $DL_{SL} = N_{SL}$.

$$DL_{SL} = \text{the number of minority instances whose safe levels are equal to SL} \tag{Equation 2}$$

A larger value for the density level with respect to a small SL means that a greater number of minority instances are spread out along the border of a minority region. In contrast, the minority core is crowded with more minority instances when the density level w.r.t. a huge SL is higher. The interpretation of the values of both levels is explained in Table 2. Note that while the SL is a property of each minority instance, the density level is a property of the whole dataset.

We can now define density function ρ based on the density level described above. As has been shown in Equation 3, the density function is assigned to each minority instance m for the density level of the dataset with respect to the SL of m. For example, in Table 1, in the dataset D described above, if m is one of the instances whose SL is 4, then the density function of m will be 10, which equals the density level associated with SL 4. To prevent confusion among all symbols, we use the capital letters DL to represent density level and the Greek letter ρ to indicate the density function. Accordingly, all of this is summarized in Table 3.

$$\rho(m) = DLSL_{(m)} \tag{Equation 3}$$

To analyze the density function, m is located where a lot of minority instances are formed to be a class core if ρ returns a large value and $SL(m)$ is high. On the other hand, both the return value and SL are low when a few minority instances that include m are spread around the border or a minority region. The idea behind the FLEX-SMOTE algorithm is to generate synthetic instances closer to minority instances with higher density function values because such instances are located in a denser region that is considered safer for over-sampling than the latter.

Our technique over-samples a minority class using an approach based on SMOTE, in which a synthetic instance is generated along the line segment connecting a given minority instance (c), and an instance is selected randomly from among the k-nearest neighbors within its class (s). In contrast to SMOTE, which equally prioritizes minority instances, FLEX-SMOTE prioritizes minority instances whose density functions exceed a user-specified threshold.

FLEX-SMOTE is exhibited in Figure 6. The algorithm initializes an empty set E in the first step. In the subsequent loop, each minority instance m is considered in turn. If the density function of m and a randomly chosen neighbor n are both above the threshold τ, then a new synthetic instance is generated and added to the set E. This synthetic instance is positioned along the line segment connecting m and n, with a probability distribution based on the density function at m and n. When the algorithm terminates, its output is the set E of synthetic instances. This set is then merged into the original dataset, resulting in an over-sampled dataset.

Theorem 1. Symmetric property of the 4th and 5th over-sampling cases.

Proof:

We can illustrate this in Figure 8. From the 4th case, it can be stated that

$$0 \leq \text{gap}_1 \leq \frac{1}{\phi_1}$$

Because $\phi_1 = \frac{SL(m_1)}{SL(n_1)}$

$$0 \leq \text{gap}_1 \leq \frac{1}{\frac{SL(m_1)}{SL(n_1)}}$$

So, $1 \geq 1 - \text{gap}_1 \geq 1 - \frac{1}{\frac{SL(m_1)}{SL(n_1)}}$

Note that $\text{gap}_1 + \text{gap}_2 = 1$.

$$1 - \frac{SL(n_1)}{SL(m_1)} \leq \text{gap2} \leq 1$$

We assume that the two cases are symmetrical. Thus, $m_1 = n_2$ and $n_1 = m_2$

$$1 - \frac{SL(m_2)}{SL(n_2)} \leq \text{gap2} \leq 1$$

Because $\phi_2 = \frac{SL(m_2)}{SL(n_2)}$

$$1 - \phi_2 \leq \text{gap2} \leq 1.$$

Finally, we can prove that the two cases are symmetrical.

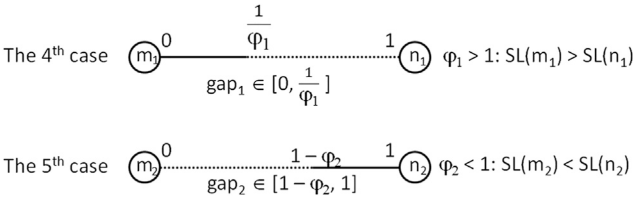

The 4th case
m_1
0
$\frac{1}{\varphi_1}$
1
n_1
$\varphi_1 > 1$: SL(m_1) > SL(n_1)
$\text{gap}_1 \in [0, \frac{1}{\varphi_1}]$
The 5th case
m_2
0
$1 - \varphi_2$
1
n_2
$\varphi_2 < 1$: SL(m_2) < SL(n_2)
$\text{gap}_2 \in [1 - \varphi_2, 1]$

Figure 8. The example shows the symmetric cases of cases 4 and 5

The locations where synthetic instances are generated in a FLEX-SMOTE dataset are illustrated in Figure 7. In this figure, white dots and black dots represent original minority instances and synthetic instances, respectively. As is shown in the figure, FLEX-SMOTE has the effect of over-sampling crowded minority instances. This allows FLEX-SMOTE to handle a minority class with arbitrarily shaped clusters. In general, a minority class may be dense in more than one region, for example, in both the core and borderline regions. FLEX-SMOTE is designed to handle such cases, wherein traditional techniques, such as SMOTE, Borderline-SMOTE, or Safe-Level-SMOTE, tend not to work well.

The terminology of the algorithm is as follows:

D is a set of the original dataset including both minority and majority instances.
E is a set of synthetic instances and is merged with *D* to be an over-sampled set.
k is the number of nearest neighbors used in the over-sampling step.
SL is the safe level of an instance computed from the number of minority neighbors.
DL is the density level of an instance computed from the number of all instances with the same *SL* of the instance.
ρ is the density function
τ is the threshold that needs to be adjusted.
m is considered a minority instance.
s is a synthetic instance generated in line 19.

In the algorithm, the 4th and 5th over-sampling cases are symmetrical. In other words, if the *SL*s between a pair of instances are exchanged, then the gap lengths remain unchanged. An example of this is shown in Table 4. We have provided evidence of this property in Theorem 1. The key idea behind this is illustrated in Figure 8. This figure shows a dashed line segment connecting the two instances that is parameterized from 0 to 1. The gap in which a synthetic instance can be generated in each case is shown as a solid line; the gaps have the same length and emanate from the instance with a higher *SL* in each case.

The threshold is one parameter that we need to adjust. It is used in line 8 of the algorithm. The two instances will be considered for the over-sampling step if, and only if, the density levels of the *SL*s of the two instances are greater than the threshold. This means that the two instances are in regions that are dense enough with respect to threshold τ.

Table 5. Descriptions of experimental techniques

Techniques	Abbreviations	Sampled regions
None	ORG	none (ORG is an original dataset without over- or under-sampling)
Over-sampling		
Random over-sampling	OVR	throughout a minority region (OVR generates duplicate instances)
SMOTE	SMT	throughout a minority region (SMT produces synthetic instances)
Borderline-SMOTE	BRD	border of a minority region
Safe-Level-SMOTE	SFE	all but neglect the minority border
DBSMOTE	DBS	all but focus on a minority core
Under-sampling		
Random under-sampling	UND	throughout a majority region
Tomek links	TMK	border of a majority region (TMK erases noises or borderlines)
MUTE	MUT	border of a majority region (MUT deletes only noises)

Conclusion

In this paper, we have proposed a new over-sampling technique called the FLEX-SMOTE. We use the term "flexible" because our technique can be applied to several different minority class distributions. FLEX-SMOTE automatically adjusts to generate minority instances into varied regions from the concept "where there's substantial, there's synthetic instances." Consequently, FLEX-SMOTE is able to adapt to many types of imbalanced datasets. The advantage of our method is that users do not need to apply trial and error as they do with many other kinds of SMOTEs. Rather, they can directly use FLEX-SMOTE to remedy the class imbalance problem. For future research, we plan to apply a similar concept to the design of a new under-sampling technique. This technique would remove majority instances by examining the distribution of majority classes and remove different numbers of instances in different regions.

Interestingly, when a dataset contained most instances of a very high *SL*, FLEX outperformed other techniques. Remarkably, when considering Segment, Satimage, and Letter in Table S9, FLEX achieved the best average result in Tables S10 and S11. However, FLEX and DBS results were comparable only to those

Table 6. Descriptions of experimental classifiers

Classifiers	Abbreviations	Types
C4.5	C4.5	decision tree
Multilayer perceptron	MLP	neural network
RIPPER	RIP	rule-based classifier
Naive Bayes	NB	probabilistic classifier
k-Nearest neighbor	KNN	instance-based classifier
Support vector machine	SVM	discriminative classifier
Logistic regression	LGB	linear classifier
Random forest	RF	ensemble of classifiers

Table 7. Descriptions of experimental datasets

Dataset	Minority class	Abbreviations	Features	Instances	Minority percentage (%)
Pima Indians Diabetes	tested positive for diabetes	Pima	8	768	34.90
Haberman's Survival	patient died within 5 years	Haberman	3	306	26.47
Image Segmentation	window	Segment	19	2,310	14.29
Ecoli	imU	Ecoli	7	336	10.42
Stalog (Landsat satellite)	damp gray soil	Satimage	36	6,435	9.36
Glass Identification	vehicle windows	Glass	9	214	7.94
Letter Recognition	H	Letter	16	20,000	3.67
Page Blocks Classification	picture	PB	10	5,473	2.10
Yeast	VAC	Yeast	8	1,484	2.02
Abalone	18	Abalone	10	4,177	1.01

of Satimage. A safe conclusion would then be that FLEX is suitable for any dataset with this characteristic.

With regard to the limitations of this study, this algorithm was designed to deal only with continuous features. However, if a dataset contains categorical features, then we will then need to apply data conversion. Importantly, in future research work, we intend to further improve this technique to handle nominal/ordinal features.

EXPERIMENTAL PROCEDURES

In this section, we have presented the results of an experiment testing the relative performance of FLEX-SMOTE[30] and the other SMOTE variants described above. In addition, we included three under-sampling methods in the experiment to compare the performance of under- and over-sampling in a single region. Due to the premise that algorithms that are under-sampled can be prevented by removing significant information from a majority core, an under-sampled region in our experiment would only be representative of a majority border. To evaluate the predictive rate, we applied various types of classifiers, configured with their default parameters using the software package Weka. Weka is a collection of machine learning algorithms. The algorithm contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. Lists of all the over- and under-sampling techniques and all experimental classifiers[31] are shown in Tables 5 and 6, respectively.

Table 7 lists the collection of UCI datasets[32] used in this experiment. These datasets come from various domains and have differing minority percentages. We scoped this research as a two-class classification problem with each imbalanced dataset consisting of a single minority and majority class. For each dataset, we chose one rare class to be the minority class and then merged all remaining classes to be the majority class. If there were more than one rare class in some dataset, then we set each rare class as a minority class so that we could then archive separate datasets from the same original file. In total, we tested 10 different experimental datasets.

Table 8 presents our experimental parameters as follows. We chose a value of $k = 5$ for the k-nearest neighbors step in the various SMOTEs because 5 is not only a commonly used value for this approach, but it is also considered a default setting in several class imbalance works.[16,23,33,14,15,29,17,18,34] For Borderline-SMOTE, we followed the recommendation that the number of borderline instances should be approximately half of the number of minority instances. We deny DBSCAN's clusters with noise because over-sampled minority instances do not include noise; additionally, we have balanced the number of synthetic instances for all over-sampling techniques in order to directly compare their predictive performance. Specifically, we configured all over-sampling techniques to achieve a 100% over-sampling rate, which meant that datasets were double sized. In contrast, under-sampling was configured to 50%, which meant that the datasets were down-sized by a factor of two. We weighted equally between precision and recall in F-value so as not to bias our results toward one or the other. We ran all models using the Weka tool with the default parameters; thus, all compared techniques used the same settings for each classifier.

Table 8. Descriptions of parameter settings

Parameter	Algorithm/measure/software	Meaning	Value
k	KNN	the number of nearest neighbors computed in an over-sampling process	5
m	Borderline-SMOTE	the number of nearest neighbors computed in the process of detecting borderline instances	tuned so that the number of borderline instances was about half of the minority class
Eps	DBSCAN	the maximum radius of the neighborhood from a core instance	tuned so that the lowest value that discovers a single cluster without noise
$MinPts$	DBSCAN	the minimum number of instances required to form a density-based cluster	k
O %	random over-sampling and SMOTEs	over-sampling percentage	100%
U %	random under-sampling	under-sampling percentage	50%
β	F-value	the weight between precision and recall	1
All	Weka	classifiers under this software	default
τ	FLEX	threshold	setting $k = 0$ to 5

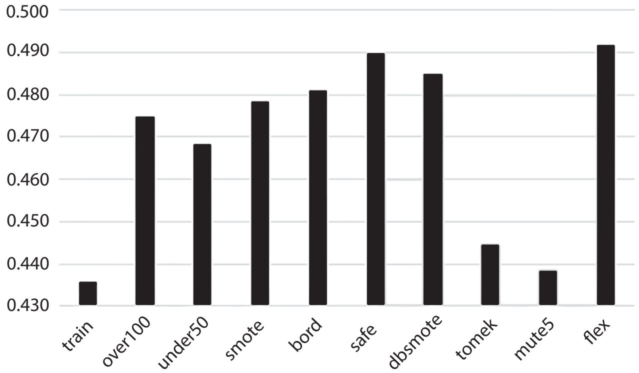


Figure 9. The results show the overall average F-value of decision tree, perceptron, RIPPER, naive Bayes, KNN, SVM, logistic regression, and random forest

KNN, *k*-nearest neighbor; SVM, support vector machine.

For each dataset, all instances have been shuffled. Next, each dataset was split up into a training set (ratio 2/3) and a test set (ratio 1/3). However, we did not split Statlog (Landsat satellite) because the UCI website distributed two separate sets. To avoid the randomness of synthetic instances, three independent runs were performed, and the results were determined by the median of the selected performance measure.

To summarize the results shown in Tables S1 and S2 and Figures 9 and 10, FLEX not only achieved a superior F-value with the random forest classifier but also achieved the largest average F-value. Interestingly, FLEX was able to improve the F-value for the decision tree classifier on the Abalone dataset, even though this dataset was determined to be the most imbalanced. Notably, FLEX also achieved the highest AUC. Both under-sampling techniques, TMK and MUT, failed to improve both the F-value and AUC. As is shown in the rankings in Tables S3 and S4, FLEX was the best technique for all classifiers in terms of F-value, while FLEX also obtained a satisfactory ranking in terms of AUS. In this paper, the average value represented an overall result. Remarkably, we did not only consider the average but also determined that FLEX indicated the best possible option.

We have applied the Wilcoxon signed-rank test to test statistical significance for the paired differences; the details are shown in Table 9. This was used to test FLEX against the other techniques with respect to F-value and AUC. FLEX significantly outperformed the others, except for SMOTE, on AUC in Tables S5 and S6. Note that minority instances are rarely distributed normally, so the Wilcoxon test is appropriate because it does not assume normality. The Wilcoxon test was determined to be safer than paired t tests in this research work.[35] Additionally, we also administered the Friedman test on all SMOTEs in Tables S7 and S8. The results were considered significant when $p < 0.05$.

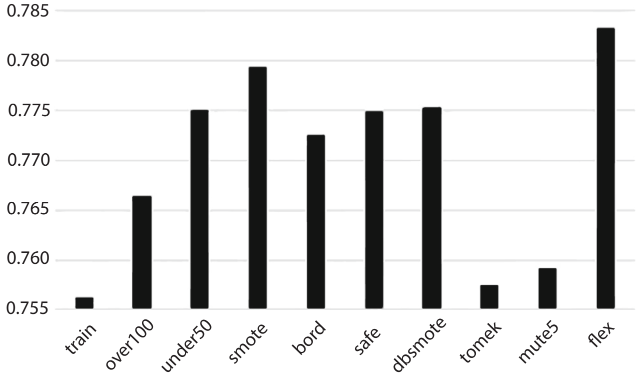


Figure 10. Overall average AUCs of various classifiers are revealed here

Table 9. Descriptions of Wilcoxon signed-rank tests

Symbol	Variable	Value	Meaning
H_0	bull hypothesis	–	there is no difference of results between FLEX and its comparative technique.
H_1	alternative hypothesis	–	there is a difference
α	significance level	5%	it is used to verify that the experimental results are statistically significant or not
p value	two-tailed probability	$\leq\alpha$	H_0 is rejected
		$>\alpha$	H_0 is accepted

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Ekkarat Boonchieng (ekkarat.boonchieng@cmu.ac.th).

Materials availability

This study did not generate new materials.

Data and code availability

The dataset is available from http://archive.ics.uci.edu/datasets. The source code is available from https://figshare.com/articles/journal_contribution/FLEX-SMOTE/23629662.[30] All other data reported in this paper will be shared by the lead contact upon request.

ACKNOWLEDGMENTS

This research is supported by Fundamental Fund 2025, Chiang Mai University.

AUTHOR CONTRIBUTIONS

Conceptualization, methodology, software, writing – original draft, C.B.; formal analysis, writing – review and editing, investigation, data curation, supervision, project administration, funding acquisition, lead researcher, E.B.; resources, visualization, V.C.; validation, writing – review and editing, D.L.

DECLARATION OF INTERESTS

The authors declare no competing interests.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2024.101073.

Received: February 14, 2023
Revised: June 9, 2023
Accepted: September 12, 2024
Published: October 9, 2024

REFERENCES

1. Chawla, N.V. (2010). Data Mining for Imbalanced Datasets: An Overview. In Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, eds. (Springer), pp. 875–886.
2. Garcia, V., Sánchez, J.S., Mollineda, R.A., Alejo, R., and Sotoca, J.M. (2007). The class imbalance problem in pattern classification and learning. In IV Taller Nacional de Minería de Datos y Aprendizaje (TAMIDA 2007), pp. 283–291.
3. Han, J., Kamber, M., and Pei, J. (2011). Data Mining: Concepts and Techniques, 3rd Edition (Morgan Kaufman).
4. Japkowicz, N., and Stephen, S. (2002). The Class Imbalance Problem: A systematic Study. Intell. Data Anal. *6*, 429–449.
5. Nguyen, G.H., Bouzerdoum, A., and Phung, S.L. (2009). Learning Pattern Classification Tasks with Imbalanced Data Sets. In Pattern recognition, P. Yin, ed., pp. 193–208.
6. Yang, Q., and Wu, X. (2006). 10 Challenging problems in data mining research. Int. J. Inf. Technol. Decis. Making *05*, 597–604.
7. Elreedy, D., Atiya, A.F., and Kamalov, F. (2023). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. Mach. Learn. *113*, 4903–4923. https://doi.org/10.1007/s10994-022-06296-4.
8. He, H., and Garcia, E.A. (2009). Learning from Imbalanced Data. IEEE Trans. Knowl. Data Eng. *21*, 1263–1284.
9. He, H., and Ma, Y. (2013). Imbalanced Learning: Foundations, Algorithms, and Applications (Wiley).
10. Chawla, N.V. (2010). Data Mining for Imbalanced Datasets: An Overview (Data Mining and Knowledge Discovery Handbook), pp. 875–886.
11. Khor, K.-C., Ting, C.-Y., and Phon-Amnuaisuk, S. (2012). A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection. Appl. Intell. *36*, 320–329.
12. Japkowicz, N. (2000). The Class Imbalance Problem: Significance and Strategies. In The 2000 International Conference on Artificial Intelligence (Las Vegas), pp. 111–117.
13. Tetko, I.V., Livingstone, D.J., and Luik, A.I. (1995). Neural Network Studies. 1. Comparison of Overfitting and Overtraining. J. Chem. Inf. Comput. Sci. *35*, 826–833.
14. Chawla, N.V., Bowyer, K.W., Hall, L.O., and Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-Sampling TEchnique. J. Artif. Intell. Res. *16*, 321–357.
15. Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In the 2005 International Conference on Intelligent Computing, Hefei, China. Lecture Notes in Computer Science, *3644*, D.-S. Huang, X.-P. Zhang, and G.-B. Huang, eds. *(Springer), pp. 878–887*.
16. Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2009). Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-sampling TEchnique for handling the class imbalanced problem. In Advances in knowledge discovery and data mining: 13th Pacific-Asia conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 proceedings 13, T. Theeramunkong, B. Kijsirikul, N. Cercone, and T.-B. Ho, eds. (Springer), pp. 475–482.
17. Bunkhumpornpat, C., and Sinapiromsaran, K. (2014). Safe Level Graph for Majority Under-sampling Techniques. Chiang Mai J. Sci. *41*, 1419–1428.
18. Bunkhumpornpat, C., and Subpaiboonkit, S. (2013). Safe Level Graph for Synthetic Minority Over-sampling Techniques. In 13th International Symposium on Communications and Information Technologies (Samui Island), pp. 570–575.
19. Buckland, M., and Gey, F. (1994). The Relationship between Recall and Precision. J. Am. Soc. Inf. Sci. *45*, 12–19.
20. Bradley, A.P. (1997). The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recogn. *30*, 1145–1159.
21. Drummond, C., and Holte, R.C. (2003). C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling beats Over-Sampling. In The ICML 2003 Workshop on Learning from Imbalanced Data Sets II, pp. 1–8.
22. Prati, R.C., Batista, G.E.A.P.A., and Monard, M.C. (2004). Class Imbalances versus Class Overlapping: an Analysis of a Learning System Behavior. In The 3rd Mexican International Conference on Artificial Intelligence, Mexico City, Mexico. Lecture Notes in Artificial Intelligence, *2972*, R. Monroy, G. Arroyo, L.E. Sucar, and H. Sossa, eds.*, pp. 312–321*.
23. Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2012). DBSMOTE: Density-Based Synthetic Minority Over-sampling Technique. Appl. Intell. *36*, 664–684.
24. Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In The 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231.
25. Li, X., and Liu, Q. (2024). DDSC-SMOTE: an imbalanced data oversampling algorithm based on data distribution and spectral clustering. J. Supercomput. *80*, 17760–17789. https://doi.org/10.1007/s11227-024-06132-7.
26. Zhang, Y., Deng, L., Huang, H., and Wei, B. (2024). An improved SMOTE based on center offset factor and synthesis strategy for imbalanced data classification. J. Supercomput. *80*, 22479–22519. https://doi.org/10.1007/s11227-024-06287-3.
27. Sahlaoui, H., Alaoui, E.A.A., Agoujil, S., and Nayyar, A. (2024). An empirical assessment of smote variants techniques and interpretation methods in improving the accuracy and the interpretability of student performance models. Educ. Inf. Technol. *29*, 5447–5483. https://doi.org/10.1007/s10639-023-12007-w.
28. Kubat, M., Holte, R., and Matwin, S. (1997). Learning When Negative Examples Abound. In The 9th European Conference on Machine Learning (ECML 1997), pp. 146–153.
29. Tomek, I. (1976). Two Modifications of CNN. IEEE Transaction on Systems, Man and Cybernetics *6*, 769–772.
30. Bunkhumpornpat, C. FLEX-SMOTE. Figshare. https://figshare.com/articles/journal_contribution/FLEX-SMOTE/23629662.
31. Witten, I.H., Frank, E., and Hall, M.A. (2011). Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition (Morgan Kaufman).
32. The UC Irvine Machine Learning Repository. https://archive.ics.uci.edu.
33. Bunkhumpornpat, C., Sinapiromsaran, K., and Lursinsap, C. (2011). MUTE: Majority Under-sampling TEchnique. In The 8th International Conference on Information, Communications, and Signal Processing. Singapore.
34. Bunkhumpornpat, C., and Sinapiromsaran, K. (2015). CORE: Core-based Synthetic Minority Over-sampling and Borderline Majority Under-sampling Technique. Int. J. Data Min. Bioinf. *12*, 44–58.
35. Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. *7*, 1–30.