A Review on Oversampling Techniques for Solving the Data Imbalance Problem in Classification

Tharinda Dilshan Piyadasa, Kasun Gunawardana

***Abstract*— The data imbalance problem is a widely explored area in the Machine Learning domain. With the rapid advancement of computing infrastructure and the incessant increase in the amount and variety of data generated, the data imbalance problem has prevailed and reshaped with the requirement for novel approaches to address it. Among the different approaches that exist to address the data imbalance problem, such as data-level and algorithmic-level, data-level approaches are more popular among the scientific community due to their classifier-independent nature. When investigating current trends in data-level approaches, it is evident that oversampling is a technique frequently explored due to its adaptability to scenarios where extreme data imbalance is present. This paper presents a review of different oversampling techniques with a comprehensive analysis of the strategies that have been used along with possible areas that looks promising to explore further to develop more advanced oversampling techniques.**

***Keywords*— Data Imbalance Problem, Classification Analysis, Oversampling.**

I. Introduction

Data mining and knowledge discovery have become indispensable in the contemporary age of big data for making accurate decisions and predictions. Classification analysis is one of the most commonly employed data mining tasks for various market and engineering problems, such as bankruptcy prediction, network intrusion detection, fraud detection, and software fault detection, where classifiers are trained to discriminate between the different classes representing the problem [1]. When using traditional classifiers to carry out the said tasks, it can be observed that these classifiers perform well over evenly distributed data. This is due to the fact that traditional classifiers are designed to increase accuracy with no notion of the distribution of data [2]. However, in the real world, data collected for classification analysis are usually class (or Data) imbalanced. In the context of classification analysis, class imbalance refers to classification problems where the dataset contains at least one class with significantly fewer samples than other classes in the dataset. In a two-class classification problem, the class with the fewest samples is called the minority class, and the other class is called the majority class [3]. The class disproportionate among these samples is identified using the Imbalance Ratio (IR), which can vary from dataset to dataset. This metric simply represents the ratio between the majority and minority class samples.

In many practical applications of classification analysis, the minority class represents the positive examples or the target class where the adverse effect of false-negative predictions is much higher than false-positive predictions [1]. For example, when considering credit card fraud detection, there can be thousands of regular transactions for a single fraudulent transaction, making the target class the minority class in the dataset. Suppose a regular transaction is flagged as a fraudulent transaction (false positive) by a trained model. In that case, it can later be resolved using further examinations. Still, on the other hand, if a fraudulent transaction is incorrectly classified as a regular transaction (false negative), which is the usual behavior of traditional classifiers on imbalanced datasets, the primary intention of the classifier is futile. The justification behind this behavior is that, in extreme imbalance scenarios where positive examples are under-represented, they are often mistaken for noise, outliers, or allocated to the majority class, ignoring the importance of their characteristics, leading traditional learning models to favor the majority class heavily [4].

Another significant observation of class imbalance is that, regardless of the poor performance of standard classifiers on the minority class, the classifier would still make predictions with higher accuracy depending on the imbalance ratio of the classes. For example, suppose the imbalance ratio of a binary class dataset is 9:1 (for nine samples in the majority class, there is only one minority sample). The classifier can acquire an accuracy of 90% by classifying all the samples into the majority class, which is a decent accuracy when considering a standard classifier [5]. In practical applications, the imbalance ratio can be much higher than the ratio depicted in the above example. It is also evident that accuracy is not a suitable evaluation metric to evaluate a standard classifier when datasets are imbalanced as the importance of the minority class is ignored.

A. Addressing the Data Imbalance Problem

The approaches used to overcome the data imbalance problem can be categorized into three groups as represented in Fig 1: External approaches (data level), Internal approach es (algorithmic level), and Hybrid approaches.

The external approaches focus on balancing the dataset either by removing the majority class samples through undersampling or adding minority class samples through oversampling. It is also possible to combine oversampling and undersampling to form hybrid sampling methods. The objective of external approaches is to reduce the imbalance ratio to achieve a favorable distribution among the classes.

Correspondence:Tharinda Dilshan Piyadasa (E-mail: tharindad7@gmail.com) Received:18-07-2022 Revised:23.01.2023 Accepted: 26-02-2023

Tharinda Dilshan Piyadasa and Kasun Gunawardana are from University of Colombo School of Computing, Sri Lanka. (tharindad7@gmail.com, kgg@ucsc.cmb.ac.lk)

DOI: http://doi.org/10.4038/icter.v16i1.7260

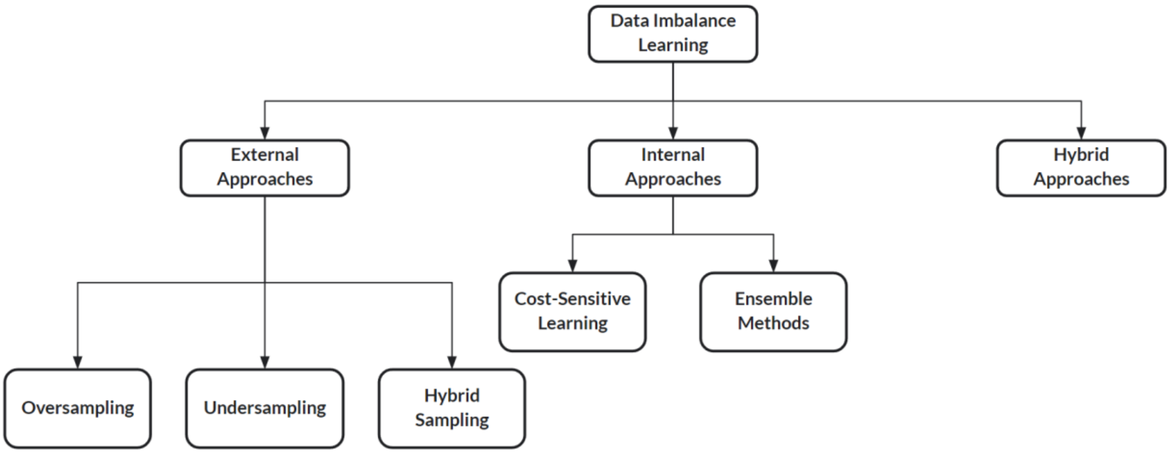

Fig. 1 Approaches to address the data imbalance problem.

Even though most of the proposed external approaches resample the dataset until the number of samples in each class is equal, studies such as [6] demonstrate that it is not always required to maintain a 50:50 class distribution when resampling. However, there is no hard and fast rule to decide on a favorable imbalance ratio as it can vary depending on the domain and the type of classifier used.

Internal approaches involve developing and improving the underlying classification algorithm without altering the dataset involved [7]. There are mainly two ways that internal approaches address the data imbalance problem . The first method is cost-sensitive learning, where the classifier is modified such that the misclassification of minority class samples is heavily penalized compared to the misclassification of majority class samples. The second and most popular internal approach is to incorporate ensemble-based classifiers where multiple weak classifiers are combined to improve the performance of the overall classification algorithm. Apart from these methods, there have also been algorithmic classifier modifications proposed in past years to improve the classifier performance on classifiers like Support Vector Machines (SVM), Extreme Learning Machines (ELM), and Neural Networks (NN). Moreover , internal approaches can also be combined with external approaches to derive hybrid approaches that incorporate both advantages and disadvantages of internal and external approaches [1][8].

When comparing the approaches to address the Data Imbalance problem, it is apparent that researchers prefer external approaches over internal approaches mainly due to the classifier independence [5]. In external approaches, since only the dataset is modified , it gives the freedom to select any suitable classifier for the classification task. However, in the case of internal approaches, as the internal structure/ algorithm of the classifier is modified to address the imprecise classification of minority class samples , the dataset is heavily dependent on the modified classifier . Nevertheless , it is impractical to use the same classification algorithm with every dataset in different contexts. Therefore, on the basis of generalizability , it is reasonable to presume that external approaches provide an added advantage over internal approaches.

B. *Overview of External Approaches*

As aforementioned, external approaches to solving the data imbalance problem are heavily favored in the field of research due to classifier independence. When considering oversampling and undersampling, both methods have their own advantages and disadvantages. The main drawback of oversampling is that it risks generating synthetic data that can lead to overfitting. This can be caused by generating synthetic samples that closely resemble original samples or by incorrectly positioning (overlapping other classes) synthetic samples in the data space. In the case of undersampling, it risks excluding important information from the dataset, such as samples that are crucial when deciding the decision boundaries or samples that contain a higher weight in representing a particular class or a feature. Apart from the exclusion of important information, undersampling can also suffer from data scarcity after resampling if the minority class contains extremely fewer samples.

Throughout the past years, many studies have been carried out to investigate methods and mechanisms to mitigate these drawbacks from external approaches. For example, the most intuitive technique to add or remove data to/from a given class is by performing random selections. These primitive techniques have evolved and improved over time to address their foundational drawbacks by combining more complex techniques and statistical and probabilistic methods.

When comparing oversampling and undersampling, even though oversampling leads to overfitting, it is possible to detect it during the earlier stages of training using straightforward approaches such as using a good train test split and observing the change in testing error compared to the training error. However, in the case of important information exclusion caused by undersampling, although it might work well with the resampled dataset, the classifier trained with excluded samples can lead to many misclassifications with the introduction of new data samples. Mohammed et al. [9] validate this assertion, where several state-of-the-art classifiers are used to evaluate oversampled and undersampled datasets. The authors have concluded that compared to undersampling, oversampling of datasets leads to a more accurate classification.

There are numerous studies, including [10] and [11], that review oversampling techniques by conducting experiments and comparing the results to provide a comprehensive evaluation of the performance of different oversampling techniques in practical settings. However, the objective of these studies seems to be finding the better technique out of a set of available techniques based on a systematic analysis, and they provide limited information on the approaches and methodologies used in such techniques. As a result, a researcher may find it challenging to comprehend the underlying strategies of an oversampling technique, which, in turn, would lead to failure in addressing its limitations.

This paper provides a comprehensive review of some popular oversampling techniques used to address the data imbalance problem, highlighting their strategies and potential areas for further improvement. The aim of this study is to provide insights and guidance for researchers and practitioners in the field of machine learning who are interested in developing more robust oversampling techniques to address the data imbalance problem.

The rest of this paper is organized as follows. Section II provides a comprehensive review of existing oversampling techniques, highlighting their strategies when performing oversampling. Section III presents the key findings of the review, emphasizing the factors that need to be considered when formulating new oversampling techniques, followed by the conclusion in section IV.

II. ANALYSIS OF OVERSAMPLING TECHNIQUES

When selecting studies for the review, a deliberate decision was made to include oversampling techniques that are widely recognized and used in the machine learning community. The rationale behind this choice was that these methods have been proven to be successful and efficient in previous studies, such as [10] and [11], and that they are readily accessible and available in popular machine learning libraries like scikit-learn. By incorporating these well-established oversampling techniques, this study aims to ensure that the review reflects the best practices and standards in the field.

Oversampling approaches generate synthetic minority class samples and combine them with the existing dataset, resulting in a new dataset that is more appropriate for training. The most intuitive form of oversampling is random oversampling, where minority class samples are randomly selected and duplicated without any specific selection standard.

Random oversampling can be effective for machine learning algorithms influenced by skewed distributions in instances where the overall size of the dataset is small and the imbalance is not that significant [9]. However, in cases where the dataset is heavily imbalanced, or the number of minority class samples is insufficient to train a decent classifier, random oversampling can risk classifier overfitting during training due to repeated duplication of the minority samples. Despite the implementation simplicity and fast execution, which is ideal for large and complex datasets, the lack of generalizability and high likelihood of overfitting in random oversampling has led researchers to look for more robust (robustness is denoted as the ability to oversample without introducing any bias in this context) alternative oversampling techniques.

When exploring literature on oversampling, the most widely used techniques in the scientific community are SMOTE [12] and its variants. SMOTE stands for Synthetic Minority Oversampling Technique, where the algorithm generates a synthetic sample along the line segment that joins a randomly selected minority class sample and one of its K nearest neighbors. In SMOTE, the value of K is a parameter that should be specified prior to its application, and minority class samples are randomly chosen from the set of K-Nearest Neighbors based on the amount of oversampling required. The operation of the SMOTE algorithm is further elaborated in Fig. 2, where (a) The majority class and minority class samples are represented in blue and green colors, respectively. (b) A minority class sample is randomly selected (black), and its K-nearest neighbors (3 in the image) are selected. (c) A new synthetic sample (red) is generated on the line that joins the randomly selected minority class sample and its nearest neighbor.

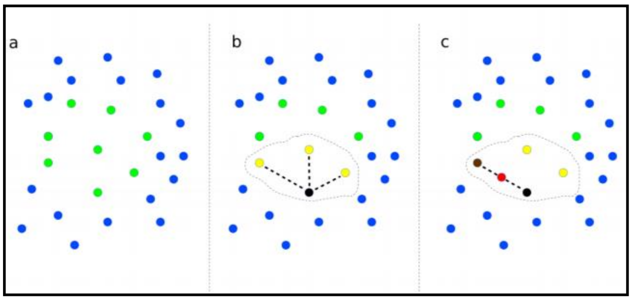

Fig. 2 Graphical representation of SMOTE algorithm [13]

As the synthetic samples generated by SMOTE are not duplicates of already existing samples, they are more generalizable than samples generated through random oversampling, reducing the risk of overfitting. However, due to the random selection of minority class samples with a uniform probability for oversampling, densely populated minority class areas become more condensed while sparsely populated minority class areas remain sparse. This behavior of SMOTE manages to address the between-class imbalance (imbalance between multiple classes), while the within-class imbalance (multiple dense or sparse regions of the same class) is ignored. Another drawback of the SMOTE algorithm is the generation of noisy samples. If a new synthetic sample is generated between an existing noisy sample and its nearest neighbor, there is a high probability that the newly generated sample will also be noisy. This is because the SMOTE algorithm has no notion of overlapping class regions when generating synthetic samples [1][4]. Throughout the years, the SMOTE algorithm has been modified to address its drawbacks and limitations.

In [14], Batista et al. apply SMOTE to oversample the minority class, followed by applying Tomek Links to increase the class separation near the decision boundary. The authors state that the class clusters are sometimes ill-defined during oversampling as the minority class samples may enter the majority class area and that interpolating minority class instances can enlarge the minority cluster, introducing noisy minority samples deep in the majority area, which is harmful and can lead to overfitting. Tomek Links act as a data cleaning mechanism in this technique, where overlapping majority and minority class samples are removed to form well-defined class clusters. This can be considered a hybrid technique as it applies both oversampling and undersampling to the dataset.

The research work presented in [15] is another hybrid technique where extensive data cleaning based on misclassifications is applied through ENN (Edited Nearest Neighbor) on an oversampled dataset. ENN is similar to Tomek Links, but it is more aggressive as it removes any sample (majority or minority) from the training set that its three nearest neighbors misclassify, creating more distinguishable class spaces with clear separation along the decision boundary. The study also states that oversampling strategies lead to more accurate classifiers than strategies derived through undersampling.

Geometric SMOTE [16] is another extension of SMOTE that generates synthetic samples near selected minority class samples in a geometric region instead of linear interpolation. While this selected region is a hyper-sphere in its default configuration, G-SMOTE deforms it to a hyper-spheroid and eventually to a line segment, simulating the SMOTE process in the last instance. Geometric SMOTE addresses two main issues in SMOTE: generation of noisy samples and generation of samples that belong to the same sub-cluster. The above issues are addressed by identifying safe areas to synthesize new samples and varying the number of minority samples generated. The authors claim that the ability of G-SMOTE to produce a variety of synthetic minority data in safe regions of the input space while aggressively boosting their diversity is the rationale for its performance gain.

Safe-Level-SMOTE [17] follows a similar approach to SMOTE but considers the nearby majority class samples when generating synthetic minority class samples. Safe levels are computed using nearest neighbor minority samples, and synthetic samples are generated such that they lie closer to minority class samples (safe area). The study tries to address the overgeneralization problem encountered by SMOTE due to arbitrary generalization of the minority class territory neglecting the majority class, which can lead to an increased likelihood of class mixing in the case of highly skewed class distributions.

Borderline-SMOTE [18] is another variation of SMOTE that generates synthetic minority class samples only within the decision boundary that separates the classes. In contrast to SMOTE, Borderline-SMOTE identifies minority class samples that lie within the vicinity of the majority class samples and prevents the generation of noisy synthetic samples based on those. The authors declare that most classification algorithms strive to understand the boundaries of each class as precisely as possible during the training process to obtain a better prediction, making the samples far from the borderline less significant compared to the samples that lie within the vicinity of the class borders. Furthermore, the study presents two versions of Borderline-SMOTE, Borderline-SMOTE1, which generates new synthetic samples between borderline minority samples and its K-nearest minority neighbors, and Borderline-SMOTE2, which generates new synthetic samples between borderline minority samples and its K-nearest minority as well as K-nearest majority neighbors.

ADASYN [19] is a density-based oversampling technique where the density of minority samples in a neighbourhood is considered when generating new synthetic minority class samples. The main intuition of ADASYN is to utilize a density distribution as a criterion to determine the number of new synthetic samples that should be generated for each minority sample. The density distribution considers the learning difficulty of each of the minority class samples and generates more synthetic samples around samples that are more difficult to learn than those that are simpler to learn. Even though ADASYN is capable of enhancing hard-to-learn minority sample areas, it is sensitive to outliers because of the possibility of misinterpreting noisy samples, which usually occur in low densities, as harder-to-learn samples, associating them with higher weights. A summary of SMOTE and its variants elaborated above are presented in Table I.

When examining the process in which the aforementioned methods have approached the problem, it is evident that they are focused on balancing the number of samples in the dataset classes. The imbalance between the dataset classes that split them into majority and minority classes is called the between-class imbalance. By default, all the resampling techniques are designed to address the between-class imbalance through oversampling, undersampling, or hybrid sampling. However, when comparing with vanilla SMOTE, it can be observed that most of the above techniques attempt in refining the output of the SMOTE algorithm by regulating the areas of sample generation and eliminating noisy synthetic samples to preserve the decision boundary that separates the classes.

The samples near the decision boundary undoubtedly represent the most crucial samples for any classification task. Despite the importance of the decision boundary, as depicted in Fig. 3 (B), the samples generated near the boundary through oversampling often tend to distort the class separation, generating noisy samples that overlap with the majority class samples. The reason for the generation of noisy samples in the decision boundary is caused by the use of the same sample generation strategy throughout the data space, which is not designed to preserve the decision boundary. The oversampling techniques mentioned above address this issue and emphasize preserving the decision boundary when generating new synthetic minority class samples.

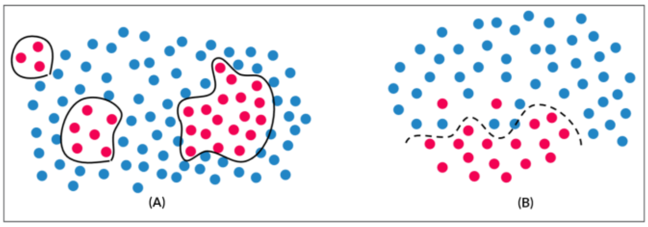

Fig. 3 (A) Occurrence of multiple disjuncts of minority class samples with varying densities. (B) Noisy minority class samples distort the decision boundary by overlapping with majority class samples

Moreover, when considering real-life datasets, there can also be instances where multiple dense or sparse clusters of minority class samples are present within the data distribution, as illustrated in Fig. 3 (A).

The existence of multiple disjuncts of minority class samples is referred to as the within-class imbalance, and it can lead to an extreme lack of representation of crucial minority class features. Oversampling techniques that randomly select minority samples to generate new synthetic samples, such as SMOTE, fail to resolve the within-class imbalance, resulting in a skewed minority class distribution

[20]. Therefore, it is important to address both between-class and within-class imbalances when addressing the data imbalance. The simultaneous removal of both these imbalances minimizes the classifier bias toward bigger sub-clusters by decreasing the influence of the bigger sub-cluster error on the total error [21].

TABLE I
SUMMARY OF SMOTE AND ITS VARIANTS

Reference	Approach	Summary
[12]	*Oversampling:* combines random sampling with the K nearest neighbor algorithm.	- A minority class sample is selected at random. - A synthetic sample is generated on the line that joins the random minority sample and its nearest neighbor.
[14]	*Hybrid Sampling:* refines the output of SMOTE by applying Tomek Links.	- Applies SMOTE followed by Tomek Links. - Tomek Links remove overlapping majority and minority samples. - Increase the class separation near the decision boundary.
[15]	*Hybrid Sampling:* refines the output of SMOTE by applying Edited Nearest Neighbor (ENN).	- Extensive data cleaning based on misclassification. - Removes any sample that its 3 nearest neighbors misclassify. - ENN is more aggressive than Tomek Links.
[16]	*Oversampling:* modifies the sample generation strategy of SMOTE by identifying safe regions.	- Generates samples in a geometric region instead of linear interpolation. - Prevents the generation of noisy samples by identifying safe areas and varying the number of samples generated.
[17]	*Oversampling:* modifies the sample generation strategy of SMOTE by computing safe levels.	- Considers nearby majority class samples when generating new synthetic minority class samples. - Safe-levels computed using nearest neighbor minority samples.
[18]	*Oversampling:* modifies SMOTE to generate new samples only within the decision boundary.	- Generates minority class samples only within the decision boundary. - Ignores minority class samples that lie within the majority class samples during synthesis.
[19]	*Oversampling:* uses the density around minority class samples to determine the number of synthetic samples to be generated.	- Density of minority class samples in a neighborhood is considered when generating new samples. - Heavily sensitive to outliers.

Further looking into oversampling techniques reveals another set of studies that use a different strategy to deal with the data imbalance problem.

MWMOTE [22] is a popular SMOTE-based oversampling technique. It first locates hard-to-learn minority class samples (samples near the decision boundary) using the majority class samples near the decision boundary and uses the Euclidean distance from these nearest majority class samples to assign them weights. This weighing mechanism ensures that higher weights are assigned to samples closer to the decision boundary than others. The authors highlight the fact that the presence of within-class imbalance and small disjuncts of the minority class can lead to performance degradation in classifiers and, therefore, similar to the weighing of hard-to-learn minority samples near the decision boundary, the samples of smaller clusters are given greater weights to reduce the within-class imbalance. Finally, a modified hierarchical clustering approach is used to create synthetic samples from the weighed minority class samples making sure the generated samples reside within the minority class region to avoid noisy sample generation.

Cluster SMOTE [23] uses K-means to cluster the minority class and applies SMOTE within the identified clusters. This approach makes sure that the generated synthetic samples always lie inside naturally occurring clusters of the minority class samples. The study claims that the existence of a small number of minority class samples is challenging when forming decent class borders, and addressing this limitation by accurate class region and border definition would enable trivial classification. Since these class regions are unknown and impossible to infer through given data, K-means is used to approximate the minority region, followed by applying SMOTE to each identified cluster. This study is explicitly designed to address the imbalance in network intrusion datasets and only uses two intrusion datasets to evaluate.

[24] presents a clustering-based oversampling technique designed to address the within and between class imbalances,

avoiding the generation of noisy synthetic samples. Initially, the algorithm clusters the input space using K-means clustering and filters out the cluster with a higher number of minority samples for oversampling. The number of synthetic samples to be generated is then dispersed, with more samples being assigned to clusters with a low density of minority samples. Finally, SMOTE is used to obtain the required ratio of minority and majority samples in each of the filtered clusters. The authors rationalize cluster-based oversampling as one of the strategies that aim to minimize the within-class imbalance while also reducing the between-class imbalance, facilitating the oversampling technique to identify the most effective areas of the input space to generate synthetic samples.

DBSMOTE [25] is another density-based oversampling technique that uses the DBSCAN algorithm to partition the minority class samples. SMOTE is used to generate synthetic samples between the shortest path that join minority class samples with a pseudo-centroid of a minority cluster, avoiding the generation of outliers or noisy samples. As a result, synthetic samples are generated in such a way that they are dense around the centroid and are sparse further away from the centroid. The authors claim that a real-world dataset with proximate data clusters can be described by a normal distribution, dense at the centroid and sparse towards the boundary and that a classifier can correctly identify samples near the centroid as it identifies the area around the centroid as a class. Based on the above observations, DBSMOTE is designed to oversample the minority class area around the centroid because it is too sparse to be recognized by a classifier.

CURE-SMOTE [26] works by clustering the minority class samples using the CURE hierarchical clustering algorithm followed by noise and outlier removal. It then randomly generates synthetic minority class samples along the line segment that joins representative points and the center point. In CURE hierarchical clustering, each sample is assumed to represent a cluster, where local clustering is used to combine these samples to form the clusters present in the input space. The study justifies CURE hierarchical clustering, stating that it is more efficient for large datasets with varying shapes of data distributions than K-means clustering, which is only suitable for spherically distributed datasets. Further, it is stated that the combination of clustering and merging operations tends to eliminate noise with reduced complexity as it eliminates the need to remove the furthest created synthetic samples (noisy samples) after applying SMOTE.

A-SUWO (Adaptive Semi-Unsupervised Weighted Oversampling) [27] and its improved version, IA-SUWO [28], cluster the minority class samples using a semi-unsupervised hierarchical clustering approach and use the classification complexity and cross-validation of each sub-cluster to decide the optimal size to oversample. Both A-SUWO and IA-SUWO aim to generate synthetic samples near minority class instances that lie close to the decision boundary with lower densities.

[29] presents a probability-based cluster expansion oversampling technique that uses a model-based clustering mechanism (MCLUST) to identify sub-clusters present in the dataset. The method also uses K-Nearest Neighbor based noise removal prior to clustering to reduce the oversampling of noisy samples and equal posterior probability after clustering to identify the boundary of the identified sub-clusters. Finally, synthetic minority class samples are generated in the enclosed region of the class separating boundary. As suggested by the authors, the main goal of this technique is to assign equal weight to all sub-clusters of the minority class that would otherwise be overlooked due to the skewness of the distribution. The cluster/density based oversampling techniques elaborated above are summarized in Table II.

In order to address the within-class imbalance, it is necessary to identify different regions within the data space where oversampling is effective. The above studies show that clustering and density-based techniques are popular approaches that researchers use to identify such areas. After the identification of significant areas to oversample, it is possible to use traditional oversampling techniques to generate synthetic samples. The clustering-based oversampling techniques introduced above emphasize the importance of addressing the within-class imbalance when formulating oversampling techniques.

A. Oversampling High-Dimensional Data

Further inspecting the aforementioned oversampling techniques that address the data imbalance, it is evident that most of the techniques are based on clustering algorithms such as K-means, DBSCAN, and hierarchical clustering, combined with heuristics based on Euclidean distance. Therefore, the majority of these approaches rely on heuristic methods that apply in two-dimensional space (Euclidean space) when generating synthetic data, whereas practical scenarios often consist of high-dimensional data [30]. Additionally, when the number of features in the dataset (dimensionality of data) increases, the data points become sparser or farther apart (Fig. 4), making the nearest neighbor problem ill-defined [31]. This behavior is called the “curse of dimensionality” [32]. As a result, in higher dimensional space, the use of heuristics based on Euclidean distance becomes ineffective, and the assumption of well-defined clusters fails, generating noisy synthetic samples.

A common strategy that can be adopted when formulating oversampling techniques that use clustering mechanisms and heuristics based on Euclidean distance is to reduce the dimensionality of the original input space. Principal Component Analysis (PCA), Multidimensional Scaling (MDS), and Self-Organizing Maps are some common dimensionality reduction techniques practitioners use. In recent years, Self-Organizing Maps [33] based resampling techniques have been extensively explored in the community.

Self-Organizing Map based Oversampling (SOMO) [30] generates a clustered two-dimensional representation of the input space by applying the SOM algorithm. Clusters are filtered to perform oversampling by calculating the density of minority class samples in each cluster. SMOTE is applied to generate synthetic minority class samples within the filtered clusters and between neighboring clusters, addressing both within and between class imbalances. The authors have identified and addressed a few inefficiencies of existing oversampling techniques, namely, the generation of noisy instances that infiltrate the majority region, the generation of duplicate samples, and the use of heuristics based on the assumption that the input space has a simple manifold structure. SOMO is capable of generating more effective synthetic samples by investigating the manifold structure of the input space, exploiting the topology-preserving property of Self-Organizing Maps.

[1] proposes an imbalance dataset resampling technique by combining Self-Organizing Maps and Genetic Algorithms. The technique uses two Self-Organizing Maps to perform oversampling on the minority class and undersampling on the majority class. The clusters derived from Self-Organizing Maps identify the regions where majority and minority class samples are dense. The filtered clusters are then utilized to derive a set of rankings among majority and synthetic minority samples, which evaluate the positive impact of their removal or inclusion in the training data, respectively. The ideal rates of exclusion and inclusion for each accepted criterion are obtained using a Genetic Algorithm that considers the performance of a random classifier for a given training dataset in the context of imbalance classification via the fitness function. The authors claim that the capabilities of Self-Organizing Maps to preserve the distribution and topology of the input data lead to the conservation of the natural spatial

TABLE II
SUMMARY OF CLUSTER/DENSITY BASED OVERSAMPLING TECHNIQUES

Reference	**Approach**	**Summary**
[22]	*Oversampling:* combines hierarchical clustering with SMOTE to address both within and between class imbalances.	- Identifies hard-to-learn minority class samples and assigns them weights based on the nearest majority class samples. - Makes sure the generated samples fall into some minority class cluster
[23]	*Oversampling:* uses K-means to approximate the minority region, followed by applying SMOTE to each identified cluster.	- Uses K-means to cluster the minority class and applies SMOTE within the identified clusters. - Makes sure generated samples lie inside naturally occurring clusters.
[24]	*Oversampling:* uses K-means to identify and filter clusters with high minority class density, followed by applying SMOTE.	- Uses K-means to identify clusters and assigns weights based on the minority class density in each cluster. - Generates more samples in clusters with low minority class densities.
[25]	*Oversampling:* combines the DBSCAN algorithm with SMOTE to generate synthetic samples such that the dataset is dense at the centroid and sparse towards the boundary.	- Uses the DBSCAN algorithm to partition the minority class samples. - Generate synthetic samples between the lines that join minority class samples with a pseudo-centroid of a minority cluster.
[26]	*Oversampling:* combines CURE hierarchical clustering with noise and outlier removal so that the samples generated after using SMOTE are more precise.	- Uses CURE hierarchical clustering algorithm followed by noise and outlier removal. - Addresses datasets that have clusters of varying shapes and sizes.
[27][28]	*Oversampling:* uses a semi-supervised hierarchical clustering algorithm to generate synthetic samples around minority class instances that lie close to the decision boundary with lower densities.	- Clusters the minority class using a hierarchical clustering approach. - Oversample size is decided from classification complexity and cross-validation of each sub-cluster.
[29]	*Oversampling:* combines a model-based clustering mechanism with KNN-based noise removal.	- Uses MCLUST to identify sub-clusters present in the dataset. - The Goal is to assign equal weights to all minority class sub-clusters that would otherwise be overlooked due to skewness of the distribution.

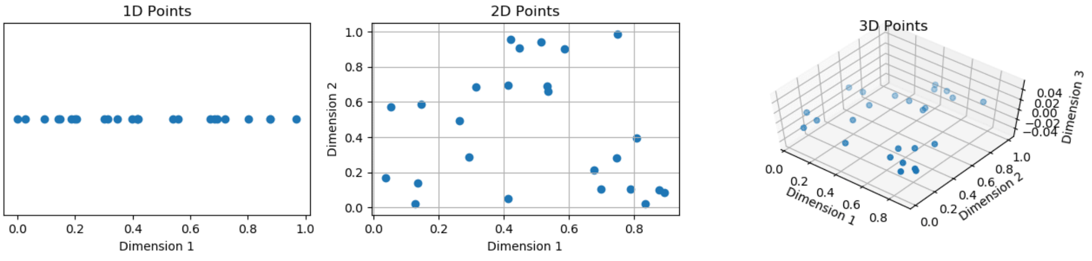

Fig. 4 Data points become sparser as dimensionality increases [34].

relationship among samples at the cluster level , and the optimization capabilities of Genetic Algorithms result in maximization of classifier performance , improving the overall resampling operation.

[35] uses a customized SOM-SMOTE algorithm to address the imbalance in clutter data when addressing the clutter suppression in search radars. The authors have identified two limitations in the SMOTE algorithm, random sample selection and ignoring the data distribution during interpolation, resulting in samples that are not representative enough. The study addresses the above limitation using a combined Self-Organizing Map and SMOTE algorithm that clusters the minority class samples into several subsets using a Self-Organizing Map and interpolates synthetic samples around the cluster centers using SMOTE. The authors also highlight the ability of Self-Organizing Maps to preserve the topology of higher dimensional clutter data, resulting in synthetic samples with distribution characteristics similar to original data.

From the above studies, it can be assumed that the ability of Self-Organizing Maps to address the within-class imbalance as a clustering algorithm, along with the ability to reduce the dimensionality of data while preserving the topology of the input space, are the main reasons for its widespread popularity as an excellent candidate to address the data imbalance problem.

III. Discussion

When analyzing oversampling techniques that address the class imbalance problem, it is possible to identify key factors that contribute to the success of an oversampling technique. Throughout the literature, it can be observed that every oversampling technique attempts to adopt one or more of these factors during its strategy formulation.

Considering the variations of the SMOTE algorithm that have been introduced as improved versions of vanilla SMOTE [12], it is evident that most of the proposed techniques such as [14], [15], Safe-Level-SMOTE [17], and Borderline-SMOTE [18] try to preserve the boundary region that separates the minority and majority classes. Compared to the vanilla SMOTE, the higher classification accuracies of these techniques demonstrate the importance of preserving the boundary region when formulating an oversampling technique.

Further exploring contemporary oversampling techniques, it can also be observed that clustering-based approaches are more popular among researchers. This is because, apart from preserving the boundary region, it is also essential to address the within-class imbalance in the dataset (all the resampling techniques address the between-class imbalance by default). Clustering algorithms do not necessarily address the within-class imbalance unless they are explicitly designed to address it.

Based on the above observations, it is possible to identify three constraints that need to be simultaneously satisfied when formulating an oversampling technique to generate an optimal resampled dataset.

1) Addressing the between-class imbalance: The between-class imbalance represents the typical imbalance scenario where there is a significant difference between the number of samples in the dataset classes. All the resampling techniques attempt to address the between-class imbalance by oversampling, undersampling, or hybrid-sampling. There is no optimal imbalance ratio that needs to be reached when resampling an imbalanced dataset. However, [6] states that a 35:65 class distribution can achieve a higher classification performance compared to a 50:50 class distribution when the classes are heavily imbalanced. This is an area that is still being investigated.

2) Addressing the within-class imbalance: Within-class imbalance demonstrates the imbalance within the minority class due to the existence of multiple disjuncts of minority class samples with varying densities (Fig. 3A) that can lead to an extreme representation deficiency of essential characteristics of the minority class. The majority of the oversampling techniques that randomly select minority class samples to generate new synthetic samples, such as SMOTE, fail to address the within-class imbalance, leaving the minority class distribution skewed. When analyzing oversampling techniques capable of addressing the within-class imbalance, it can be observed that they are based on clustering approaches. The use of clustering approaches is an obvious design choice as they provide the capability to analyze the spatial location of the minority class to determine the suitable areas to generate new synthetic samples.

3) Preserving the boundary region when generating synthetic samples: The boundary region represents the area that separates two or more classes. As mentioned previously, the most crucial samples in any classification task are the samples that reside near the boundary region. When considering oversampling techniques, the synthetic samples generated near the decision boundary often distort the class separation, generating noisy samples that overlap with the majority class samples. This behavior is caused due to the use of the same sample generation strategy throughout the data space, which is not designed to preserve the decision boundary. However, studies such as [14], [15], [16], [17], and [18] emphasize the importance of preserving the boundary region when generating new synthetic samples. The decision boundary preservation can be achieved either by using a separate sample generation strategy near the boundary region or by refining the synthetically generated samples to remove noisy samples generated near the decision boundary.

Even though there are oversampling techniques that address different combinations of the above three constraints, almost all the proposed oversampling techniques do not address all three constraints together.

Aside from addressing the constraints mentioned above when formulating an oversampling approach, it is also preferable to pay special attention to the curse of dimensionality. As elaborated in the previous section, many of the currently available oversampling approaches are unable to handle the curse of dimensionality, resulting in poor performance on high dimensional datasets. We believe addressing the above constraints along with a proper clustering algorithm or a dimensionality reduction technique is a promising research avenue to investigate further.

IV. Conclusion

The data imbalance problem is one of the most well-defined problems in the Machine Learning domain that has been addressed throughout the past decades. With the emergence of Big Data, traditional techniques to address the data imbalance problem have been challenging, and the necessity of new and improved techniques to address the imbalance has created promising research avenues in many practical domains.

This paper reviews numerous research work that has attempted to address the data imbalance problem by oversampling the minority class samples. We identify several subsets of oversampling techniques and highlight different approaches adopted by them to discover suitable samples/areas to oversample and strategies used to generate new synthetic samples. Based on these studies it is evident that some oversampling techniques focus on preserving the decision boundary by refining the oversampled output or by restricting sample generation in certain areas. It is also possible to identify studies that use clustering and density-based techniques to prevent the generation of noisy samples and alleviate the occurrence of disjuncts of minority class samples with varying densities. Furthermore, the review also presents the challenges faced by traditional oversampling techniques on high-dimensional data and suggest different techniques that can be utilized to address them.

By analyzing various strategies adopted in the scientific community for oversampling, we have identified three key constraints that need to be satisfied when developing state-of-the-art oversampling techniques,

1) Addressing the between-class imbalance: represents the typical imbalance scenario where there is a significant difference between the number of samples in the dataset classes.

2) Addressing the within-class imbalance: represents the imbalance within the minority class due to the existence of multiple disjuncts of minority class samples with varying densities.

3) Preserving the boundary region when generating synthetic samples: the boundary region represents the area that separates two or more classes. It is required to make sure that the synthetic samples do not distort the decision boundary and overlap with samples in other classes.

Along with the above constraints, being attentive to the curse of dimensionality and addressing it would lead to a more optimal resampling. Based on these findings, researchers can develop more robust oversampling techniques in the future.

References

[1] M. Vannucci and V. Colla, Imbalanced datasets resampling through self organizing maps and genetic algorithms. Springer International Publishing, 2019, vol. 1000. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-20257-6_34

[2] S. Maheshwari, J. Agrawal, and S. Sharma, “A New approach for Classification of Highly Imbalanced Datasets using Evolutionary Algorithms,” International Journal of Scientific & Engineering Research, vol. 2, no. 7, 2011. [Online]. Available: http://www.ijser.org

[3] T. Jo and N. Japkowicz, “Class imbalances versus small disjuncts,” ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 40–49, 2004.

[4] V. García, J. S. Sánchez, A. I. Marqués, R. Florencia, and G. Rivera. 2020. Understanding the apparent superiority of over-sampling through an analysis of local information for class-imbalanced data. Expert Systems with Applications 158, December (2020). https://doi.org/10.1016/j.eswa.2019.113026

[5] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, “An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics,” Information Sciences, vol. 250, pp. 113–141, Nov. 2013, doi: 10.1016/j.ins.2013.07.007.

[6] T. M. Khoshgoftaar, M. Golawala, and J. V. Hulse. 2007. An empirical study of learning from imbalanced data using random forest. Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI 2 (2007), 310–317. https://doi.org/10.1109/ICTAI.2007.46

[7] U. Bhowan, M. Johnston, and M. Zhang. 2012. Developing New Fitness Functions in Genetic Programming for Classification With Unbalanced Data. 42, 2 (2012), 406–421.

[8] J. Gao, K. Liu, B. Wang, D. Wang, and Q. Hong. 2021. An improved deep forest for alleviating the data imbalance problem. Soft Computing 25, 3 (2021), 2085–2101. https://doi.org/10.1007/s00500-020-05279-8

[9] R. Mohammed, J. Rawashdeh, and M. Abdullah. 2020. Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. 2020 11th International Conference on Information and Communication Systems, ICICS 2020 May (2020), 243–248. https://doi.org/10.1109/ICICS49469.2020.239556

[10] D. Elreedy and A. F. Atiya, “A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance,” Information Sciences, vol. 505, pp. 32–64, Dec. 2019, doi: https://doi.org/10.1016/j.ins.2019.07.070.

[11] A. Gosain and S. Sardana, “Handling class imbalance problem using oversampling techniques: A review,” IEEE Xplore, Sep. 01, 2017. https://ieeexplore.ieee.org/abstract/document/8125820 (accessed May 17, 2022).

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16, 2 (jun 2002), 321–357. https://doi.org/10.1613/jair.953

[13] M. Schubach, M. Re, P. N. Robinson, and G. Valentini. 2017. Imbalance-aware machine learning for predicting rare and common disease-associated non-coding variants. Scientific reports 7, 1 (2017), 1–12.

[14] G. E. A. P. A. Batista, A. L. C. Bazzan, and M. C. Monard. 2003. Balancing Training Data for Automated Annotation of Keywords: a Case Study. In Proceedings of the Second Brazilian Workshop on Bioinformatics January (2003), 35–43. http://www.cs.waikato.ac.nz/

[15] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6, 1 (2004), 20–29. https://doi.org/10.1145/1007730.1007735

[16] G. Douzas and F. Bacao. 2017. Geometric SMOTE: Effective oversampling for imbalanced learning through a geometric extension of SMOTE. (2017), 1–22. http://arxiv.org/abs/1709.07377

[17] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. 2009. Safe-level-SMOTE: Safe-level-synthetic minority oversampling technique for handling the class imbalanced problem. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5476 LNAI (2009), 475–482. https://doi.org/10.1007/978-3-642-01307-2_43

[18] H. Han, W. Y. Wang, and B. H. Mao. 2005. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In Advances in Intelligent Systems and Computing. Vol. 683. 878–887. https://doi.org/10.1007/11538059_91

[19] H. He, Y. Bai, E. A. Garcia, and S. Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. Proceedings of the International Joint Conference on Neural Networks 3 (2008), 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

[20] C. T. Lin, T. Y. Hsieh, Y. T. Liu, Y. Y. Lin, C. N. Fang, Y. K. Wang, G. Yen, N. R. Pal, and C. H. Chuang. 2018. Minority Oversampling in Kernel Adaptive Subspaces for Class Imbalanced Datasets. IEEE Transactions on Knowledge and Data Engineering 30, 5 (2018), 950–962. https://doi.org/10.1109/TKDE.2017.2779849

[21] S. A. Shahee and U. Ananthakumar. 2018. An adaptive oversampling technique for imbalanced datasets. Vol. 10933 LNAI. Springer International Publishing. 1–16 pages. https://doi.org/10.1007/978-3-319-95786-9_1

[22] S. Barua, M. M. Islam, X. Yao, and K. Murase. 2014. MWMOTE - Majority weighted minority oversampling technique for imbalanced data set learning. IEEE Transactions on Knowledge and Data Engineering 26, 2 (2014), 405–425. https://doi.org/10.1109/TKDE.2012.232

[23] D. A. Cieslak, N. V. Chawla, and A. Striegel. 2006. Combating imbalance in network intrusion datasets. 2006 IEEE International Conference on Granular Computing JANUARY 2006 (2006), 732–737. https://doi.org/10.1109/grc.2006.1635905

[24] F. Last, G. Douzas, and F. Bacao. 2017. Oversampling for Imbalanced Learning Based on K-Means and SMOTE. (2017), 1–19. https://doi.org/10.1016/j.ins.2018.06.056

[25] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. 2012. DBSMOTE: Density-based synthetic minority over-sampling technique. Applied Intelligence 36, 3 (2012), 664–684. https://doi.org/10.1007/s10489-011-0287-y

[26] L. Ma and S. Fan. 2017. CURE-SMOTE algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. BMC Bioinformatics 18, 1 (2017), 1–18. https://doi.org/10.1186/s12859-017-1578-z

[27] I. Nekooeimehr and S. K. L. Yuen. 2016. Adaptive semi-unsupervised weighted oversampling (A-SUWO) for imbalanced datasets. Expert Systems with Applications 46 (2016), 405–416. https://doi.org/10.1016/j.eswa.2015.10.031

[28] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang. 2020. IA-SUWO: An Improving Adaptive semi-unsupervised weighted oversampling for imbalanced classification problems. Knowledge-Based Systems 203, June (2020), 106116. https://doi.org/10.1016/j.knosys.2020.106116

[29] S. A. Shahee and U. Ananthakumar. 2018. Probability Based Cluster Expansion Oversampling Technique for Imbalanced Data. (2018), 77–90. https://doi.org/10.5121/csit.2018.80607

[30] G. Douzas and F. Bacao. 2017. Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning. Expert Systems with Applications 82 (2017), 40–52. https://doi.org/10.1016/j.eswa.2017.03.073

[31] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, C. Beeri, and P. Buneman. 1999. When Is ”Nearest Neighbor” Meaningful?, 217–235 pages. http://www.springerlink.com/content/04p94cqnbge862kh/

[32] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 1973 (2001), 420–434. https://doi.org/10.1007/3-540-44503-x_27

[33] T. Kohonen. 1982. Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, 1 (1982), 59–69. https://doi.org/10.1007/BF00337288

[34] N. B. Subramanian, “Why High Dimensional Data are a Curse?”, https://aiaspirant.com/curse-of-dimensionality (accessed June 4, 2022).

[35] X. Zhang, W. Wang, X. Zheng, Y. Ma, Y. Wei, M. Li, and Y. Zhang. 2019. A Clutter Suppression Method Based on SOM-SMOTE Random Forest. In 2019 IEEE Radar Conference (RadarConf). IEEE, 1–4. https://doi.org/10.1109/RADAR.2019.8835836