



Deep generative approaches for oversampling in imbalanced data classification problems: A comprehensive review and comparative analysis

Mozafar Hayaian Shirvan^a, Mohammad Hossein Moattar^{a,*}, Mehdi Hosseinzadeh^{b,c,**}

^a Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

^b School of Computer Science, Duy Tan University, Da Nang, Vietnam

^c Jadara University Research Center, Jadara University, Irbid, Jordan

HIGHLIGHTS

- The article investigates oversampling strategies for imbalanced data classification.
- Compares between deep methodologies and conventional oversampling techniques.
- Identifies of difficulties and restrictions associated with applying deep approaches.
- Also, recommends future research directions and addresses challenges in the problem.

ARTICLE INFO

Keywords:

Imbalanced Data Classification
Oversampling
Deep generative models
Generative adversarial networks
Variational autoencoders

ABSTRACT

There are inherent issues with classifying imbalanced data, especially in classifying minority class samples. With an emphasis on the use of deep generative methodologies, this study offers a thorough investigation of oversampling strategies for imbalanced data classification. This paper begins with a summary of unbalanced data categorization and the need for oversampling techniques. Then traditional approaches including SMOTE, ADASYN, and random oversampling are introduced and discussed. This study then discusses deep generative models and how oversampling may be used to address imbalanced data problem using Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). A comparative study between deep generative and conventional oversampling techniques is performed concerning a comprehensive evaluation of the difficulties, restrictions, and possible risks associated with applying deep generative approaches. The paper concludes with recommendations for future researches and highlights the need for addressing challenges in oversampling approaches for imbalanced data classification.

1. Introduction

Imbalanced data classification refers to machine learning tasks where the distribution of classes in the dataset is uneven, with far fewer examples belonging to one minority class compared to other majority classes [1,2]. This imbalance poses significant challenges for training accurate classifiers, as most algorithms optimize for overall accuracy which biases models towards the majority class [3], ignoring or misclassifying the under-represented minority [4]. The two basic strategies for dealing with imbalanced data classification are classifier-level and data-level approaches. To address this issue, classifier-level strategies

use ensemble methods and cost-sensitive learning, whereas data-level solutions mostly use resampling methodologies and synthetic data generation.

Data-level solutions to address the class imbalance usually involve resampling the dataset [5], either by undersampling the majority classes or oversampling the minority class, to balance the class distribution [6,7]. Recently, deep generative models like generative adversarial networks (GANs) [8,9,10,11] and variational autoencoders (VAEs) [12] have emerged as a promising approach for oversampling by synthetically generating new minority class examples [13,14,15]. These models can capture multidimensional distributions of classes from limited data

* Corresponding author.

** Corresponding author at: School of Computer Science, Duy Tan University, Da Nang, Vietnam.

E-mail addresses: Hayaianmozafar@gmail.com (M. Hayaian Shirvan), moattar@mshdiau.ac.ir (M.H. Moattar), mehdihosseinzadeh@duytan.edu.vn (M. Hosseinzadeh).

<https://doi.org/10.1016/j.asoc.2024.112677>

Received 18 March 2024; Received in revised form 11 November 2024; Accepted 22 December 2024

Available online 3 January 2025

1568-4946/© 2025 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

and produce diverse realistic samples [1,2,16]. In addition to these approaches, a novel method called Generative Adversarial Network Synthesis for Oversampling (GANSO) [17] has been proposed, which integrates the concept of vector Markov Random Fields (vMRF) with the adversarial learning framework of GANs to synthesize new samples. By incorporating structural information through the vMRF model, GANSO aims to generate realistic minority class instances, even in scenarios with extreme data scarcity.

Compared to conventional oversampling techniques like SMOTE [4] which interpolate between minority examples, deep generative oversampling can create more varied data points while preserving intrinsic data characteristics [1,7,18]. GANSO, in particular, leverages the vMRF model to capture relevant structural information and generate synthetic instances that maintain the original data properties [17]. Augmenting real training data with such synthesized minority examples helps classifiers better learn patterns of under-represented classes [19]. Experiments have shown improved classifier performance on imbalanced datasets after oversampling them using deep generative models across domains like image analysis [20], network security [11,21] medical diagnosis [22,23] and industrial defect detection [24].

However, deep generative models have some limitations. GANs struggle with training stability and mode collapse [25,26] generating limited sample diversity. VAEs tend to overfit to data and exhibit posterior collapse [12], affecting generalization. Hybrid models like VAE-GANs [27] and medGAN [22] aim to address these issues but add architectural complexity [28]. In general, challenges remain in effectively covering minority class data distributions, especially with extreme imbalance ratios [29]. Carefully designed evaluation schemes are also needed to truly demonstrate improved real-world performance [30]. Recent focus has been on conditional architectures for guided generation [31], attention mechanisms [24], and architectural innovations [32] to improve sample quality and diversity. Domain specific tailored solutions have also emerged, like 3D-HyperGAMO for spectral-spatial satellite image data [33].

In addition to alleviating the class imbalance problem, deep generative models have shown promise in addressing the broader challenge of data scarcity, where there is an insufficient number of training samples across all classes, not just the minority class. The extent to which these deep generative approaches can be applied to scenarios of general data scarcity, beyond just class imbalance, is an important consideration. Techniques like proxy learning curve analysis can be used to quantify the degree of data scarcity and estimate the required sample sizes for each class [34]. By exploring the application of deep generative methods to data-scarce regimes, regardless of class distributions, their true capabilities in augmenting limited datasets can be better understood.

In this paper, we provide a comprehensive review of various minority oversampling [35] methods using deep generative models including GANs, VAEs, conditional GANs (CGANs), conditional VAEs (CVAEs), Balancing GAN (BAGAN), Adversarially Regularized Autoencoder (ARAE), Gaussian Generative Adversarial Network (G-GAN), Minority Oversampling Generative Adversarial Network (MoGAN), 3D-HyperGAMO framework, Improved VAEGAN, and Generative Adversarial Network Synthesis for Oversampling (GANSO). Their relative strengths and weaknesses are analyzed compared to traditional oversampling techniques like SMOTE, ADASYN and random oversampling in terms of the ability to effectively capture characteristics of the rare minority data for intelligent diverse sample generation. This assists with tackling the significant problem of imbalanced class distributions in training datasets across a variety of domains [36].

2. Imbalanced data classification

Learning from uneven, skewed data where one or more classes have significantly fewer samples poses an innate challenge for machine learning models [37,38,39,40]. Besides accuracy, metrics like specificity, sensitivity, precision and recall get affected for under-represented

classes [41]. A labeled dataset is defined as $(\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_i, y_i), i = 1, 2, \dots, n\}$, where \mathbf{x}_i represents the features of the i th observation and $y_i \in \{1, 2, \dots, L\}$ gives its class label, with L being the total number of classes. In the labeled dataset (\mathbf{X}, \mathbf{Y}) , ρ_{ℓ} is defined as the fraction of samples belonging to class ℓ :

$$\rho_{\ell} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{(y_i=\ell)}, \ell = 1, 2, \dots, L,$$

where $\mathbf{1}_{(A)}$ is the indicator function of the event A , and n is the total number of samples.

The dataset (\mathbf{X}, \mathbf{Y}) is considered imbalanced if:

$$\min_{\ell} \rho_{\ell} \ll \max_{\ell} \rho_{\ell}.$$

This inequality highlights that the minimum fraction of samples for one class (indexed by ℓ) is much smaller than the maximum fraction for another class (indexed by \mathcal{J}). The severity of imbalance is described by the fraction of minority classes $\frac{|\mathcal{m}|}{L}$ and the imbalance ratio (IR), defined as:

$$\text{IR} = \frac{\max_{\ell} \rho_{\ell}}{\min_{\ell} \rho_{\ell}}.$$

The core issue is that most algorithms inherently maximize overall accuracy, virtually ignoring minority categories during training [42,43]. Both data-level and algorithm-level solutions attempt addressing this long-standing problem [44,45,46].

Earlier works focused extensively on sampling-based remedies involving replicating or removing training instances [46,47]. Under-sampling diminishes samples from majority classes through random elimination. But it risks losing vital characteristics [48]. Simple random oversampling augments minority data by duplicating samples identical to existing ones. However, identical replicated data hardly provides new information to models, often making them prone to only memorizing training instances [49,50,51].

Oversampling techniques like SMOTE interpolate synthetic new data points between existing minority samples rather than just making extra copies [52,53]. This expands the minority class decision boundary rather than overfitting limited points. Borderline-SMOTE and ADASYN further guide oversampling to emphasize complex neighborhoods with significant class overlaps [54,55]. However, a key limitation is that all interpolation methodologies only consider local relationships during sample generation rather than capturing global distributions [56,57].

The latest breakthroughs in deep generative modeling allow approximating actual minority class distributions instead of just expanding sample clusters [58,59]. Models like generative adversarial networks (GANs) and variational autoencoders (VAEs) can realistically simulate data characteristics rather than simply extra/interpolating limited samples [27,60,61]. Conditional variants facilitate targeted generation for designated difficult minority classes [62,63]. Augmenting real data with such synthetically oversampled global data characteristics can significantly improve model generalization capability over localized oversampling [2,64,65].

3. Oversampling techniques

Oversampling refers to the technique of increasing the number of samples from the minority class in an imbalanced dataset to balance out the class distribution. It is a data-level approach to tackle the problem of learning from uneven, skewed data where one class is under-represented compared to other classes [1,66,67].

By replicating or generating additional synthetic minority class samples and combining them with the original dataset, oversampling reduces the extent of imbalance in the training data fed to machine learning models. This enables classifiers to better learn the patterns of minority classes that may have been previously ignored due to their

small sample size [2,68]. The rest of this section continues with a short introduction on the main approaches in this regard.

In the past two decades, oversampling techniques have gone through a revolution as depicted in Fig. 1. This revolution can be split into two periods: the Traditional Methods Era (2000–2014) and Deep Generative Models Era (2014–present). The traditional era commenced with the use of rudimentary methods such as Random sampling methods and progressed to the use of more advanced methods such as SMOTE [4] and ADASYN [55], which exploited interpolation and density based synthetic sample generation approaches. The field witnessed a drastic transformation with the introduction of deep generative models starting from 2014 beginning with GANs and VAEs. However, that period also ushered in more complex approaches such as CGAN and more advanced architectures such as BAGAN [86], ARAE [88] and medGAN [22]. More recently, GANSO [17], MoGAN [95], 3D-HyperGAMO [33], and Improved VAEGAN [32] have improved these approaches by addressing some domain and architectural specific issues. These methods and how they helped solve the class imbalance problem are mentioned in the following sections.

3.1. Synthetic minority oversampling technique (SMOTE)

SMOTE generates new synthetic minority data points by interpolating between several nearest minority class neighbors but it focuses only on minority data and ignores class overlaps [4].

The SMOTE algorithm first identifies the minority class examples in the imbalanced dataset. For each minority class sample, it calculates the K_{NN} nearest minority class neighbors [69]. The parameter K_{NN} is typically set to 5. This uses a distance metric like Euclidean distance between two samples in feature space. Next, the algorithm randomly chooses one of those K_{NN} minority class neighbors. Then, it creates a new synthetic data point along the line segment joining the current minority sample and its selected neighbor. This synthetic point will have attribute values interpolated between the two points based on a random number between 0 and 1. This process repeats for each minority class sample as needed, generating new synthetic neighbors until the minority class is sufficiently oversampled to achieve a desired level of balance with the majority class.

The key insight is that rather than arbitrarily replicating minority class samples, SMOTE creates new examples based on characteristics of existing minority points. This helps the decision regions associated with the minority class examples to grow larger and less specific [57,70].

3.2. Adaptive synthetic sampling (ADASYN)

ADASYN (Adaptive Synthetic Sampling Approach for Imbalanced Learning) [55] is an algorithm for generating synthetic minority class examples to balance imbalanced datasets. The key idea behind ADASYN is to adaptively generate more synthetic data samples for those minority examples that are harder to learn, according to their distributions. In that sense, ADASYN is similar to SMOTE – both utilize synthetic data generation to overcome class imbalance. However, SMOTE produces the same number of synthetic samples for each minority example, while ADASYN decides the number of synthetic examples based on the distribution density \hat{r}_i of each minority example i , generating more examples for “difficult” cases. As such, ADASYN not only reduces bias resulting from the imbalanced data distribution, but also enables adaptive shifting of the decision boundary toward difficult examples.

The algorithm takes as input the imbalanced training dataset D_{tr} with m examples $\{x_i, y_i\}$, where x_i is a data instance, $y_i \in Y = \{1, -1\}$ is the class label, m_s is the number of minority class examples, and m_l is the number of majority class examples. Then, the algorithm proceeds with the following steps:

1. Calculate degree of class imbalance according to Eq. (1) [55]:

$$d = m_s/m_l \quad (1)$$

where $d \in (0, 1]$.

2. If $d < d_{th}$ (where d_{th} is a preset threshold):

- a) Calculate number of synthetic minority samples to generate based on Eq. (2) [55]:

$$S_{syn} = (m_l - m_s) \times \beta \quad (2)$$

here, $\beta \in [0, 1]$ controls desired balance level after generation.

- b) For each minority example x_i , find K_{NN} nearest neighbors and calculate ratio r_i using Eq. (3) [55]:

$$r_i = \Delta_i/K_{NN}, \quad i = 1, \dots, m_s \quad (3)$$

where Δ_i is number of majority class neighbors.

- c) Normalize r_i as \hat{r}_i distribution:

$$\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$$

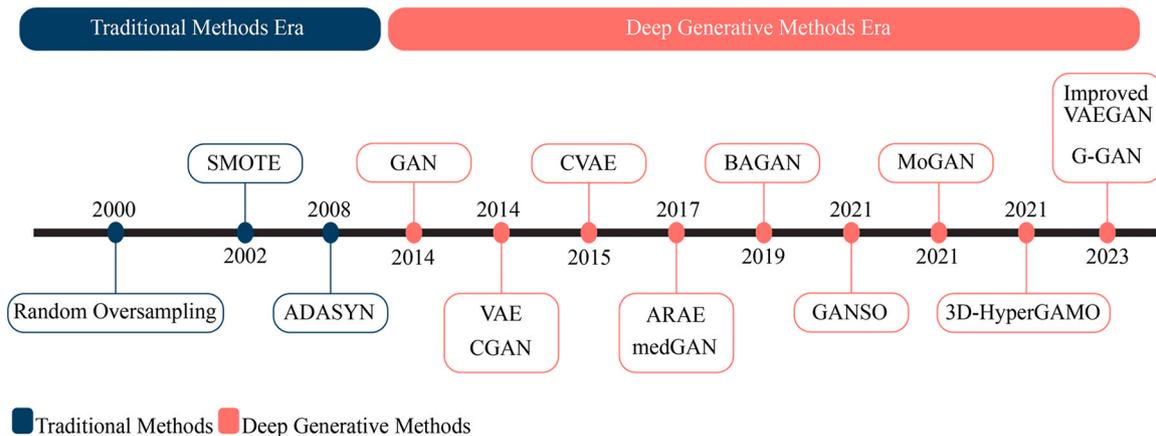


Fig. 1. Research Timeline of Oversampling Evolution from Traditional to Deep Generative Approaches (2000–2024): Timeline showing the progression of oversampling techniques from Traditional Methods Era to Deep Generative Models Era. The evolution demonstrates the transformation from basic approaches like Random Oversampling and SMOTE to sophisticated deep learning-based methods such as GANs, VAEs, and their variants for handling imbalanced data classification problems.

- d) Compute synthetic samples to generate for each \mathbf{x}_i based on Eq. (4) [55]:

$$\mathbf{g}_i = \hat{r}_i \times S_{\text{syn}} \quad (4)$$

- e) Generate g_i synthetic samples for each \mathbf{x}_i :

- Randomly pick a neighbor \mathbf{x}_{zi}
- Generate sample according to Eq. (5) [55]:

$$\mathbf{s}_i = \mathbf{x}_i + (\mathbf{x}_{zi} - \mathbf{x}_i) \times \lambda \quad (5)$$

- where λ is a random number in [0,1].

So, in summary, ADASYN decides the number of synthetic samples per minority example adaptively based on density distribution \hat{r}_i to reduce bias and also shift decision boundary focus towards difficult examples. The key parameters are K_{NN} , β , d_{th} and λ .

The adaptive generation of synthetic training examples enables ADASYN to reduce bias [56], focus more on difficult minority cases, and improve learning performance for imbalanced classification providing some advantages over algorithms like SMOTE.

3.3. Random oversampling

Random oversampling [4] is a simple method to deal with the class imbalance problem in the training data. In this method, the minority class examples (e.g. positive class) are randomly selected and duplicated to sufficiently increase their numbers, making the class distribution more balanced [71].

A common enhancement to random oversampling is adding a small amount of noise, typically Gaussian, to the replicated samples. This technique improves the diversity of the augmented dataset, helping to mitigate some of the limitations associated with simple duplication.

The main differences between random oversampling and advanced algorithms like SMOTE and ADASYN are:

1. SMOTE and ADASYN generate new synthetic samples while random oversampling just replicates the existing samples.
2. SMOTE and ADASYN generate new samples by considering the feature space neighborhood of existing minority class samples. But random oversampling works without taking into account relationships between samples.
3. SMOTE and ADASYN add new information to the dataset while random oversampling does not provide any new information, just duplication of existing data points.

In general, SMOTE and ADASYN are considered more advanced and smarter approaches compared to simple random oversampling. They try to expand the minority class decision boundary by interpolating new points between minority class samples lying close together. This forces the model to create larger and less specific decision regions, rather than smaller and more specific regions.

The main advantage of random oversampling is that it is simple and takes almost no computation. However, this advantage comes at the price of no new information being added to the model. The duplicated samples can also cause overfitting [72]. SMOTE and ADASYN are attempts to generate new information rather than just replication.

4. Deep generative approaches

Deep generative models [1,7] can learn the underlying distribution of a dataset $\mathbf{X} = \{\mathbf{x}_i, i = 1, 2, \dots, n\}$ where \mathbf{x}_i represents the features of the i th sample. The goal is to train a model that can generate new

synthetic samples resembling the original data. Two key classes of models are reviewed. Recent advances in deep generative models provide powerful new oversampling methods for imbalanced data [73,74]. Two prominent categories of models are GANs and VAEs.

4.1. Generative adversarial networks (GANs)

GANs [1,8,75] train a generator network to produce synthetic samples that are indistinguishable from real samples by a discriminator network [76,77,78]. The training process is adversarial with the two networks competing against each other [79,80,81]. GANs can capture complex distributions and generate realistic samples. Conditional GANs (CGANs) [82,83] allow conditioning the model on class labels which is useful for targeted oversampling of minority classes. The general architecture and workflow of a GAN is illustrated in Fig. 2.

GANs consist of two main components [17]. The first is a generator G_ϕ , which takes as input a random noise variable $\mathbf{z} \sim q_z(\mathbf{z})$ and aims to generate a synthetic observation \mathbf{x}' that closely resembles those belonging to \mathbf{X} . The second is a discriminator D_θ that has the task of determining whether an observation is real or one that is generated by G_ϕ . The learning objectives of G_ϕ and D_θ are thus in opposition to one another and can be written as a min-max problem, given by Eq. (6) [1]:

$$\mathcal{L}_{GAN}(\mathbf{x}; \phi, \theta) = \min_{\phi} \max_{\theta} (E_{\mathbf{x} \sim p_X(\mathbf{x})} [\log D_\theta(\mathbf{x})] + E_{\mathbf{z} \sim q_z(\mathbf{z})} [\log(1 - D_\theta(G_\phi(\mathbf{z})))]), \quad (6)$$

$\mathcal{L}_{GAN}(\mathbf{x}; \phi, \theta)$ represents the GAN loss function, where ϕ are the parameters of the generator G , and θ are the parameters of the discriminator D . E_x denotes expectation over real data distribution, E_z denotes expectation over noise distribution, p_x is the real data distribution, and q_z is the noise distribution. Once trained, synthetic observations are generated by first sampling $\mathbf{z} \sim q_z(\mathbf{z})$ and then passing this \mathbf{z} to the trained G_ϕ such that $\mathbf{x}' = G_\phi(\mathbf{z})$.

4.2. Variational autoencoders (VAEs)

VAEs [1,12,21] learn an explicit latent space representation of the input data using probabilistic encoder and decoder networks. By sampling points from the learned latent distribution and decoding, new samples can be generated from the prior training distribution. Conditional VAEs (CVAEs) enable conditioning on class labels for minority class oversampling. VAEs are premised on \mathbf{x} being generated by a random process involving a latent random variable \mathbf{z} . Specifically, the process is such that an observation of \mathbf{z} is first sampled from the prior distribution $p_\theta(\mathbf{z})$, which in turn is used to sample an observation of \mathbf{x} from the conditional distribution $p_\theta(\mathbf{x}|\mathbf{z})$.

The goal of the VAE is to obtain approximate maximum likelihood or maximum a posteriori estimates of the parameters θ in situations where both the marginal likelihood $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})d\mathbf{z}$ and the posterior $p_\theta(\mathbf{z}|\mathbf{x})$ are intractable. It does so by utilizing the distribution $q_\phi(\mathbf{z}|\mathbf{x})$ as an approximation to intractable $p_\theta(\mathbf{z}|\mathbf{x})$, and maximizing the variational lower bound for $p_\theta(\mathbf{x})$ given by Eq. (7) [1]:

$$\mathcal{L}_{VAE}(\mathbf{x}; \phi, \theta) = \max_{\phi, \theta} (E_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{K}\mathbb{L}(q_\phi(\mathbf{z}|\mathbf{x}) \parallel p_\theta(\mathbf{z}))), \quad (7)$$

where $\mathbb{K}\mathbb{L}$ denotes the Kullback-Leibler divergence, a measure of difference between two probability distributions.

Once the VAE is trained, a synthetic observation \mathbf{x}' is generated by first sampling $\mathbf{z} \sim p_\theta(\mathbf{z})$ and subsequently sampling \mathbf{x}' from the trained probabilistic decoder $p_\theta(\mathbf{x}|\mathbf{z})$.

Fig. 3 shows the architecture of a typical VAE. The encoder maps the input \mathbf{x} to a probability distribution $q_\phi(\mathbf{z}|\mathbf{x})$, from which latent variables \mathbf{z} are sampled. The decoder then maps these latent variables to another probability distribution $p_\theta(\mathbf{x}'|\mathbf{z})$, which generates the reconstructed output \mathbf{x}' . The model aims to minimize the difference between the input

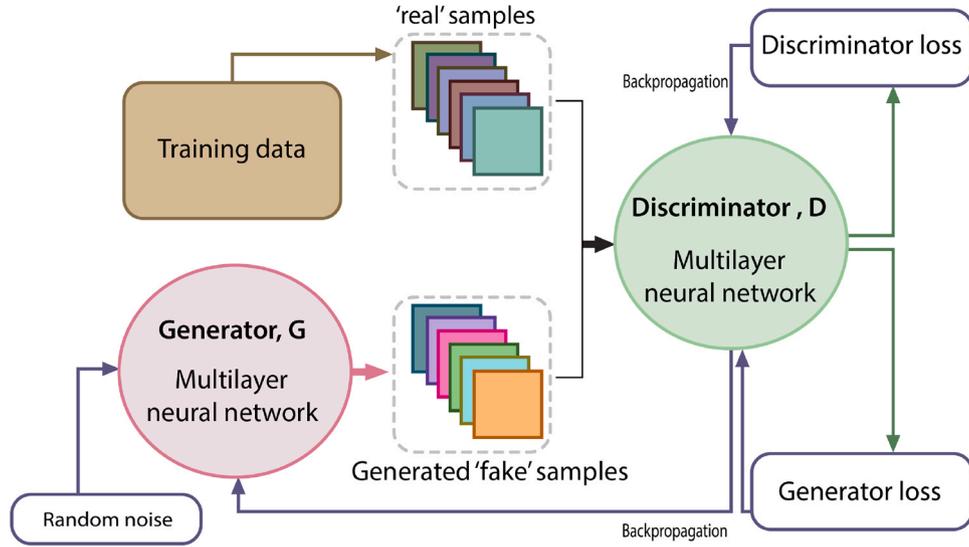


Fig. 2. General flowchart of a Generative Adversarial Network (GAN) architecture: The GAN consists of two main components: a Generator (G) and a Discriminator (D), both implemented as multilayer neural networks. The Generator takes random noise as input and produces synthetic 'fake' samples. The Discriminator receives both real samples from the training data and fake samples from the Generator, attempting to distinguish between them. The Discriminator's output is used to compute both the Discriminator loss and the Generator loss. These losses are then used to update the respective networks through backpropagation. This adversarial process continues iteratively, with the Generator improving at creating realistic fake samples and the Discriminator becoming better at distinguishing real from fake, until an equilibrium is reached.

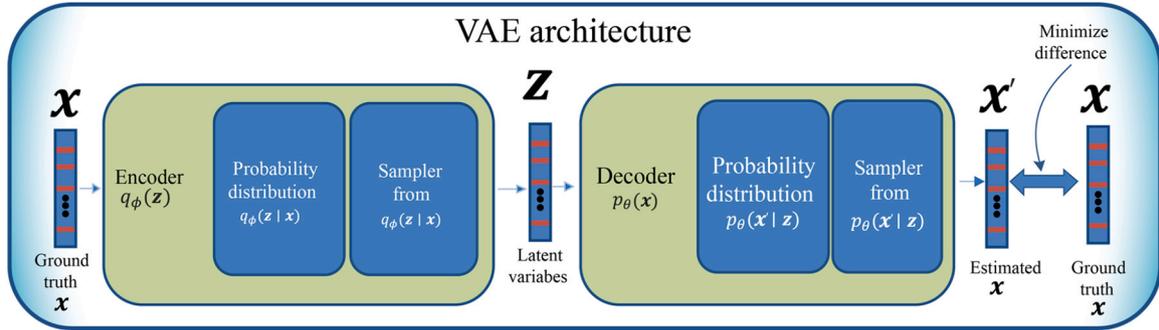


Fig. 3. VAE architecture: The model consists of an encoder that maps input \mathbf{x} to a latent distribution, a sampling step to generate latent variables \mathbf{z} , and a decoder that maps \mathbf{z} to the reconstructed output \mathbf{x}' . The goal is to minimize the difference between \mathbf{x} and \mathbf{x}' [21].

\mathbf{x} and the reconstructed output \mathbf{x}' .

4.3. Deep conditional generative models for oversampling

The following sections explore various deep conditional generative models developed to effectively oversample the minority classes in imbalanced datasets. These models aim to generate synthetic samples for the minority classes, thereby reducing the imbalance and improving the performance of machine learning algorithms trained on such datasets [1,7].

4.4. Conditional GANs (CGAN)

The conditional GAN (CGAN) [1,31,84] extends the classical GAN by conditioning both the generator and discriminator on the class label \mathbf{y} , as illustrated in Fig. 4. The learning objective represents the same two-player minimax game as Eq. (6), but now both models require \mathbf{y} as input.

$$\mathcal{L}_{\text{CGAN}}(\mathbf{x}, \mathbf{y}, \phi, \theta) = \min_{\phi} \max_{\theta} (E_{\mathbf{x} \sim P_X} [\log D_{\theta}(\mathbf{x}, \mathbf{y})] + E_{\mathbf{z} \sim q_{\phi}} [\log (1 - D_{\theta}(G_{\phi}(\mathbf{z}, \mathbf{y}), \mathbf{y}))]),$$

once trained, to sample a class \mathbf{y} observation, firstly $\mathbf{z} \sim q_{\mathbf{z}}(\mathbf{z})$ is sampled

and then both \mathbf{z} and \mathbf{y} are passed to the conditional generator G_{ϕ} to produce $\mathbf{x}' = G_{\phi}(\mathbf{z}, \mathbf{y})$. The CGAN was applied to handle class imbalance in [2]. In addition, the auxiliary classifier GAN (ACGAN) [85] and the balancing GAN (BAGAN) are two other conditional GAN variants capable of producing class-dependent synthetic observations, as discussed in [86] respectively.

4.5. Conditional VAEs (CVAE)

A variant of VAEs is proposed [1,27] that aims to learn class-dependent distributions, referred to as the conditional VAE (CVAE). Conditionalizing the VAE merely requires a shift in the objective from learning parameters (θ, ϕ) that maximize a lower bound on $p_{\theta}(\mathbf{x})$ to instead learning (θ, ϕ) which maximize a similar lower bound for the conditional distribution $p_{\theta}(\mathbf{x}|\mathbf{y})$ given by Eq. (8) [1]:

$$\mathcal{L}_{\text{CVAE}}(\mathbf{x}, \mathbf{y}; \theta, \phi) = \max_{(\phi, \theta)} \left(E_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y})} [\log p_{\theta}(\mathbf{x}|\mathbf{z}, \mathbf{y})] - \mathbb{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y}) \parallel p_{\theta}(\mathbf{z}|\mathbf{y})) \right), \quad (8)$$

$\mathcal{L}_{\text{CVAE}}(\mathbf{x}, \mathbf{y}; \theta, \phi)$ represents the CVAE loss function, where θ are the parameters of the generative model (decoder), and ϕ are the parameters of the inference model (encoder). The encoder $q_{\phi}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ aims to

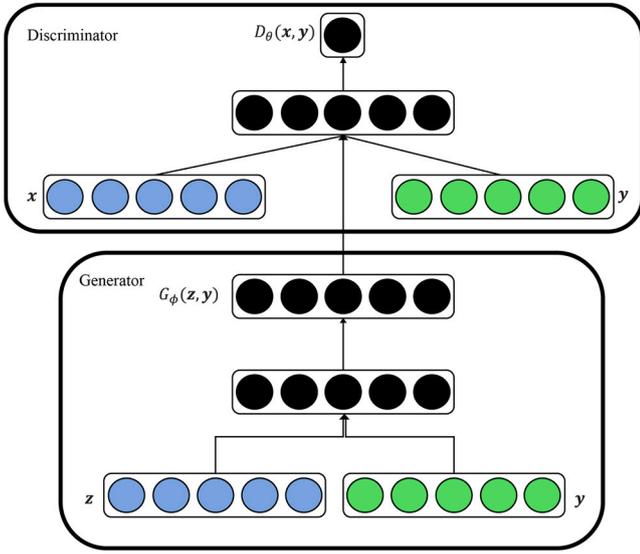


Fig. 4. Architecture of a Conditional Generative Adversarial Network (CGAN): The generator $G_\phi(z, y)$ takes random noise z and class label y as inputs to produce synthetic data. The discriminator D receives both real data x and generated data, along with the class label y , and tries to distinguish between real and fake samples. Both networks are conditioned on the class label y to enable class-specific generation and discrimination [31].

approximate the true posterior $p_\theta(x|z, y)$, while the decoder $p_\theta(x|z, y)$ generates the output conditioned on both z and y .

Observe that z conditioned on x under the q_ϕ measure is independent of y such that $q_\phi(z|x, y) = q_\phi(z|x)$. It also bears mentioning that other versions of the CVAE differ from the one described above, as discussed in [21] and [27].

Fig. 5 shows the architecture of a CVAE, showing how the label information y is incorporated into both the encoder and decoder processes.

4.6. Balancing generative adversarial network (BAGAN)

BAGAN [86] is a GAN model used for oversampling minority classes in imbalanced image classification data to restore balance. BAGAN leverages an autoencoder along with the GAN structure to enable

class-conditional image generation. An autoencoder is first pretrained in an unsupervised manner on all the training images. The decoder of the autoencoder is used to initialize the generator G in BAGAN. And its encoder is used to initialize the first layers of the discriminator D . This initialization enables learning a class conditioning in the latent space of G , where each class c is modeled via a multivariate Gaussian distribution $N_c = \mathcal{N}(\mu_c, \Sigma_c)$ estimated from encoded samples of that class. During adversarial training, G receives as input random latent vectors $Z_c \sim \mathcal{N}_c$ corresponding to class label c , and tries to generate realistic fake images of that class. While D tries to classify generated and real images as either fake or belonging to one of the classes. Therefore, BAGAN uses the autoencoder to facilitate conditional generation of minority class images in order to oversample those classes and restore the balance of an imbalanced dataset. The key mathematical concepts are: autoencoder training via l_2 loss optimization; modeling each class using the multivariate Gaussian $N_c = \mathcal{N}(\mu_c, \Sigma_c)$ and adversarial training of G and D via sparse categorical cross-entropy loss. The BAGAN training approach is organized in the three steps showed in Fig. 6.

4.7. Medical generative adversarial network (medGAN)

medGAN [22,30,87] is a generative model proposed to generate synthetic electronic health records (EHR). It combines a generative adversarial network (GAN) with an autoencoder to handle high-dimensional multi-label discrete variables representing events in EHRs like diagnoses, medications, procedures etc. An autoencoder is first pretrained in a supervised way to learn salient features of the discrete variables. The decoder of the autoencoder is then used to initialize the generator G of the GAN. And its encoder initializes the first layers of the discriminator D . This allows G to generate distributed representations of patient records, which are then decoded to synthetic patient records by the pretrained decoder Dec . D tries to discriminate between real patient records and synthetic ones decoded through $Dec(G(z))$. Mini-batch averaging is also used to avoid mode collapse and improve diversity of generated samples. Therefore, medGAN leverages the autoencoder to enable effective generation of multi-label discrete EHR data. The key mathematical aspects include: training the autoencoder via MSE loss defined in Eq. (9) [22]:

$$\frac{1}{m} \sum_{i=0}^m \|\mathbf{x}_i - \mathbf{x}'_i\|_2^2 \quad (9)$$

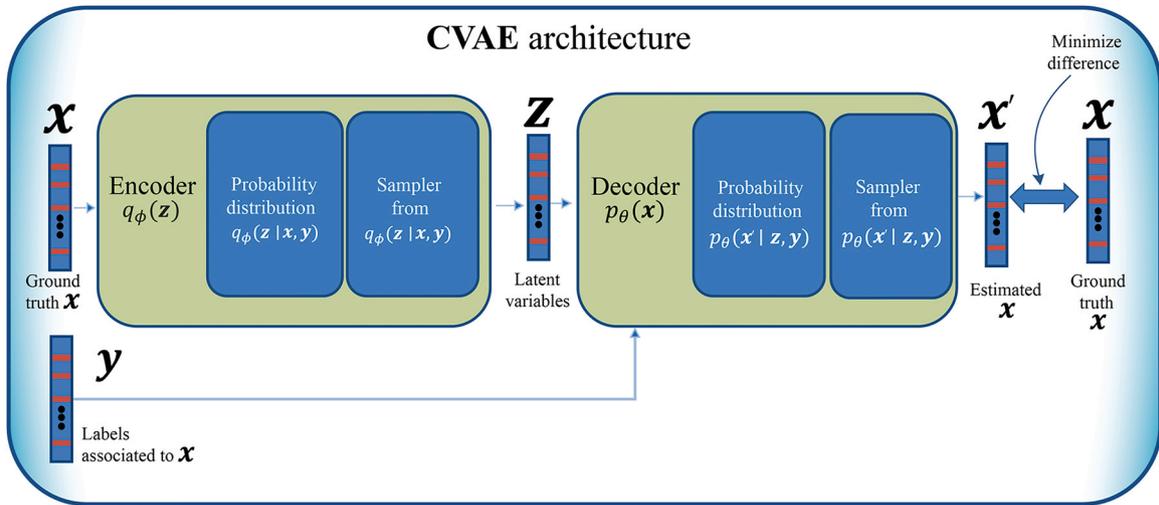


Fig. 5. CVAE architecture: This diagram illustrates the structure of a Conditional Variational Autoencoder. The encoder transforms input data x and associated labels y into a probability distribution $q_\phi(z|x, y)$. Latent variables z are sampled from this distribution. The decoder then uses both z and y to generate a probability distribution $p_\theta(x|z, y)$, from which the reconstructed data x' is sampled. The model aims to minimize the difference between x and x' , while utilizing label information to improve reconstruction quality and enable class-conditional generation [21].

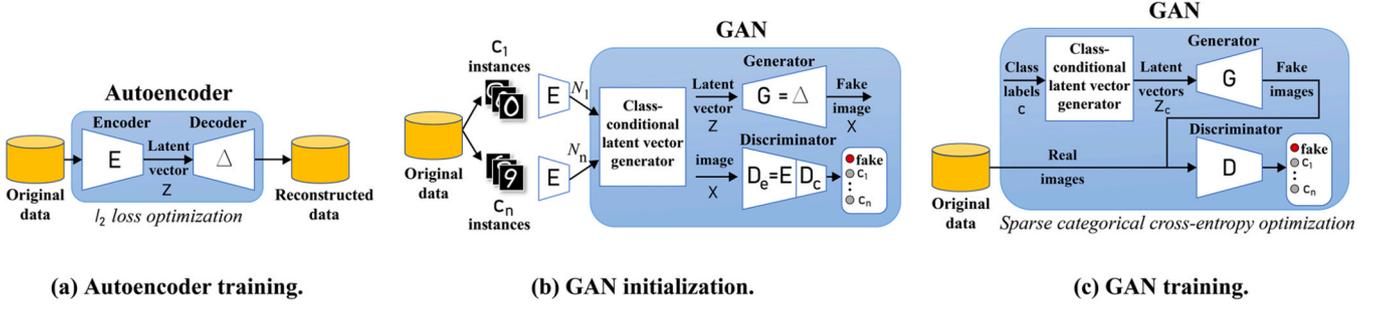


Fig. 6. The three training steps of the BAGAN methodology: (a) Autoencoder training on all data classes, (b) GAN initialization using the pretrained autoencoder components, and (c) GAN training for class-conditional image generation. The autoencoder's decoder initializes the GAN generator, while its encoder initializes the first layers of the discriminator [86].

or cross-entropy loss Eq. (10) [22]:

$$\frac{1}{m} \sum_{i=0}^m x_i \log x'_i + (1 - x_i) \log(1 - x'_i), \quad (10)$$

where $x'_i = Dec(enc(x_i))$

m is the number of samples in the mini-batch and x'_i is the reconstructed output of the autoencoder for the i th sample. enc represents the encoder function and Dec represents the decoder function of the autoencoder.

Adversarial training objective for GAN is similar to:

$$\phi_g \leftarrow \phi_g + \alpha \nabla_{\phi_g} \frac{1}{m} \sum_{i=0}^m \log D(G(z_i)),$$

where ϕ are the parameters of the generator G , α is the learning rate, z_i is the i th noise input.

But with discrete decoding; and mini-batch averaging to provide useful statistics to D to encourage diversity. The equations forming the mathematical foundation for medGAN's working are shown in Fig. 7.

4.8. Adversarially Regularized Autoencoder (ARAE)

The Adversarially Regularized Autoencoder (ARAE) [88,89,90] is an innovative model that integrates a discrete autoencoder with a latent representation regularized by a Generative Adversarial Network (GAN).

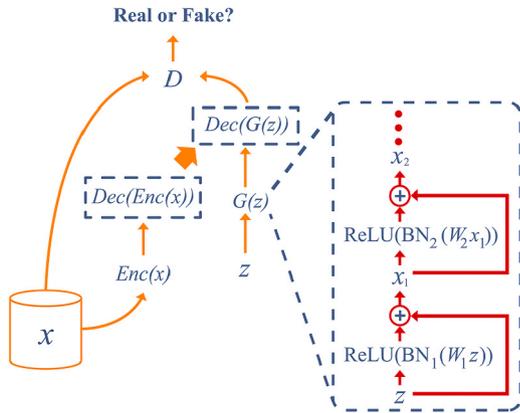


Fig. 7. Architecture of medGAN: The model consists of an autoencoder (encoder enc and decoder Dec), a generator G , and a discriminator D . Real patient data x is encoded and reconstructed through the autoencoder. The generator G , which includes multiple layers with non-linear activations, takes random noise z as input and produces synthetic data, which is then processed by the decoder Dec . The discriminator D evaluates both real and synthetic data to distinguish between them. The generator and decoder are trained together to produce realistic synthetic patient records that can fool the discriminator [22].

This architecture comprises an encoder function enc_ϕ that maps inputs (such as text sequences) to a continuous latent space z , and a conditional decoder $p_\theta(x|z)$ that reconstructs the original discrete input x from the latent code z .

A key feature of ARAE is the regularization of the latent code z . This is achieved by aligning its distribution p_Q with a prior distribution p_z through adversarial training. In this process, a critic/discriminator f_w is trained to differentiate between samples from p_Q and p_z . Concurrently, the encoder enc_ϕ is trained adversarially against f_w to minimize the Wasserstein distance between p_Q and p_z . The prior distribution p_z can be either fixed (e.g., Gaussian) or flexible (e.g., generated by a neural network generator G).

This approach effectively minimizes an upper bound on the total variation distance between the model distribution p_θ and the true data distribution p_x , enabling ARAE to effectively model complex discrete distributions [88].

The training process of ARAE alternates between three main steps:

1. Training the autoencoder (encoder and decoder) to minimize the reconstruction loss \mathcal{L}_{rec} . This step is expressed mathematically in Eq. (11) [88], which shows the objective function for minimizing the reconstruction loss:

$$\min_{\phi, \psi} \mathcal{L}_{rec}(\phi, \psi) = E_{x \sim p_x} [-\log p_\theta(x|enc_\phi(x))]. \quad (11)$$

2. Training the critic f_w to approximate the Wasserstein distance W between distributions. Eq. (12) [88] represents this step, demonstrating how the critic is trained to maximize the difference between the expected values of the critic's output for real and generated samples:

$$\max_{w \in \mathcal{W}} \mathcal{L}_{cri}(w) = E_{x \sim p_x} [f_w(enc_\phi(x))] - E_{z \sim p_z} [f_w(\tilde{z})], \quad (12)$$

where \tilde{z} represents samples drawn from the prior distribution p_z .

3. Adversarially training the encoder/generator to minimize W . This final step is formulated in Eq. (13) [88], which illustrates the objective function for training the encoder to minimize the Wasserstein distance:

$$\min_{\phi} \mathcal{L}_{enc}(\phi) = E_{x \sim p_x} [f_w(enc_\phi(x))] - E_{z \sim p_z} [f_w(\tilde{z})]. \quad (13)$$

These equations collectively form the mathematical foundation of the ARAE model, enabling it to effectively learn and generate complex discrete distributions.

4.9. Gaussian generative adversarial network (G-Gan)

G-GAN [29] utilizes the framework of Wasserstein Generative

Adversarial Network (WGAN) which demonstrates better training stability over regular GAN models [91,92,93]. In WGAN, the goal is to match the distribution of real positive samples and fake samples generated by the model, under the optimal transport metric of Wasserstein distance. This is formulated as the objective function shown in Eq.

$$\max_w \mathbb{E}_{x \sim p_x} [f_w(x)] - \mathbb{E}_{z \sim p_z} [f_w(G_\phi(z))], \quad (14)$$

[29]:

$$\max_{w,\theta} \mathbb{E}_{x \sim p_x} [f_w(x)] - \mathbb{E}_{z \sim p_z} [f_w(G_\phi(z))], \quad (14)$$

where W are the parameters of the critic function f_w , ϕ are the parameters of the generator function G , p_x is the real data distribution, and P_z is the prior noise distribution.

G-GAN [29] is a novel oversampling method to generate additional minority class (positive) samples for imbalanced classification problems. G-GAN is based on the WGAN framework which has better training stability compared to regular GANs. Prior knowledge about the distribution of positive samples is incorporated by fitting a Gaussian distribution to the minority class data. This Gaussian distribution is used to sample some of the input noise vectors to the generator. The input noise vectors are a mix of samples from the fitted Gaussian distribution and uniform random noise. This increases the diversity of the generated samples. Multiple G-GAN models are trained using bagging on subsets of the positive samples. This further improves diversity and reduces overfitting. The multiple G-GAN models generate synthetic positive samples, which are combined to balance out the class distribution. The fitted Gaussian distribution used to model positive samples Eq. (15) [29] is:

$$N(\mathbf{z}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p) = \frac{1}{(2\pi)^{1/2} |\boldsymbol{\Sigma}_p|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu}_p)^T \boldsymbol{\Sigma}_p^{-1} (\mathbf{z} - \boldsymbol{\mu}_p)\right), \quad (15)$$

where $\boldsymbol{\mu}_p$ and $\boldsymbol{\Sigma}_p$ are the mean and covariance matrix fit to the positive samples.

The mixed input noise vectors Eq. (16) [29] are defined as:

$$\mathbf{Z} = \left\{ \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{\lfloor \frac{b}{2} \rfloor}, \mathbf{z}_{\lfloor \frac{b}{2} \rfloor + 1}, \dots, \mathbf{z}_b \right\}, \quad (16)$$

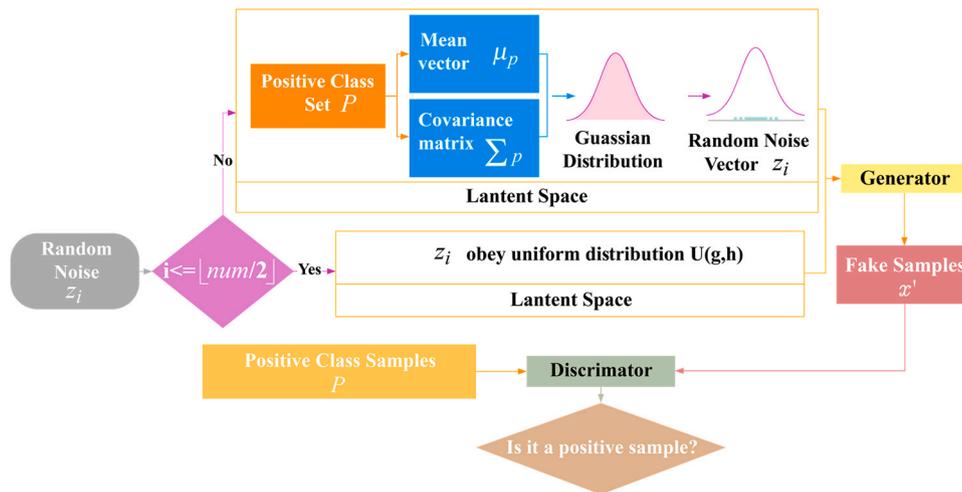


Fig. 8. Diagram of G-GAN's latent space generation and sample creation process: The model uses a combination of Gaussian and uniform distributions to generate the latent space input for the generator. The positive class samples are used to fit a Gaussian distribution, providing the mean vector and covariance matrix. Random noise vectors are sampled from either this Gaussian distribution or a uniform distribution $U(\mathbf{g}, \mathbf{h})$, depending on their index. The generator uses this mixed latent space input to create fake samples, which are then evaluated by the discriminator along with real positive samples [29].

where $\mathbf{Z}, \mathbf{z}_i, i = 1, 2, \dots, \lfloor b/2 \rfloor$ are sampled from the uniform distribution $U(\mathbf{g}, \mathbf{h})$ and the remaining \mathbf{z}_i are sampled from the Gaussian distribution $N(\mathbf{z}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$. The overall architecture of G-GAN is shown in Fig. 8.

Fig. 9 shows the G-GAN algorithm with Bagging approach – training multiple models on subsets of the data.

The key steps of the G-GAN algorithm are as follows:

First, a Gaussian distribution $N(\mathbf{z}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ is fitted to the positive samples, where $\boldsymbol{\mu}_p$ and $\boldsymbol{\Sigma}_p$ represent the mean and covariance matrix fit to the positive data.

Next, batches of input noise vectors \mathbf{Z} are generated using as explained previously, Eq. (16) samples half the noise vectors from a uniform distribution $U(\mathbf{g}, \mathbf{h})$ and the other half from the fitted Gaussian. The WGAN generator and discriminator networks are then trained. The generator tries to produce synthetic samples that fool the discriminator, while the discriminator tries to tell apart real and fake samples.

Once training converges, the trained generator is used to produce new synthetic positive samples. The bagging technique [94] is applied by re-training the G-GAN models on different subsets of positive data. This trains multiple G-GAN models. The synthetic samples from all the G-GAN models are aggregated.

Finally, a subset is randomly sampled from this aggregate set to balance out the original class distribution.

4.10. Minority oversampling generative adversarial network (MoGAN)

$$p_g(\mathbf{x}) = \pi p_{m_i}(\mathbf{x}) + \pi(1 - p_x(\mathbf{x})), \quad (17)$$

MoGAN [95] (Minority Oversampling Generative Adversarial Network) contains two interdependent networks, a generative network G and a discriminator network D . The generative network of MoGAN acts as an efficient oversampling technique to generate synthetic minority samples. It incorporates the majority class distribution when generating new minority samples in order to restore balance in the imbalanced dataset. MoGAN learns from all available data distributions instead of interfering with only the majority data distribution. This prevents overfitting and leads to better variety in the generated samples. The generative network G uses a mixture data distribution to generate minority samples as shown by Eq. (17) [95]:

where $p_g(\mathbf{x})$ is the generated data distribution, $p_{m_i}(\mathbf{x})$ is the majority class distribution, $p_x(\mathbf{x})$ is the real data distribution, and π is a mixing parameter controlling the contribution of each distribution.

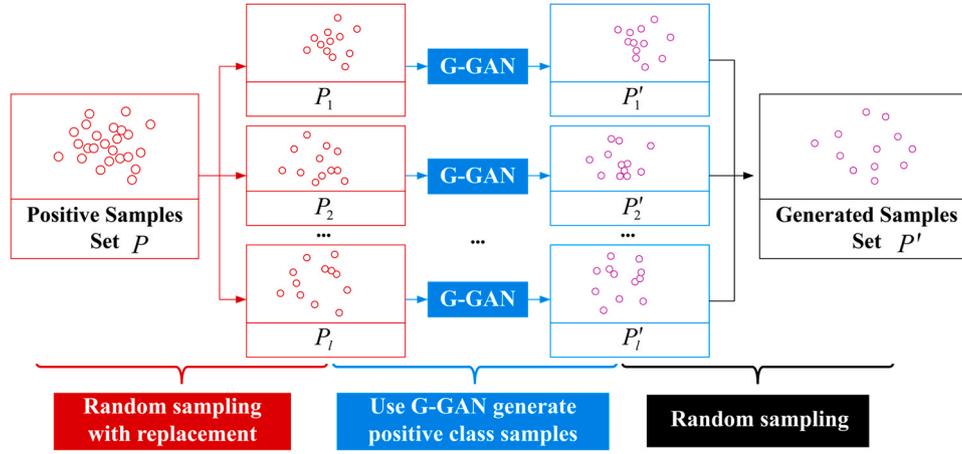


Fig. 9. G-GAN with Bagging strategy: Multiple G-GAN models are trained on randomly sampled subsets of the positive sample set P . Each G-GAN generates a set of new positive samples P' . The final generated sample set P' is created by randomly sampling from the combined output of all the G-GAN models. This approach aims to increase diversity and reduce overfitting in the generated samples [29].

The discriminator network D in MoGAN acts as a classifier to distinguish between real/fake samples as well as classify them into normal vs faulty samples simultaneously. The optimal discriminator for the false positive rate is given by Eq. (18) [95]:

$$D = \frac{p_X(\mathbf{x})}{p_X(\mathbf{x}) + p_g(\mathbf{x})} \quad (18)$$

The training objective for the generator is formulated as shown in Eq. (19) [95]:

$$\min_{\phi_g} E_{z \sim p_z} [\log (1 - D(G(\mathbf{z}; \phi_g); \theta_d))], \quad (19)$$

where ϕ_g and θ_d are the parameters of the generator and discriminator respectively, and p_z is the prior noise distribution and E_x denotes expectation over noise distribution.

To handle the mixture data distribution and generate minority samples in low-density regions, MoGAN employs a specialized loss function based on KL divergence, as represented by Eq. (20) [95]:

$$\min_{\phi_g} - \Gamma(p_g(\mathbf{x}; \phi_g) + E_{x \sim p_g(x; \phi_g)} \log_{P_{Mj}}(\mathbf{x})) \quad (20)$$

$$\text{where; } Loss(G) = \min_{\phi_g} \rho[P_{Mj}(\mathbf{x}) > \delta] + \rho[E_{x \sim p_f(x)} [f(G(\mathbf{z}; \phi_g))]]$$

where E_x denotes expectation over the respective data distribution, $\Gamma(\cdot)$, $\rho(\cdot)$ are the marginal and joint entropy function respectively, $P_{Mj}(\mathbf{x})$ is the probability of \mathbf{x} belonging to the majority class, and δ is a threshold, $p_f(\mathbf{x})$ is the distribution of fault samples, and f is a feature extraction

function.

These equations collectively define the core of the MoGAN model, enabling it to generate diverse and realistic minority samples while maintaining a balance between classes in fault diagnosis tasks. Fig. 10 shows the overall architecture of MoGAN.

4.11. 3D-HyperGAMO

The 3D-HyperGAMO architecture presented in Fig. 11 aims to address the class imbalance problem for hyperspectral image (HSI) classification [33,96]. It utilizes a generative adversarial approach to minority oversampling [97] in order to balance the training data by generating additional samples for classes with limited representations.

The key components of the 3D-HyperGAMO architecture include:

- 3D Patch Extractor: Extracts 3D cubes from the HSI data to retain full spectral information;
- Conditional Feature Mapping Unit: Maps noise vectors to intermediate feature representations conditioned on class labels;
- Input: Random noise vector (\mathbf{z}), One-hot encoded class label vector (\mathbf{l});
- Convex 3D Patch Generator Unit (PGU): Generates new 3D HSI patches for minority classes using the intermediate features and real samples. The number of patches generated for class i is given by Eq. (21) [33];

$$\lambda_i^g = m_s - m_l \quad (21)$$

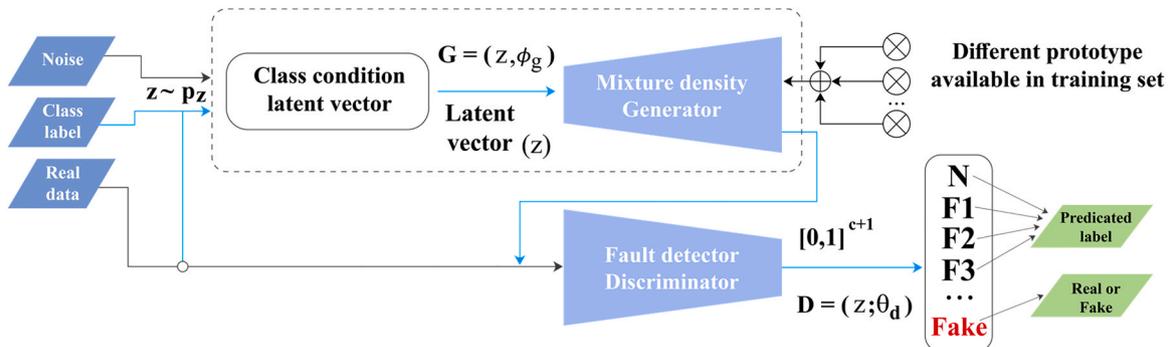


Fig. 10. Architecture of the proposed MoGAN model for imbalanced fault diagnosis: The generator G takes noise \mathbf{z} , class labels, and real data as inputs to produce a mixture density output. The discriminator D acts as both a classifier and fault detector, distinguishing between real/fake samples and predicting fault labels. N, F1, F2, F3, etc. represent normal and different fault classes [95].

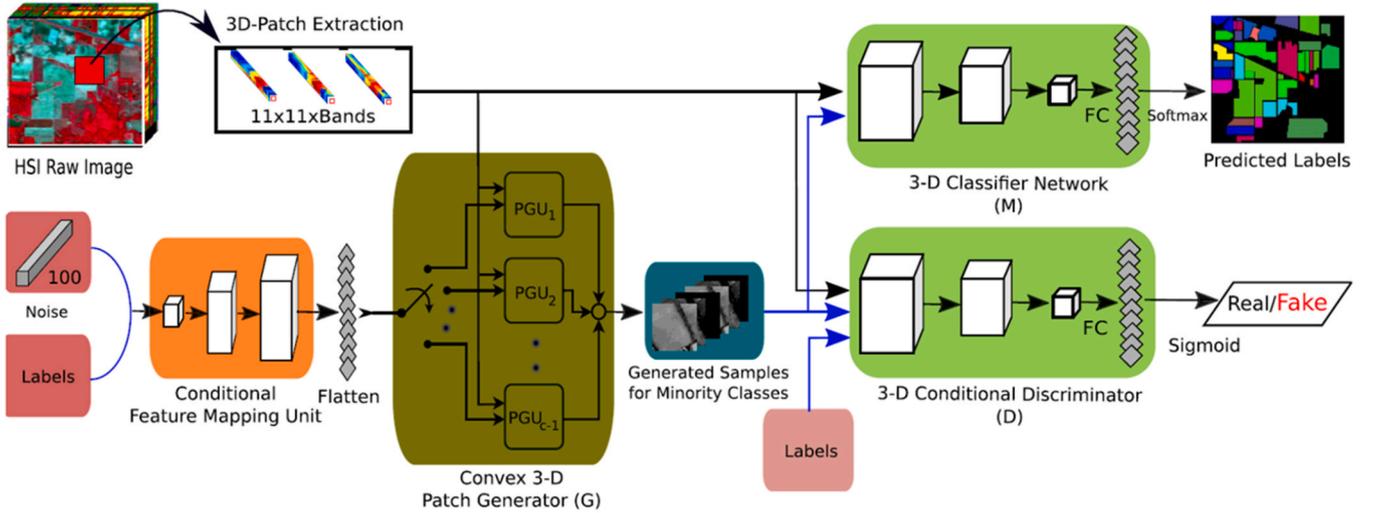


Fig. 11. Architecture of the proposed 3D-HyperGAMO model for addressing class imbalance in hyperspectral image classification: The model includes a 3D patch extractor, a conditional feature mapping unit, a convex 3D patch generator (G) for minority classes, a 3D conditional discriminator (D), and a 3D classifier network (M). This approach generates new samples for minority classes to improve classification performance on imbalanced datasets [33].

where m_l is the number of samples in the majority class, and m_s is the number of available samples in the i minority class.

- 3D Conditional Discriminator: Distinguishes between real and generated 3D patches;
- 3D Classification Network: Categorizes original and generated patches into classes.

The conditional feature mapping unit and PGU enable controlled generation of new minority class training samples. The discriminator provides adversarial supervision for realistic patches.

The PGU generates a class-specific feature matrix F_m for creating new patches using Eq. (22) [33]:

$$F_m = I_m \cdot (F_m)^T \quad (22)$$

where I_m is the random feature matrix and $(F_m)^T$ is the transpose of F_m .

The overall objective function is a minimax game between the generator (G), discriminator (D), and classifier (M) networks. By optimizing this function, 3D-HyperGAMO synthesizes additional training samples to mitigate imbalance and enhance classification performance.

4.12. Improved variational autoencoder generative adversarial network (improved VAEGAN)

Improved Variational Autoencoder Generative Adversarial Network (improved VAEGAN [32]) is a new oversampling method for handling imbalanced data classification problems. It is an improvement over the original VAEGAN model by adding an extra encoder, aiming to enhance the model's representation ability to generate more realistic and diverse minority class data for oversampling.

Specifically, the improved VAEGAN has two encoders E1 and E2 that encode the input real data \mathbf{x} into mean (μ_1, μ_2) and variance (σ_1^2, σ_2^2) codes respectively. By fusing the outputs of the two encoders, the latent code \mathbf{z} is generated, which is then decoded by the decoder to generate fake samples \mathbf{x}' . Fig. 12 shows the flowchart of improved VAEGAN.

$$h'(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{(\mathbf{x}-\mu_0)^2}{2\sigma_0^2}} \quad (23)$$

The fusion of the two encoder outputs is achieved by multiplying their probability density functions, as shown in Eq. (23) [32]:

where $h'(\mathbf{x})$ represents the fused probability density function, σ_0 is

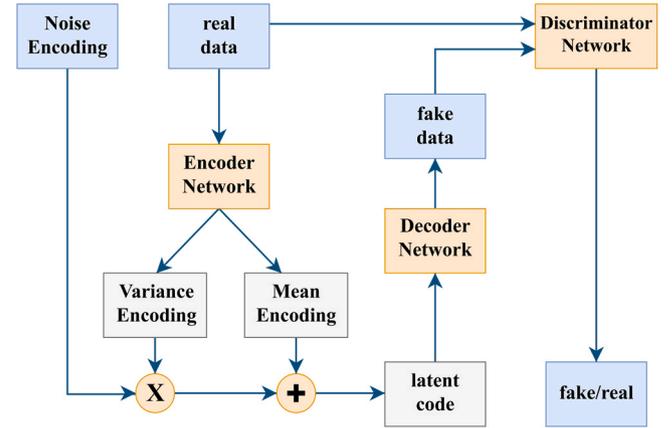


Fig. 12. Flowchart of VAEGAN: Real input data is fed into the encoder and decoder components to produce synthesized false samples. The discriminator then attempts to evaluate the authenticity of the generated samples versus the real data [32].

the fused standard deviation, and μ_0 is the fused mean.

The mean μ_0 of the fused distribution is calculated as shown in Eq. (24) [32]:

$$\mu_0 = \frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \quad (24)$$

The variance σ_0^2 of the fused distribution is determined by Eq. (25) [32]:

$$\sigma_0^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \quad (25)$$

The latent code \mathbf{z} is then sampled from the distribution $N(\mu_0, \sigma_0^2)$, where N represents a normal distribution.

The improved VAEGAN model is used to generate new minority class samples by training the encoder(s)-decoder flow. The generated samples are combined with the original imbalanced dataset to form an augmented balanced dataset for training classifiers. Experiments demonstrate improved classification performance compared to original VAEGAN and other oversampling methods.

4.13. Generative adversarial network synthesis for oversampling (GANSO)

GANSO [17] is a novel oversampling method that utilizes Generative Adversarial Networks (GANs) and vector Markov Random Fields (vMRFs) to synthesize realistic instances from a limited number of original samples. It aims to address the challenge of training classifiers with extremely small datasets by incorporating structural information into the generative process. The GANSO architecture consists of two main components: a generative block and a discriminative block, as illustrated in Fig. 13. The generative block uses the vMRF model to synthesize surrogates by applying the Graph Fourier Transform (GFT) on an extended graph representation of the data. The discriminative block implements a linear discriminant on features measuring clique similarities between the synthesized and original instances. The two blocks engage in an iterative competition until the discriminator can no longer distinguish between the synthetic and original samples.

The key steps of the GANSO algorithm are as follows:

1. Construct an extended undirected graph $G(V_e, E_e, \mathbf{A}_e)$ from the original graph $G(V, E, \mathbf{A})$, where each sample is assigned to a vertex in V_e , and edges in E_e connect vertices corresponding to samples from the same or connected segments in the original graph.
2. Generate an initial synthetic instance $\mathbf{s}_{(m,0)}$ by applying the GFT on a selected original instance $\mathbf{x}_{(m)}$. The GFT is computed using the eigenvectors \mathbf{U}_e of the extended graph Laplacian matrix \mathbf{L}_e and a diagonal matrix Φ with random sign changes, as shown in Eq. (26) [17]:

$$\mathbf{s}_{(m,0)} = \mathbf{U}_e \Phi \mathbf{U}_e^H \mathbf{x}_{(m)} \quad (26)$$

where \mathbf{U}_e contains the eigenvectors of \mathbf{L}_e , Φ is a diagonal matrix with random ± 1 values, and \mathbf{U}_e^H is the Hermitian transpose of \mathbf{U}_e .

3. Compute feature vectors $\mathbf{k}_{(n)}^x$ and $\mathbf{k}_{(n)}^s$ based on similarities between maximal cliques of original and synthetic instances. The feature vectors are defined in Eq. (27) [17]:

$$\begin{aligned} \mathbf{k}_{(n)}^x &= [k(\mathbf{x}_{(m)}^1, \mathbf{x}_{(n)}^1) \dots k(\mathbf{x}_{(m)}^c, \mathbf{x}_{(n)}^c)]^T \\ \mathbf{k}_{(n)}^s &= [k(\mathbf{s}_{(m)}^1, \mathbf{x}_{(n)}^1) \dots k(\mathbf{s}_{(m)}^c, \mathbf{x}_{(n)}^c)]^T \end{aligned} \quad (27)$$

where $k(\mathbf{x}^c, \mathbf{y}^c)$ is a similarity measure between cliques, \mathbf{x}^c and \mathbf{s}^c represent the cliques of the original and synthetic instances, respectively, and C is the total number of maximal cliques.

4. Train the discriminator by finding the optimal linear discriminant coefficients $\mathbf{w}_{opt,i}$ to distinguish between original and synthetic instances. The optimization problem is formulated in Eq. (28) [17]:

$$\underbrace{\mathbf{K}}_{(2N \times C)} \cdot \underbrace{\mathbf{w}}_{(C \times 1)} = \underbrace{\mathbf{v}}_{(2N \times 1)} \quad \mathbf{K} = [\mathbf{k}_{(1)}^x \dots \mathbf{k}_{(N)}^x \mathbf{k}_{(1)}^s \dots \mathbf{k}_{(N)}^s]^T \quad \mathbf{v} = \begin{bmatrix} \& \mathbf{1}_N \\ \& -\mathbf{1}_N \end{bmatrix} \quad (28)$$

where \mathbf{K} is a matrix containing the feature vectors $\mathbf{k}_{(N)}^x$ and $\mathbf{k}_{(N)}^s$, \mathbf{w}_{opt} is the vector of optimal discriminant coefficients, and \mathbf{v} is a vector of labels (+1 for original instances and -1 for synthetic instances). The solution is given by Eq. (29) [17]:

$$\mathbf{w}_{opt} = \mathbf{K}^T (\mathbf{K} \mathbf{K}^T)^{-1} \mathbf{v} \quad (29)$$

5. If the discriminator achieves an error probability close to 0.5, accept the current synthetic instance. Otherwise, proceed to the generator step.
6. Update the generator by computing optimal correcting factors $\mathbf{f}_{opt,i}$ to modify the synthetic instance and deceive the discriminator. The corrected feature vectors are computed as shown in Eq. (30) [17]:

$$\mathbf{k}_{(n,1)}^s \equiv \mathbf{k}_{(n,0)}^s \mathbf{f}_0 = [k(\mathbf{s}_{(m,0)}^1, \mathbf{x}_{(n)}^1) \cdot \mathbf{f}_0^1 \dots k(\mathbf{s}_{(m,0)}^c, \mathbf{x}_{(n)}^c) \cdot \mathbf{f}_0^c]^T \quad (30)$$

where \mathbf{f}_0 is a vector of correcting factors for each clique. The optimal correcting factors are obtained by solving the system of equations in Eq. (31) [17]:

$$\begin{aligned} [\mathbf{k}_{(1,0)}^s \cdot \mathbf{f}_0 \dots \mathbf{k}_{(N,0)}^s \cdot \mathbf{f}_0]^T \cdot \mathbf{w}_{opt,0} &= [\mathbf{1}_N] \Leftrightarrow [\mathbf{k}_{(1,0)}^s \cdot \mathbf{w}_{opt,0} \dots \mathbf{k}_{(N,0)}^s \cdot \mathbf{w}_{opt,0}]^T \cdot \underbrace{\mathbf{f}_0}_{\mathbf{1}_N} \\ &= [\mathbf{1}_N] \end{aligned} \quad (31)$$

The solution is given by Eq. (32) [17]:

$$\begin{aligned} \mathbf{f}_{opt,0} &= \mathbf{K}_{w,0}^s (\mathbf{K}_{w,0}^s \mathbf{K}_{w,0}^s)^{-1} [\mathbf{1}_N] \quad \mathbf{K}_{w,0}^s \\ &= [\mathbf{k}_{(1,0)}^s \cdot \mathbf{w}_{opt,0} \dots \mathbf{k}_{(N,0)}^s \cdot \mathbf{w}_{opt,0}]^T \end{aligned} \quad (32)$$

where $\mathbf{K}_{w,0}^s$ is a matrix containing the dot products between the feature vectors $\mathbf{k}_{(N,0)}^s$ and the discriminant coefficients $\mathbf{w}_{opt,0}$.

7. Generate corrected cliques $\mathbf{s}_{(m,i+1)}^c = \text{sign}(f_{opt,i}^c) \mathbf{s}_{(m,i)}^c$ and reconstruct the updated synthetic instance $\mathbf{s}_{(m)}$.
8. Repeat steps 4–7 for a predefined number of iterations or until the discriminator cannot distinguish the synthetic instance.

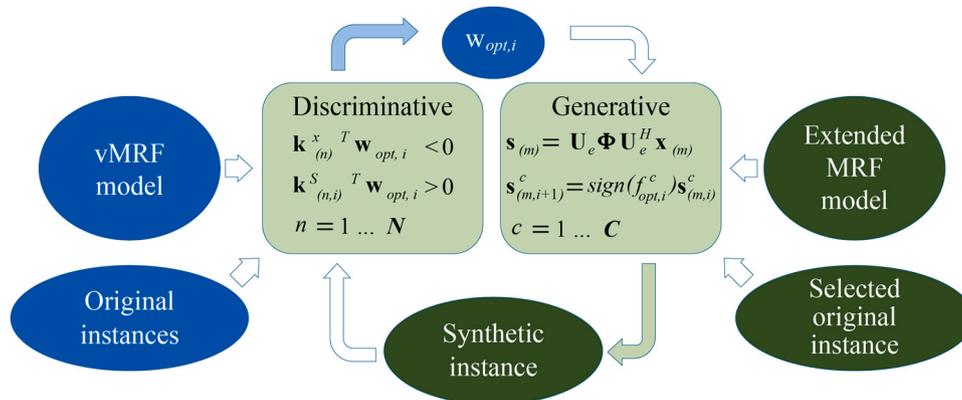


Fig. 13. Schematic diagram of the GANSO architecture: The generative block synthesizes instances using vMRFs and Graph Fourier Transform, while the discriminative block evaluates their authenticity. White arrows indicate inputs; solid arrows show outputs. The iterative process continues until synthetic instances are indistinguishable from original ones, enabling effective oversampling for imbalanced datasets [17].

By incorporating structural information through the vMRF model and leveraging the adversarial learning paradigm, GANSO can effectively generate realistic synthetic instances from a limited number of original samples. Experimental results on simulated and real-world datasets demonstrate GANSO's ability to improve classifier performance in scenarios with extreme data scarcity, outperforming traditional oversampling methods like SMOTE.

In summary, GANSO is a promising deep generative approach for oversampling that addresses the challenges of training classifiers with extremely small datasets by integrating GAN-based adversarial learning with vMRF-based structural information. Its innovative architecture and mathematical formulation make it a valuable tool for enhancing the performance of machine learning models in data-scarce domains.

4.14. Discussions

Deep generative models can be harder to train due to additional neural network components. The models may also require more data compared to simpler oversampling techniques. Overall, though, deep generative minority oversampling is a very promising direction for handling highly imbalanced datasets across a variety of problem domains.

The bar chart in Fig. 14 shows the growing attention towards using Generative Adversarial Networks (GANs) for oversampling in recent years.

As observed, the frequency of occurrence mention of the keywords "GAN" and "oversampling" in scientific papers has notably increased since 2018. Our analysis indicates this rising trend reflects intensifying efforts to harness GANs capabilities to address data imbalance by synthetically oversampling minority classes.

However, open challenges remain around ensuring training stability, diversity among generated samples, and rigorous, transparent benchmarking. Further research and development are vital to demonstrate performance improvements over current approaches on real-world problems. Evaluating advancements on class-specific metrics is critical.

Table 1 presents a comparative analysis of various deep generative models that have been employed for minority oversampling to handle class imbalance. Key criteria used for comparison include the type of data the models have been applied on, the datasets utilized in the referenced studies, the evaluation metrics measured, the reported performance results, disadvantages or limitations of the methods, and their advantages. In addition to the models discussed, the Generative Adversarial Network Synthesis for Oversampling (GANSO) approach has shown promising results in generating realistic minority class samples by incorporating vector Markov Random Field (vMRF) structural information into the GAN framework. GANSO has been applied to various data types, including EEG, fMRI, and ECG signals, and has demonstrated the ability to improve classifier performance in scenarios with extreme

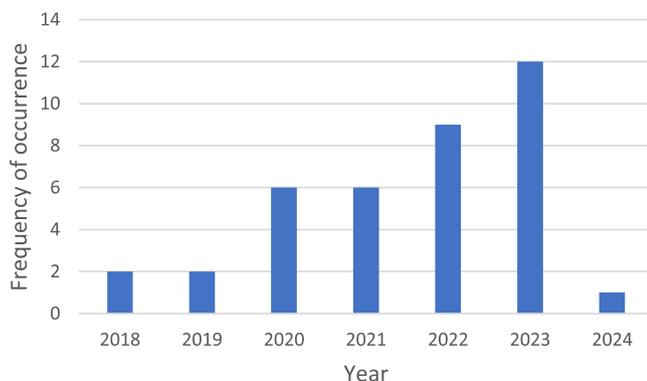


Fig. 14. Growth of Attention to GAN for Oversampling as suggested by Scopus indexed articles.

data scarcity.

As can be observed, techniques like generative adversarial networks (GANs), variational autoencoders (VAEs), conditional GANs (CGANs), conditional VAEs (CVAEs), and Balancing GAN (BAGAN) have been tested on image data, while Adversarially Regularized Autoencoder (ARAE), Gaussian Generative Adversarial Network (G-GAN), and Medical GAN (medGAN) handle textual, numerical and electronic health record data respectively. Performance is quantified using metrics like overall precision, recall, F1-scores, group-mean accuracy, balanced accuracy and others depending on context. GANSO, being a relatively new approach, has been evaluated using metrics such as classification accuracy and learning curves on real-world datasets, showcasing its potential for enhancing minority class representation.

The results showcase some relative strengths and weaknesses of the models. For instance, CGANs achieve stable training but worst test performance among comparable GAN techniques as per the MNIST case study. In contrast, CVAEs produce better F1-scores but have higher computational overhead. GANSO, on the other hand, leverages the vMRF model to capture structural information and generate realistic samples, but may face challenges in scalability and training complexity as the data dimensionality increases. Comparisons on criteria such as sample quality, diversity, computational complexity, scalability and training stability highlight areas needing improvement across models.

The structured analysis of various minority oversampling techniques using deep generative models and their characteristics facilitated through Table 1 sets the context for further sections discussing current challenges, shortcomings, comparative analysis and future opportunities in this rapidly evolving subfield of machine learning research.

5. Challenges of oversampling with deep generative approaches

Each of explained models has its own distinct advantages and limitations w.r.t scalability, training complexity and potential biases which should be considered based on data type and application requirements.

5.1. Scalability

GANs are very popular as generative content models for high quality and high-resolution images, but their main challenge still remains scalability to very large dataset sizes. Increasing image sizes significantly increases the number of parameters required to generate samples, leading to increased training time and computational costs. In addition to hardware limitations, the efficiency of very high dimensional neural networks also decreases. VAEs also face scalability challenges with increasing data dimensions, as the number of parameters and the complexity of the latent space increases rapidly, reducing model efficiency. Although the extra conditional input in CVAE partially controls this issue, it is still challenging for large datasets.

BAGAN utilizes both VAE and GAN components, so it has even higher computational costs compared to independent GAN and VAE. Since it requires separate training of VAE and GAN components, it can lead to a significant drop in efficiency at large data dimensions. ARAE has the advantage that unlike GAN and VAE, it does not require model structure adjustments for different input sizes and uses a standard architecture for different datasets. But very long sequence inputs still cause scalability issues leading to reduced efficiency.

G-GAN is designed specifically for small-sized low-dimensional numerical data, so when volumes become very large or input dimensions become too high, it faces serious scalability constraints. MoGAN benefits from convolutional architectures that scale well to large image data. It also uses group normalization that accelerates the training. On the other hand, medGAN employs convolutional networks suitable for large images and uses techniques like importance sampling for better computational efficiency. 3D-HyperGAMO is designed for hyperspectral imagery data containing large volumes of spatio-spectral information, and utilizes convolutional networks capable of processing such large

Table 1
Comparative Analysis of Deep Generative Models for Minority Oversampling.

Model	Data Type	Dataset	Evaluation Metric	Result	Performance at Different Imbalance Ratios (IR)	Disadvantages	Advantages		
SMOTE [1]	Image	MNIST	Prec.(Overall) Recall(Overall) F1 Score (Overall)	0.920 0.906 0.905	IR = 80 (F1 =0.905) IR = 320 (F1 =0.392)	Sensitive to hyperparameters	Best performance in 4 out of 12 experiments		
RANDOM [1]	Image	MNIST	Prec.(Overall) Recall(Overall) F1 Score (Overall)	0.919 0.904 0.902	IR = 80 (F1 =0.868) IR = 320 (F1 =0.529)	May overfit	Best performance in 2 out of 12 experiments		
CGAN [1]	Image	MNIST	Prec.(Overall) Recall(Overall) F1 Score (Overall)	0.900 0.873 0.869	IR = 80 (F1 =0.816) IR = 320 (F1 =0.147)	Worst performance	Stable training		
CVAE [1]	Image	MNIST	Prec.(Overall) Recall(Overall) F1 Score (Overall)	0.916 0.909 0.908	IR = 80 (F1 =0.908) IR = 320 (F1 =0.662)	High computational cost	Best performance in 6 out of 12 experiments		
ADASYN [30]	Tabular (discrete)	Adult Dataset	F1 Score (Mean) (Std)	0.811 0.055	IR = 0.28 (F1 =0.537) IR = 0.33 (F1 =0.727)	Applicable only for classification tasks	Simple and fast oversampling		
VAE [30]	Tabular (discrete)	Adult Dataset	F1 Score (Mean) (Std)	0.820 0.039	IR = 0.28 (F1 =0.535) IR = 0.001 (F1 =0.820)	More difficult To train compared to GANs	Stable training process		
GAN [30]	Tabular (discrete)	Adult Dataset	F1 Score (Mean) (Std)	0.820 0.069	IR = 0.33 (F1 =0.734) IR = 0.001 (F1 =0.820)	Risk of mode collapse	Generate sharp samples		
medGAN [30]	Tabular (discrete)	Adult Dataset	F1 Score (Mean) (Std)	0.822 0.042	IR = 0.28 (F1 =0.534) IR = 0.001 (F1 =0.822)	Insufficient training data	Privacy Preservation, faster model prototyping		
BAGAN [86]	Image	MNIST CIFAR10 Flowers GTSRB	SSMI couples Accuracy (min-max)	0.2–0.4 0.1–0.6 0.1–0.7 0.1	0.99 0.70 0.75 0.97	IR= 1.67 (Accuracy = 99 %) IR= 40 (Accuracy = 96.5 %)	Requires further validation	Robustness against data imbalance, prevents mode collapse	
ARAE [88]	Binary images	Text MNIST Yelp SNLI	Reverse PPL Forward Accuracy PPL	82.2 44.3	81.8 % 70.9 %	Performance at different IR ratios not reported.	Sensitive to hyperparameters, Challenging for complex structures	Smoother latent space, Manipulation ability, Semi-supervised performance	
G-GAN [29]	Numerical	11 standard datasets from KEEL database	G-mean	0.9152	IR = 1.84 (G-mean=0.6115) IR = 66.67 (F1 =0.9997)	Not suitable for highly imbalanced numerical data	Better performance compared to 11 other common balancing algorithms		
MoGAN [95]	Time series	IMS Dataset WTFF Dataset IPF Dataset	G-mean Precision F-measure Recall Balanced Accuracy	0.9789 0.9427 0.9064 0.8902 99.23 0.8946 0.8730 0.8902 97.28	IR = 1:3 (F1 = 0.95) IR = 1:1000 (F1 = 0.71)	High computational complexity	Better performance compared to existing methods		
3D-HyperGamo [33]	Hyperspectral image	Indian Pines (IP) Kenedy Space Center (KSC) University of Pavia (UP) Botswana (BW)	Overall Accuracy Average Accuracy Kappa (x100)	86.96 95.31 93.9 97.43	78.72 92.26 93.29 97.4	85.17 94.78 91.86 97.22	IR = 10 (OA=95.19 %) IR = 123 (OA=85.95 %)	High computational complexity due to the use of deep neural networks	Significantly improves classification performance for minority classes by oversampling them during training
Improved VAEGAN [32]	Numerical	Credit card fraud detection dataset (Number Ng=33700 of generated examples is varied)	AUC Precision Recall F1 score	0.98476 0.8630 0.8302 0.8463		IR = 0.25 (F1 = 0.8775) IR = 100 (F1 = 0.8793)	More complex architecture compared to original VAEGAN	Generates more diverse data for minority class significantly improves precision, f1 score, combines strengths of VAE and GAN	
GANSO [17]	Time series, Images	EEG (Barcelona Test), fMRI (OpenNeuro), ECG (UCD Sleep Apnea Database)	Probability of error	0.5–0.2–0.4 0.47–0.28 0.5 - < 0.3		IR ratios not reported.	Requires structural assumptions (vMRF model)	Effective for very small datasets (3–5 samples), outperforms SMOTE	

data.

GANSO, being a GAN-based approach, also faces scalability challenges as the dimensionality of the data increases. The computational complexity of training the generative and discriminative blocks in GANSO grows with the size of the extended graph representation, which can impact its efficiency when dealing with high-dimensional data such as fMRI. However, for lower-dimensional signals like EEG and ECG, GANSO may be less affected by scalability issues compared to models handling image data. The incorporation of the vMRF model in GANSO introduces additional parameters and complexity, further impacting its scalability as the number of nodes and edges grows with the data dimensionality.

On the other hand, improved VAEGAN suffers from similar scalability issues as VAE and GAN with increasing data dimensions. While the dual encoders compensate for this constraint to some extent, the training complexity grows rapidly making large scale training infeasible.

5.2. Training complexity

GAN training is challenging in itself. Phenomena such as mode collapse, non-convergence or high variational training, and sensitivity to hyperparameters are common issues. The simple GAN training mechanism of the interaction between the two networks of generator and discriminator also induces and exacerbates these training problems. Training CGAN is similar to GAN and involves the interaction between the two networks, hence facing similar issues including mode collapse, training instability and sensitivity to hyperparameter selection. The extra conditional input can provide some guided training.

VAEs face more complexity during inference and loss computation compared to GANs. So, the optimization requires techniques like KL annealing to prevent problems like instability and divergence. CVAE also needs specialized techniques like KL annealing for stable optimization similar to regular VAE. BAGAN training provides more stability owing to the autoencoder component used along with GAN. But it requires a two-step training process for the separate VAE and GAN sections adding to computational overhead and training time. Moreover, considerable exploration of the large parameter space is needed to attain harmonious tuning between the two sections.

ARAE training is more stable than regular GAN training since it uses an autoencoder for density estimation but very sensitive tuning of the equilibrium weighting factor λ is needed to prevent overfitting and properly capture useful features from the data, increasing model and hyperparameter tuning complexity.

The usage of the bagging technique and prior distribution information allows G-GAN training to be more stable and guided minimizing mode collapse. medGAN aims to mitigate common GAN training challenges using techniques like importance estimation, group normalization and generating data from a mixture of distributions leading to more stable and faster training, but issues like mode collapse still persist.

3D-HyperGAMO employs techniques like input noise injection and multi-scale prediction objectives for more robust training. It also uses a separate classifier for the generated data putting less pressure on the model itself. The usage of dual encoders increases computational complexity in improved VAEGAN and attaining harmonious tuning between them necessitates exploration of a vast parameter space. But the enhanced representation power somewhat compensates for this limitation.

GANSO, being a GAN-based approach, shares similar training challenges such as mode collapse, non-convergence, and sensitivity to hyperparameters. The iterative competition between the generative and discriminative blocks in GANSO can be computationally demanding, especially when dealing with large extended graph representations. The incorporation of the vMRF model adds an additional layer of complexity, requiring careful tuning of hyperparameters to ensure convergence and stability. The use of the Graph Fourier Transform (GFT) for generating synthetic samples also contributes to the training

complexity, as it involves eigendecomposition of the extended graph Laplacian matrix. However, the structured nature of the vMRF model can provide some regularization and guidance during the training process, potentially mitigating issues like mode collapse to some extent. Nevertheless, finding the right balance between the competing objectives of the generative and discriminative blocks remains a challenge in GANSO, similar to other GAN-based methods.

5.3. Potential biases

Since the primary goal of the generator in GAN is to fool the discriminator, it may be more prone to just imitating particular samples of the training data that it finds easiest to fool the discriminator with, failing to be representative of the actual data diversity especially for complex datasets with multiple modes.

CGAN has a similar goal of fooling the discriminator, hence likely to overfit to such biased patterns from the data like GAN [98,99]. VAEs also face the risk of overfitting to the observed training samples because of the reconstruction performed using the latent space, potentially leading to poor generalization ability. CVAE may also overfit to the data as it performs reconstruction of the dependent variable y , requiring proper hyperparameter tuning to prevent such bias.

BAGAN combines GAN and VAE that compensate for each other's limitations - VAE reconstructs data images and estimates the latent space increasing output sample diversity, while GAN's interactive adversarial training guarantees generated samples to appear realistic. So, this combination reduces chances of overfitting and bias.

ARAE uses an autoencoder, so like other generative content models based on autoencoders, it also faces considerable risks of overfitting to observed training samples and losing generalization power. Appropriate mechanisms like tuning the weighting of the equilibrium loss are needed to reduce bias. The usage of prior distribution information about minorities and bagging allows G-GAN's generator to be trained for producing samples more related to minorities. NOISE allows increasing diversity of produced samples and stretching coverage.

medGAN relies on strategies like estimating importance, generating data from a mixture of distributions instead of direct minorities distribution to enhance diversity and reduce bias of produced patterns. 3D-HyperGAMO employs a separate classifier to classify the generated samples, input noise injection, and multi-scale prediction as techniques to reduce pattern bias and improve their diversity. Using dual encoders and tuning their fusion by the VAE and GAN parts can potentially reduce bias in improved VAEGAN But risks of overfitting still exist necessitating proper tuning.

In the case of GANSO, the generator's objective of fooling the discriminator may lead to a bias towards generating samples that are easily confused with the real data, rather than capturing the full diversity of the minority class. The vMRF model used in GANSO aims to incorporate structural information and dependencies among the data points, which can help in generating more representative samples. However, if the assumed vMRF structure does not accurately reflect the true underlying relationships in the data, it may introduce its own biases. The iterative adversarial training process in GANSO can potentially mitigate some of these biases by continuously updating the generator based on the discriminator's feedback. Nevertheless, careful tuning of hyperparameters and regularization techniques may be necessary to strike a balance between generating realistic samples and maintaining diversity. The limited number of training samples in the minority class can also make GANSO vulnerable to overfitting, as the generator may focus on replicating specific patterns present in the observed data. Techniques such as data augmentation, cross-validation, and early stopping can be employed to reduce overfitting and improve generalization. Additionally, monitoring the diversity of the generated samples and using evaluation metrics that capture both the quality and variability of the synthetic data can help in identifying and mitigating potential biases in GANSO.

6. Shortcomings and potential risks

6.1. Generalization

GANs cannot cover full data diversity due to problems like mode collapse that reduce variation. They likely miss infrequent but critical minority cases. CGANs share the same deficiencies as GANs. The conditional input provides some guided training but does not ensure generalization or minority coverage. VAEs face latent space overfitting, hampering generalization to new test data. Posterior collapse reduces their output diversity. CVAEs risk overfitting on reconstruction, limiting generalization. High KL costs can make them disregard the latent code, affecting diversity. BAGAN combines GAN & VAE strengths to improve generalization but requires extensive tuning. Like autoencoders, ARAE can overfit to training data, requiring preventative tuning. G-GAN has scalability issues that constrain its generalization. Strategies in medGAN help generalization, but problems like mode collapse persist. Input injection for 3D-HyperGAMO does not guarantee accurate conformity to actual distribution. MoGAN leverages techniques like group normalization for stability and convolutional architectures for scalability. However, it shares GAN risks like mode collapse that limit full generalization, especially on minority classes. The dual encoder approach of improved VAEGAN aims for compensation but high complexity affects reliable large-scale generalization. GANSO's generalization ability depends on the effectiveness of the vMRF model in capturing the underlying structure of the minority class. If the assumed vMRF does not accurately represent the true data dependencies, GANSO may struggle to generate diverse and representative samples that generalize well to unseen data. The limited number of minority class instances can also hinder GANSO's generalization capability, as it may overfit to specific patterns in the training data.

6.2. Overfitting

GAN tends to overfit to particular training samples that easily fool the discriminator, losing diversity and generalization capability. CGAN shares the overfitting tendencies of regular GANs. VAE risks overfitting in its latent space, hampering generalization. Posterior collapse exacerbates this. CVAE can overfit on reconstruction of the dependent variable. High KL costs increase risks of posterior collapse and overfitting. BAGAN combines GAN and VAE strengths, compensating limitations to reduce overfitting. But extensive tuning for optimal fusion of the components is computationally demanding. Like autoencoders, ARAE can easily overfit to training data. Sensitive tuning of equilibrium loss weighting is required. G-GAN employs bagging and prior distribution information for more guided training with less overfitting. medGAN uses techniques like importance estimation from mixture densities to minimize overfitting risks. MoGAN exploits group normalization and convolutional architectures that provide regularization but does not eliminate underlying GAN overfitting risks [100]. Using a separate classifier in 3D-HyperGAMO reduces reliance on the model alone, mitigating overfitting pressures. Tuning dual encoders in improved VAEGAN targets reducing overfitting but significant tuning is essential. GANSO, like other GAN-based models, is susceptible to overfitting, especially when trained on a limited number of minority class samples. The generator may memorize specific patterns in the training data instead of learning the overall distribution, leading to poor generalization. The vMRF model in GANSO can provide some regularization, but it may not completely eliminate the overfitting risk if the assumed structure is too simplistic or fails to capture the true data dependencies.

In general, adequate regularization and tuning controls are vital in all models to account for overfitting risks. Complex models like BAGAN and GANSO require extensive tuning focus. Techniques like ARAE's equilibrium tuning, G-GAN's guided training, GANSO's vMRF regularization, and MoGAN's normalizer regularization help but no model is inherently immune to overfitting risks.

6.3. Evaluation metrics

GANs need customized metrics testing class-specific performance, biases, feature space coverage, and real-world impact compared to benchmarks. Accuracy metrics masking poor minority class handling and purely sample similarity judgments overlooking problems are inadequate. CGANs share concerns, needing minority-class focused analysis. Aggregate performance metrics are insufficient.

VAEs require careful class-level metrics analyzing minority handling to prevent inflated perceived metrics from overfitting issues like posterior collapse. Aggregate similarity metrics overlook generalization difficulties and must be supplemented with classification performance impact analysis. CVAEs conceal overfitting problems if reconstruction similarity alone is used without class-specific impact studies. Imbalanced classification metrics are mandatory.

BAGAN's high compute requirements affect properly tuning and cross-validating it for reliability. Customized hybrid evaluators analyzing GAN and VAE components separately on metrics like sample diversity before integrated model assessments are imperative. Proper analysis of fusion tuning impact is critical. ARAE needs customized frameworks ensuring analysis of training data memorization issues, equilibrium tuning impact, biases, and feature space coverage through both automated metrics and qualitative human assessment. Over-reliance on aggregate similarity metrics must be avoided.

G-GAN has narrow specific applicability limiting evaluations on real complex data. Tailored analysis for distribution characteristics of under-represented minorities is key. medGAN's group normalizers need balanced class-specific multi-scale evaluations. Mixture density training necessitates metrics ensuring minority coverage. 3D-HyperGAMO's separate classifier keeps models isolated necessitating customized evaluators revealing true enhancements in generative model capability. MoGAN's deep 3D-CNNs require procedures testing model handling when real-world variability is introduced in new unseen imbalanced cases. Extensive dual encoder tuning means improved VAEGAN needs customized non-aggregated metrics analyzing fusion model impact, not just similarity metrics.

For GANSO, evaluation metrics should focus on assessing the quality and diversity of the generated minority class samples and their impact on improving classification performance. Class-specific metrics, such as precision, recall, and F1-score, should be used to evaluate the classifier's performance on the minority class after oversampling. Additionally, metrics that capture the diversity and representativeness of the generated samples, such as MMD or FID, can provide insights into how well GANSO captures the underlying distribution. Visual inspection and domain expert feedback can also help assess the realism and plausibility of the generated samples in the specific application context.

7. Comparative analysis

Traditional oversampling methods such as SMOTE, ADASYN and random oversampling take a simple approach of rebalancing class distributions in imbalanced datasets by increasing minority class samples through replication or generation from existing samples. While computationally cheap and intuitive, they tend to overfit limited minority data and lack diversity. Their scope is also mostly restricted to low-dimensional data. In contrast, deep generative models like GANs, VAEs, CGANs, CVAEs, BAGAN, medGAN, ARAE, G-GAN, MoGAN, 3D-HyperGAMO, Improved VAEGAN, and GANSO leverage complex neural networks to implicitly model intricate data distributions and sample new, realistic and diverse points from it. By capturing the authentic underlying statistics, they significantly reduce overfitting and provide better generalization over traditional techniques. These powerful black-box models can scale to high-dimensional visual data, but require extensive expertise and computing resources for effective training and tuning due to architectural complexity.

An important observation from Table 1 is the varying performance of

different methods across imbalance ratios (IR). Traditional methods like SMOTE and Random oversampling show high sensitivity to imbalance ratios, with SMOTE's F1-scores dropping significantly from 0.905 at IR= 80–0.392 at IR= 320. Deep generative approaches demonstrate varying levels of resilience to high imbalance ratios. Some models show remarkable stability, such as Improved VAEGAN maintaining consistent performance (F1 \approx 0.87) from IR= 0.25 to IR= 100, and BAGAN showing minimal degradation (Accuracy dropping only 2.5 % from IR=1.67 to IR=40). However, others like CGAN exhibit high sensitivity with F1-score declining substantially from 0.816 at IR= 80–0.147 at IR= 320. Models like MoGAN and 3D-HyperGAMO show moderate sensitivity, with MoGAN's F1-score decreasing from 0.95 to 0.71 at extreme ratios (IR=1:1000) and 3D-HyperGAMO's accuracy declining from 95.19 % to 85.95 % as IR increases from 10 to 123. Notably, some models like medGAN and G-GAN demonstrate low sensitivity, even showing improved performance at higher imbalance ratios. This varying response to imbalance ratios suggests that while some deep generative models can effectively handle class imbalances, their effectiveness varies significantly across different architectures and domains. These findings align with the sensitivity ratings provided in Table 2, where deep learning-based methods show a spectrum from low to high sensitivity to imbalance ratios, with newer architectures generally showing better resilience to high imbalance scenarios.

To facilitate a multi-dimensional comparison of these oversampling strategies, we present Table 2 which evaluates both traditional and deep generative approaches across several key criteria. These criteria were selected to provide a comprehensive view of each method's strengths and limitations in addressing the challenges of imbalanced data classification.

The criteria in Table 2 offer insights into various aspects of each method's performance and applicability. The number of parameters indicates model complexity and computational demands, ranging from non-parametric approaches to architectures with large-scale parameters. This parameter count directly impacts both training time and memory requirements, with larger models typically requiring extended training periods on specialized hardware, while smaller models can be trained more quickly on standard computing resources. Data scarcity robustness indicates performance with limited samples. The data type handled shows the versatility of each approach. Sensitivity to high imbalance ratio is crucial for severely skewed datasets. Generated data diversity is essential for creating representative synthetic samples. Scalability assesses the method's applicability to large-scale problems.

Specifically, GANs can produce high quality, realistic image samples

with intricate visual details and diversity through their moderate-sized generator-discriminator networks, but adversarial training often causes mode collapse and diminished diversity. VAEs enable interpretable latent manipulation and reasoning with their compact encoder-decoder structure, but reconstruct complex images poorly. CGANs and CVAEs, with their larger architectures through additional class embedding layers, improve conditional sample specificity yet still inherit GAN training instability issues like mode collapse. BAGAN effectively captures the heterogeneous and sparse spatial-spectral distributions of satellite imagery with few minority samples, utilizing substantial parameters shared between autoencoder and GAN components, but requires data preprocessing and feature extraction and cannot operate on raw data. medGAN employs a medium-sized architecture and is robust to severe imbalance in medical records but has binary classification constraints and requires structured data. ARAE can generate diverse texts conditioned on labels through its LSTM-based framework but demands complex architectures. G-GAN, with its lightweight structure, generates new minority samples from imbalanced numerical data, though it struggles with extreme imbalance ratios. MoGAN enables simultaneous machine fault detection and classification through specialized convolutional architectures but is domain limited. 3D-HyperGAMO improves spectral-spatial satellite image classification performance but needs more complex neural architectures and higher computational costs. Improved VAEGAN combines two generators for better fraud sample generation at the cost of increased complexity over vanilla VAEs and GANs. GANSO maintains a highly efficient structure through its vMRF-based approach while combining the strengths of GANs and vector Markov Random Fields to synthesize realistic minority class samples even with extremely small training sets. It demonstrates improved classifier performance over traditional methods when dealing with very limited data through iterative adversarial training, though it requires careful vMRF structure design. GANSO shows particular promise for biomedical applications involving EEG, fMRI, and ECG data, where obtaining large labeled datasets is often difficult and expensive, despite potential scalability challenges with high-dimensional data.

Among traditional methods, SMOTE is a simple and easy to implement method for increasing the number of minority samples but overfits limited minority data and lacks diversity of generated samples as well as ineffectiveness in higher dimensions and continuous data like images. While ADASYN performs targeted sampling based on difficulty of learning and prevents overfitting, scarce diversity of generated samples is observed. Both amplify existing samples. Random oversampling establishes implementation simplicity and class balance but likely causes

Table 2
Multi-dimensional Comparison of Oversampling Strategies.

Model	Number of Parameters	Data Scarcity Robustness	Data Type Handled	Sensitivity to High Imbalance Ratio	Generated Data Diversity	Scalability
SMOTE	N/A (Non-parametric)	Low	Numerical, Low-dimensional	High	Low	High
ADASYN	N/A (Non-parametric)	Low	Numerical, Low-dimensional	Medium	Low	High
RANDOM	N/A (Non-parametric)	Low	Image, Text	High	Low	High
GAN	~1.1 M	Medium	Image, Text	Medium	High	Medium
CGAN	~13.18 M	Medium	Image, Text, Conditional	High	High	Medium
VAE	~0.85 M	Medium	Image, Text	Medium	Medium	Medium
CVAE	~13.34 M	Medium	Image, Text, Conditional	Medium	Medium	Medium
BAGAN	~13.15 M	High	Image	Low	High	Low
medGAN	~5 M	High	Discrete Medical Data	Low	High	Medium
ARAE	~1.65 M	Medium	Text, Discrete Data	Medium	High	Medium
G-GAN	~3.2 K	High	Numerical, Low-dimensional	Low	Medium	Low
MoGAN	~1.5 M	High	Time Series	Medium	High	Medium
3D-HyperGAMO	~8.5 M	High	Hyperspectral Images	Medium	High	Low
Improved VAEGAN	~131 K	High	Numerical	Low	High	Medium
GANSO	~494	High	Time Series, Image	Medium	High	Medium

overfitting due to replication without adding new information to training data.

Comparing these methods presents several challenges due to the diversity of data types, imbalance ratios, and application domains. This diversity makes it impractical to provide a single set of experimental results that would be universally applicable or meaningful across all scenarios. Performance can vary significantly based on these factors, making direct comparisons complex. For instance, while deep generative models excel in high-dimensional spaces, traditional methods may be more effective for simpler, low-dimensional data. The choice of method often depends on the specific requirements of the task, available computational resources, and the nature of the dataset. Given these challenges, a more effective approach to comparing these methods involves analyzing their theoretical foundations, computational requirements, and reported performance across various case studies in the literature, rather than attempting to provide a single set of experimental results. Consequently, we have focused on comparing the general characteristics and applicability of these methods across various scenarios.

In summary, while traditional techniques provide an accessible starting point, they falter on complex high-dimensional data. Deep generative models can accurately model intricate data distributions for intelligent sampling but require greater expertise and computing resources to harness their power. These complementary paradigms can be innovatively combined into robust and sample-efficient solutions that augment minority classes effectively across the data complexity spectrum.

8. Future directions and recommendations

For BAGAN future efforts could focus on applying the methodology to additional classification tasks and datasets with imbalanced training data to further validate its usefulness. Researchers could also explore modifications or extensions to the BAGAN framework itself to generate higher quality synthetic images and improve classifier performance.

For medGAN priority areas are developing a sequential version that can model patient data over time and expanding the types of data that medGAN can synthesize, including lab results, demographics, and free-text notes. This will enhance the realism and usefulness of the generated outputs.

Researchers could evaluate ARAE's scalability on more complex long-form discrete sequences like documents to test robustness. Reducing gaps between practical training objectives and theoretical optimal transport objectives could enhance performance. Additional latent space regularization and structuring may enable more controlled manipulations with improved interpretability. Comparing ARAE against a wider range of recent discrete latent variable models would better highlight ARAE's strengths and limitations to guide progress.

In terms of recommendations, researchers could release ARAE code implementations to accelerate research. Testing model resilience on real-world discrete datasets would also evaluate scalability. Clearly reporting differences between theoretical and practical training objectives would support reproducibility.

Further suggestions include adapting ARAE for more discrete sequence modeling tasks beyond initial experiments to demonstrate versatility. Enhancing discretization schemes for modalities like video or audio could widen applications. Developing specialized ARAE model architectures and training procedures for different data types could also improve results. Comprehensive benchmarks gauging reconstruction quality, manipulation precision, and computational efficiency would identify areas needing work.

Researchers can stretch the idea of G-GAN by extracting more information to inject into the GAN, adapting it for multi-class imbalance problems, and improving stability for small numerical datasets. Comparing against newer GAN-based solutions will also help advance this approach.

For MoGAN reducing dimensionality could improve fault diagnosis under different conditions. Creating a new loss function to handle mixture data distributions would enrich generator performance and improve the discriminator's detection abilities.

With 3D-HyperGAMO testing how the model generalizes to new hyperspectral datasets and comparing against recent deep learning techniques like attention-based networks could further guide architecture optimization. Researchers could also explore semi-supervised extensions and adapt the 3D oversampling idea to other modalities.

For the improved VAEGAN boosting recall and AUC via model architecture changes and advanced sampling strategies should be priorities. Testing on more real-world problems will demonstrate broad usefulness across classification tasks involving imbalance.

For GANSO, future research could focus on optimizing the vMRF structure design for different data types and domains. Investigating adaptive or learnable vMRF structures could enhance the method's flexibility and generalization ability. Extending GANSO to handle multi-class imbalance problems and exploring its applicability to other data modalities beyond EEG, fMRI, and ECG would broaden its impact. Researchers could also work on improving GANSO's scalability to higher-dimensional data and larger datasets. Comparative studies against state-of-the-art few-shot learning methods would help position GANSO within the broader context of learning from limited data. Additionally, developing theoretical guarantees for GANSO's performance under different data distributions and imbalance ratios could provide valuable insights. Exploring the integration of GANSO with other deep learning architectures, such as attention mechanisms or graph neural networks, might lead to improved performance in specific domains.

9. Conclusion

This survey has analyzed various oversampling techniques using deep generative models for handling class imbalance in machine learning. Compared to traditional methods like SMOTE and ADASYN that simply replicate or interpolate minority samples, advanced deep models such as GANs, VAEs, and their variants (CGANs, CVAEs, BAGAN, medGAN, ARAE, G-GAN, MoGAN, 3D-HyperGAMO, Improved VAEGAN, and GANSO) have demonstrated significant potential in capturing intricate minority data distributions for intelligent, diverse sample generation.

These deep generative approaches have shown promising results across various domains, including image classification, medical diagnostics, and fraud detection. Their ability to generate high-quality, realistic samples that maintain the underlying statistical properties of minority classes represents a substantial advancement in addressing class imbalance. For instance, models like BAGAN and 3D-HyperGAMO have shown particular promise in handling imbalanced image datasets, while medGAN and GANSO have demonstrated effectiveness in generating synthetic medical data and time series, respectively.

However, it is important to acknowledge that these advanced methods also present new challenges. Scalability issues due to architectural complexity can constrain real-world applicability, especially for very large datasets. The delicate training procedures of some models, prone to instability and overfitting, necessitate extensive tuning and expertise. Additionally, the use of aggregate evaluation metrics may sometimes mask model deficiencies, calling for more nuanced, class-specific assessments.

Despite these challenges, the field has made substantial progress. Innovative solutions combining complementary paradigms, such as the integration of GANs with vector Markov Random Fields in GANSO, or the dual encoder approach in improved VAEGAN, showcase the potential for oversampling techniques that can scale across data complexity levels. These advancements are particularly crucial as datasets grow more imbalanced and complex, necessitating robust, efficient, and transparent oversampling methods for unbiased model development.

Looking forward, addressing these critical shortcomings is

imperative before large-scale deployment. Innovative solutions combining complementary paradigms could lead to oversampling techniques that scale across data complexity. As datasets grow more imbalanced, developing robust, efficient and transparent oversampling is vital for unbiased model development. Techniques accurately representing minority groups in training data will be integral to ensuring fair and ethical AI systems.

In conclusion, while challenges remain, the progress in deep generative oversampling techniques represents a significant step forward in addressing class imbalance. As these methods continue to evolve, they hold great promise for enabling more accurate and fair machine learning models across a wide range of applications. The path forward involves interdisciplinary efforts uniting deep generative modeling expertise with domain knowledge and ethical considerations. By continuing to innovate and refine these techniques, we can work towards AI systems that are not only powerful but also equitable and representative of all data classes.

Ethical and informed consent for data used

No ethical concern exists.

Funding

The authors declare that no funding was received for this research.

CRediT authorship contribution statement

Mozafar Hayaieian Shirvan: Writing – original draft, Investigation, Formal analysis. **Mohammad Moattar:** Writing – review & editing, Methodology, Formal analysis. **Mehdi Hosseinzadeh:** Writing – review & editing, Resources.

Declaration of Competing Interest

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

Acknowledgements

Not applicable.

Authors contribution statement

All authors contribute to the preparation of the article.

Use of experimental animals, and human participants

No animals and human participants were involved in this study in any kind.

Data Availability

No data was used for the research described in the article.

References

- [1] V.A. Fajardo, et al., On oversampling imbalanced data with deep conditional generative models, *Expert Syst. Appl.* 169 (May 2021), <https://doi.org/10.1016/j.eswa.2020.114463>.
- [2] G. Douzas, F. Bacao, Effective data generation for imbalanced learning using conditional generative adversarial networks, *Expert Syst. Appl.* 91 (Jan. 2018) 464–471, <https://doi.org/10.1016/j.eswa.2017.09.030>.
- [3] R.A. Nugraha, H.F. Pardele, A. Subekti, Oversampling based on generative adversarial networks to overcome imbalance data in predicting fraud insurance claim, *Kuwait J. Sci.* 49 (2022), <https://doi.org/10.48129/kjs.splml.19119>.
- [4] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” 2002.
- [5] T. Feizi, M.H. Moattar, H. Tabatabaee, M2GDL: Multi-manifold guided dictionary learning based oversampling and data validation for highly imbalanced classification problems, *Inf. Sci.* 682 (2024) 121280, <https://doi.org/10.1016/j.ins.2024.121280>.
- [6] T. Feizi, M.H. Moattar, H. Tabatabaee, A multi-manifold learning based instance weighting and under-sampling for imbalanced data classification problems, *J. Big Data* 10 (1) (Dec. 2023) 1–36, <https://doi.org/10.1186/S40537-023-00832-2/TABLES/24>.
- [7] M. Buda, A. Maki, M.A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, *Neural Netw.* 106 (Oct. 2018) 249–259, <https://doi.org/10.1016/J.NEUNET.2018.07.011>.
- [8] L.J. Goodfellow et al., “Generative Adversarial Networks,” Jun. 2014, [Online]. Available: (<http://arxiv.org/abs/1406.2661>).
- [9] A.K. Gangwar, V. Ravi, WiP: Generative Adversarial Network for Oversampling Data in Credit Card Fraud Detection, in: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11952, LNCS, 2019, pp. 123–134, https://doi.org/10.1007/978-3-030-36945-3_7.
- [10] W. Jo, D. Kim, OBGAN: Minority oversampling near borderline with generative adversarial networks, *Expert Syst. Appl.* 197 (Jul. 2022), <https://doi.org/10.1016/j.eswa.2022.116694>.
- [11] O. Dayan, L. Wolf, F. Wang, Y. Harel, Optimizing AI for Mobile Malware Detection by Self-Built-Dataset GAN Oversampling and LGBM, *Proc. 2023 IEEE Int. Conf. Cyber Secur. Resil., CSR 2023* (2023) 60–65, <https://doi.org/10.1109/CSR57506.2023.10224927>.
- [12] D.P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” Dec. 2013, [Online]. Available: (<http://arxiv.org/abs/1312.6114>).
- [13] C. Liu, X. Wang, K. Wu, J. Tan, F. Li, W. Liu, Oversampling for imbalanced time series classification based on generative adversarial networks, 2018 IEEE 4th Int. Conf. Comput. Commun., ICC 2018 (2018) 1104–1108, <https://doi.org/10.1109/CompComm.2018.8780808>.
- [14] W. Bouzeraib, A. Ghenai, N. Zeghib, A Multi-Objective Genetic GAN Oversampling: Application to Intelligent Transport Anomaly Detection, *Proc. - 2020 IEEE 22nd Int. Conf. High. Perform. Comput. Commun., IEEE 18th Int. Conf. Smart City IEEE 6th Int. Conf. Data Sci. Syst., HPCC-SmartCity-DSS 2020* (2020) 1142–1149, <https://doi.org/10.1109/HPCC-SmartCity-DSS50907.2020.00148>.
- [15] T. Miftahshudur, B. Grieve, H. Yin, Permuted KPCA and SMOTE to Guide GAN-based oversampling for imbalanced HSI Classification, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens* 17 (2024) 489–505, <https://doi.org/10.1109/JSTARS.2023.3326963>.
- [16] H. Ba, “Improving Detection of Credit Card Fraudulent Transactions using Generative Adversarial Networks,” Jul. 2019, [Online]. Available: (<http://arxiv.org/abs/1907.03355>).
- [17] A. Salazar, L. Vergara, G. Safont, Generative adversarial networks and markov random fields for oversampling very small training sets, *Expert Syst. Appl.* 163 (Jan. 2021), <https://doi.org/10.1016/j.eswa.2020.113819>.
- [18] S. Barutcu, A.K. Katsaggelos, and D. Gürsoy, “A Deep Generative Approach to Oversampling in Ptychography,” Jul. 2022, [Online]. Available: (<http://arxiv.org/abs/2207.14392>).
- [19] M. Dierolf, et al., Ptychography & lensless X-ray imaging, *Europhys. N.* 39 (1) (Jan. 2008) 22–24, <https://doi.org/10.1051/EPN:2008003>.
- [20] J. Miao, R.L. Sandberg, C. Song, Coherent x-ray diffraction imaging, *IEEE J. Sel. Top. Quantum Electron.* 18 (1) (2012) 399–410, <https://doi.org/10.1109/JSTQE.2011.2157306>.
- [21] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot, *Sens. (Switz.)* 17 (9) (Sep. 2017), <https://doi.org/10.3390/s17091967>.
- [22] E. Choi, S. Biswal, B. Malin, J. Duke, W.F. Stewart, and J. Sun, “Generating Multi-label Discrete Patient Records using Generative Adversarial Networks,” Mar. 2017, [Online]. Available: (<http://arxiv.org/abs/1703.06490>).
- [23] A. Salazar, L. Vergara, G. Safont New applications of an oversampling method based on generative adversarial networks *Proc. - 2020 Int. Conf. Comput. Sci. Comput. Intell., CSCI 2020*, in *Proceedings - 2020 International Conference on Computational Science and Computational Intelligence, CSCI 2020*, 2020, pp. 1699 – 1701. doi: 10.1109/CSCI51800.2020.00314.2020.
- [24] W. Zhang, X. Li, X.D. Jia, H. Ma, Z. Luo, X. Li, Machinery fault diagnosis with imbalanced data using deep generative adversarial networks, *Measurement* 152 (Feb. 2020) 107377, <https://doi.org/10.1016/J.MEASUREMENT.2019.107377>.
- [25] M. Arjovsky and L. Bottou, “Towards Principled Methods for Training Generative Adversarial Networks,” Jan. 2017, [Online]. Available: (<http://arxiv.org/abs/1701.04862>).
- [26] C. Liao, M. Dong, Acwgan: an auxiliary classifier wasserstein gan-based oversampling approach for multi-class imbalanced learning, *Int. J. Innov. Comput., Inf. Control* 18 (3) (2022) 703–721, <https://doi.org/10.24507/ijic.18.03.703>.
- [27] K. Sohn, X. Yan, and H. Lee, “Learning Structured Output Representation using Deep Conditional Generative Models.”
- [28] Z. Han, C. Shen, Y. Zhang, H. Wang, L. Yu, Data-driven fault detection of rotating machinery using synthetic oversampling and generative adversarial network, *Proc. SPIE - Int. Soc. Opt. Eng.* (2023), <https://doi.org/10.1117/12.2674215>.
- [29] Y. Zhang, Y. Liu, Y. Wang, J. Yang, An ensemble oversampling method for imbalanced classification with prior knowledge via generative adversarial

- network, *Chemom. Intell. Lab. Syst.* 235 (Apr. 2023), <https://doi.org/10.1016/j.chemolab.2023.104775>.
- [30] R.D. Camino, R. State, and C.A. Hammerschmidt, "Oversampling Tabular Data with Deep Generative Models: Is it worth the effort?" [Online]. Available: (<http://archive.ics.uci.edu/ml/datasets/adult>).
- [31] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," Nov. 2014, [Online]. Available: (<http://arxiv.org/abs/1411.1784>).
- [32] Y. Ding, W. Kang, J. Feng, B. Peng, A. Yang, Credit card fraud detection based on improved variational autoencoder generative adversarial network, *IEEE Access* 11 (2023) 83680–83691, <https://doi.org/10.1109/ACCESS.2023.3302339>.
- [33] S.K. Roy, J.M. Haut, M.E. Paolletti, S.R. Dubey, A. Plaza, Generative adversarial minority oversampling for spectral-spatial hyperspectral image classification, *IEEE Trans. Geosci. Remote Sens.* 60 (2022), <https://doi.org/10.1109/TGRS.2021.3052048>.
- [34] A. Salazar, L. Vergara, E. Vidal, A proxy learning curve for the Bayes classifier, *Pattern Recognit.* 136 (Apr. 2023), <https://doi.org/10.1016/j.patcog.2022.109240>.
- [35] A. Koivu, M. Sairanen, A. Airola, T. Pahikkala, Synthetic minority oversampling of vital statistics data with generative adversarial networks, *J. Am. Med. Inform. Assoc.* 27 (11) (Nov. 2020) 1667–1674, <https://doi.org/10.1093/jamia/ocaa127>.
- [36] Z. Wei, Y. Fu, W. Shi, D. Chen Oversampling algorithm based on generative adversarial network in Proceedings of SPIE - The International Society for Optical Engineering, 2023, 10.1117/12.2684591.
- [37] A. Fernández, S. García, F. Herrera, and N.V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," 2018.
- [38] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: review of methods and applications, *Expert Syst. Appl.* 73 (May 2017) 220–239, <https://doi.org/10.1016/j.eswa.2016.12.035>.
- [39] Z. Pu, D. Cabrera, R.-V. Sánchez, M. Cerrada, C. Li, J.V. de Oliveira, Exploiting generative adversarial networks as an oversampling method for fault diagnosis of an industrial robotic manipulator, *Appl. Sci.* (Switz.) 10 (21) (2020) 1–21, <https://doi.org/10.3390/app10217712>.
- [40] Y. Guo, G. Xiong, Z. Li, J. Shi, M. Cui, G. Gou Combating imbalance in network traffic classification using gan based oversampling 2021 IFIP Netw. Conf., IFIP Netw. 2021, in 2021 IFIP Networking Conference, IFIP Networking 2021, 2021. doi: 10.23919/IFIPNetworking52078.2021.94727772021.
- [41] A. Fernández, V. López, M. Galar, M.J. Del Jesus, F. Herrera, Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches, *Knowl. Based Syst.* 42 (2013) 97–110, <https://doi.org/10.1016/j.knsys.2013.01.018>.
- [42] H. He, E.A. Garcia, Learning from imbalanced data, *IEEE Trans. Knowl. Data Eng.* 21 (9) (2009) 1263–1284, <https://doi.org/10.1109/TKDE.2008.239>.
- [43] W. Han, Z. Huang, S. Li, Y. Jia, Distribution-sensitive unbalanced data oversampling method for medical diagnosis, *J. Med Syst.* 43 (2) (2019) 1–10, <https://doi.org/10.1007/S10916-018-1154-8>.
- [44] J. Van Hulse, T.M. Khoshgoftaar, and A. Napolitano, "Experimental Perspectives on Learning from Imbalanced Data."
- [45] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A Rev. Ensembles Cl. imbalance Probl.: Bagging-, Boost-, Hybrid-. Based Approaches (2012), <https://doi.org/10.1109/TSMC.2011.2161285>.
- [46] B. Krawczyk, Learning from imbalanced data: open challenges and future directions, *Prog. Artif. Intell.* 5 (4) (2016) 221–232, <https://doi.org/10.1007/S13748-016-0094-0/TABLES/1>.
- [47] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, *SIGKDD Explor. News.* 6 (1) (2004) 20–29.
- [48] S. Ndichu, T. Ban, T. Takahashi, D. Inoue Security-Alert Screening with Oversampling Based on Conditional Generative Adversarial Networks Proc. - 2022 17th Asia Jt. Conf. Inf. Secur., AsiaJCS 2022, in Proceedings - 2022 17th Asia Joint Conference on Information Security, AsiaJCS 2022, 2022, pp. 1 – 7. doi: 10.1109/AsiaJCS57030.2022.000112022.
- [49] W. Juanjuan, X. Mantao, W. Hui, Z. Jiwu, Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding, in: International Conference on Signal Processing Proceedings, 3, ICSP, 2007, <https://doi.org/10.1109/ICOSP.2006.345752>.
- [50] Q. Kang, X.S. Chen, S.S. Li, M.C. Zhou, A noise-filtered under-sampling scheme for imbalanced classification, *IEEE Trans. Cyber* 47 (12) (2017) 4263–4274, <https://doi.org/10.1109/TCYB.2016.2606104>.
- [51] W.W.Y. Ng, J. Hu, D.S. Yeung, S. Yin, F. Roli, Diversified sensitivity-based undersampling for imbalance classification problems, *IEEE Trans. Cyber* 45 (11) (2015) 2402–2412, <https://doi.org/10.1109/TCYB.2014.2372060>.
- [52] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," 2005.
- [53] D.A. Cieslak, T.R. Hoens, N.V. Chawla, W.P. Kegelmeyer, Hellinger distance decision trees are robust and skew-insensitive, *Data Min. Knowl. Discov.* 24 (1) (2011) 136–158, <https://doi.org/10.1007/S10618-011-0222-1>.
- [54] H.M. Nguyen, E.W. Cooper, K. Kamei, Borderline over-sampling for imbalanced data classification, *Int J. Knowl. Eng. Soft Data Parad.* 3 (1) (2011) 4, <https://doi.org/10.1504/IJKESDP.2011.039875>.
- [55] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, *Proc. Int. Jt. Conf. Neural Netw.* (2008) 1322–1328, <https://doi.org/10.1109/IJCNN.2008.4633969>.
- [56] G. Kovács, Smote-variants: a python implementation of 85 minority oversampling techniques, *Neurocomputing* 366 (2019) 352–354, <https://doi.org/10.1016/j.neucom.2019.06.100>.
- [57] R. Blagus, L. Lusa, SMOTE for high-dimensional class-imbalanced data, *BMC Bioinforma.* 14 (2013) 106, <https://doi.org/10.1186/1471-2105-14-106>.
- [58] B. Das, N.C. Krishnan, D.J. Cook, RACOG and wRACOG: two probabilistic oversampling techniques, *IEEE Trans. Knowl. Data Eng.* 27 (1) (2015) 222–234, <https://doi.org/10.1109/TKDE.2014.2324567>.
- [59] S. Bond-Taylor, A. Leach, Y. Long, C.G. Willcocks, Deep generative modelling: a comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models, *IEEE Trans. Pattern Anal. Mach. Intell.* 44 (11) (2022) 7327–7347, <https://doi.org/10.1109/TPAMI.2021.3116668>.
- [60] X. Mao, Q. Li, H. Xie, R.Y.K. Lau, Z. Wang, S.P. Smolley, Least squares generative adversarial networks, *Proc. IEEE Int. Conf. Comput. Vis.* 2017-October (2017) 2813–2821, <https://doi.org/10.1109/ICCV.2017.304>.
- [61] A. Creswell, A.A. Bharath, Inverting the generator of a generative adversarial network, *IEEE Trans. Neural Netw. Learn Syst.* 30 (7) (2016) 1967–1974, <https://doi.org/10.1109/TNNLS.2018.2875194>.
- [62] Y. Sun, M.S. Kamel, A.K.C. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, *Pattern Recognit.* 40 (12) (2007) 3358–3378, <https://doi.org/10.1016/j.patcog.2007.04.009>.
- [63] Y.O. Lee, J. Jo, J. Hwang Application of deep neural network and generative adversarial network to industrial maintenance: A case study of induction motor fault detection Proc. - 2017 IEEE Int. Conf. Big Data, Big Data 2017, Vol. 2018-Jan., Proceedings - 2017 IEEE International Conference on Big Data, Big Data 2017, vol. 2018-January, pp. 3248–3253, Jul. 2017, doi: 10.1109/BIGDATA.2017.8258307. Jul. 2017, 3248325310.1109/BIGDATA.2017.8258307.
- [64] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification, *Neurocomputing* 321 (Dec. 2018) 321–331, <https://doi.org/10.1016/j.neucom.2018.09.013>.
- [65] T. Zhou, W. Liu, C. Zhou, L. Chen GAN-based semi-supervised for imbalanced data classification 2018 4th Int. Conf. Inf. Manag., ICIM 2018; 2018 4th International Conference on Information Management, ICIM 2018, pp. 17–21, Jun. 2018, doi: 10.1109/INFOMAN.2018.8392662. Jun. 2018, 172110.1109/INFOMAN.2018.8392662.
- [66] J. Yang, Z. Jiang, T. Pan, Y. Chen, W. Pedrycz, Oversampling method based on GAN for tabular binary classification problems, *Intell. Data Anal.* 27 (5) (2023) 1287–1308, <https://doi.org/10.3233/JDA-220383>.
- [67] G. Kovács, An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets, *Appl. Soft Comput.* 83 (Oct. 2019), <https://doi.org/10.1016/j.asoc.2019.105662>.
- [68] Y. Yan, R. Liu, Z. Ding, X. Du, J. Chen, Y. Zhang, A parameter-free cleaning method for SMOTE in imbalanced classification, *IEEE Access* 7 (2019) 23537–23548, <https://doi.org/10.1109/ACCESS.2019.2899467>.
- [69] N. Verbiest, E. Ramentol, C. Cornelis, F. Herrera, Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection, *Appl. Soft Comput.* 22 (2014) 511–517, <https://doi.org/10.1016/j.asoc.2014.05.023>.
- [70] S. Maldonado, J. López, C. Vairetti, An alternative SMOTE oversampling strategy for high-dimensional datasets, *Appl. Soft Comput.* 76 (Mar. 2019) 380–389, <https://doi.org/10.1016/j.asoc.2018.12.024>.
- [71] M. Perez-Ortiz, P.A. Gutierrez, P. Tino, C. Hervás-Martínez, Oversampling the minority class in the feature space, *IEEE Trans. Neural Netw. Learn Syst.* 27 (9) (Sep. 2016) 1947–1961, <https://doi.org/10.1109/TNNLS.2015.2461436>.
- [72] F.A. Ghaleb, F. Saeed, M. Al-Sarem, S.N. Qasem, T. Al-Hadhrani, Ensemble synthesized minority oversampling-based generative adversarial networks and random forest algorithm for credit card fraud detection, *IEEE Access* 11 (2023) 89694–89710, <https://doi.org/10.1109/ACCESS.2023.3306621>.
- [73] H. Tan, Tabular GAN-based oversampling of imbalanced time-to-event data for survival prediction 2023 8th Int. Conf. Cloud Comput. Big Data Anal., ICCCBDA 2023; in 2023 8th International Conference on Cloud Computing and Big Data Analytics, ICCCBDA 2023, 2023, pp. 376 – 380. doi: 10.1109/ICCCBDA56900.2023.101548832023.
- [74] J. Kim, H. Park, Reduced CNN model for face image detection with gan oversampling, *Lect. Notes Netw. Syst.* 279 (2022) 232–241, https://doi.org/10.1007/978-3-030-79728-7_23.
- [75] N. Abedzadeh, M. Jacobs GANMCMCRO: A generative adversarial network markov chain Monte Carlo random oversampling algorithm for imbalance datasets; in International Conference on Web Information Systems and Technologies, WEBIST - Proceedings, 2023, pp. 587 – 594. doi: 10.5220/00122596000035842023.
- [76] J.-H. Oh, J.Y. Hong, J.-G. Baek, Oversampling method using outlier detectable generative adversarial network, *Expert Syst. Appl.* 133 (2019) 1–8, <https://doi.org/10.1016/j.eswa.2019.05.006>.
- [77] S. Yang, Y. Zhou, X. Chen, C. Deng, C. Li, Fault diagnosis of wind turbines with generative adversarial network-based oversampling method, *Meas. Sci. Technol.* 34 (4) (2023), <https://doi.org/10.1088/1361-6501/acad20>.
- [78] A. Majeed, S.O. Hwang, CTGAN-MOS: conditional generative adversarial network based minority-class-augmented oversampling scheme for imbalanced problems, *IEEE Access* 11 (2023) 85878–85899, <https://doi.org/10.1109/ACCESS.2023.3303509>.
- [79] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, F.Y. Wang, Generative adversarial networks: Introduction and outlook, *IEEE/CAA J. Autom. Sin.* 4 (4) (2017) 588–598, <https://doi.org/10.1109/JAS.2017.7510583>.
- [80] A. Anand, K. Gorde, J.R. Antony Moniz, N. Park, T. Chakraborty, B.-T. Chu Phishing URL detection with oversampling based on text generative adversarial networks Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018; in Proceedings -

- 2018 IEEE International Conference on Big Data, Big Data 2018, 2018, pp. 1168–1177. doi: 10.1109/BigData.2018.86225472018.
- [81] M.S. Munia, M. Nourani, S. Houari, Biosignal Oversampling Using Wasserstein Generative Adversarial Network. 2020 IEEE International Conference on Healthcare Informatics, ICHI 2020, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, <https://doi.org/10.1109/ICHI48887.2020.9374315>.
- [82] J. Engelmann, S. Lessmann, Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning, *Expert Syst. Appl.* 174 (2021), <https://doi.org/10.1016/j.eswa.2021.114582>.
- [83] E. Nazari, P. Branco On oversampling via generative adversarial networks under different data difficulty factors; in *Proceedings of Machine Learning Research*, 2021, pp. 76 – 89. [Online]. Available: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85120163757&partnerID=40&md5=5f4e969e9bc436314021043f34fb39fc2021,7689>[Online]. Available.
- [84] Y. Dong, H. Xiao, Y. Dong, SA-CGAN: An oversampling method based on single attribute guided conditional GAN for multi-class imbalanced learning, *Neurocomputing* 472 (2022) 326–337, <https://doi.org/10.1016/j.neucom.2021.04.135>.
- [85] J. Hao, C. Wang, H. Zhang, G. Yang Annealing Genetic GAN for Minority Oversampling 31st Br. Mach. Vis. Conf., BMVC 2020; in 31st British Machine Vision Conference, BMVC 20202020[Online]. Available:).
- [86] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, “BAGAN: Data Augmentation with Balancing GAN,” Mar. 2018, [Online]. Available: (<http://arxiv.org/abs/1803.09655>).
- [87] B. Abbey, et al., Keyhole coherent diffractive imaging, *Nat. Phys.* 4 (5) (2008) 394–398, <https://doi.org/10.1038/nphys896>.
- [88] J. Zhao, Y. Kim, K. Zhang, A.M. Rush, Y. LeCun, Adversarially Regularized Autoencoders, 35th Int. Conf. Mach. Learn., ICML 2018 13 (2017) 9405–9420. Accessed: Dec. 26, 2023. [Online]. Available: (<https://arxiv.org/abs/1706.04223v3>).
- [89] I.O. Tolstikhin, O. Bousquet, S. Gelly, B. Schölkopf, “Wasserstein Auto-Encoders,” *Int. Conf. Learn. Represent.* (2017).
- [90] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial Autoencoders, *Elem. Dimens. Reduct. Manifold Learn.* (2015) 577–596, https://doi.org/10.1007/978-3-031-10602-6_21.
- [91] M. Arjovsky, S. Chintala, L. Bottou, *Gener. Advers. Netw.* (2017).
- [92] X. Liu, T. Li, R. Zhang, D. Wu, Y. Liu, Z. Yang, A GAN and feature selection-based oversampling technique for intrusion detection, *Secur. Commun. Netw.* 2021 (2021), <https://doi.org/10.1155/2021/9947059>.
- [93] X. Yao, K. Li, L. Yu, Imbalanced corporate bond default modeling using generative adversarial networks oversampling techniques, *Xitong Gongcheng Lilun yu Shijian/Syst. Eng. Theory Pract.* 42 (10) (2022) 2617–2634, <https://doi.org/10.12011/SETP2021-2328>.
- [94] E. Farahbakhsh, J. Maughan, R.D. Müller, Prospectivity modelling of critical mineral deposits using a generative adversarial network with oversampling and positive-unlabelled bagging, *Ore Geol. Rev.* 162 (2023), <https://doi.org/10.1016/j.oregeorev.2023.105665>.
- [95] M. Zareapoor, P. Shamsolmoali, J. Yang, Oversampling adversarial network for class-imbalanced fault diagnosis, *Mech. Syst. Signal Process* 149 (Feb. 2021), <https://doi.org/10.1016/j.ymssp.2020.107175>.
- [96] S.S. Mullick, S. Datta, S. Das, Generative Adversarial Minority Oversampling, in: *International Conference on Computer Vision, 2019-October*, IEEE, 2019, pp. 1695–1704, <https://doi.org/10.1109/ICCV.2019.00178>.
- [97] Y. Zhan, D. Hu, Y. Wang, X. Yu, Semisupervised Hyperspectral Image Classification Based on Generative Adversarial Networks, *IEEE Geosci. Remote Sens. Lett.* 15 (2) (2018) 212–216, <https://doi.org/10.1109/LGRS.2017.2780890>.
- [98] J. Kim, H. Park OA-GAN: Overfitting avoidance method of GAN oversampling based on xAI Int. Conf. Ubiquitous Future Netw., ICUFN; in *International Conference on Ubiquitous and Future Networks, ICUFN, 2021*, pp. 394 – 398. doi: 10.1109/ICUFN49451.2021.95285942021.
- [99] J. Wang, L. Yao Unrolled GAN-based oversampling of credit card dataset for fraud detection 2022 IEEE, in 2022 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2022, 2022, pp. 858 – 861. doi: 10.1109/ICAICA54878.2022.98444212022.
- [100] J. Tao, G. Wei, Y. Song, J. Dou, W. Mu, Oversampling algorithm based on gradient penalty generative adversarial network, *Shanghai Ligong Daxue Xuebao/J. Univ. Shanghai Sci. Technol.* 45 (3) (2023) 235–243, <https://doi.org/10.13255/J.CNKI.JUSST.20220307005>.