FG-SMOTE: Towards Fair Node Classification with Graph Neural Network

Zichong Wang[1], Zhipeng Yin[1], Yuying Zhang[1], Liping Yang[2], Tingting Zhang[3], Niki Pissinou[1], Yu Cai[4], Shu Hu[5], Yun Li[6], Liang Zhao[7], and Wenbin Zhang[1*]
[1] Florida International University, Miami, FL, USA
[2] University of New Mexico, Albuquerque, NM, USA
[3] University of South Florida, Tampa, FL, USA
[4] Michigan Technology University, Houghton, MI, USA
[5] Purdue University, Indianapolis, IN, USA
[6] i4AI Ltd, London, United Kingdom
[7] Emory University, Atlanta, GA, USA

ABSTRACT

Graph generative models have become increasingly prevalent across various domains due to their superior performance in diverse applications. However, as their application rises, particularly in high-risk decision-making scenarios, concerns about their fairness are intensifying within the community. Existing graph-based generation models mainly focus on synthesizing minority nodes to enhance the node classification performance. However, by overlooking the node generation process, this strategy may intensify representational disparities among different subgroups, thereby further compromising the fairness of the model. Moreover, existing oversampling methods generate samples by selecting instances from corresponding subgroups, risking overfitting in those subgroups owing to their underrepresentation. Furthermore, they fail to account for the inherent imbalance in edge distributions among subgroups, consequently introducing structural bias when generating graph structure information. To address these challenges, this paper elucidates how existing graph-based sampling techniques can amplify real-world bias and proposes a novel framework, *Fair Graph Synthetic Minority Oversampling Technique* (FG-SMOTE), aimed at achieving a fair balance in representing different subgroups. Specifically, FG-SMOTE starts by removing the identifiability of subgroup information from node representations. Subsequently, the embeddings for simulated nodes are generated by sampling from these subgroup information desensitized node representations. Lastly, a fair link predictor is employed to generate the graph structure information. Extensive experimental evaluations on three real graph datasets show that FG-SMOTE outperforms the state-of-the-art baselines in fairness while also maintaining competitive predictive performance.

1. INTRODUCTION

Graph data is pervasive in real-world applications, such as the financial markets [48], biological networks [38], and social networks [31]. To extract node features and understand the intricate graph structures, various graph mining algorithms have been developed over the past decades [26, 37, 7, 18, 24, 54]. Among these, graph generative models have become crucial components of the graph Machine Learning (ML) framework, serving purposes such as data augmentation [5], anomaly detection [1], and recommendation [32]. For instance, in scenarios like financial fraud detection [27] and rare disease prediction [34], graph generative models can produce synthetic samples from minority community to enhance model training, thereby improving the model's ability to generalize over real-world data [56].

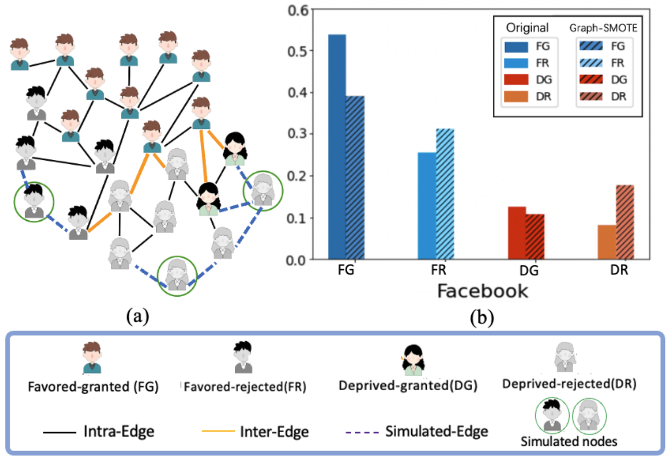

Figure 1: An illustrative example demonstrating the bias introduced by current graph generative models and its implications in the Facebook dataset.

As graph generative models become increasingly prevalent in high-risk decision-making scenarios such as credit scoring [28], recommendation [13], and healthcare [22], concerns about their fairness are intensifying [25]. Indeed, recent literature has proven graph generative models may inadvertently inherit or even exacerbate biases from real-world data [34, 34, 57]. This issue stems from distributional differences in subgroups defined by *sensitive attributes* (*e.g.*, gender or race) and labels, within graph datasets. Such dis-

*Corresponding author
Email: {ziwang, wenbin.zhang}@fiu.edu

parities can lead graph generative models to inherit or even exacerbate biases and pass them on to downstream classification tasks. In fact, despite their potential, existing graph generative models primarily focus on improving node classification performance without adequately addressing fairness implications. For instance, GraphSMOTE [56] addresses class label imbalance by generating synthetic nodes through interpolation among underrepresented groups and employs a pre-trained edge generator to establish their graph structure. Despite the remarkable effectiveness, it inadvertently intensifies disparities among subgroups, compromising the fairness of the model. To illustrate, consider credit card applications through social networking information [10] depicted in Figure 1 (a). Given the inherent higher proportion of rejected labels (*i.e.*, gray) for the deprived group (*e.g.*, females) in the dataset, GraphSMOTE generates more synthetic samples with rejected labels (surrounded by green circles) for the deprived group. Consequently, the synthetic graph depicts a reduced proportion of favorable outcomes for the deprived group, further ingraining the stereotype associating the deprived subgroup with rejected labels [45]. As a practical example, as shown in Figure 1 (b), GraphSMOTE is applied to the Facebook dataset aiming to balance the label distribution. Initially, 56.7% of samples in the deprived group had the granted label, compared to 67.8% in the favored subgroup—an 11.1% disparity. While it balances the class distribution, it further exacerbates the disparities between subgroups; the new distribution showed 36.2% favorable outcomes for the deprived subgroup and 55.6% for the favored subgroup, widening the gap to a 19.4% disparity.

Additionally, existing graph balancing techniques cannot simply be applied to simultaneously balance class labels and sensitive attribute distribution. Still, in the example shown in Figure 1 (b), the favored-granted subgroup represents 53.3% of the samples, while the deprived-granted subgroup accounts for a mere 12.4%. This evident discrepancy suggests that existing efforts to balance subgroup representation could result in the recurrent selection of samples from the deprived subgroup for synthetic sample generation. This repetitive process risks undermining sample diversity, particularly the underrepresented deprived-granted subgroup, thereby increasing the likelihood of overfitting and exacerbating existing biases.

Furthermore, existing graph generative models utilize pre-trained edge predictors to determine the graph structure for generated nodes [40]. However, biases present in the actual edge distribution can skew this process. For instance, as illustrated in Figure 1 (a), applicants are more frequently connected (*i.e.*, black line) with nodes that share the same sensitive attributes. Consequently, the graph structure generated by the model (*i.e.*, purple line) inherits and exacerbates this bias by preferentially connecting applicants with similar sensitive attributes, inadvertently introducing structural bias (*i.e.*, nodes with same sensitive attributes are excessively linked) and fostering group segregation [34].

In this paper, we investigate these challenges concerning the fairness of graph generative models. Specifically, we examine the disparities in representativeness among different subgroups and the structural biases evident in the generated graphs. We also delve into the challenges posed by sample selection in graph generative models, especially when dealing with subgroups with limited sample sizes. This domain remains largely uncharted and presents three distinct challenges: **i) Complexity of Graph Data.** Unlike tabular data, graph data does not follow the principle of Independent and Identically Distributed (I.I.D.) due to the inherent interconnections between each node and its neighbors [38]. Consequently, generating samples demands not only the creation of high-quality node features but also the accurate representation of graph structure information. **ii) Multiple sources of bias in graph data.** Biases can originate from the sensitive attribute of the nodes themselves or from the structural relationships between them [33]. Therefore, graph generative models should mitigate these multiple biases simultaneously to achieve equitable representation. **iii) Simultaneously balancing class labels and sensitive attributes.** In contrast to graph generative models driven solely by performance, fairness-aware generation necessitates simultaneous consideration of both class labels and sensitive attributes. In addition, applying conventional balancing methods directly is ineffective and may even worsen existing biases, a sophisticated design is thus essential.

To address the above challenges, we introduce a novel framework, *Fair Graph Synthetic Minority Oversampling Technique (FG-SMOTE)*, which aims at synthesizing new samples and their connections while mitigating inherent graph bias for fair node classification. *To the best of our knowledge, this is the first work that simultaneously accounts for node distribution and structural bias to generate fair graphs.* Specifically, FG-SMOTE begins by generating node embeddings for each node through aggregation. It then imposes performance and fairness constraints to ensure that the learned node embeddings remain invariant to sensitive attributes while retaining as much node information as possible. This strategy enables the following graph augmentation to select sample templates from a wider pool of sensitive information de-identified similar node embeddings. Following this, FG-SMOTE employs graph augmentation techniques, *i.e.*, cluster interpolation in the embedding space, ensuring the node generation process remains unaffected by intra-class similarity and inter-class differences. Lastly, FG-SMOTE constructs graph structural information using a fair link predictor, which is designed to prevent structural bias that might arise when generating edges based on feature similarity. Overall, integrating the strengths of both graph augmentation and graph generation, FG-SMOTE addresses the largely unattended bias issues in increasingly prevalent graph generative models, contributing to the development of a first-of-its-kind fair graph generation methodology. The key contributions of this work can be summarized as follows:

- We define a new research direction in generating fair graphs, focusing on addressing both node distributional and graph structural biases within the graphs.
- We introduce FG-SMOTE to simultaneously mitigate these inherent graph biases while generating fair graphs for equitable node classification, thereby providing essential complements to the existing literature on graph fairness.
- Extensive experimental results on three real-world graph datasets, evaluated using 11 metrics, demonstrate the effectiveness of FG-SMOTE in striking a balance between fairness and predictive performance.

2. RELATED WORK

Graph Generative Models. Existing graph generation techniques can typically be classified into two primary categories [55]: i) Node generation and ii) Edge generation. Node generation aims to extend existing oversampling methods to graph data. For instance, Oversampling [4] amplifies the representation of minority nodes by directly duplicating their embeddings from real datasets. On the other hand, GraphSMOTE [56] generates synthetic nodes for minority classes by interpolating between real embeddings, then generates the graph topology using a pre-trained edge generator. In the realm of edge generation, the goal is to capture and replicate essential structural properties of graphs. For example, Netgan [3], taking inspiration from the GAN framework [14], generates synthetic random walks while discriminating between synthetic and real random walks sampled from the input graph. GraphVAE [29] generates multiple smaller graphs and utilizes a subgraph matching algorithm to stitch them into a full-sized graph, akin to the original. GraphRNN [47] conceptualizes a graph as a sequential assembly of nodes and edges, learning this process with autoregressive models. However, their common oversight is a singular focus on performance without due consideration for fairness, which leads to the introduction of distributional or structural bias and further degrading the model's fairness.

Fairness in Graph. Existing work on graph fairness primarily focuses on improving equitable node classification [8, 41, 42, 43]. This area typically falls into two categories: individual fairness and group fairness. Specifically, individual fairness [49, 36, 12] requires that similar individuals (*e.g.*, measured by Euclidean distance) in the input space should have similar probability distributions in the output space. In contrast, group fairness [9, 15, 52] requires classifiers to have similar performance (*e.g.*, prediction accuracy) across different subgroups. More recently, there has been an emphasis on tackling structural bias in graph learning. For instance, Fairwalk [24] counters link prediction bias by adjusting the probability of random walks, ensuring equal selection chances for various subgroups. Moreover, FG^{2}AN [34] addresses biases in graph structure generation by equalizing performance across different node degrees and reducing linkage disparities among different subgroups. Nevertheless, current methodologies fall short of mitigating node distribution biases arising during the node generation.

Contrary to previous research, our approach aims to mitigate the bias in the node generation while simultaneously accounting for structural bias. Specifically, FG-SMOTE focuses on enhancing the representation of various subgroups: i) By generating samples from deprived communities, we aim to ensure consistent representation across different subgroups by a global sampling strategy; ii) By employing a fair link predictor, FG-SMOTE effectively mitigates structural biases within the graph generation model.

3. NOTIONS

In this paper, we use bold uppercase characters (*e.g.*, $\mathbf{A}$) to denote matrices, bold lowercase letters (*e.g.*, $\mathbf{s}$) to denote vectors, uppercase calligraphic characters (*e.g.*, $\mathcal{V}$) to denote sets, and normal lowercase letters (*e.g.*, s) to denote scalars. In addition, we represent the i-th row, j-th column, and (i, j)-th entry of any matrix, such as A as $A_{[i,:]}$, $A_{[:,j]}$, and $A_{i,j}$, respectively. we use lowercase bold vectors with index to represent the row vector of a matrix (*e.g.*, $a_i = A_{[i,:]}$). Moreover, we use $|\cdot|$ to denote the absolute value operator. Furthermore, we consider input graph as undirected and unweighted $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V}$ denote the set of nodes, $\mathcal{E}$ ($\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$) represent the set of edges, and $\mathcal{X} \in \mathbb{R}^{n \times d}$ (n = $|V|$) represent the set of node features, where x_i represents the features of the node v_i. We let $\mathbf{A}$ denote the adjacency matrix of the input graph $\mathcal{G}$, where $a_{i,j}$ takes on the value 1 if there exists an edge $i \to j$, and 0 otherwise. Meanwhile, each node v_i has a sensitive attribute, we utilize $\mathbf{S} \in \{0, 1\}^{N \times 1}$ to represent the sensitive attributes, where s_i represents whether or not a given individual v_i is a member of the deprived set. Note that $s_i \in x_i$, and we let S^- denote the deprived subgroup and S^+ the favored subgroup. For every node v_i, the ego graph is G_{v_i}. The ego graph shows the direct neighbors and their connections for a specific node in the larger graph. Without restricting the generality, We let $L = \{v_1, v_2, \ldots, v_{|L|}\}$ represents the set of labeled vertices. The associated ground-truth labels are represented by Y = $\{y_1, y_2, \ldots, y_{|L|}\}$, where y_i is the label for vertex v_i. Furthermore, $U = \{v_{|L|+1}, v_{|L|+2}, \ldots, v_{|L|+|U|}\}$ represents the set of unlabeled vertices. Predicted labels for these vertices are represented as $\hat{y}$. Notably, $L \bigcup U = \mathcal{V}$. Additionally, for the convenience of the study, we assume that all sensitive attributes and labels are binary.

4. METHODOLOGY

4.1 FG-SMOTE: In a Nutshell

Figure 2 provides an overview of FG-SMOTE, showcasing its three essential modules: the Sensitive Information De-identification Module (highlighted in blue rectangle), the Data Augmentation Module (highlighted in orange rectangle), and the Fair Link Prediction Module (highlighted in green rectangle). Specifically, the Sensitive Information De-identification Module is designed to minimize the identifiability of sensitive attributes in node embeddings, while preserving as much other information as feasible by establishing three constraints (*c.f.* Section 4.2). Subsequently, the Data Augmentation Module generates samples for the underrepresented subgroups by interpolating node representations that are obtained from sensitive information de-identified embedding, ensuring balanced representation across different subgroups (*c.f.* Section 4.3). Lastly, the Fair Link Prediction Module establishes graph structure information for each synthesized sample while mitigating structural bias (*c.f.* Section 4.4). The following sections detail each of them.

4.2 Sensitive Information De-identification Module

Existing graph generative models mainly utilize node embeddings derived from encoder aggregation for oversampling, designed to balance representational differences according to class labels. However, this strategy cannot directly be applied for fair sampling, which involves distributional disparities across both sensitive attributes and labels, complicating each other. Specifically, synthetic samples are often chosen from underrepresented subgroups to balance distributional disparities. This practice can lead to the repeated generation of samples that either already exist in the training data or closely resemble existing ones, due to de-

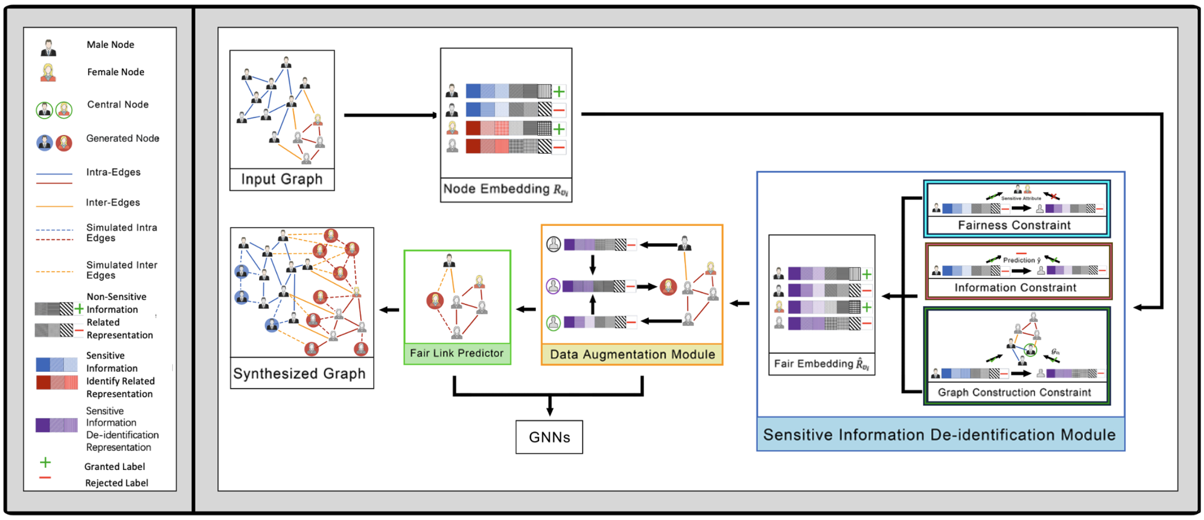

Figure 2: Overview of the proposed FG-SMOTE framework.

prived subgroups' lack of representation in the data. As a result, synthetic samples generated for these deprived subgroups lack diversity [23]. This deficiency in diversity predisposes the model to overfitting on these subgroups. To track this issue, FG-SMOTE introduces global sampling with the goal of enhancing the diversity of generated samples for underrepresented subgroups. In essence, rather than restricting the selection of sample embeddings to specific subgroup (*e.g.*, deprived rejected subgroup), we are able to select sample embeddings from the entire pool of sensitive information de-identified similar node embedding (*e.g.*, not only deprived rejected subgroup but also favored rejected subgroup), thus vastly expanding the available choices. As exemplified in the Data Augmentation Module in Figure 2, the chosen two embeddings for producing the synthetic sample (*e.g.*,) is derived from distinct subgroups (*e.g.*, and). This occurs because these two embeddings resemble each other after the de-identification of sensitive attribute information. To achieve this, we map each node's embedding into a new representation space. This transformation is designed to obscure any information indicating whether the individual belongs to a deprived subgroup, while preserving as much of the remaining information as possible with the following three constraints: the fairness constraint, the information constraint, and the graph construction constraint.

Fairness Constraint. The fairness constraint is designed to prevent the retention of information about the sensitive attribute, *i.e.,* favored and deprived subgroups, in node representations for fair sampling, thereby preventing the introduction of additional biases between various subgroups. Practically, within the light blue part of this module, while the original node embedding reveals information about nodes belonging to a specific group (✓), this information becomes indiscernible after applying the fairness constraint (X). Adhering to the embedding perspective, the original representation space distinctly reveals whether a node belongs to a specific subgroup. This is evident in the Node Embedding R_{v_i} in Figure 2, where sensitive information-related embeddings distinguishing between male and female are represented by red and blue colors, respectively. Subsequently, these embeddings converge towards uniform representation in the Fair Embedding $\hat{R}_{v_i}$, depicted in purple, thereby obscuring their distinct sensitive information. To achieve this, we map the node representation R_{v_i} to a new space that cannot identify whether a node belongs to a specific subgroup. Specifically, we represent each node's information in this new space using a set of prototypical probabilistic mappings; for each prototype $\rho = \hat{R}_k$, where ρ is a multinomial random variable, each value k corresponds to an instance from the intermediate set of prototypes (the dimensionality of $\hat{R}_k$ is identical to that of R_{v_i}). Hence, a node representation does not contain sensitive group information if it has the same probability of appearing in the deprived subgroup (S^-) as in the favorable subgroup (S^+), which can be mathematically represented as $P(\rho = \hat{R}_k | R \in S^+) = P(\rho = \hat{R}_k | R \in S^-)$. To efficiently map the node representation R to ρ, a natural probability mapping using the softmax function is defined as:

$$P(\rho = k|R) = \frac{\exp(-d(R, \hat{R}_k))}{\sum_{j=1}^{K} \exp(-d(R, \hat{R}_j))} \tag{1}$$

where $d(\cdot)$ is a distance metric (*e.g.*, Euclidean distance) and K denotes the number of prototypes.

Building on this, we measure the difference in the probability of each prototype in different sensitive groups. Hence, the group statistical parity difference (GSD) is formally defined as $GSD = |GP^+ - GP^-|$, where group mapping probability (GP) is defined as follows:

$$\begin{cases} GP^+ = \dfrac{1}{|v_i|^+} \displaystyle\sum_{v_i \in S^+} \sum_{k=1}^{K} P(\rho = \hat{R}_k | R_S \in S^+) \\ GP^- = \dfrac{1}{|v_i|^-} \displaystyle\sum_{v_i \in S^-} \sum_{k=1}^{K} P(\rho = \hat{R}_k | R_S \in S^-) \end{cases} \tag{2}$$

Finally, the fairness constraint is defined as follows:

$$\mathcal{L}_F = GSD + \sum_{i=1}^{n} (R_{v_i} - \hat{R}_{v_i})^2 \tag{3}$$

where $\hat{R}_{v_i}$ is the reconstructions of R_{v_i}. This constraint encourages the model to encode all information contained within the raw features except for any information that could lead to biased learning.

Information Constraint. For each node, v_i, the obtained node representation $\hat{R}_{v_i}$ should capture important node features and graph topology information to retain utility for downstream tasks. In other words, for node v_i, the model can be trained on the representations to make accurate label predictions (*i.e.*, $\hat{R}_{v_i} \rightarrow y_i$). As illustrated in the maplered part of this module, the information constraint ensures the retention of the necessary information to accurately predict (✓) the label in both the new node embedding, *i.e.*, $\hat{R}_{v_i}$, and the original node embedding, *i.e.*, R_{v_i}. Hence, the objective of the information constraint is to minimize the loss of the prediction model, as shown in Equation 4:

$$\mathcal{L}_I = \frac{1}{|\mathcal{V}_L|} \sum_{v_i \in \mathcal{V}_L} -(y_i \log(\hat{y_i}) + (1 - y_i) \log(1 - \hat{y_i})) \quad (4)$$

where y_i is the one-hot encoding of the ground-truth label of v_i. Note that current graph generative models mainly focus on R_{v_i}, thus leading to biased sampling towards deprived subgroups.

Graph Construction Constraint. For each node v_i, another objective is to ensure that its embedding accurately represents the node itself. This involves accurately reconstructing (✓) the ego-graph of node v_i, denoted as $\mathcal{G}_{v_i}$, depicted in the deep green part of this module, using the new node embedding $\hat{R}_{v_i}$. To this end, the graph construction constraint is formalized as the graph structure reconstruction loss ($\mathcal{L}_G$) as below:

$$\mathcal{L}_G = \frac{1}{|\mathcal{E}^+| + |\mathcal{E}^-|} \sum_{e_{ij} \in \mathcal{E}} L(e_{ij}, \hat{e}_{ij}) \quad (5)$$

where $\mathcal{E}^+$ and $\mathcal{E}^-$ denote the sets of sampled positive and negative edges, respectively, and $L(\cdot)$ denotes the cross-entropy loss function. In addition, e_{ij} denotes the actual connection status between nodes v_i and v_j, while the predicted probability of a connection between these nodes is given by $\hat{e}_{ij} = \sigma(\rho_i \rho_j^T)$, where $\sigma(\cdot)$ is the sigmoid function. Notable, due to the scarcity of positive edges, a negative edge is randomly selected to add to the set of negative edges for every positive edge acquired.

In essence, the introduction of the graph construction constraint serves as a precaution against noise infiltration into node representations. This ensures that the reconstructed ego-graph, $\mathcal{G}_{v_i}$, remains uncorrupted, thereby refining the quality of node generation.

4.3 Data Augmentation Module

With the fair embedding, FG-SMOTE carries out global oversampling to mitigate the rooted distributional disparities, following the general idea of SMOTE algorithm [6]. Specifically, for each node v_i, its nearest neighboring nodes bearing the same label are determined using the $Pick(\cdot)$ function, which is based on the Euclidean distance in the input space and is mathematically represented by Equation 6:

$$Pick(v_i) = \{\forall v_j \in \mathcal{G} | \text{argmin} \left\| \hat{R}_{v_i} - \hat{R}_{v_j} \right\| \} \quad \text{s.t.} \quad Y_i = Y_j \quad (6)$$

where v_i denotes the selected node, while v_j represents the neighboring node of v_i that shares the same class label but their sensitive attribute might differ.

Using the selected node embeddings of the samples and their respective nearest neighbors, we can formally define the embedding of the synthetic sample as: $R_{gen} = \alpha \hat{R}_{v_i} + (1-\alpha) \hat{R}_{v_j}$, where α is a random variable in the range $[0, 1]$, to control the similarity of synthetic node embedding is close to one instance or its close neighbor. Given that both $\hat{R}_{v_i}$ and $\hat{R}_{v_j}$ belong to the same class and are in close spatial proximity, the resulting synthetic sample R_{gen} inherits their properties and remains within the same class. The generated sample's sensitive attribute information is subsequently proportionally assigned to ensure consistent representation across different subgroups. Notably, our approach employs the SMOTE as a representative example due to its status as a foundational and widely adopted method in the realm of oversampling. However, it's important to emphasize that our methodology is versatile and can be adapted to various oversampling techniques beyond SMOTE.

4.4 Fair Link Predictor

After generating synthetic nodes to address the distributional disparities, we face another challenge: the generated nodes are not connected to the input graph $\mathcal{G}$, rendering them as isolated nodes. Consequently, it is imperative to produce graph structure information for each of these nodes to integrate them to $\mathcal{G}$. Existing methodologies harness the edge distribution of the real data to train link predictors, subsequently generating the required graph structure information. Essentially, the existence of a link between any two nodes is contingent upon their similarity according to the weighted inner product. The predicted relation ($E_{(v_i,v_j)}$) between node v_i and v_j is define as follows:

$$E_{(v_i,v_j)} = \frac{\exp(\sigma(\hat{R}_{v_i} \cdot \mathbf{Z} \cdot \hat{R}_{v_j}))}{\sum \exp(\sigma(\hat{R}_{v_i} \cdot \mathbf{Z} \cdot \hat{R}_{v_j}))} \quad (7)$$

where the parameter matrix $\mathbf{Z}$ contains the interaction between node v_i and v_j. Next, the edge generator's loss function is represented as:

$$\mathcal{L}_{IE} = \|\mathbf{E} - \mathbf{A}\|_F^2 \quad (8)$$

where $\mathbf{E}$ predicts the connections between nodes in $\mathcal{V}$.

However, this approach is performance-driven and could exacerbate the connection disparities between deprived and favored subgroups [47]. Specifically, given that most connections in the input graph $\mathcal{G}$ are intra-group connections, this leads to amplified inter-group connection disparity in the topology learned by existing graph generation models. Considering the growing practice of using social networks for credit scoring as an example: users from deprived groups may receive unjustly reduced credit scores since a significant portion of their social network consists of inter-group connections with lower credit scores. This ultimately leads to biased credit assessments against the deprived group [44].

Such a disparity can be quantified as the model performance difference between inter-group and intra-group links in $\mathcal{G}$, referred to as the consistency difference (CD), *i.e.,* CD $= |L_{inter} - L_{intra}|$, where L_{inter} represents the loss of inter-group connection, and L_{intra} denotes the loss associated with intra-group connection:

$$\begin{cases} L_{inter} = \|E_{inter} - A_{inter}\|_F^2 \\ L_{intra} = \|E_{intra} - A_{intra}\|_F^2 \end{cases} \tag{9}$$

where E_{inter} and E_{intra} denote predicted connections between inter-group and intra-group nodes, while A_{inter} and A_{intra} represent actual connections within inter-group and intra-group nodes, respectively.

Integrating these components, the loss of fair link predictor is formally defined as: $\mathcal{L}_C = \mathcal{L}_{IE} + \text{CD}$

4.5 Final Joint Learning Framework

Assembling the previously discussed modules, the final objective function of FG-SMOTE, which consists of four parts and is controlled by the tunable hyperparameters a, b, and c to balance the contributions of the various elements in the overall objective function, is depicted in Equation 10:

$$\begin{aligned} \min\ \mathcal{L}_{total} &= \mathcal{L}_I + a\mathcal{L}_G + b\mathcal{L}_F + c\mathcal{L}_C \\ &= \frac{1}{|\mathcal{V}_L|} \sum_{v_i \in \mathcal{V}_L} -(y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i})) \\ &+ a\frac{1}{|\mathcal{E}^+| + |\mathcal{E}^-|} \sum_{e_{ij} \in \mathcal{E}} L(e_{ij}, \hat{e}_{ij}) + b|GP^+ - GP^-| \\ &+ b\sum_{i=1}^{n}(R_{v_i}, -\hat{R}_{v_i})^2 + c\|\mathbf{E} - \mathbf{A}\|_F^2 + c|L_{inter} - L_{intra}| \end{aligned} \tag{10}$$

where the first term, $\mathcal{L}_I$, aims to minimize the prediction loss; the second term, $\mathcal{L}_G$, design to minimize the reconstruction loss for the node representations R with the function $L(\cdot)$ denoting cross-entropy; The third term, $\mathcal{L}_F$, aiming to desensitize the representation R, thereby enable global sampling to mitigate node representation bias; The last term, $\mathcal{L}_C$, aims to promote the fairness of the edge predictor and avoid introducing structural bias.

5. EXPERIMENT

5.1 Experimental Settings

Datasets. Three real-world graph datasets with varying characteristics are used for thorough evaluations: i) The **Facebook** dataset [20] contains the Ego Network of social networks, where each node represents a user, and if there is an edge between two nodes, it means that the two users are Facebook friends with each other. The sensitive attribute is gender, and the aim is to predict whether users are in the same social circle. ii) The **Pokec-z** [30] is derived from a popular social network in Slovakia. Nodes denote users with features such as gender, age, interest, etc. Edge represents the friendship between users. Considering the region as the sensitive attribute, the task is to predict the working field of the users. iii) The **Credit** dataset [20] contains payment default information for each individual. In this dataset, each node represents an individual, and each edge between the two nodes indicates a similarity in their payment methods. The sensitive attribute is age, with the aim of predicting whether their default payment is to use a credit card. The detailed statistical information for each graph dataset is shown in Table 1.

Table 1: Summary of the datasets used in the experiments.

Dataset	Facebook	Pokec-z	Credit	Bail
Vertices	1,034	67,796	30,000	18,876
Edges	26,749	651,856	137,377	311,870
Average Degree	51.7	19.2	10	34
Sensitive Attribute	Gender	Region	Age	Race

Evaluation Metrics. Eleven metrics encompassing node classification performance, graph generation quality, and their respective fairness considerations are employed for comprehensive evaluation:

i) Node Classification Metrics. Three metrics are adopted to evaluate the node classification performance, *i.e.,* accuracy, F1-Score, and AUC. For all of them, higher values correspond to better performance. In terms of fairness, the evaluation is based on Statistical Parity Differences (SPD) [19] and Equal Opportunity Differences (EOD) [15]. For both of them, an absolute value close to 0 indicates optimal fairness, while larger values denote more significant discrimination.

ii) Edge Generation Evaluation Metrics. Three metrics are employed to evaluate the quality of the generated graph. Specifically, the difference between the following properties of the generated graph and the original graph are measured [58]: 1) Mean Degree Difference (MD): The average node degree; 2) Edge Distribution Entropy Difference (EDED): The relative edge distribution entropy of $\mathcal{G}$; 3) Gini Difference (GD): The Gini coefficient of the degree distribution. Moreover, drawing from prior work [34], three fair edge generation metrics are adopted to gauge disparities when generating edges for favored and deprived subgroups: 1) Average Degree Difference (ADD): Evaluates the disparity in network clustering difference between deprived and favored node subgroups; 2) Equal Edge Distribution Entropy (EEDE): Quantifies the disparity between the relative edge distribution entropy of the favored and deprived subgroups; 3) Equal Gini (EG): Evaluates the disparity in the Gini coefficient of the degree distribution across different subgroups. For all of these six metrics, smaller values indicate better predictive performance and fairness.

Baselines. To evaluate the efficacy of FG-SMOTE, nine state-of-the-art graph generative models from different perspectives are compared: Oversampling [4], Embed-SMOTE [2], GraphSMOTE [56], FairGNN [11], Graphair [21], FairAGG [59], RFCGNN [39], FDGNN [35], and FG^{2}AN [34]. Specifically, the first three models are performance-driven, focusing on oversampling the minority class to enhance the classifier's performance. On the other hand, the subsequent six are fairness-aware by design. Among them, **FairGNN** employs adversarial learning to cultivate GNNs adhering to group fairness criteria; **Graphhair** aims to generate fair graph data using adver-

Table 2: Comparison results of FG-SMOTE with baseline methods across real-world datasets. In each row, the best result is indicated in bold, while the runner-up result is marked with an underline.

Dataset	Metrics \ Methods	Oversampling	Embed-SMOTE	GraphSMOTE	FairGNN	Graphair	FairAGG	RFCGNN	FDGNN	**FG-SMOTE**
Facebook	**Accuracy** (↑)	0.721 ± 0.013	**0.747 ± 0.017**	0.743 ± 0.015	0.667 ± 0.024	0.563 ± 0.015	0.668 ± 0.021	0.663 ± 0.019	0.713 ± 0.017	0.744 ± 0.023
	F1-Score (↑)	0.729 ± 0.024	0.730 ± 0.019	0.743 ± 0.024	0.673 ± 0.021	0.619 ± 0.022	0.683 ± 0.022	0.692 ± 0.028	0.724 ± 0.031	**0.749 ± 0.015**
	AUC(↑)	0.769 ± 0.055	0.783 ± 0.045	**0.788 ± 0.067**	0.687 ± 0.031	0.581 ± 0.003	0.658 ± 0.031	0.745 ± 0.033	0.756 ± 0.041	0.782 ± 0.004
	SPD(↓)	0.157 ± 0.065	0.131 ± 0.042	0.0116 ± 0.045	0.083 ± 0.027	0.043 ± 0.021	0.045 ± 0.024	0.041 ± 0.023	0.039 ± 0.015	**0.037 ± 0.019**
	EOD(↓)	0.132 ± 0.032	0.114 ± 0.027	0.103 ± 0.048	0.067 ± 0.024	0.027 ± 0.017	0.031 ± 0.011	0.025 ± 0.033	0.023 ± 0.019	**0.022 ± 0.012**
Pokec-z	**Accuracy** (↑)	0.728 ± 0.015	**0.755 ± 0.008**	0.740 ± 0.012	0.724 ± 0.021	0.655 ± 0.004	0.735 ± 0.032	0.698 ± 0.091	0.724 ± 0.013	0.739 ± 0.013
	F1-Score (↑)	0.791 ± 0.012	0.817 ± 0.012	**0.828 ± 0.015**	0.687 ± 0.033	0.647 ± 0.019	0.716 ± 0.005	0.776 ± 0.029	0.789 ± 0.032	0.824 ± 0.022
	AUC(↑)	0.723 ± 0.015	0.756 ± 0.025	0.788 ± 0.017	0.761 ± 0.011	0.653 ± 0.013	0.731 ± 0.013	0.738 ± 0.033	0.747 ± 0.005	**0.789 ± 0.004**
	SPD(↓)	0.116 ± 0.032	0.103 ± 0.028	0.098 ± 0.026	0.038 ± 0.022	0.035 ± 0.008	**0.022 ± 0.014**	0.031 ± 0.007	0.030 ± 0.017	0.024 ± 0.007
	EOD(↓)	0.101 ± 0.031	0.095 ± 0.026	0.081 ± 0.017	0.033 ± 0.029	0.032 ± 0.006	0.028 ± 0.011	0.027 ± 0.013	**0.022 ± 0.011**	0.025 ± 0.013
Credit	**Accuracy** (↑)	0.755 ± 0.017	0.774 ± 0.013	0.781 ± 0.017	0.687 ± 0.012	0.531 ± 0.024	0.653 ± 0.034	0.735 ± 0.017	0.736 ± 0.020	**0.801 ± 0.022**
	F1-Score (↑)	0.802 ± 0.028	0.834 ± 0.011	**0.871 ± 0.018**	0.783 ± 0.043	0.728 ± 0.072	0.747 ± 0.042	0.849 ± 0.049	0.861 ± 0.048	0.868 ± 0.052
	AUC(↑)	0.734 ± 0.015	0.741 ± 0.015	**0.771 ± 0.027**	0.711 ± 0.074	0.758 ± 0.047	0.721 ± 0.022	0.743 ± 0.033	0.747 ± 0.031	0.763 ± 0.014
	SPD(↓)	0.161 ± 0.035	0.117 ± 0.037	0.153 ± 0.037	0.123 ± 0.036	0.085 ± 0.034	0.074 ± 0.047	0.074 ± 0.047	0.056 ± 0.024	**0.051 ± 0.014**
	EOD(↓)	0.127 ± 0.035	0.103 ± 0.047	0.108 ± 0.047	0.115 ± 0.042	0.088 ± 0.035	0.056 ± 0.021	0.064 ± 0.016	0.047 ± 0.016	**0.044 ± 0.013**

sarial learning to deceive a discriminator to achieve fairness; **FairAGG** implements a fair aggregation scheme based on the Shapley value to ensure group fairness; **RFCGNN** learns a fair node representation by identifying counterfactual instances and sensitive attribute-related information masking; **FDGNN** utilizes counterfactual samples to learn disentangled node representation to mitigate the multi-source biases, and **FG^{2}AN** is designed to foster fair edge generation for equitable graph structure information.

5.2 Implementation Details

For FG-SMOTE, we employ the Adam optimizer [16] with a learning rate of $lr = 0.001$, setting $epochs = 1000$ and a weight decay of 1×10^{-5}. For all baseline methods except FG^{2}AN, we use a 1-layer GCN [17] with 16 hidden dimensions as the model backbone, coupled with a linear layer for classification. We ensure fairness and optimal performance across all models by tuning the hyperparameters based on each method's performance on the validation set. For the FG^{2}AN method, we followed the author's instructions and configured the number of transformer heads to 4, set the learning rate to $lr = 0.01$, and the walk length to $T = 10$. All models are implemented with PyTorch and PyTorch-Geometric.

5.3 Experiment Results

The following five research questions are investigated to comprehensively evaluate FG-SMOTE.

RQ1: How does FG-SMOTE perform? FG-SMOTE is benchmarked against eight state-of-the-art baselines using four real-world graph datasets. Each experiment is repeated 10 times, with the average performance reported. To ensure a fair comparison, the selection of hyperparameters for all methods was optimized based on their performance on a validation set, with the results summarized in Table 2. Note that FG^{2}AN generates graph structure only and is thus excluded from RQ1. As we can see, FG-SMOTE demonstrates remarkable advantages in both fairness and performance. Out of 9 performance comparisons, FG-SMOTE achieved the highest ranking in 3 and was the second-highest in 4. In the 6 fairness comparisons, FG-SMOTE was ranked first 4 times and second twice. Specifically, FG-SMOTE considers fairness and representativeness simultaneously to balance the distribution of class labels and sensitive attributes while adopting a global sampling strategy that effectively sidesteps overfitting risks. Hence, FG-SMOTE emerges as an equitable option for socially sensitive graph-based applications.

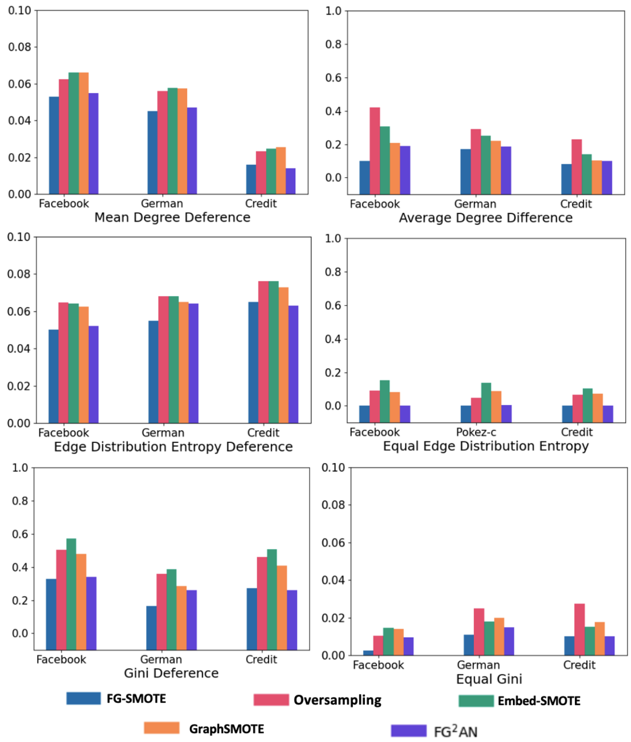

Figure 3: Comparative results of FG-SMOTE and baselines generated graph quality.

RQ2: What is the quality of graphs generated by FG-SMOTE? FG-SMOTE is compared with three graph sampling techniques and a fair graph structure generation method with the results shown in Figure 3. Note that FairGNN, Graphair, FairAGG, RFCGNN, and FDGNN are designed for classification tasks and do not generate nodes or edges, rendering a comparison with them infeasible. Overall, FG-SMOTE generates high-quality graphs while ensuring that the generated graph structure information is devoid of discrimination information. This is because existing graph sampling techniques overlook the disparity in edge distribution when generating graph structure information, resulting in stronger intra-connections and group isolation. In contrast, FG-SMOTE, equipped with a fair link predictor, effectively addresses both intra-group and inter-group distribution disparities while ensuring performance.

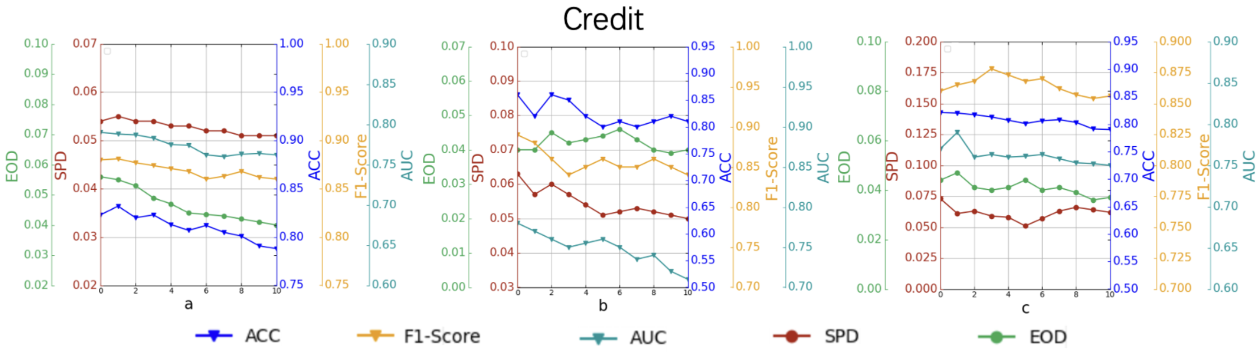

Figure 4: Exploring hyperparameters study results in the Credit dataset.

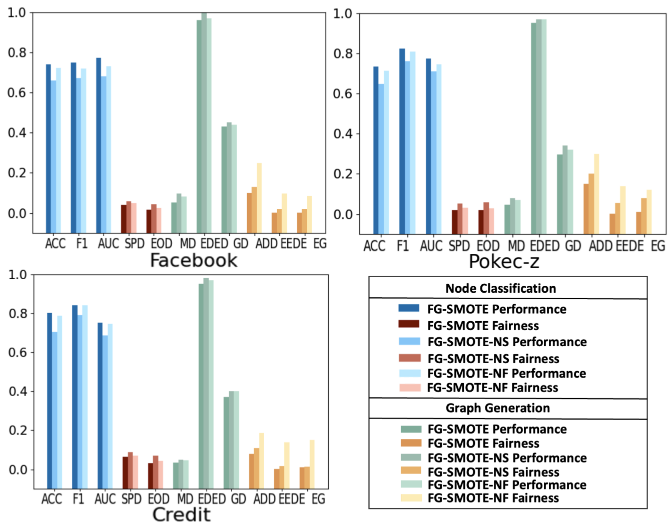

Figure 5: Ablation study results for FG-SMOTE, FG-SMOTE-NS, and FG-SMOTE-NF.

RQ3: What is the impact of each module in the FG-SMOTE framework on its overall performance and fairness? Various ablation studies are conducted to answer RQ3. First, the FG-SMOTE-NS variant is created by removing the Sensitive Information De-identification Module to evaluate the effectiveness of this module. In this setting, node embeddings obtained from a standard GNN encoder are directly used for data augmentation aimed at enhancing the representation of under-represented subgroups. The results are shown in Figure 5, highlighting a deterioration in both performance and fairness metrics compared to the FG-SMOTE. This drop is due to the limitation on synthetic sample selection, which is limited to corresponding subgroups, leading to reduced sample diversity. This limitation increases the model's susceptibility to overfitting, negatively impacting generalizability, performance, fairness, and ultimately, the quality of the constructed graph.

Next, the Fair Link Prediction Module's impact was assessed by evaluating the FG-SMOTE-NF variant, which excludes this module. As we can see from the results shown in Figure 5, FG-SMOTE-NF shows a decrease in fairness, demonstrating the importance of the Fair Link Prediction Module. Without this module, the model generates a biased graph structure, causing groups sharing the same sensitive attributes to be linked more closely. As a result, the model's node predictions are more influenced by sensitive attributes, leading to discriminatory decisions. Additionally, the lack of differentiation between inter- and intra-group connection losses in the edge prediction exacerbates intra-group connectivity, intensifying structural bias in the generated graph.

RQ4: How do hyperparameters affect the performance and fairness of FG-SMOTE? Three hyperparameters a, b, and c within FG-SMOTE are investigated for their roles in governing node reconstruction performance, fair node generation, and the structural bias of the framework, respectively. Figure 4 delineates the effects of modulating these hyperparameters, varying as $\{0, 1, \ldots, 10\}$, on both the performance and fairness of FG-SMOTE. Specifically, an increase in a will increase model fairness but at the cost of some predictive performance degradation. This is attributed to the enhanced ability of node representations to learn the graph's structural information, which, in turn, reduces bias arising from unequal neighbor distribution. In addition, an increase of b results in a minor decrease in performance while fairness improves. Specifically, a higher b value amplifies the framework's capability to desensitize node embeddings, thus curtailing bias introduced during node generation. As for c, increasing its value initially improves fairness, and beyond a certain threshold, fairness declines while the impact on performance remains minimal. This is due to enhancing c bolsters the model's proficiency in accurately generating graph structural, thus circumventing the infusion of structural bias.

6. CONCLUSION

This paper is driven by growing concerns over discriminatory practices in graph generation models. Unlike existing techniques, the proposed FG-SMOTE focuses on addressing node distribution and structural biases prevalent in real-world graph applications. It employs constraints related to fairness, performance, and graph structure reconstruction to ensure both sensitive information de-identification and preservation of critical prediction information. Moreover, by employing global sampling, FG-SMOTE enhances the diversity of generated nodes, ensuring a balanced representation of class labels and sensitive attributes. Experimental results on four real graphs validate FG-SMOTE's effectiveness in prediction performance and fairness. In addition, this paper explores a new research direction and lays the groundwork for future advancements in fair graph generation.

Acknowledgement

This work was supported in part by the National Science Foundation (NSF) under Grant No. 2245895.

7. REFERENCES

[1] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. "RTM: Laws and a recursive generator for weighted time-evolving graphs". In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE. 2008, pp. 701–706.

[2] Shin Ando and Chun Yuan Huang. "Deep over-sampling framework for classifying imbalanced data". In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10*. Springer. 2017, pp. 770–785.

[3] Aleksandar Bojchevski et al. "Netgan: Generating graphs via random walks". In: *International conference on machine learning*. PMLR. 2018, pp. 610–619.

[4] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. "A systematic study of the class imbalance problem in convolutional neural networks". In: *Neural networks* 106 (2018), pp. 249–259.

[5] Deepayan Chakrabarti and Christos Faloutsos. "Graph mining: Laws, generators, and algorithms". In: *ACM computing surveys (CSUR)* 38.1 (2006), 2–es.

[6] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[7] Flavio Chierichetti et al. "Fair clustering through fairlets". In: *Advances in neural information processing systems* 30 (2017).

[8] Manvi Choudhary, Charlotte Laclau, and Christine Largeron. "A survey on fairness for machine learning on graphs". In: *arXiv preprint arXiv:2205.05396* (2022).

[9] Alexandra Chouldechova. "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments". In: *Big data* 5.2 (2017), pp. 153–163.

[10] Zhibo Chu, Zichong Wang, and Wenbin Zhang. "Fairness in Large Language Models: A Taxonomic Survey". In: *ACM SIGKDD Explorations Newsletter, 2024* (2024), pp. 34–48.

[11] Enyan Dai and Suhang Wang. "Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 2021, pp. 680–688.

[12] Cynthia Dwork et al. "Fairness through awareness". In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226.

[13] Wenqi Fan et al. "Graph neural networks for social recommendation". In: *The world wide web conference*. 2019, pp. 417–426.

[14] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[15] Moritz Hardt, Eric Price, and Nati Srebro. "Equality of opportunity in supervised learning". In: *Advances in neural information processing systems* 29 (2016).

[16] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[17] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016).

[18] Matthäus Kleindessner et al. "Guarantees for spectral clustering with fairness constraints". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3458–3467.

[19] Tai Le Quy et al. "A survey on datasets for fairness-aware machine learning". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 12.3 (2022), e1452.

[20] Jure Leskovec and Julian Mcauley. "Learning to discover social circles in ego networks". In: *Advances in neural information processing systems* 25 (2012).

[21] Hongyi Ling et al. "Learning fair graph representations via automated data augmentations". In: *International Conference on Learning Representations (ICLR)*. 2023.

[22] Ziad Obermeyer et al. "Dissecting racial bias in an algorithm used to manage the health of populations". In: *Science* 366.6464 (2019), pp. 447–453.

[23] Joonhyung Park, Jaeyun Song, and Eunho Yang. "Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification". In: *International Conference on Learning Representations*. 2021.

[24] Tahleen Rahman et al. "Fairwalk: Towards fair graph embedding". In: (2019).

[25] Nripsuta Ani Saxena, Wenbin Zhang, and Cyrus Shahabi. "Missed opportunities in fair AI". In: *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM. 2023, pp. 961–964.

[26] Prithviraj Sen et al. "Collective classification in network data". In: *AI magazine* 29.3 (2008), pp. 93–93.

[27] Shubhranshu Shekhar, Neil Shah, and Leman Akoglu. "Fairod: Fairness-aware outlier detection". In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*. 2021, pp. 210–220.

[28] Valentina Shumovskaia et al. "Linking bank clients using graph neural networks powered by rich transactional data: Extended abstract". In: *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. 2020, pp. 787–788.

[29] Martin Simonovsky and Nikos Komodakis. "Graphvae: Towards generation of small graphs using variational autoencoders". In: *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*. Springer. 2018, pp. 412–422.

[30] Lubos Takac and Michal Zabovsky. "Data analysis in public social networks". In: *International scientific conference and international workshop present day trends of innovations*. Vol. 1. 6. 2012.

[31] Huaiyu Wan et al. "Aminer: Search and mining of academic social networks". In: *Data Intelligence* 1.1 (2019), pp. 58–76.

[32] Xiang Wang et al. "Neural graph collaborative filtering". In: *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 2019, pp. 165–174.

[33] Zichong Wang and Wenbin Zhang. "Group Fairness with Individual and Censorship Constraints". In: *27th European Conference on Artificial Intelligence*. 2024.

[34] Zichong Wang et al. ": Fairness-aware graph generative adversarial networks". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2023, pp. 259–275.

[35] Zichong Wang et al. "Advancing Graph Counterfactual Fairness through Fair Representation Learning". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer Nature Switzerland. 2024, pp. 40–58.

[36] Zichong Wang et al. "Graph Fairness via Authentic Counterfactuals: Tackling Structural and Causal Challenges". In: *ACM SIGKDD Explorations Newsletter, 2025* (2025).

[37] Zichong Wang et al. "History, Development, and Principles of Large Language Models-An Introductory Survey". In: *AI and Ethics, 2024* (2024).

[38] Zichong Wang et al. "Individual Fairness with Group Awareness Under Uncertainty". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer Nature Switzerland. 2024, pp. 89–106.

[39] Zichong Wang et al. "Mitigating multisource biases in graph neural networks via real counterfactual samples". In: *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE. 2023, pp. 638–647.

[40] Zichong Wang et al. "Preventing Discriminatory Decision-making in Evolving Data Streams". In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. 2023.

[41] Zichong Wang et al. "Toward Fair Graph Neural Networks via Real Counterfactual Samples". In: *Knowledge and Information Systems* (2024), pp. 1–25.

[42] Zichong Wang et al. "Towards Fair Graph Pooling with Group and Individual Awareness". In: *proceedings of the AAAI conference on artificial intelligence*. 2025.

[43] Zichong Wang et al. "Towards fair machine learning software: Understanding and addressing model bias through counterfactual thinking". In: *arXiv preprint arXiv:2302.08018* (2023).

[44] Yanhao Wei et al. "Credit scoring with social network data". In: *Marketing Science* 35.2 (2016), pp. 234–258.

[45] Depeng Xu et al. "Fairgan: Fairness-aware generative adversarial networks". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 570–575.

[46] Zhipeng Yin, Zichong Wang, and Wenbin Zhang. "Improving Fairness in Machine Learning Software via Counterfactual Fairness Thinking". In: *Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings*. 2024, pp. 420–421.

[47] Jiaxuan You et al. "Graphrnn: Generating realistic graphs with deep auto-regressive models". In: *International conference on machine learning*. PMLR. 2018, pp. 5708–5717.

[48] Si Zhang et al. "Hidden: hierarchical dense subgraph detection with application to financial fraud detection". In: *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM. 2017, pp. 570–578.

[49] Wenbin Zhang. "AI fairness in practice: Paradigm, challenges, and prospects". In: *Ai Magazine* (2024).

[50] Wenbin Zhang, Tina Hernandez-Boussard, and Jeremy Weiss. "Censored fairness through awareness". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 37. 12. 2023, pp. 14611–14619.

[51] Wenbin Zhang and Eirini Ntoutsi. "Faht: an adaptive fairness-aware decision tree classifier". In: *arXiv preprint arXiv:1907.07237* (2019).

[52] Wenbin Zhang and Jeremy C Weiss. "Fairness with censorship and group constraints". In: *Knowledge and Information Systems* 65.6 (2023), pp. 2571–2594.

[53] Wenbin Zhang and Jeremy C Weiss. "Longitudinal fairness with censorship". In: *proceedings of the AAAI conference on artificial intelligence*. Vol. 36. 11. 2022, pp. 12235–12243.

[54] Wenbin Zhang et al. "Fairness amidst non-iid graph data: A literature review". In: *arXiv preprint arXiv:2202.07170* 2 (2022).

[55] Wenbin Zhang et al. "Individual Fairness under Uncertainty". In: *26th European Conference on Artificial Intelligence*. 2023, pp. 3042–3049.

[56] Tianxiang Zhao, Xiang Zhang, and Suhang Wang. "Graphsmote: Imbalanced node classification on graphs with graph neural networks". In: *Proceedings of the 14th ACM international conference on web search and data mining*. 2021, pp. 833–841.

[57] Lecheng Zheng et al. "Fairgen: Towards fair graph generation". In: *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE. 2024, pp. 2285–2297.

[58] Dawei Zhou et al. "A data-driven graph generative model for temporal interaction networks". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 401–411.

[59] Yuchang Zhu et al. "FairAGG: Toward Fair Graph Neural Networks via Fair Aggregation". In: *IEEE Transactions on Computational Social Systems* (2024).