Technical Report: Fair Clustering

Juli Bakagianni
Department of Computer Science and Engineering, University of Ioannina, Ioannina, Greece
Foteini Plakoutsi
Department of Computer Science and Engineering, University of Ioannina, Ioannina, Greece

Abstract

We study the question of fair clustering under the balance notion metric, where each group must have approximately equal representation in every cluster as in the whole dataset. Introducing a straightforward yet robust approach, we refine the outputs of clustering algorithms, which inherently lack fairness optimization, to produce fair clusters. A key contribution of this study lies in our benchmarking analysis. We systematically compare the trade-off between fairness and clustering quality of our fair clustering method across multiple datasets and embedding models with various parameter settings, that adjust for homophily and the structural equivalence similarity, incorporating also different vanilla clustering algorithms into our comparisons. We assess cluster quality using two distinct evaluation metrics: modularity which measures the overall quality of clustering structure, and the silhouette score which measures the quality of individual data points within clusters.

1 Introduction

Machine Learning (ML) has emerged as a powerful tool for addressing numerous significant challenges, many of which carry substantial societal implications. However, ML models, that trained on vast datasets which have been found to contain biases against both individuals and minority groups, can further amplify biases. To address these concerns and rectify biased models, recent efforts in ML research have focused on developing approaches that prioritize fairness constraints [6, 7, 12, 16, 26, 27].

Our study specifically examines clustering, a fundamental unsupervised learning task aimed at partitioning a dataset. Clustering is not only integral for organizing data but also commonly employed in feature generation and enhancement. Therefore, it is imperative to address biases and unfairness when evaluating the quality of clusters. Fairness considerations vary based on applications and legal contexts, leading to a diverse range of fairness notions. Chhabra et al. [3] introduced a comprehensive categorization of fairness definitions, including group-level, individual-level, algorithm-agnostic, and algorithm-specific fairness.

In our study, we adopt the **balance notion** as our fairness criterion, a widely used definition falling under the group-level and algorithmic-agnostic categories. Initially introduced by Chierichetti et al. [4], the balance notion asserts that within every cluster, each group should be represented in approximately the same fraction as in the entire dataset. It entails calculating the ratio between the proportion of total group members in the dataset and their proportion within a cluster. The balance of the clustering is then determined by the minimum value obtained across all clusters and protected groups. As such, the balance metric yields values between 0 and 1, with higher values indicating a fairer clustering outcome.

Formulating the balance notion, let there be m groups. Then, define r and r_a to be the proportion of samples of the dataset belonging to group b and the proportion of samples in cluster $a \in [k]$ belonging to group b. Then define another ratio for this cluster and group as $R_{a,b} = r/r_a$. The balance fairness notion is then defined over all clusters and protected groups as:

$$balance := \min_{a \in [k], b \in [m]} \min\{R_{a,b}, \frac{1}{R_{a,b}}\}$$

Ensuring fairness in ML models can occur at three stages of the learning pipeline [2, 18]: 1) before-training, 2) during-training, or 3) after-training phases. In our investigation, we focus on fairness on the after-training phase, where a *post-processing* procedure adjusts the clustering outputs of the vanilla clustering algorithm to adhere to specified fairness criteria.

The structure of the report reflects the stages of our pipeline. Section 2 presents the social network datasets used in our study. Section 3 details the node embeddings employed to map the graphs into a lower-dimensional embedding space, while Section 4 outlines the vanilla clustering algorithms used for initial clustering. Subsequently, Section 5 elaborates on the post-processing method applied to transform the vanilla clusters into fair clusters that align with the balance notion. Section 6 provides the experimental setup, while Section 7 presents the experimental results of our approach. Finally, in Section 8, we summarize the conclusions drawn from our study.

2 Datasets

We conduct experiments on four datasets that are network graphs, including one synthetic graph and three real social network graphs. Each of these datasets is annotated with information about vertex group assignments. While the synthetic dataset comes also annotated with cluster assignments, the three real social networks lack ground truth cluster information. To address this, we apply the Louvain algorithm [1] to obtain the number of clusters for these networks. Subsequently, we utilize this fixed number of clusters to perform clustering using other algorithms, eliminating the need to search for the optimal number of clusters per algorithm.

The synthetic dataset is an instantiation of a variation proposed by Kleindessner et al. [15] of the well-known stochastic block model (SBM; [14]). The SBM is a widely used random graph model for assessing clustering algorithm performance. In the traditional SBM, there is a ground-truth clustering of the vertex set $V = [n]$ into k clusters. In a random graph generated from the model, the probability of connection between two vertices i and j depends solely on the clusters to which i and j belong. In the variant proposed by Kleindessner et al. [15], which we use, the SBM is modified so that vertices belong to groups, and within each cluster, each group is represented in the same proportion as in the entire dataset V. This modification ensures a fair, balanced ground-truth clustering, following the balance notion (see Section 1).

The next two datasets represent high school friendship networks [17]. Vertices correspond to students and are divided into two groups based on gender. The FRIENDSHIPNET dataset comprises 127 vertices and 396 edges, where an edge indicates reported friendship between two students. On the other hand, FACEBOOKNET consists of 155 vertices and 1412 edges, where an edge represents friendship on Facebook.

The last dataset is the DRUGNET dataset, a network representing acquaintanceships among drug users, constructed after a two-year study of ethnographic observations of people's drug habits [25]. The network consists of 185 edges and 265 vertices divided into two groups based on gender.

The statistics for the four datasets mentioned above are presented in Table 1. The details of the synthetic dataset pertain to our implementation of the SBM model variant. A diverse range of ratios is observed for cluster and group balance: the synthetic dataset is balanced, the FACEBOOKNET exhibits nearly balanced characteristics in terms of both cluster and group balance. In contrast, DRUGNET and FRIENDSHIPNET are considered imbalanced.

3 Node embeddings

We work with social network datasets aiming to capture individuals' structural information within the network, such as their neighborhoods and community affiliations. To represent each person in a lower-dimensional vector, we leverage three node embedding algorithms: node2vec [13], deepwalk [20], and fairwalk [21]. These algorithms are inspired by the word2vec skip-gram model [19], which is a language model that maximizes the co-occurrence probability among the words that appear within a window in a sentence. Therefore there are proximity-preserving node embeddings methods, as they generate sequences of neighboring nodes through random walks, distilling representations based on local neighborhood information [23].

Node2vec [13] algorithm employs a 2nd order random walk to generate network neighborhoods for nodes, using parameters p and q to control exploration speed and neighborhood retention. Higher p values reduce revisits, fostering moderate graph exploration, while lower p values promote backtracking for proximity to the starting node. A high q biases the walk toward nodes close to the previous step, while a low q encourages outward exploration. Configuring p and q yields diverse similarities between nodes, enabling tuning for homophily (nodes in the same community) or structural equivalence (similar connection patterns).

DeepWalk [20] algorithm learns feature representations by simulating uniform random walks. The sampling strategy in DeepWalk can be viewed as a special case of node2vec with p = 1 and q = 1 [13].

Both node2vec and deepwalk can produce fairness-unaware representations, capturing inherent biases in the network. To explore fairness-aware embeddings, we employ the Fairwalk algorithm [21], an extension of node2vec that considers sensitive attributes during random walks. Fairwalk generates more diverse network neighborhood representations [10].

4 Vanilla Clustering Algorithms

For the vanilla clustering component of our approach, we conducted experiments with four clustering algorithms. Following the taxonomy proposed by Fahad et al. [9], we used two partitioning-based algorithms - K-means and K-medoids. These type of algorithms iteratively identify similarity among intra-cluster points based on their distances from the cluster centroid [11]. Additionally, we employed an hierarchical-based algorithm, the agglomerative clustering, where the bottom-up approach starts with each data point as a separate cluster. By employing a specific distance metric, the proximity between two points is calculated, and the closest pairs are progressively merged into a single cluster. This iterative process continues until all data points are combined into a single cluster [11]. Lastly, we incorporated a density-based clustering algorithm — spectral clustering. Such methods

DATASET	NO. VERTICES	NO. EDGES	NO. CLUSTERS	CLUSTER BALANCE RATIO	NO. GROUP	GROUP BALANCE RATIO
SYNTHETIC DATASET	240	6366	4	1.	2	1.
FRIENDSHIPNET	127	396	8	0.21	2	0.49
FACEBOOKNET	155	1412	6	0.44	2	0.72
DRUGNET	185	265	11	0.29	2	0.25

Table 1: Statistics for the datasets, including vertex count (no. vertices), edge count (no. edges), number of clusters (no. clusters) determined by the Louvain algorithm, cluster balance ratio, number of groups (no. groups), and group balance ratio. The balance ratio is computed as $\frac{\min(C_1,..,C_k)}{\max(C_1,..,C_k)}$, where $C_1,..,C_k$ represent either the clusters or the groups.

leverage the density of data points in the data space to form clusters [11].

Considering the necessity to incorporate the centroid of each cluster (see Section 5), and recognizing that the agglomerative and spectral clustering algorithms do not inherently provide the centroid of each cluster, we used the NearestCentroid classifier [24] for these two algorithms. This classifier represents each class by the centroid of its members, akin to the label updating phase of the KMeans algorithm. It's worth noting that this classifier may face challenges with non-convex classes, although such non-convexity is not observed in our networks.

5 Post-processing Method

In our approach to fair community detection, we implement a post-processing step within the learning pipeline. This involves adjusting the clustering outputs to ensure fairness. Initially, a vanilla clustering algorithm is applied (see Section 4), yielding initial clustering results. Subsequently, these results are refined to generate fair clusters, as the vanilla algorithm itself does not explicitly optimize for both clustering accuracy and fairness simultaneously.

Our post-processing method, outlined in Algorithm 5, takes as input the initial clustering assignments of vertices, the distances of vertices from their cluster centroids (provided by the initial clustering step), and the group membership of each vertex. The output of this method is the revised clustering assignments, optimized for fairness according to the balance notion metric (detailed in Section 1).

To transform the initial clusters into fair clusters, we identify clusters that are unfair. Within these clusters, we extract points that belong to an over-represented group (i.e., a group with a proportion in the cluster exceeding its proportion in the dataset). Points from over-represented groups are then reassigned to the closest unfair cluster where the represented group is underrepresented. This reassignment process continues until the balance criterion is satisfied (with a customizable error threshold) or until a manually specified iteration limit is reached. An issue that we did not consider in the current implementation of our algorithm is that we do not take into account the imbalance of the size of the clusters, therefore the re-assignments may lead to empty clusters.

Algorithm 1 Post-processing step

procedure POST-PROCESSING-STEP($vanilla\_assign, colors, dist\_centers, max\_stop$)
  $nodes \leftarrow len(vanilla\_assign)$
  $fair\_fraction \leftarrow \frac{\sum l}{nodes}, \forall l \in colors, \forall k \in n\_clusters$
  $current\_fraction \leftarrow \frac{\sum colors\_in\_k}{nodes\_k}, \forall k \in n\_clusters$
  $diff\_fraction \leftarrow \frac{current\_fraction}{fair\_fraction}, \forall k \in n\_clusters$
  $fair\_assign \leftarrow copy(vanilla\_assign)$
  $stop \leftarrow 0$
  while $any(diff\_fraction) \geq err$ or $stop \leq max\_stop$ **do**
    for $\forall k \in n\_clusters$ **do**
      for $\forall a \in unique(colors)$ **do**
        if $diff\_fraction[k,a] \geq err$ **then**
          $max\_dist\_point \leftarrow \max(dist\_center_k)$ ▷ This is a comment
          $unfair\_under\_clusters \leftarrow diff\_fraction_a \leq 0.$
          $nearest\_cluster \leftarrow asc\_sort(dist\_centers[max\_dist\_point, unfair\_under\_clusters])[0]$
          $fair\_assign[max\_dist\_point] \leftarrow nearest\_cluster$
          $current\_fraction \leftarrow \frac{\sum colors\_in\_k}{nodes\_k}, \forall k \in n\_clusters$
          $diff\_fraction \leftarrow \frac{current\_fraction}{fair\_fraction}, \forall k \in n\_clusters$
        end if
      end for
    end for
  end while
  return $fair\_assign$
end procedure

6 Experimental Setup

Preprocessing is applied to all real network datasets, i.e. FRIENDSHIPNET, FACEBOOKNET, and DRUGNET, to remove nodes with unknown group attributes and to retain only the largest connected component for analysis. The dataset statistics presented in Table 1 are derived after this preprocessing step. Additionally, the synthetic dataset is created by translating the MATLAB implementation of the SBM model variation, obtained from the code repository associated with the paper[1], into Python.

For node2vec and deepwalk algorithms, we employ the implementations from the karate-club GitHub repository[2], while fairwalk embeddings are generated using the implementation from another GitHub repository[3]. Default parameter values are retained for all algorithms, maintaining a fixed walk_length parameter of 15. We explore the homophily and structural equivalence hypotheses by experimenting with parameters p and q for node2vec and fairwalk algorithms. Specifically, for homophily, we explore settings (p, q) = [(1, 0.5), (0.5, 1)], and for structural equivalence, we consider settings (p, q) = [(2, 1), (1, 2)].

In the case of clustering algorithms and the NearestNeighbor algorithm, implementations from the scikit-learn library[4] are employed, with default parameter values retained.

Concerning the evaluation metrics employed, the first is the balance notion metric, which evaluates the clustering assignments with respect to the balance notion (see Section 1). The next two metrics evaluate the clustering quality, and are the graph modularity [5], and the silhouette score [22]. The final metric is the normalized mutual information (nmi) [8], a normalized version of mutual information. We use it to measure the agreement between the ground truth (and where the ground truth clustering is missing, with the Louvain algorithm's clustering) and either the vanilla clustering or the fair clustering assignments.

7 Experimental Results

First, we analyze and interpret the results individually for each dataset, followed by summarizing and drawing conclusions based on the overall findings.

Synthetic Dataset Results. The outcomes for the synthetic dataset are presented in Table 2. Optimal results are achieved when considering both quality and fairness constraints, striking a balance between these two aspects. Notably, the most favorable trade-off between fairness and clustering quality is observed with node2vec embeddings (p=0.5, q=1) and kmedoids clustering, exhibiting the highest fair balance. Although the modularity experiences a slight decrease of 0.87% from its peak value, the silhouette score declines by 110.2%. However, this reduction in silhouette score is justified considering the overall trade-offs to achieve fair clustering.

FRIENDSHIPNET Results. The outcomes for the FRIENDSHIPNET dataset are detailed in Table 3. Optimal results are achieved when considering both quality and fairness constraints, striking a balance between these two aspects. The optimal trade-off between fairness and clustering quality is observed with deepwalk embeddings and agglomerative clustering, displaying the highest fair balance, while the modularity and silhouette values approach their peaks (decreasing by 6.9% and 15.2%, respectively, compared to their highest values). The normalized mutual information (nmi) between the Louvain algorithm assignments and the fair clustering assignments demonstrates close alignment.

FACEBOOKNET Results. The results for the FACEBOOKNET dataset are outlined in Table 4. The trade-off balance between fairness and clustering quality is apparent with node2vec embeddings (p=1, q=2) and kmedoids clustering, showcasing the highest fair balance. However, this comes at the expense of a 25.4% decrease in modularity and a 72.9% decrease in silhouette score compared to their highest values. Achieving a balance between fairness and clustering quality proved challenging in this dataset, as increasing fairness led to a rapid decrease in clustering quality. This trend can be attributed to the low balance score for the vanilla clustering, indicating significant unfairness in the initial clusters. Rectifying this unfairness incurred a cost in clustering quality.

DRUGNET Results. The outcomes for the DRUGNET dataset are presented in Table 5. The optimal trade-off between fairness and clustering quality is achieved with node2vec embeddings (p=2, q=1) and kmeans clustering, displaying the highest modularity and silhouette score with a slight 0.48% decrease in balance score. Notably, the vanilla modularity achieves its peak, and the vanilla silhouette is slightly 0.38% lower than its highest value.

Table 6 summarizes the best evaluation outcomes across all datasets. Notably, fairwalk embeddings fail to strike the optimal balance between fairness and clustering quality. While they may achieve high balance metric scores, metrics assessing clustering quality remain notably low. Conversely, deepwalk and particularly node2vec embeddings yield the most promising results. Regarding the p and q parameters of node2vec, it is observed that embeddings tuned towards structural equivalence generally yield superior results, except in the synthetic dataset. In this dataset, however, nodes are highly connected, suggesting that the homophily setting may be more effective. Among the vanilla clustering algorithms, the partitioning-based algorithms, i.e. the K-means

[1] https://github.com/matthklein/fair_spectral_clustering
[2] https://github.com/benedekrozemberczki/karateclub
[3] https://github.com/tjzetty/fairwalk
[4] https://scikit-learn.org/stable/

EMBED	CL_ALG	VANILLA_BALANCE	FAIR_BALANCE	GT_MODULARITY	VANILLA_MODULARITY	FAIR_MODULARITY	GT_SIL	VANILLA_SIL	FAIR_SIL	VANILLA_NMI	FAIR_NMI
node2vec_1_2	kmeans	0.000	0.000	0.250	0.388	0.288	-0.005	0.066	0.005	0.019	0.022
node2vec_1_2	kmedoids	0.692	0.764	0.250	0.302	0.279	-0.005	0.033	0.025	0.028	0.024
node2vec_1_2	agglomerative clust.	0.100	0.923	0.250	0.393	0.165	-0.005	0.055	-0.028	0.021	0.038
node2vec_1_2	spectral clustering	0.030	1.000	0.250	0.380	0.107	-0.005	0.082	-0.039	0.017	0.017
node2vec_1_0.5	kmeans	0.000	(-100%) 0.000	0.250	0.368	**0.344**	-0.005	0.069	(-22.4%) 0.038	0.018	0.022
node2vec_1_0.5	kmedoids	0.282	1.000	0.250	0.342	0.208	-0.005	0.041	-0.014	0.024	0.028
node2vec_1_0.5	agglomerative clust.	0.111	1.000	0.250	0.372	0.165	-0.005	0.057	-0.037	0.015	0.023
node2vec_1_0.5	spectral clustering	0.065	0.241	0.250	0.377	0.245	-0.005	0.075	-0.030	0.032	0.041
node2vec_0.5_1	kmeans	0.034	1.000	0.250	0.392	0.206	-0.007	0.068	-0.046	0.006	0.013
node2vec_0.5_1	kmedoids	0.148	**1.000**	0.250	(-0.87%) 0.341	0.248	-0.007	0.041	(-110.2%) -0.005	0.029	0.041
node2vec_0.5_1	agglomerative clust.	0.070	0.203	0.250	0.359	0.240	-0.007	0.049	-0.006	0.020	0.023
node2vec_0.5_1	spectral clustering	0.057	0.429	0.250	0.380	0.133	-0.007	0.083	-0.034	0.020	0.034
node2vec_2_1	kmeans	0.000	(-100.%) 0.000	0.250	0.374	(-3.78%) 0.331	-0.005	0.075	**0.049**	0.038	0.017
node2vec_2_1	kmedoids	0.453	0.548	0.250	0.313	0.304	-0.005	0.033	0.027	0.022	0.021
node2vec_2_1	agglomerative clust.	0.160	0.169	0.250	0.351	0.280	-0.005	0.070	0.009	0.019	0.029
node2vec_2_1	spectral clustering	0.062	0.125	0.250	0.390	0.308	-0.005	0.079	-0.002	0.024	0.012
deepwalk	kmeans	0.082	0.519	0.250	0.264	0.147	-0.019	0.129	-0.021	0.008	0.010
deepwalk	kmedoids	0.122	0.125	0.250	0.244	0.224	-0.019	0.080	0.018	0.010	0.012
deepwalk	agglomerative clust.	0.125	1.000	0.250	0.263	0.112	-0.019	0.119	-0.074	0.010	0.035
deepwalk	spectral clustering	0.152	0.500	0.250	0.243	0.195	-0.019	0.175	0.012	0.016	0.008
fairwalk_1_2	kmeans	0.000	0.000	0.250	0.374	0.331	-0.005	0.075	0.049	0.038	0.017
fairwalk_1_2	kmedoids	0.453	0.548	0.250	0.313	0.304	-0.005	0.033	0.027	0.022	0.021
fairwalk_1_2	agglomerative clust.	0.160	0.169	0.250	0.351	0.280	-0.005	0.070	0.009	0.019	0.029
fairwalk_1_2	spectral clustering	0.062	0.125	0.250	0.390	0.308	-0.005	0.079	-0.002	0.024	0.012
fairwalk_1_0.5	kmeans	0.000	0.000	0.250	0.374	0.331	-0.005	0.075	0.049	0.038	0.017
fairwalk_1_0.5	kmedoids	0.453	0.548	0.250	0.313	0.304	-0.005	0.033	0.027	0.022	0.021
fairwalk_1_0.5	agglomerative clust.	0.160	0.169	0.250	0.351	0.280	-0.005	0.070	0.009	0.019	0.029
fairwalk_1_0.5	spectral clustering	0.062	0.125	0.250	0.390	0.308	-0.005	0.079	-0.002	0.024	0.012
fairwalk_0.5_1	kmeans	0.000	0.000	0.250	0.374	0.331	-0.005	0.075	0.049	0.038	0.017
fairwalk_0.5_1	kmedoids	0.453	0.548	0.250	0.313	0.304	-0.005	0.033	0.027	0.022	0.021
fairwalk_0.5_1	agglomerative clust.	0.160	0.169	0.250	0.351	0.280	-0.005	0.070	0.009	0.019	0.029
fairwalk_0.5_1	spectral clustering	0.062	0.125	0.250	0.390	0.308	-0.005	0.079	-0.002	0.024	0.012
fairwalk_2_1	kmeans	0.000	0.000	0.250	0.374	0.331	-0.005	0.075	0.049	0.038	0.017
fairwalk_2_1	kmedoids	0.453	0.548	0.250	0.313	0.304	-0.005	0.033	0.027	0.022	0.021
fairwalk_2_1	agglomerative clust.	0.160	0.169	0.250	0.351	0.280	-0.005	0.070	0.009	0.019	0.029
fairwalk_2_1	spectral clustering	0.062	0.125	0.250	0.390	0.308	-0.005	0.079	-0.002	0.024	0.012

Table 2: Evaluation results for the synthetic dataset, considering various node embeddings and p, q parameter configurations for the node2vec and fairwalk algorithms (embed column, where the value of p is the first number in the name of the embedding and the value of q is the second). The results encompass multiple clustering algorithms. Best results per metric are highlighted in bold, and for the superior outcomes, we indicate the difference from the highest value for the other metrics. The best result is enclosed with a red border, signifying the optimal trade-off between cluster quality and fairness.

and the K-medoids, generally produce more cohesive clusters, facilitating their transformation into fair clusters. In the case of the FRIENDSHIPNET dataset, agglomerative clustering demonstrates superior performance. Concerning the clustering quality of the datasets themselves, FRIENDSHIPNET and DRUGNET exhibit the most favorable structures in terms of modularity and silhouette score, whereas the synthetic dataset and FACEBOOKNET, characterized by high connectivity, demonstrate lower cluster quality.

8 Conclusions

In this study, we introduced a novel yet straightforward fair clustering method, acting as a post-processing step after generating cluster solutions of high quality that lack optimization for fairness. We evaluated our approach using the fairness metric proposed by Chierichetti et al. [4] alongside two clustering quality metrics. We employed three distinct node embedding models to generate the embedding space for our datasets, adjusting model parameters to account for homophily and structural equivalence similarity. Additionally, we investigated the impact of fair node embeddings generated from a node embedding model that takes into account the sensitive attribute of the points. For the initial vanilla clustering step, we integrated four different clustering algorithms. The combination of the embedding models and the clustering algorithms resulted in 36 unique models per dataset for evaluation.

Our benchmarking analysis revealed that fair node embeddings do not consistently yield high-quality clusters. Despite achieving high balance metric scores, metrics assessing clustering quality remained notably low. Conversely, deepwalk and particularly node2vec embeddings showed the most promising results. Regarding the node embedding model's configuration that adjust the embeddings towards the homophily or the structural equivalence similarity, embeddings tuned towards structural equivalence generally outperformed others, except in the synthetic dataset, where nodes exhibit high connectivity, suggesting that the homophily setting may be more effective. Among the vanilla clustering algorithms, partitioning-based methods, such as K-means and K-medoids, generally produced more cohesive clusters, facilitating their transformation into fair clusters.

For future research, we aim to enhance our post-processing method to address the imbalance among the size of the clusters generated by clustering algorithms. Additionally, we seek to determine the optimal number of clusters per clustering algorithm in cases where ground truth clusters are unavailable, rather than relying on the clusters generated by the Louvain algorithm.

embed	cl_alg	lv_balance	vanilla_balance	fair_balance	lv_modularity	vanilla_modularity	fair_modularity	lv_sil	vanilla_sil	fair_sil	vanilla_nmi	fair_nmi
node2vec_1_2	kmeans	0.735	0.595	**0.972**	0.705	0.702	0.653	0.305	0.277	0.204	0.824	0.706
node2vec_1_2	kmedoids	0.735	0.000	0.756	0.705	0.709	0.663	0.305	0.244	0.173	0.761	0.654
node2vec_1_2	agglomerative clust.	0.735	0.735	(-1.75%) 0.955	0.705	0.733	**0.698**	0.305	0.293	(-30%) 0.212	**0.944**	**0.797**
node2vec_1_2	spectral clustering	0.735	0.496	0.827	0.705	0.706	0.650	0.305	0.264	0.178	0.807	0.656
node2vec_1_0.5	kmeans	0.735	0.427	0.672	0.705	0.721	0.650	0.296	0.297	0.163	0.779	0.646
node2vec_1_0.5	kmedoids	0.735	0.000	0.529	0.705	0.635	0.562	0.296	0.186	0.077	0.668	0.558
node2vec_1_0.5	agglomerative clust.	0.735	0.735	0.968	0.705	0.735	0.667	0.296	0.313	0.220	0.898	0.754
node2vec_1_0.5	spectral clustering	0.735	**0.756**	0.756	0.705	0.723	0.683	0.296	0.311	0.240	0.849	0.779
node2vec_0.5_1	kmeans	0.735	0.477	0.756	0.705	0.683	0.631	0.289	0.219	0.138	0.774	0.629
node2vec_0.5_1	kmedoids	0.735	0.000	0.000	0.705	0.656	0.589	0.289	0.200	0.138	0.767	0.626
node2vec_0.5_1	agglomerative clust.	0.735	0.744	**0.972**	0.705	**0.737**	0.679	0.289	0.307	0.199	0.913	0.722
node2vec_0.5_1	spectral clustering	0.735	0.661	0.864	0.705	0.691	0.650	0.289	0.265	0.177	0.742	0.650
node2vec_2_1	kmeans	0.735	0.606	0.882	0.705	0.706	0.640	0.306	0.255	0.151	0.830	0.671
node2vec_2_1	kmedoids	0.735	0.504	0.909	0.705	0.714	0.616	0.306	0.269	0.143	0.765	0.645
node2vec_2_1	agglomerative clust.	0.735	**0.756**	0.962	0.705	0.729	0.690	0.306	0.317	0.261	0.883	0.767
node2vec_2_1	spectral clustering	0.735	0.735	0.772	0.705	0.717	0.669	0.306	0.318	0.215	0.851	0.706
deepwalk	kmeans	0.735	0.496	0.000	0.705	0.667	0.626	0.234	**0.380**	0.272	0.687	0.575
deepwalk	kmedoids	0.735	0.000	0.661	0.705	0.671	0.633	0.234	0.330	0.262	0.660	0.592
deepwalk	agglomerative clust.	0.735	0.551	**0.972**	0.705	0.712	(-6.9%) 0.677	0.234	0.333	(-15.2%) 0.257	0.765	0.723
deepwalk	spectral clustering	0.735	0.496	(-11.8%) 0.864	0.705	0.626	(-15.5%) 0.583	0.234	0.368	**0.303**	0.682	0.580
fairwalk_1_2	kmeans	0.735	0.606	0.882	0.705	0.706	0.640	0.306	0.255	0.151	0.830	0.671
fairwalk_1_2	kmedoids	0.735	0.504	0.909	0.705	0.714	0.616	0.306	0.269	0.143	0.765	0.645
fairwalk_1_2	agglomerative clust.	0.735	**0.756**	0.962	0.705	0.729	0.690	0.306	0.317	0.261	0.883	0.767
fairwalk_1_2	spectral clustering	0.735	0.735	0.772	0.705	0.717	0.669	0.306	0.318	0.215	0.851	0.706
fairwalk_1_0.5	kmeans	0.735	0.606	0.882	0.705	0.706	0.640	0.306	0.255	0.151	0.830	0.671
fairwalk_1_0.5	kmedoids	0.735	0.504	0.909	0.705	0.714	0.616	0.306	0.269	0.143	0.765	0.645
fairwalk_1_0.5	agglomerative clust.	0.735	**0.756**	0.962	0.705	0.729	0.690	0.306	0.317	0.261	0.883	0.767
fairwalk_1_0.5	spectral clustering	0.735	0.735	0.772	0.705	0.717	0.669	0.306	0.318	0.215	0.851	0.706
fairwalk_0.5_1	kmeans	0.735	0.606	0.882	0.705	0.706	0.640	0.306	0.255	0.151	0.830	0.671
fairwalk_0.5_1	kmedoids	0.735	0.504	0.909	0.705	0.714	0.616	0.306	0.269	0.143	0.765	0.645
fairwalk_0.5_1	agglomerative clust.	0.735	**0.756**	0.962	0.705	0.729	0.690	0.306	0.317	0.261	0.883	0.767
fairwalk_0.5_1	spectral clustering	0.735	0.735	0.772	0.705	0.717	0.669	0.306	0.318	0.215	0.851	0.706
fairwalk_2_1	kmeans	0.735	0.606	0.882	0.705	0.706	0.640	0.306	0.255	0.151	0.830	0.671
fairwalk_2_1	kmedoids	0.735	0.504	0.909	0.705	0.714	0.616	0.306	0.269	0.143	0.765	0.645
fairwalk_2_1	agglomerative clust.	0.735	**0.756**	0.962	0.705	0.729	0.690	0.306	0.317	0.261	0.883	0.767
fairwalk_2_1	spectral clustering	0.735	0.735	0.772	0.705	0.717	0.669	0.306	0.318	0.215	0.851	0.706

Table 3: Evaluation results for the FRIENDSHIPNET dataset, considering various node embeddings and p, q parameter configurations for the node2vec and fairwalk algorithms (embed column, where the value of p is the first number in the name of the embedding and the value of q is the second). The results encompass multiple clustering algorithms. Best results per metric are highlighted in bold, and for the superior outcomes, we indicate the difference from the highest value for the other metrics. The best result is enclosed with a red border, signifying the optimal trade-off between cluster quality and fairness.

EMBED	CL_ALG	LV_BALANCE	VANILLA_BALANCE	FAIR_BALANCE	LV_MODULARITY	VANILLA_MODULARITY	FAIR_MODULARITY	LV_SIL	VANILLA_SIL	FAIR_SIL	VANILLA_NMI	FAIR_NMI
node2vec_1_2	kmeans	0.373	0.373	0.726	0.527	**0.554**	0.441	0.265	0.268	0.141	**0.939**	0.650
node2vec_1_2	kmedoids	0.373	**0.477**	**0.978**	0.527	0.462	(-25.4%) 0.384	0.265	0.152	(-72.9%) 0.093	0.566	0.456
node2vec_1_2	agglomerative clust.	0.373	0.298	0.646	0.527	0.545	0.447	0.265	0.259	0.091	0.850	0.639
node2vec_1_2	spectral clustering	0.373	0.000	0.000	0.527	0.512	0.458	0.265	0.262	0.208	0.697	0.568
node2vec_1_0.5	kmeans	0.373	0.361	(-53.9%) 0.451	0.527	0.553	**0.515**	0.231	0.235	(-48.1%) 0.178	0.938	**0.782**
node2vec_1_0.5	kmedoids	0.373	0.251	0.795	0.527	0.456	0.392	0.231	0.153	0.064	0.682	0.517
node2vec_1_0.5	agglomerative clust.	0.373	0.341	0.734	0.527	0.547	0.452	0.231	0.228	0.048	0.848	0.619
node2vec_1_0.5	spectral clustering	0.373	0.000	0.000	0.527	0.511	0.442	0.231	0.268	0.206	0.677	0.568
node2vec_0.5_1	kmeans	0.373	0.351	0.000	0.527	0.543	0.437	0.249	0.237	0.111	0.880	0.612
node2vec_0.5_1	kmedoids	0.373	0.140	0.000	0.527	0.418	0.349	0.249	0.170	0.104	0.546	0.356
node2vec_0.5_1	agglomerative clust.	0.373	0.341	0.923	0.527	0.543	0.371	0.249	0.237	0.037	0.785	0.508
node2vec_0.5_1	spectral clustering	0.373	0.000	0.000	0.527	0.512	0.444	0.249	0.265	0.202	0.697	0.494
node2vec_2_1	kmeans	0.373	0.344	0.000	0.527	0.543	0.398	0.255	0.242	0.095	0.807	0.580
node2vec_2_1	kmedoids	0.373	0.227	0.000	0.527	0.495	0.429	0.255	0.173	0.114	0.671	0.496
node2vec_2_1	agglomerative clust.	0.373	0.367	0.815	0.527	0.530	0.407	0.255	0.242	0.054	0.784	0.542
node2vec_2_1	spectral clustering	0.373	0.000	0.397	0.527	0.513	0.441	0.255	0.273	0.164	0.713	0.528
deepwalk	kmeans	0.373	0.361	**0.978**	0.527	0.444	0.238	0.184	0.212	-0.095	0.612	0.401
deepwalk	kmedoids	0.373	0.000	0.000	0.527	0.388	0.305	0.184	0.189	0.065	0.551	0.410
deepwalk	agglomerative clust.	0.373	0.000	0.000	0.527	0.442	0.307	0.184	0.260	0.053	0.602	0.357
deepwalk	spectral clustering	0.373	0.000	(-100%) 0.000	0.527	0.469	(-16.3%) 0.431	0.184	**0.358**	**0.343**	0.596	0.535
fairwalk_1_2	kmeans	0.373	0.344	0.000	0.527	0.543	0.398	0.255	0.242	0.095	0.807	0.580
fairwalk_1_2	kmedoids	0.373	0.227	0.000	0.527	0.495	0.429	0.255	0.173	0.114	0.671	0.496
fairwalk_1_2	agglomerative clust.	0.373	0.367	0.815	0.527	0.530	0.407	0.255	0.242	0.054	0.784	0.542
fairwalk_1_2	spectral clustering	0.373	0.000	0.397	0.527	0.513	0.441	0.255	0.273	0.164	0.713	0.528
fairwalk_1_0.5	kmeans	0.373	0.344	0.000	0.527	0.543	0.398	0.255	0.242	0.095	0.807	0.580
fairwalk_1_0.5	kmedoids	0.373	0.227	0.000	0.527	0.495	0.429	0.255	0.173	0.114	0.671	0.496
fairwalk_1_0.5	agglomerative clust.	0.373	0.367	0.815	0.527	0.530	0.407	0.255	0.242	0.054	0.784	0.542
fairwalk_1_0.5	spectral clustering	0.373	0.000	0.397	0.527	0.513	0.441	0.255	0.273	0.164	0.713	0.528
fairwalk_0.5_1	kmeans	0.373	0.344	0.000	0.527	0.543	0.398	0.255	0.242	0.095	0.807	0.580
fairwalk_0.5_1	kmedoids	0.373	0.227	0.000	0.527	0.495	0.429	0.255	0.173	0.114	0.671	0.496
fairwalk_0.5_1	agglomerative clust.	0.373	0.367	0.815	0.527	0.530	0.407	0.255	0.242	0.054	0.784	0.542
fairwalk_0.5_1	spectral clustering	0.373	0.000	0.397	0.527	0.513	0.441	0.255	0.273	0.164	0.713	0.528
fairwalk_2_1	kmeans	0.373	0.344	0.000	0.527	0.543	0.398	0.255	0.242	0.095	0.807	0.580
fairwalk_2_1	kmedoids	0.373	0.227	0.000	0.527	0.495	0.429	0.255	0.173	0.114	0.671	0.496
fairwalk_2_1	agglomerative clust.	0.373	0.367	0.815	0.527	0.530	0.407	0.255	0.242	0.054	0.784	0.542
fairwalk_2_1	spectral clustering	0.373	0.000	0.397	0.527	0.513	0.441	0.255	0.273	0.164	0.713	0.528

Table 4: Evaluation results for the FACEBOOKNET dataset, considering various node embeddings and p, q parameter configurations for the node2vec and fairwalk algorithms (embed column, where the value of p is the first number in the name of the embedding and the value of q is the second). The results encompass multiple clustering algorithms. Best results per metric are highlighted in bold, and for the superior outcomes, we indicate the difference from the highest value for the other metrics. The best result is enclosed with a red border, signifying the optimal trade-off between cluster quality and fairness.

EMBED	CL_ALG	LV_BALANCE	VANILLA_BALANCE	FAIR_BALANCE	LV_MODULARITY	VANILLA_MODULARITY	FAIR_MODULARITY	LV_SIL	VANILLA_SIL	FAIR_SIL	VANILLA_NMI	FAIR_NMI
node2vec_1_2	kmeans	0.395	0.000	0.734	0.711	0.757	0.697	0.208	0.210	0.123	0.677	0.574
node2vec_1_2	kmedoids	0.395	0.000	0.000	0.711	0.729	0.627	0.208	0.078	0.042	0.590	0.416
node2vec_1_2	agglomerative clust.	0.395	0.000	0.000	0.711	0.771	0.681	0.208	0.231	0.107	**0.791**	0.608
node2vec_1_2	spectral clustering	0.395	0.000	0.571	0.711	0.710	0.635	0.208	0.243	0.125	0.700	0.554
node2vec_1_0.5	kmeans	0.395	0.395	0.623	0.711	0.783	0.711	0.236	0.247	0.118	0.739	0.621
node2vec_1_0.5	kmedoids	0.395	0.000	0.000	0.711	0.720	0.645	0.236	0.165	0.094	0.644	0.495
node2vec_1_0.5	agglomerative clust.	0.395	0.000	0.571	0.711	0.776	0.664	0.236	0.255	0.046	0.719	0.528
node2vec_1_0.5	spectral clustering	0.395	0.000	0.000	0.711	0.720	0.656	0.236	0.246	0.154	0.683	0.564
node2vec_0.5_1	kmeans	0.395	0.395	**0.894**	0.711	0.787	(-0.96%) 0.722	0.209	0.235	(-23.3%) 0.125	0.785	**0.661**
node2vec_0.5_1	kmedoids	0.395	0.000	0.000	0.711	0.734	0.654	0.209	0.081	0.057	0.575	0.435
node2vec_0.5_1	agglomerative clust.	0.395	0.000	0.000	0.711	0.773	0.708	0.209	0.239	0.156	0.760	0.620
node2vec_0.5_1	spectral clustering	0.395	0.000	0.000	0.711	0.705	0.634	0.209	0.231	0.128	0.645	0.534
node2vec_2_1	kmeans	0.395	0.395	(-0.48%) 0.890	0.711	**0.789**	**0.729**	0.238	0.259	**0.163**	0.751	0.637
node2vec_2_1	kmedoids	0.395	0.000	0.000	0.711	0.721	0.677	0.238	0.119	0.115	0.623	0.569
node2vec_2_1	agglomerative clust.	0.395	0.000	0.894	0.711	0.776	0.694	0.238	**0.260**	0.100	0.790	0.620
node2vec_2_1	spectral clustering	0.395	0.000	0.000	0.711	0.749	0.698	0.238	0.252	0.135	0.749	0.632
deepwalk	kmeans	0.395	**0.467**	0.584	0.711	0.744	0.679	0.166	0.181	0.094	0.660	0.554
deepwalk	kmedoids	0.395	0.000	0.000	0.711	0.663	0.608	0.166	0.068	0.043	0.526	0.454
deepwalk	agglomerative clust.	0.395	0.367	0.791	0.711	0.762	0.674	0.166	0.200	0.069	0.717	0.553
deepwalk	spectral clustering	0.395	0.000	0.000	0.711	0.634	0.579	0.166	0.086	0.060	0.580	0.437
fairwalk_1_2	kmeans	0.395	0.395	0.890	0.711	0.789	0.729	0.238	0.259	0.163	0.751	0.637
fairwalk_1_2	kmedoids	0.395	0.000	0.000	0.711	0.721	0.677	0.238	0.119	0.115	0.623	0.569
fairwalk_1_2	agglomerative clust.	0.395	0.000	0.894	0.711	0.776	0.694	0.238	0.260	0.100	0.790	0.620
fairwalk_1_2	spectral clustering	0.395	0.000	0.000	0.711	0.749	0.698	0.238	0.252	0.135	0.749	0.632
fairwalk_1_0.5	kmeans	0.395	0.395	0.890	0.711	0.789	0.729	0.238	0.259	0.163	0.751	0.637
fairwalk_1_0.5	kmedoids	0.395	0.000	0.000	0.711	0.721	0.677	0.238	0.119	0.115	0.623	0.569
fairwalk_1_0.5	agglomerative clust.	0.395	0.000	0.894	0.711	0.776	0.694	0.238	0.260	0.100	0.790	0.620
fairwalk_1_0.5	spectral clustering	0.395	0.000	0.000	0.711	0.749	0.698	0.238	0.252	0.135	0.749	0.632
fairwalk_0.5_1	kmeans	0.395	0.395	0.890	0.711	0.789	0.729	0.238	0.259	0.163	0.751	0.637
fairwalk_0.5_1	kmedoids	0.395	0.000	0.000	0.711	0.721	0.677	0.238	0.119	0.115	0.623	0.569
fairwalk_0.5_1	agglomerative clust.	0.395	0.000	0.894	0.711	0.776	0.694	0.238	0.260	0.100	0.790	0.620
fairwalk_0.5_1	spectral clustering	0.395	0.000	0.000	0.711	0.749	0.698	0.238	0.252	0.135	0.749	0.632
fairwalk_2_1	kmeans	0.395	0.395	0.890	0.711	0.789	0.729	0.238	0.259	0.163	0.751	0.637
fairwalk_2_1	kmedoids	0.395	0.000	0.000	0.711	0.721	0.677	0.238	0.119	0.115	0.623	0.569
fairwalk_2_1	agglomerative clust.	0.395	0.000	0.894	0.711	0.776	0.694	0.238	0.260	0.100	0.790	0.620
fairwalk_2_1	spectral clustering	0.395	0.000	0.000	0.711	0.749	0.698	0.238	0.252	0.135	0.749	0.632

Table 5: Evaluation results for the DRUGNET dataset, considering various node embeddings and p, q parameter configurations for the node2vec and fairwalk algorithms (embed column, where the value of p is the first number in the name of the embedding and the value of q is the second). The results encompass multiple clustering algorithms. Best results per metric are highlighted in bold, and for the superior outcomes, we indicate the difference from the highest value for the other metrics. The best result is enclosed with a red border, signifying the optimal trade-off between cluster quality and fairness.

DATASET	EMBED	CL_ALG	GT|LV_BALANCE	VANILLA_BALANCE	FAIR_BALANCE	GT|LV_MODULARITY	VANILLA_MODULARITY	FAIR_MODULARITY	GT|LV_SIL	VANILLA_SIL	FAIR_SIL	VANILLA_NMI	FAIR_NMI
SYNTHETIC DATASET	node2vec_0.5_1	kmedoids	1.	0.148	**1.000**	0.250	(-0.87%) 0.341	0.248	-0.007	0.041	(-110.2%) -0.005	0.029	0.041
FRIENDSHIPNET	deepwalk	agglomerative clust.	0.735	0.551	**0.972**	0.705	0.712	(-6.9%) 0.677	0.234	0.333	(-6.6%) 0.257	0.765	0.723
FACEBOOKNET	node2vec_1_2	kmedoids	0.373	**0.477**	**0.978**	0.527	0.462	(-25.4%) 0.384	0.265	0.152	(-72.9%) 0.093	0.566	0.456
DRUGNET	node2vec_2_1	kmeans	0.395	0.395	(-0.48%) 0.890	0.711	**0.789**	**0.729**	0.238	0.259	**0.163**	0.751	0.637

Table 6: Top evaluation outcomes across all datasets. The percentages in parentheses indicate the reduction in the result from its peak value.

References

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[2] Simon Caton and Christian Haas. Fairness in machine learning: A survey. *ACM Computing Surveys*, 2020.

[3] Anshuman Chhabra, Karina Masalkovaitė, and Prasant Mohapatra. An overview of fairness in clustering. *IEEE Access*, 9:130698–130720, 2021.

[4] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. *Advances in neural information processing systems*, 30, 2017.

[5] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6): 066111, 2004.

[6] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*, 2018.

[7] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on fairness, accountability and transparency*, pages 119–133. PMLR, 2018.

[8] Pablo A Estévez, Michel Tesmer, Claudio A Perez, and Jacek M Zurada. Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2):189–201, 2009.

[9] Adil Fahad, Najlaa Alshatri, Zahir Tari, Abdullah Alamri, Ibrahim Khalil, Albert Y Zomaya, Sebti Foufou, and Abdelaziz Bouras. A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, 2(3): 267–279, 2014.

[10] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1):1–51, 2023.

[11] Attri Ghosal, Arunima Nandy, Amit Kumar Das, Saptarsi Goswami, and Mrityunjoy Panday. A short review on different clustering techniques and their applications. *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, pages 69–83, 2020.

[12] Ben Green. Fair" risk assessments: A precarious approach for criminal justice reform. In *5th Workshop on fairness, accountability, and transparency in machine learning*, pages 1–5, 2018.

[13] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[14] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

[15] Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. In *International Conference on Machine Learning*, pages 3458–3467. PMLR, 2019.

[16] Lydia T Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, pages 3150–3158. PMLR, 2018.

[17] Rossana Mastrandrea, Julie Fournet, and Alain Barrat. Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys. *PloS one*, 10(9): e0136497, 2015.

[18] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[20] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[21] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards fair graph embedding. 2019.

[22] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[23] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *Journal of Complex Networks*, 9(2):cnab014, 2021.

[24] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

[25] Margaret R Weeks, Scott Clair, Stephen P Borgatti, Kim Radda, and Jean J Schensul. Social networks of drug users in high-risk sites: Finding the connections. *AIDS and Behavior*, 6:193–206, 2002.

[26] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017.

[27] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.