Survey of Network-Based AI Algorithms: Identification and Removal of Algorithmic Unfairness

MEHER SHASHWAT NIGAM*, ANOUSHKA VYAS*, and DR. NIMMI RANGASWAMY, International Institute of Information Technology, Hyderabad

Fig. 1. Fairness in AI

We present a survey of algorithmic unfairness in Network-Based AI Algorithms. We start by discussing why this survey is relevant in Section. 1, as this has been a hitherto less researched area. We study two main applications in this domain, community detection algorithms and knowledge graph-based methods. In Section. 2, we discuss how algorithmic bias can arise in the current popularly used methods. Further in Section. 3 we analyze the approaches that have been developed to reduce bias, where we focus mainly on two approaches. We also conducted an online survey which we describe in Section. 4. All relevant links can be found in Section. 6.

Additional Key Words and Phrases: Bias in AI, Bias Mitigation, Knowledge Graphs, Network-Based Algorithms, Community Detection

ACM Reference Format:

Meher Shashwat Nigam, Anoushka Vyas, and Dr. Nimmi Rangaswamy. 2021. Survey of Network-Based AI Algorithms: Identification and Removal of Algorithmic Unfairness. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/1122445.1122456

*Both authors contributed equally to this research.

Manuscript submitted to ACM

1 INTRODUCTION

AI has penetrated every aspect of our daily life. AI Algorithms make movie recommendations, suggest products to buy, and are increasingly used in high-stakes decisions in loan applications, dating and hiring. There are clear benefits to algorithmic decision-making; unlike people, machines do not become tired or bored, and can take into account orders of magnitude more factors than people can. However, like people, algorithms are vulnerable to biases that render their decisions "unfair".

With the widespread use of AI systems and applications in our everyday lives, it is important to take fairness issues into consideration while designing and engineering these types of systems. Such systems can be used in many sensitive environments to make important and life-changing decisions; thus, it is crucial to ensure that the decisions do not reflect discriminatory behavior toward certain groups or populations.

In the context of decision-making, *fairness is the absence of any prejudice or favoritism toward an individual or a group based on their inherent or acquired characteristics*. Thus, an unfair algorithm is one whose decisions are skewed toward a particular group of people. A canonical example comes from a tool used by courts in the United States to make parole decisions. The software, Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), measures the risk of a person to recommit another crime. Judges use COMPAS to decide whether to release an offender, or to keep him or her in prison. An investigation into the software found a bias against African-Americans: COMPAS is more likely to assign a higher risk score to African-American offenders than to Caucasians with the same profile. Similar findings have been made in other areas, such as an AI system that judges beauty pageant winners but was biased against darker-skinned contestants, or facial recognition software in digital cameras that overpredicts Asians as blinking. These biased predictions stem from the hidden or neglected biases in data or algorithms. In this survey, we will look at bias arising from the algorithms.

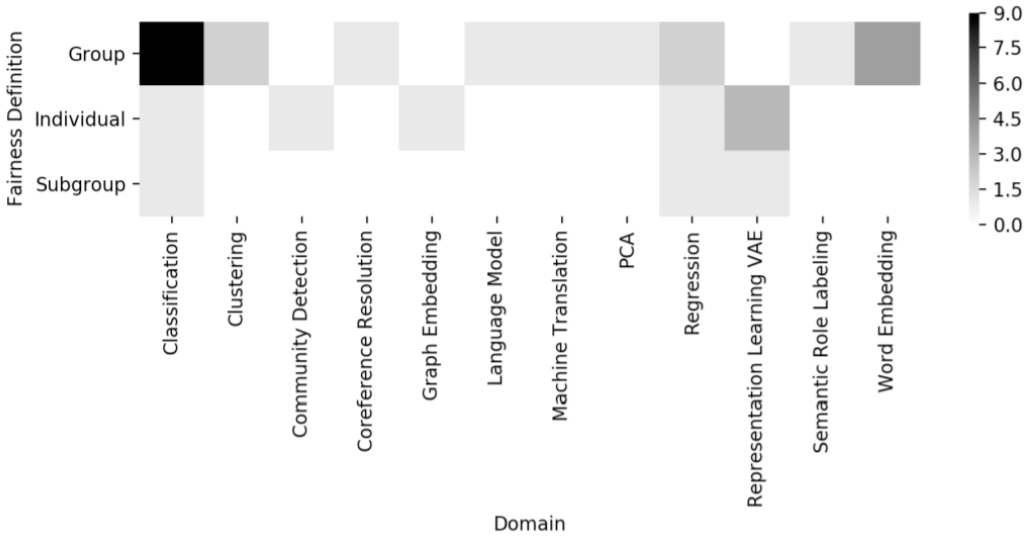

Fig. 2. Heatmap depicting distribution of previous work in fairness, grouped by domain and fairness definition. (Image courtesy [14])

Fairness definitions fall under different categories as follows:

(1) **Individual Fairness:** This means giving similar predictions to similar individuals.
(2) **Group Fairness:** This means giving similar predictions to different groups with similar properties.
(3) **Subgroup fairness:** Subgroup fairness intends to obtain the best properties of the group and individual notions of fairness. It is different than these notions but uses them in order to obtain better outcomes and asks whether this constraint holds over a large collection of subgroups.

While there have been many definitions of, and approaches to, fairness in the literature, the study in this area is anything but complete. Fairness and algorithmic bias still holds a number of research opportunities. As can be seen in Fig. 2, some subareas of machine learning, like natural language processing, to representation learning, to clustering, have all seen efforts to make their methodologies more fair but not every area has received the same amount of attention from the research community. Fig. 2 provides an overview of what has been done in different areas to address fairness — categorized by the fairness definition type and domain. Some areas (e.g., community detection and graph embeddings at the subgroup level) have received no attention in the literature, and could be fertile future research areas. This is our motivation behind surveying algorithmic unfairness in the area of community detection and knowledge graph based applications.

2 IDENTIFICATION OF BIAS IN NETWORK-BASED ALGORITHMS

2.1 Community Detection Methods

Inequalities in online communities and social networks can also potentially be another place where bias and discrimination can affect the populations. For example, in online communities users with a fewer number of friends or followers face a disadvantage of being heard in online social media. Existing community detection methods, can amplify this bias by ignoring these low-connected users in the network or by wrongfully assigning them to the irrelevant and small communities.

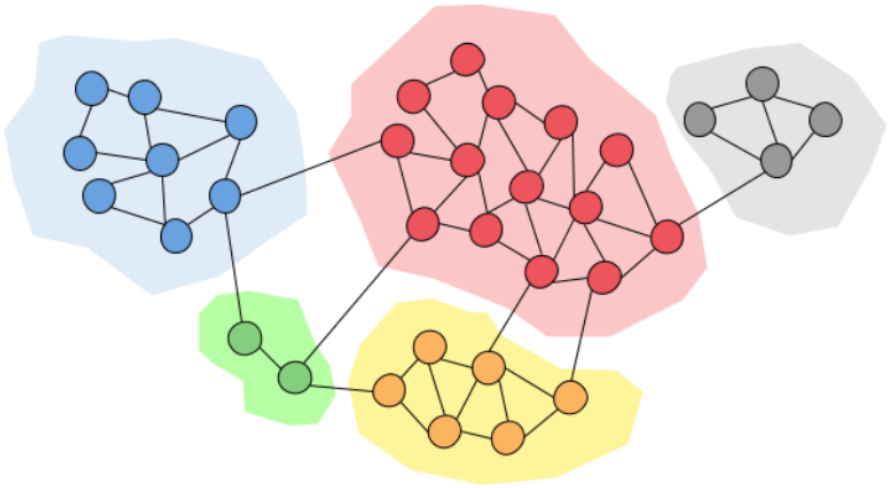

Fig. 3. Identifying communities in a network (Image courtesy [10])

2.1.1 *Problem and Applications:*

When analyzing different networks, it may be important to discover communities inside them. Community detection is a fundamental task in social network analysis [20], which identifies sub-groups within social networks. These groups can represent a variety of things including movie genre likings, political leanings, and deeply-held beliefs. Community detection techniques are useful for social media algorithms to discover people with common interests and keep them tightly connected. Traditionally, these groups are identified by searching for densely-connected groups of nodes in the graph [12], nodes in multiple communities can overlap. M. Girvan and M. Newman are two popular researchers in the domain of community detection. In one of their research, they have highlighted the community structure-property using social networks and biological networks [9]. According to them, network nodes are tightly connected in knit groups within communities and loosely connected between communities. More recently, attributed approaches go beyond merely the links to cluster nodes based upon their attributes and their network connections. [17].

2.1.2 *Difference Between Community Detection and Clustering:*

One can argue that community detection is similar to clustering. Clustering is a machine learning technique in which similar data points are grouped into the same cluster based on their attributes. Even though clustering can be applied to networks, it is a broader field in unsupervised machine learning which deals with multiple attribute types. On the other hand, community detection is specially tailored for network analysis which depends on a single attribute type called edges.

2.1.3 *Sources of Bias in the Existing Algorithms:*

Existing community detection approaches suffer from a major flaw: the inability to assign lowly-connected users into communities. One may argue that lowly-connected users are not well-blended into the social network, but is important to note that the information they provide can be crucial for better understanding the motivations and beliefs of the community. Failure to incorporate the lowly-connected users may result in biased results. For instance, studying groups within a social network, such as a Twitter retweet network, can be biased towards users whose tweets get a large number of retweets. This will lead to a biased analysis of the data in which not all users' voices are heard. Instead, those high-degree "popular" users with non-genuine retweets will end up having more voice in the study.

Consider the toy example shown in Fig. 4. Each node represents a user and the shapes represent difference of opinion in the network; the shade and different texture of each shape represent their variability of expressing opinions. Traditional community detection methods that solely focus on the highly-connected nodes would only capture the opinions of users within the dark-grey area; however, many diverse opinions are lost due to the fact that the users in the light grey area are excluded because of their low degree in the network.

2.1.4 *Types of Community Detection Algorithms:*

Although there are many different methods for the community detection [8, 16], we divide our analysis in two very broad sections:

(1) Methods that do not use node attributes for the community detection task.
(2) Methods that use node attributes for the community detection task.

We analyze and discuss a popular recent study by Mehrabi, et al. [13] and discuss it at length. The study identifies the existence of bias by showing what is omitted by existing community detection approaches. They look at two state-of-the-art approaches, CESNA [21], and the Louvain method [5]. Louvain uses only the network while assigning

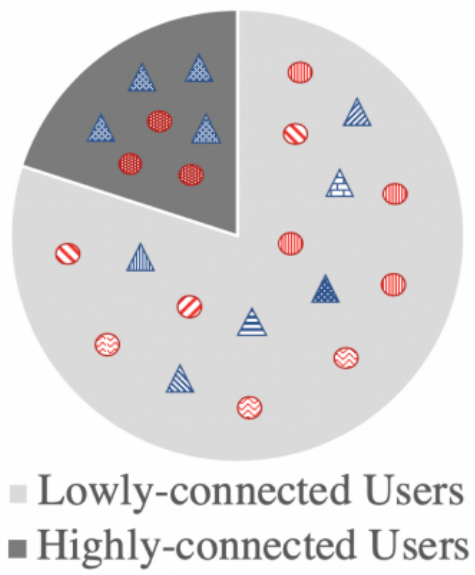

Fig. 4. Identifying communities in a network (Image courtesy [13])

communities, while CESNA uses both the network and user attributes. Later these methods are compared on two separate datasets, one based on Gamergate and one based on the 2016 U.S. Presidential Elections.

2.1.5 *Description of datasets considered:*

(1) **Gamergate [15]:** The Gamergate dataset consists of tweets posted in 2014 between months of August through October. The tweets surround the Gamergate controversy. It contains 21,441 users who collectively produced 104,914 tweets. These users fall into one of the two groups surrounding the controversy. One group consists of Gamergate supporters who are tweeting about ethics in journalism and believe that regardless of the relationship between journalists and game developers, journalists should give honest reviews to game developers. The other group, Gamergate opposers, argues that Gamergate supporters attack female game developers and also feminist critics, and that they are not concerned with ethics in journalism, but are using the opportunity to attack women in the gaming industry.
From Fig. 5 it is clear that, while there are hundreds of small communities represented by this visualization, they cluster into two major groups. In the middle, there are a handful of controversial people engaging both sides. And on the margins, some isolated people unrelated and disengaged.

(2) **2016 US Presidential Elections [4]:** This dataset contains 10,074 users who discuss the U.S. presidential election in 2016. This dataset consists of two major groups which indicate the political party of each user. This dataset comes from in which we only utilized the seed users from the whole dataset which brought our dataset size down from more than million users to 10,074 users since we required pure ground truth labels that were obtained away from the network structure and label propagation.

Fig. 5. Gamergate retweet network colored based on the network structure is shown on the left hand side, and the network colored by the ground truth labels is shown on the right hand side. The snap zooms into one of the areas, showing the disagreement between the two labeling approaches. (Image courtesy [1]). Purple nodes represent Gamergate opposers and green nodes represent Gamergate supporters.

2.2 Knowledge Graph-Based Methods

2.2.1 *Knowledge Graph:*

Knowledge Graphs (KGs) store human knowledge about the world in structured format, e.g., triples of facts or graphs of entities and relations, to be processed by AI systems.

A knowledge graph is a graph-based data model that describes real-world entities and relations between them. For example as shown in Fig. 6, "MONA LISA" and "LOUVRE" are entities and "is in" is the relation between them. An entity can also be linked to a data value, often referred to as literals, e.g., "Jan 1 1984".

Consider a simple example of Google search, like "taj mahal". Search has always essentially been about matching keywords to queries. But "taj mahal" has a much richer meaning. You might think of one of the world's most beautiful monuments, or a Grammy Award-winning musician, or possibly even a casino in Atlantic City, NJ or, depending on when you last ate, the nearest Indian restaurant (Fig. 7). The Knowledge Graph enables you to search for things, people or places that Google knows about—landmarks, celebrities, cities, sports teams, buildings, geographical features, movies, celestial objects, works of art and more and instantly get information that's relevant to your query. This is a critical first step towards building the next generation of search.

2.2.2 *Knowledge Graph in Recommendation Systems:*

Knowledge graphs are most commonly used in Recommendation Systems. Let us take the example of Netflix to

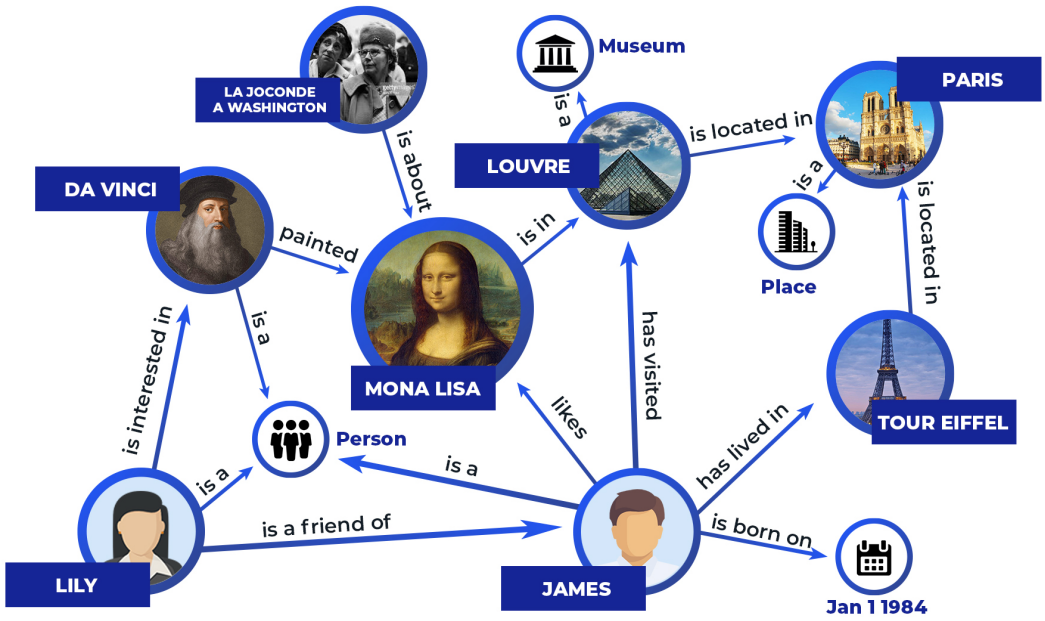

Fig. 6. An example of a knowledge graph (Image courtesy [18])

understand this better. When you visit Netflix, there are several lists of movies for you to watch including new releases and some "Top Picks for You". Netflix uses a powerful recommendation system to generate this list based on what you have watched and rated. It builds a profile of your tastes in terms of genres, plots, actors and more, and uses this profile to recommend movies that fit to your taste. Fig. 8 shows a sample knowledge graph built with the movie "Forrest Gump" as the source. Depending on the user profile, Netflix can recommend "The Terminal", as both have the same actor starring in it or it can recommend "Back to the Future" as both have the same director directing the films.

2.2.3 *Algorithmic Unfairness in Knowledge Graph-Based Methods:*

A knowledge graph is generally represented in triple form, where a triple consists of two entities and a relation, e.g. in Fig. 8, ("Cast Away". "country", "U.S."). The aim of graph embedding methods is to use these triples to learn a continuous vector representation of dimension d of all entities and relations. Knowledge graph embeddings (in their basic form, with no debiasing) are trained by optimizing the entity and relation embeddings to produce a high score for positive (true) triples, and a low score for randomly generated false triples.

To see how biases may be encoded into the embeddings of human entities, it is important to define a set of "sensitive attributes"; human characteristics which may be associated with unwanted stereotypes like, gender, ethnicity, religion and nationality. For each knowledge graph, there will be a set of relations which provide these attributes, which we term "sensitive relations".

When embeddings are trained with positive triples such as (person1, gender, male), the embedding of person1 will be updated with information related to the entity "male" in order to score this triple higher than negative triples, including (person1, gender, female). However, as gender information is now encoded in the embedding of person1, the model is also able to use this information when scoring other triples, such as (person1, profession, banker). So, in some cases

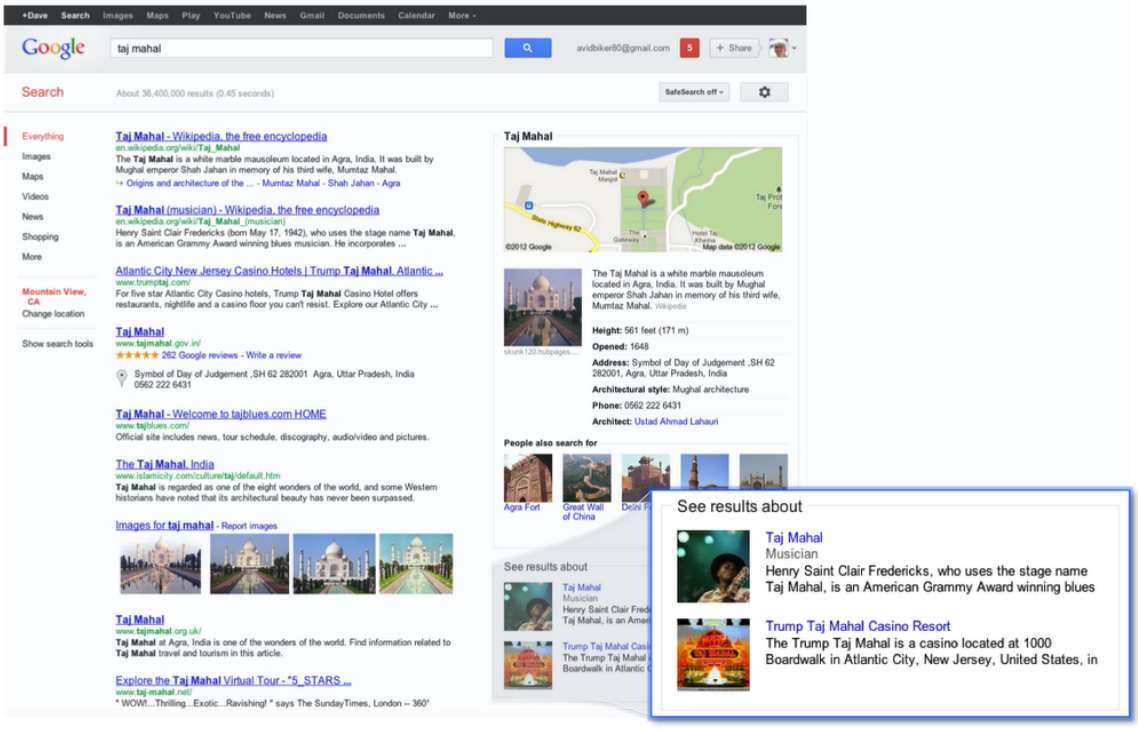

Fig. 7. A Google search of the words "taj mahal" (Image courtesy [19])

when there are many more male bankers than female bankers, the model learns to use the encoded gender information when predicting the likelihood a person is a banker, alongside other harmful stereotypes.

Thus, we need to train all human's embeddings to be neutral with respect to sensitive attributes. That is, we wish to make it impossible to predict, for example, a person's gender, from their embedding. As a result, predictions made using these embeddings (such as about profession) will also be independent of these attributes.

2.2.4 *Description of datasets considered:*

(1) **Freebase 15k - 237 [2]:** Freebase was a large collaborative knowledge base consisting of data composed mainly by its community members. It was an online collection of structured data harvested from many sources, including individual, user-submitted wiki contributions. Freebase aimed to create a global resource that allowed people (and machines) to access common information more effectively.

The Freebase 15k - 237 dataset contains knowledge base relation triples and textual mentions of Freebase entity pairs. It has a total of 592,213 triplets with 14,951 entities and 1,345 relationships. Freebase 15k - 237 is a variant of the original dataset where inverse relations are removed, since it was found that a large number of test triplets could be obtained by inverting triplets in the training set.

Taking the most common entity attribute labels, we used the 3 most common attribute labels as "sensitive" attributes. The goal in this dataset is to perform the standard knowledge base completion task, while having the

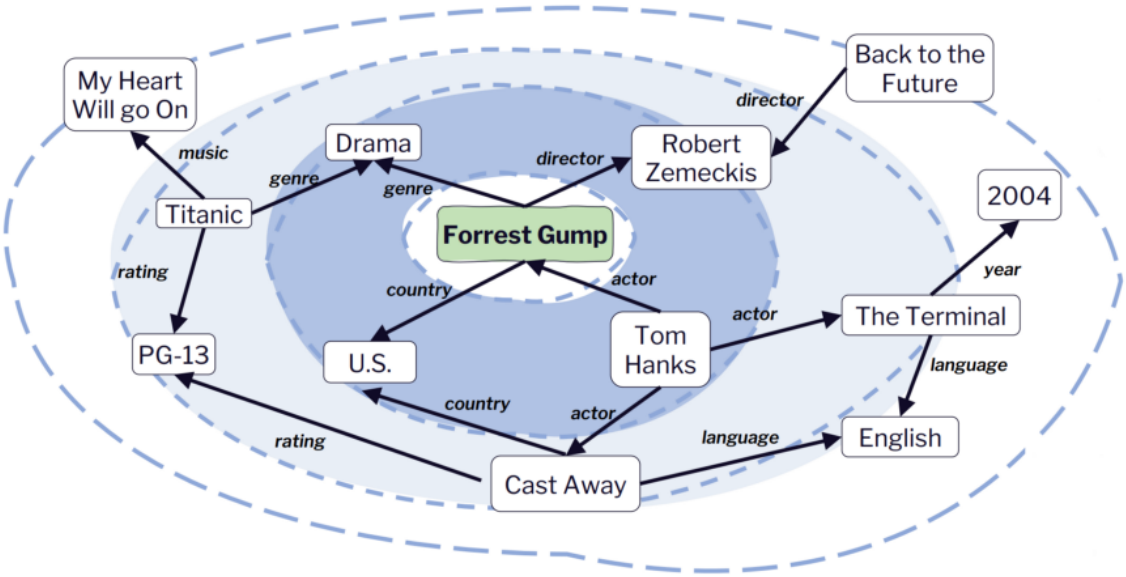

Fig. 8. The above is an example knowledge graph representing movies, the common actor, genres, country, directors, language and the complex interrelationships among them. In a knowledge graph, not only do we know what items are related to what properties, we know how they are related and impose no restrictions on what can be related and how. (Image courtesy [7])

entity embeddings be invariant with respect to these "sensitive" attribute labels. While synthetic, this dataset provides a useful reference point due to its popularity in the graph embedding literature.

(2) **MovieLens - 1M [3]:** MovieLens helps you find movies you will like. Rate movies to build a custom taste profile, then MovieLens recommends other movies for you to watch. This is a standard recommender system benchmark, where the goal is to predict the rating that users assign movies. We treat the user features (age, gender, and occupation) as sensitive attributes. This recommendation task is an edge prediction problem between users and movies, viewing the different possible ratings as different edge relations. It contains 1 million ratings from 6000 users on 4000 movies.

3 TECHNIQUES TO REMOVE BIAS FROM NETWORK-BASED ALGORITHMS

3.1 Mitigating Bias in Community Detection

It is clear from the previous section that ignoring low-degree users introduces bias in the analysis of data. In order to mitigate this, Mehrabi et al. [13] (introduced in Sec. 2.1.4) proposed an algorithm CLAN, which we discuss here in detail.

- It is an approach based on categorizing low-degree users into relevant communities instead of being ignored.
- CLAN classifies more users into relevant categories and outperforms existing state of-the-art community detection methods in terms of accuracy.

3.1.1 *CLAN Algorithm :*

Here is the pseudo-code of the algorithm:

```
Input:
```

```
1: Network
2: Threshold
Output: Communities
// Step 1: Finding the communities
C = find_communities(Network)
// Step 2: Classifying minority users into significant communities


for Ci in C do
    if count(Ci) > Threshold then
        Add Ci to training set
    else
        Add Ci to test set
    end if
    model = train(training set)
end for
predictions = model.inference(test set)
for p in predictions do
    if p.label == Ci then
        Add p.data to Ci
    end if
end for
return C
```

CLAN uses node attributes (such as text), other than network attributes (relationships between nodes) to assign lowly-connected nodes into appropriate communities and avoids creating smaller, insignificant communities. This helps avoiding bias arising from having low recall, as the aim is to not be able to only identify the popular users correctly but all the users regardless of their "popularity" in the network. CLAN can be understood as a two-step approach as is clear from the pseudo-code above. The first step is an unsupervised process to identify communities using network attributes and modularity values. Post this, using supervised classification, lowly-connected nodes are classified to the communities identified using additional node attributes (such as text). The first step could be any standard unsupervised community detection algorithm like [5], making the approach flexible.

3.1.2 ***Quantitative and Qualitative Comparison of Community Detection Algorithms:***

For comparison, F1 score and Jaccard similarity scores were calculated for each dataset with respect to ground truth labels for all methods considered. As is clear from Table. 1, CLAN outperforms the baseline methods for both of the datasets considered. Also, using CLAN there were no unlabeled nodes, while the other methods reported high number of nodes being dropped from consideration and not being classified into any community. The more unlabeled nodes a method has, the more susceptible to bias it is. Hence CLAN provides a two-fold advantage, it classifies even the lowly-connected nodes into communities, and this does not come at the cost of accuracy, rather it improves the performance scores of the algorithm.

	Gamergate Dataset		US Election dataset	
Method	F1 Score	Jaccard	F1 Score	Jaccard
CESNA	0.34	0.21	0.25	0.15
Modularity	0.43	0.28	0.75	0.60
CLAN	**0.48**	**0.32**	**0.79**	**0.65**

Table 1. The quantitative results obtained from calculating the F1 and Jaccard similarity scores with regards to the ground truth labels for each of the methods, as reported in the study by Mehrabi, et al. [13]

.

For qualitative results the study takes the example of a few sensitive tweets that would have gone undetected in the baseline algorithms but are detected by CLAN. For example for the Gamergate dataset, these tweets would have gone undetected by the baseline methods:

- Anti Gamergate: *"#gamergate is brutally put in its place by a journalist"*
- Pro Gamergate: *"#gamergate i support #gamergate and #notyourshield i stand against harassment threats and doxxing no matter who or why"*

Similarly for the US Presidential election dataset, the following tweet by a Democrat would have not been classified by the baseline methods : *"Trump wants to ban Muslim immigrants like my parents. I wrote a piece for telling him to go f*** himself."*.

The results reported in this section confirm the fact that the baseline methods have low agreement with the ground truth labels and suffer from bias towards low-degree and some users who are excluded from being labeled. Current methods could completely ignore entire communities if membership in that community is correlated with degree. For example, a community of introverts is likely to be overlooked because they do not form many links.

3.1.3 *Discussing Further Improvements:*

On completing our survey of various community detection algorithms and methods to reduce algorithmic bias in them, here we discuss a few improvements and suggestions that we came up with:

(1) To extend the current methods to be capable of hierarchical community detection method capable of detecting subgroups within larger communities, the importance of which is highlighted in a few recent works [11]. According to them, it is more interpretable and in some regimes more accurate to construct a hierarchical tree of communities instead of just partioning/clustering into non-overlapping groups.
(2) Develop algorithms that consider edge attributes and node attributes simultaneously. This will enable the methods to understand and identify much more diverse connections and relations between seemingly unrelated nodes.

3.2 Mitigating Bias in Knowledge Graph-Based Applications

The insight behind mitigating bias in the knowledge graph embeddings is that we learn a set of adversarial filters that remove information about particular sensitive attributes. One such approach is proposed in [6], and Fig. 9 described the model used to filter out the embeddings. In this approach, each of the learned filters can be optionally applied after training, so the model can flexibly generate embeddings that are invariant with respect to different combinations of sensitive attributes. For instance, in the context of social recommendations, the framework would allow one user to request that their recommendations are invariant to both their age and gender, while also allowing another user to request invariance to just their age.

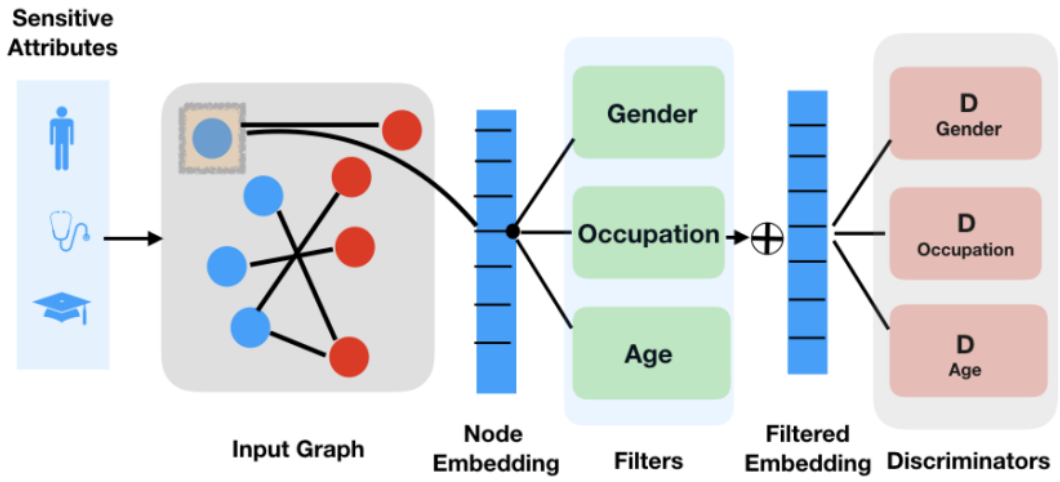

Fig. 9. The goal is to generate graph embeddings that are invariant to particular sensitive attributes (e.g., age or gender). We train a set of "filters" to prevent adversarial discriminators from classifying the sensitive information from the filtered embeddings. After training, these filters can be composed together in different combinations, allowing the flexible generation of embeddings that are invariant w.r.t. any subset of the sensitive attributes. (Image courtesy [6])

3.2.1 *Method:*

In the Fig. 9, the general case of embedding a heterogeneous or multi-relational (social) graph is considered, which is a set of directed edge triples consisting of starting node, ending node and a relation type or edge between the two. For example, the blue nodes can be considered users, red nodes are movies, and the edges represent movie ratings. Now, consider the task is to predict these edges or relations between the nodes, but we do not want age, gender and occupation to not bias the results. The edge prediction task can essentially be defined as learning a scoring function on edges which should ideally score any true edge higher than any negative edge.

Now, the next step is to encode the node to an embedding as shown in Fig. 9, where the blue node is mapped to a cell in the node embedding array. Generally, the intuition in embedding-based approaches is that the distance between two node embeddings should encode the likelihood that there is an edge between the nodes.

The invariance to sensitive attributes is enforced by introducing an adversarial loss and a technique to "filter" the embeddings generated by the encoding function. Note, that the set of sensitive attributes we want to be invariant with respect to — is not fixed across nodes; i.e., we may want to enforce invariance on different sets of sensitive attributes for different nodes. Now, to perform this filtering, there is a discriminator defined for every sensitive attribute as shown in Fig. 9. This attempts to predict the sensitive attribute from the node embeddings and essentially filters it out.

3.2.2 *Experiments:*

We have performed the experiments on the MovieLens - 1M dataset (proposed in the paper) and Freebase 15k - 237 dataset (new dataset we tried). Throughout these experiments, we compare against baseline that does not include any invariance constraints. The extensive experimental setup and analysis are available in a different report. Here, we state just the numeric results for the purpose of theoretical analysis.

Freebase 15K - 237	Baseline	With Invariance Constraints
Attribute 0	0.97	**0.80**
Attribute 1	0.99	**0.80**
Attribute 2	0.98	**0.82**

Table 2. AUC scores to calculate the effectiveness of predicting sensitive attributes in Freebase 15K - 237 dataset.

MovieLens - 1M	Baseline	With Invariance Constraints
Gender	0.71	**0.50**
Age	0.41	**0.33**
Occupation	0.14	**0.12**

Table 3. AUC score to calculate the effectiveness of predicting sensitive attribute like gender in MovieLens - 1M dataset and for age and occupation attributes the score is micro averaged F1.

AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. Thus, for binary sensitive attributes, an ideal result is an AUC score of 0.5 when attempting to predict the sensitive attributes from the learned embeddings.

The MovieLens - 1M dataset was able to achieve a reasonable tradeoff, with the near-complete removal of the sensitive information leading to a roughly 10% relative error increase on the edge prediction tasks from the baselines. In other words, on the dataset the sensitive attributes were nearly impossible to predict from the filtered embeddings, while the accuracy on the main edge prediction task was roughly 10% worse than a baseline approach that does not include the invariance constraints as is evident in Table 3. For, the age and occupation average F1 score is used as it is not a binary attribute. The closer the value of the F1 score to 0, the better is the filtering of the node embeddings.

On the Freebase15k - 237 dataset it was not possible to completely remove the sensitive information without incurring a significant decrease in accuracy on the original edge prediction task and thus the values in Table 2, reflect that as it is not close 0.5. This result is not entirely surprising, since for this dataset the "sensitive" attributes were synthetically constructed from entity type annotations, which are presumably very relevant to the main edge/relation prediction task.

4 SOCIAL SURVEY

In our study on network based algorithms, specifically knowledge graph based methods, we studied techniques that aimed at removing algorithmic bias. One such approach which we analyzed in detail was by Bose, et al.: Compositional Fairness Constraints for Graph Embeddings [6]. It discusses an algorithm for removing bias from recommendation systems by learning graph embeddings that are invariant to chosen sensitive attributes (could be age, gender, occupation etc.). This would allow users themselves decide what attributes/factors are to be used/removed while giving them recommendations, in a very flexible way. It makes the recommendation algorithm "personalized" as opposed to what has been a standard algorithm for everyone. An excerpt from the paper: "*For instance, in the context of social recommendations, our framework would allow one user to request that their recommendations are invariant to both their age and gender, while also allowing another user to request invariance to just their age.*"

A few interesting questions were raised in our discussions:

- Are users are informed enough to decide which attributes to choose, to get good recommendations ?
- Are people interested in having the option to decide which attributes influence their recommendations ?
- Are there differences of opinion between demographics about the same?

4.1 Survey Questions

To answer these question we conducted an online survey with 320 participants across various age groups and asked them the following questions:

- What is your Gender?
  - Female
  - Male
  - Non-Binary
  - Prefer not to say
- What is your age?
- Which of the following platforms do you use regularly?
  - Netflix
  - Amazon Prime
  - YouTube
  - Spotify
  - JioSaavn
- Do you know what information about you do the above platforms use in order to provide recommendations?
  - Yes
  - No
- Would you like to have an option to choose which attributes to use while giving you recommendations?
  The attributes could be your gender, region you live in, previously watched movie genres etc.
  - Yes
  - No

(If they answer yes for the previous question, one further question)

- Which of these attributes would you like to NOT influence your recommendations?
  - Gender
  - Age
  - Region you stay in
  - Previous usage history
  - People with similar watch preferences
  - The device you use
  - Time of the day you usually use the platform
  - Duration for which you use the platform

4.2 Results

There were quite a few interesting observations we made. The results have been analyzed in a separate report [1]. All data pertaining to the survey can be found in the above report.

Here we discuss a few interesting observations.

4.2.1 *Overall Statistics.*

- We surveyed around 320 people, with around 30% female respondents.
- 87% of people wanted to have the option to choose which attributes are taken into account while getting recommendations!
- 54% people considered themselves aware of the attributes that are used by popular platforms to give them recommendations.
- 50% of the people didn't want their gender to be considered.
- Around 65% people were okay with their age being used to get recommendations.

4.2.2 *Analyzing demographics : Age Based*

. We analyzed responses by separating responses of people based on age (people above and below the age of 40).

- 57% of people below 40 knew about what influences their recommendations compared to 47% of people above 40. The younger population is more aware of the factors.
- 48% of people below 40 didn't want their gender to be considered while only 68% people above 40 were okay with their gender being while getting recommendations.
- 75% of people below 40 wanted to watch what their peers were watching (people with similar watch preferences). People above 40 tended to be more individualistic and more than 40% did not want peers to influence their recommendations.
- People of all age groups were okay with their age being used to give them recommendations, with over 70% being okay with it.

4.2.3 *Analyzing demographics : Gender Based*

. We analyzed responses by separating responses of people based on gender.

- Women are more sensitive about their gender being used as compared to men. 39% of men did not want their gender to be considered compared to 50% of women.
- Over 70% of both men and women were okay with their age being used.
- Men were a bit more aware of attributes used by recommendation engines, around 60% responded they knew, compared to 49% of women.
- Almost all women, 93% wanted the option to choose the attributes being used to give recommendation, 85% men wanted to have the option.
- Both men and women wanted peer influence on their recommendations (>70% for both).

[1] Link to survey analysis

5 CONCLUSION

We present a survey of algorithmic unfairness in network based algorithms. We start by discussing why this survey is relevant in Section. 1, as this has been a hitherto less researched area. We study two main applications in this domain, community detection algorithms and knowledge graph-based methods. In Section. 2 we discuss how algorithmic bias can arise in the current popularly used methods. Further in Section. 3 we analyze the approaches that have been developed to reduce bias. We focus mainly on two approaches:

- A method to reduce bias in community detection, CLAN, where we also provide suggestions and scope for further improvement.
- A flexible knowledge graph based recommendation system for reducing bias, which allows users to choose attributes to be used for getting recommendations; for which we also provide code and analysis. We also test the algorithm on a new dataset and provide our observations on the results obtained.

For the second method described above, we conducted an online survey of around 320 participants, which we describe in Section. 4. We provide the code and demographic analysis done on the same. All relevant links can be found in the Appendix.

6 APPENDIX

Code implementation for "Compositional Fairness Constraints for Graph Embeddings" [6], described in Section. 3.2 :

- Github repository with source code
- Colab notebook for code demonstration

Links relevant to survey in Section. 4 :

- Response sheet with raw data
- Colab notebook for data analysis

REFERENCES

[1] [n.d.]. https://medium.com/message/72-hours-of-gamergate-e00513f7cf5d

[2] [n.d.]. https://www.microsoft.com/en-us/download/details.aspx?id=52312

[3] [n.d.]. https://grouplens.org/datasets/movielens/1m/

[4] Adam Badawy, Aseel Addawood, Kristina Lerman, and Emilio Ferrara. 2018. Characterizing the 2016 Russian IRA Influence Campaign. arXiv:1812.01997 [cs.SI]

[5] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (Oct 2008), P10008. https://doi.org/10.1088/1742-5468/2008/10/p10008

[6] Avishek Bose and William Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 715–724. http://proceedings.mlr.press/v97/bose19a.html

[7] Denis Gallo, Matteo Lissandrini, and Yannis Velegrakis. 2020. Personalized page rank on knowledge graphs: Particle Filtering is all you need!. In *Advances in Database Technology-EDBT 2020*, Vol. 2020. OpenProceedings.org, 447–450.

[8] Ullas Gargi, Wenjun Lu, Vahab Mirrokni, and Sangho Yoon. 2011. Large-Scale Community Detection on YouTube for Topic Discovery and Exploration. *Proceedings of the International AAAI Conference on Web and Social Media* 5, 1 (July 2011). https://ojs.aaai.org/index.php/ICWSM/article/view/14191

[9] M. Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (Jun 2002), 7821–7826. https://doi.org/10.1073/pnas.122653799

[10] Thamindu Dilshan Jayawickrama. 2021. Community Detection Algorithms. https://towardsdatascience.com/community-detection-algorithms-9bd8951e7dae

[11] Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Koen Van den Berge, Purnamrita Sarkar, Peter J. Bickel, and Elizaveta Levina. 2020. Hierarchical community detection by recursive partitioning. arXiv:1810.01509 [stat.ME]

[12] Fragkiskos D. Malliaros and Michalis Vazirgiannis. 2013. Clustering and community detection in directed networks: A survey. *Physics Reports* 533, 4 (Dec 2013), 95–142. https://doi.org/10.1016/j.physrep.2013.08.002

[13] Ninareh Mehrabi, Fred Morstatter, Nanyun Peng, and Aram Galstyan. 2019. Debiasing Community Detection: The Importance of Lowly Connected Nodes *(ASONAM '19)*. Association for Computing Machinery, New York, NY, USA, 509–512. https://doi.org/10.1145/3341161.3342915

[14] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A Survey on Bias and Fairness in Machine Learning. *arXiv e-prints*, Article arXiv:1908.09635 (Aug. 2019), arXiv:1908.09635 pages. arXiv:1908.09635 [cs.LG]

[15] Torill Elvira Mortensen. 2018. Anger, Fear, and Games: The Long Event of #GamerGate. *Games and Culture* 13, 8 (2018), 787–806. https://doi.org/10.1177/1555412016640408 arXiv:https://doi.org/10.1177/1555412016640408

[16] Symeon Papadopoulos, Ioannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. 2012. Community detection in Social Media. *Data Min. Knowl. Discov.* 24 (05 2012), 515–554. https://doi.org/10.1007/s10618-011-0224-z

[17] Yiye Ruan, David Fuhry, and Srinivasan Parthasarathy. 2012. Efficient Community Detection in Large Networks using Content and Links. arXiv:1212.0146 [cs.SI]

[18] Yashu Seth. [n.d.]. Introduction to Question Answering over Knowledge Graphs. https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/

[19] Amit Singhal. [n.d.]. Introducing the Knowledge Graph: things, not strings. https://blog.google/products/search/introducing-knowledge-graph-things-not/

[20] Lei Tang and Huan Liu. 2010. *Community Detection and Mining in Social Media*. Vol. 2. https://doi.org/10.2200/S00298ED1V01Y201009DMK003

[21] Jaewon Yang, Julian McAuley, and Jure Leskovec. 2013. Community Detection in Networks with Node Attributes. *2013 IEEE 13th International Conference on Data Mining* (Dec 2013). https://doi.org/10.1109/icdm.2013.167