Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Fair graph representation learning: Empowering NIFTY via Biased Edge Dropout and Fair Attribute Preprocessing

Danilo Franco [a], Vincenzo Stefano D'Amato [a], Luca Pasa [b], Nicolò Navarin [b,*], Luca Oneto [a,c]

[a] University of Genoa, Via Opera Pia 11a, 16145, Genova, Italy
[b] University of Padua, Via Trieste 63, 35121, Padova, Italy
[c] CINI, Via Ariosto 25, 00185, Roma, Italy

ARTICLE INFO

Keywords:
Graph neural network
Node classification
Inductive scenario
Transductive scenario
Fairness
Fair representation learning
Biased Edge Dropout
Fair Attribute Preprocessing

ABSTRACT

The increasing complexity and amount of data available in modern applications strongly demand Trustworthy Learning algorithms that can be fed directly with complex and large graphs data. In fact, on one hand, machine learning models must meet high technical standards (e.g., high accuracy with limited computational requirements), but, at the same time, they must be sure not to discriminate against subgroups of the population (e.g., based on gender or ethnicity). Graph Neural Networks (GNNs) are currently the most effective solution to meet the technical requirements, even if it has been demonstrated that they inherit and amplify the biases contained in the data as a reflection of societal inequities. In fact, when dealing with graph data, these biases can be hidden not only in the node attributes but also in the connections between entities. Several Fair GNNs have been proposed in the literature, with uNIfying Fairness and stabiliTY (NIFTY) (Agarwal et al., 2021) being one of the most effective. In this paper, we will empower NIFTY's fairness with two new strategies. The first one is a Biased Edge Dropout, namely, we drop graph edges to balance homophilous and heterophilous sensitive connections, mitigating the bias induced by subgroup node cardinality. The second one is Attributes Preprocessing, which is the process of learning a fair transformation of the original node attributes. The effectiveness of our proposal will be tested on a series of datasets with increasingly challenging scenarios. These scenarios will deal with different levels of knowledge about the entire graph, i.e., how many portions of the graph are known and which sub-portion is labelled at the training and forward phases.

1. Introduction

Nowadays, applications capable of learning from data have led to breakthroughs thanks to their practical ability to solve complex real-world problems impacting research and society at large. Modern learning algorithms are able to extract actionable information directly from complex structured data (such as graphs) and outperform humans in a wide range of decision-making scenarios, ranging from healthcare to education and cybersecurity [1–4]. These breakthroughs are mainly motivated by three reasons. The first is the increasing digitalization and datafication of all aspects of people's daily lives [5,6], with the consequent growth in data availability. The second one is the increasing availability of computing power and hardware accelerators that allowed the experimentation of models previously computationally impractical [7]. The last one is the discovery of new learning schemes that allow for more accurate and efficient learning models that can be applied directly to complex data [8–10]. In particular, these new methods relieved data scientists from the challenging and time-consuming problem of designing suitable vector-based data representations required by classical learning techniques [8,11].

Graphs, i.e., data structures that can represent patterns highlighting relationships between entities, are a direct and intuitive solution to describe complex information and, for this reason, are often used to naturally express the domain of interest in many modern applications, e.g., gene–protein interaction networks in bioinformatics, molecules in chemistry, social networks in social science, and many others [7,12,13]. The main problem of dealing with structured data is to compute a sound and meaningful representation for graph nodes that is invariant to the graph representation and also incorporates structural information relevant to the prediction task. There are two main types of tasks: (i) Predictions Over Graphs, where each example is composed of a whole graph and the learning tasks are predictions of graph properties; (ii) Predictions Over Nodes, where the dataset is composed of one or more

[*] Corresponding author.
E-mail addresses: danilo.franco@edu.unige.it (D. Franco), vincenzo.damato@unige.it (V.S. D'Amato), luca.pasa@unipd.it (L. Pasa), nnavarin@math.unipd.it (N. Navarin), luca.oneto@unige.it (L. Oneto).

https://doi.org/10.1016/j.neucom.2023.126948
Received 15 February 2023; Received in revised form 28 July 2023; Accepted 16 October 2023
Available online 20 October 2023

large graphs, and each example is a node of a graph. In this paper, we focus on task (ii). In particular, many node classification tasks are defined in the semi-supervised setting, where the graph consists of both labelled and unlabelled nodes. In a semi-supervised setting, it is also important to distinguish between the transductive approach, where the model can perform prediction only on unlabelled graph nodes that have already been observed at training time, and the inductive approach, where the model can perform prediction for previously unobserved nodes. The development of ML techniques able to directly process structured data has gained more and more attention since the first developments, such as Recursive Neural Networks [14,15] proposed in the second half of the '90s. In the 2000s, kernel methods for structured data [16] became the dominant approach to dealing with such kinds of data. In the last few years, following the success of deep neural networks in many application domains, there has been a burst of interest in developing deep learning models for graph domains characterized by remarkable technical achievements [17–25] empowering the applicability of such techniques to a broad variety of real-world problems.

Nevertheless, the achievements in the field of ML for structured data, or more naturally of ML in general, are accompanied by an increase in concerns where scholars try to explore the social and moral implications of this widespread adoption of Artificial Intelligence (AI) [26–28].

The definition of ethical concepts and behaviours dates back to earlier philosophers, but relating them to technological progress, especially to autonomous decision-making, is still an open challenge [29,30]. Early studies tried to divide the field of robot and AI ethics into mainly two branches [30]. The first one is questioning how robots and AI should be applied in order to minimize the ethical harms that can arise from poor design or misuse. In particular, it focuses on developing ethical principles and good practices for adopting new emerging technologies [31–33], such as is the case for ML. The second branch concerns how robots and AI can act ethically by learning moral behaviours. This second direction spans different fields, such as philosophy and engineering, where the research questions mix both moral requirements (such as the search for appropriate ethical values) and practical needs (such as the definition of new learning algorithms). This second research trend is often referred to as machine ethics [30]. Despite not being new to the general public thanks to scientific fiction of the calibre of Asimov's Laws of Robotics, which dates back to 1950, the research on machine ethics is still rather young. The earliest works, published less than 20 years ago, aimed at finding appropriate definitions for ethical governance, accountability, and agency, where, up to today, no general consensus has been reached yet [29,30]. In one of the earliest influential works, Moore [34] defines four categories of ethical agents, which distinguish themselves on the possibility of reasoning about ethics and the capability of avoiding unethical outcomes. Different posterior works spawned from this categorization and, in particular, some influential studies proposed some evaluation strategies for estimating the morality of autonomous systems, such as the Comparative Moral Turing Test [35] or Ethical Turing Test [36], where the machine decisions are assessed by ethicists or compared to the ones taken by humans actors.

Breaking from the fairly theoretical definitions of early publications, the latest works found in the literature of machine ethics observed how technical metrics like accuracy and computational requirements are not enough to characterize well a learning machine [37–40]. In fact, the data exploited for training these systems are naturally biased since they reflect past and present societal inequities that may even be amplified by the learning machines themselves [41]. As a result, these algorithms can strongly influence people's lives, and, eventually, the societal and ethical issues related to their use cannot be ignored and need to be explicitly addressed [42–46]. Therefore, the design of trustworthy learning algorithms that we, as humans, can trust is nowadays becoming essential [11]. Specifically, they must consider human-related metrics like fairness, robustness, privacy, and explainability [37].

In this paper, we will focus on building fair models, namely, models able to not discriminate against subgroups of the population (e.g., based on gender or ethnicity) [47]. In this context, a learning system needs to optimize possibly raw but measurable metrics of Fairness [47–49] along with the technical ones. These roughly fall under two main families [47–51]: statistical/group and individual. Group fairness aims at treating different subgroups in the population equally, while individual fairness aims at giving similar predictions to similar individuals. Among the group fairness definitions, Demographic Parity [52] and Equal Opportunity [53] are probably the most exploited in a supervised learning scenario. Counterfactual Fairness [54] is instead the most known and leveraged individual fairness definition. Once the fairness definition has been set, the problem of mitigating the bias can be tackled through three main approaches: pre-, in-, and post-processing techniques
[47,55]. The first approach tries to mitigate the bias present in the data by directly modifying them (thus, learning a fair representation), avoiding the need for changes in the training algorithms or for adjustments in the model predictions. The second approach injects fairness constraints into the training algorithms in order to select models that optimize both technical and fairness metrics. Finally, the last one is able to make a previously trained complex model fairer by mitigating the bias by acting directly on the predictions of such a model.

However, the standard bias mitigation techniques listed above are hardly applicable to learning from structured, and specifically graph, data [56]. Graph data entities and their relations are non-i.i.d., calling for the challenge of defining fairness mitigation strategies able to encompass this problem (e.g., mitigating discrimination propagation across nodes and edges of a graph) [57]. Nonetheless, the problem of learning from graphs is typically addressed by extracting a hidden representation, formally known as embedding, from which many fairness metrics can be easily evaluated. Deep Graph Networks (DGNs) proved to be a very effective solution in this context by iteratively propagating information across the graphs such that each node aggregates incoming messages from its neighbours and updates its representation [11,58,59]. Among DGN models, Graph Convolutional Networks (GCNs) gained popularity due to their simplicity, usability, and effectiveness [60]. In GCNs, nodes exploit the contextual information of their neighbourhood computed in the previous layers by using shared (hence convolutional) filters. Intuitively, stacking more layers allows a node to explore larger portions of the surrounding graph [61]. On the one hand, the embeddings extracted by GCNs are universal, in the sense that they can be leveraged by classical linear or non-linear shallow models, while, on the other, standard fairness techniques can help us extract less discriminatory graph representations [62–64]. One naive way of building fair GCNs is to employ post-processing techniques that, by definition, can be applied to any learned model [53,65]. Another naive way is to exploit in-processing techniques [66] and techniques that learn a fair representation [62–64,67,68] since, in these cases, the idea is simply to constrain the optimization objective by imposing fair models and/or representations. Classical pre-processing techniques instead cannot be naively applied to graphs: in fact, being able to modify (i.e., pre-process) the topology and the node attributes of the original graph without impacting accuracy is a challenging task [69,70]. Indeed, within the context of learning from graphs, Fairness mitigation strategies can be furtherly grouped into two families [69,71]. The first family tries to preprocess directly and debias the original graphs by balancing known graph properties between protected and unprotected groups and, thus, is task agnostic. Several works [72–74] proposed to promote group fairness by rebalancing specific paths in the input network. Specifically, they aim at balancing the nodes' appearance rates across different population subgroups in random walks. Similarly, other works achieved fair rebalancing through node sampling [68,75] or generation [76], where deep models, such as GCNs, are trained either on sub-portions or on augmented versions of the original graph with balanced populations. Another line of work acts directly on rewiring the networks' edges, thus, mitigating a possible biased topology. Following this direction, FairDrop [77] directly samples the edges for increasing communities heterophily concerning the sensitive attributes, while other works [78–80] leverage on concepts from information theory for optimizing the adjacency matrix (which depicts all the graph connections) and mitigating unfair information flows. The second family of fairness mitigation strategies in learning from graphs extracts fair and task-specific embeddings by acting directly on the learning procedure. The majority of works in this family define a regularized/constrained optimization problem that aims at improving group [64,67,70,81–83] or individual [84–86] fairness definitions. In particular, NIFTY [85] enforces a regularized loss function where any difference between the observed node embeddings and the counterfactual ones is penalized. Posing a similar problem, numerous other proposals [87–91] stem from the field of Generative Adversarial Networks [92], where the objective is posed as a contrasting optimization between a generator, which extracts graph embeddings decoupled from the sensitive information, and a discriminator, which tries to predict the sensitive attributes based on the generator output. In this second family of fairness mitigation strategies, a final line of works aims at projecting the graph embeddings into hyperplanes that are directly orthogonal to the direction of sensitive attributes [67,93], thus effectively removing any linear dependency from the extracted representations to the sensitive features.

In this study, we will build upon the work of NIFTY [85] for graph node labelling empowering it by introducing two strategies to improve the fairness of the solution while minimally impacting the final accuracy of the model. In particular, we consider the GCN exploited in NIFTY, which measures both technical (accuracy and Area Under the Receiver Operating Characteristic) and fairness metrics. We will leverage Demographic Parity, Equal Opportunity, and Counterfactual Fairness for what concerns the latter. The two strategies proposed to improve these metrics in NIFTY focus on both perturbing node attributes and modifying the graph topology. More specifically, the former is Fair Attributes Preprocessing, namely, we learn a fair transformation of the original node attributes inspired by the work of Zemel et al. [62], while the latter is Biased Edges Dropout, namely, we drop graph edges to balance homophilous and heterophilous sensitive connections mitigating the bias induced by subgroups node cardinality, inspired by the work of Spinelli et al. [77]. The effectiveness of our proposal will be tested on a series of datasets with increasingly challenging scenarios. These scenarios will deal with different levels of knowledge about the entire graph, i.e., how many portions of the graph are known and what sub-portion is labelled at the training and forward phases. In particular, in the first scenario, we assume the knowledge of a subgraph of an unknown graph where just some nodes are labelled at the training phase. Eventually, we would like to estimate the labels of the unlabelled nodes during the test phase. In the second scenario, we assume that, during the training phase, all the nodes of the subgraph are labelled and, during the test phase, a new single node whose label needs to be estimated is added to the graph.[1] In the third and last scenario, all the nodes of the sub-graph are again labelled at the training phase, but during the test phase, a new sub-graph coming from the same unknown graph needs to be fully labelled. Results on six different datasets, i.e., German [94], Credit [95], Bail [96], Pokec [97], Facebook [98], and Google Plus [98], will support our proposal with empirical evidences.

The rest of the paper is organized as follows. Section 2 reports some preliminaries necessary to understand our work and recalls the work of NIFTY [85]. Section 3 describes our proposal in detail, along with the two new fair mitigation strategies to include in NIFTY. Section 4 presents the results when applying our proposal to multiple datasets in a series of increasingly challenging scenarios. Finally, Section 5 concludes the paper.

2. Preliminaries

Let us consider a graph $\mathcal{G} = (\mathcal{V}_{\mathcal{D}} \cup \mathcal{V}_{\mathcal{U}} \cup \mathcal{V}_{\mathcal{T}}, \mathcal{E}_{\mathcal{G}}, X_{\mathcal{G}}, \boldsymbol{s}_{\mathcal{G}}, \mathcal{L}_{\mathcal{D}})$, where $\mathcal{V}_{\mathcal{D}} = \{v_1, \dots, v_n\}$ denotes the set vertices (or nodes) of the graph for which the label is known via $\mathcal{L}_{\mathcal{D}} = \{(v_1, y_{v_1}), \dots, (v_n, y_{v_n})\}$, $\mathcal{V}_{\mathcal{U}} = \{v_1, \dots, v_m\}$ the set of unlabelled nodes of $\mathcal{G}$ and $\mathcal{V}_{\mathcal{T}} = \{v_1, \dots, v_t\}$ the set of test nodes, i.e. nodes for which we are interested in computing a prediction, $\mathcal{E}_{\mathcal{G}} \subseteq \mathcal{V}_{\mathcal{G}} \times \mathcal{V}_{\mathcal{G}}$ is the set of edges, $X_{\mathcal{G}} \in \mathbb{R}^{g \times d}$ is the matrix of the d non-sensitive node attributes associated to the g nodes, and $\boldsymbol{s}_{\mathcal{G}} \in \{0,1\}^n$ is the vector of the binary-valued sensitive attributes associated to the n nodes.[2] Let $\mathcal{G}^{train}$ be the subgraph of $\mathcal{G}$ that insists only on the nodes considered during training, i.e., $\mathcal{G}_{train} = (\mathcal{V}_{\mathcal{D}} \cup \mathcal{V}_{\mathcal{U}}, \mathcal{E}_{D \cup \mathcal{U}}, X_{D \cup \mathcal{U}}, \boldsymbol{s}_{D \cup \mathcal{U}}, \mathcal{L}_{D})$, where $\mathcal{E}_{D \cup \mathcal{U}}$ is the set of edges insisting on a node in $\mathcal{V}_{\mathcal{D}}$ or $\mathcal{V}_{\mathcal{U}}$ while $X_{D \cup \mathcal{U}}$ and $\boldsymbol{s}_{D \cup \mathcal{U}}$ contain all the attributes and sensitive attributes of the considered nodes, respectively. In this work, we deal with the problem of binary classification of the nodes in a (possibly disconnected) large graph. Our goal is to learn a model f, based on $\mathcal{V}_{\mathcal{D}}$ and $\mathcal{V}_{\mathcal{U}}$, able to classify a node v in $\mathcal{V}_{\mathcal{T}}$, that can be either $\mathcal{V}_{\mathcal{T}} = \mathcal{V}_{\mathcal{U}}$, $\mathcal{V}_{\mathcal{T}} \subset \mathcal{V}_{\mathcal{U}}$ or a completely disjoint set of nodes from $\mathcal{V}_{\mathcal{U}}$. The model's predictions will be based on the node's features and topology, i.e., $\hat{y}_v = f(v, \mathcal{G})$. We define $A \in \mathbb{R}^{(n+m) \times (n+m)}$ as the adjacency matrix of $\mathcal{G}^{train}$ with elements $a_{ij} = 1$ if $(v_i, v_j) \in \mathcal{E}_{D \cup U}$ and $a_{ij} = 0$ otherwise. Note that for undirected graphs (i.e., graphs where all the edges are bidirectional), A is symmetric. We also consider $\dot{A} = A + I$ where I the identity matrix and $\bar{A} \in \mathbb{R}^{(n+m) \times (n+m)}$ as the diagonal degree matrix, where $\bar{a}_{i,i} = \sum_{j=1}^{n+m} \dot{a}_{i,j}$. The accuracy of this classification process will be measured according to two different metrics: percentage of accuracy (ACC) and Area Under the Receiver Operating Characteristic curve (AUROC) [99]. Moreover, we want this classification process to be fair [47]. In this work, as fairness metrics, we opt for the Difference of Demographic (or Statistical) Parity (DDP) [64] and the Difference of Equal Opportunity (DEO) on both positively (DEO^+) and negatively (DEO^-) labelled nodes [100], and the Counterfactual Fairness (CF) [54].

As we already mentioned in Section 1, in order to build our model, we will leverage the graph representation induced by GCNs [15,18,19,61,101–107]. The GCN exploited in our setting is the one proposed by Kipf et al. [61], whose embeddings are defined as $H = \bar{A}^{-\frac{1}{2}} \dot{A} \bar{A}^{-\frac{1}{2}} Z \Theta^h \in \mathbb{R}^{n \times r}$ where $Z \in \mathbb{R}^{n \times d}$ are the inputs and $\Theta^h \in \mathbb{R}^{d \times r}$ are its learnable parameters. Note that Z could possibly include the sensitive attribute among its features, namely $Z = [X, (\mathbf{s})]$, depending on the local legislation [108]. We will build upon the work of NIFTY [85] for graph node labelling. NIFTY aims to learn a fair and stable node representation leveraging on a particular network structure, the Siamese framework [109], which maximizes the similarity between the representations learned with the initially labelled nodes of $\mathcal{D}$ and the one learned on a perturbed version of the same nodes. The perturbation is designed to enforce fairness and stability by slightly modifying the nodes according to the following strategies:

- $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \boldsymbol{b} \circ N(0, p_\sigma)$, where $\circ$ is the Hadamard (element-wise) product, $\boldsymbol{b} \in \{0,1\}^d$ is a random masking vector drawn from a Bernoulli distribution of parameter p_b, and $N(0, p_\sigma) \in \mathbb{R}^d$ is sampled from a Gaussian distribution of parameter p_σ, namely the non-sensitive features can be perturbed with a small Gaussian noise. This perturbation should enforce stability;
- dropping a connection between two nodes (v_i, v_j) randomly chosen with probability p_d, i.e., we create another adjacency matrix $\tilde{A}$ that is similar to $\tilde{A}$ but some ones are randomly set to 0 with probability p_d. This perturbation should enforce stability;
- flipping the sensitive attributes, i.e., $s_i = \neg s_i \; \forall i \in \{1, \dots, n+m\}$, inspired by the principle behind Counterfactual Fairness [54], namely the representation of the node should not change if the

[1] The slightly more general case of adding more than one node is not explored since it falls in between the second and the third scenarios, depending on the edges connecting the new nodes with the ones available during training.

[2] The extension to multiple subgroups is outside the scope of this work.

value of its sensitive attribute changes. This perturbation should enforce fairness. We are aware that some legislations do not allow the explicit use of possible sensitive features [110] or that in some datasets the sensitive feature may not be available. In this work, however, we built upon NIFTY which requires that the sensitive attribute is present among the features fed to the network both at training and test time. Otherwise, this augmented version of the training dataset could not be computed.

Then, the Siamese network learns to minimize the discrepancy between the representations of the original nodes in $\mathcal{D}$ and the ones of a perturbed version of the same nodes according to the following criteria. Let $\boldsymbol{h}_i \in \mathbb{R}^r$ be the node representation extracted from the original node v_i with $i \in \{1, \dots, n\}$ and $\tilde{\boldsymbol{h}}_i \in \mathbb{R}^r$ be the one extracted from the perturbed version of the same node. Let $t : \mathbb{R}^r \to \mathbb{R}^r$ be a predictor function that maps the representation of the original data in the one of the corresponding perturbed version and vice versa, i.e., $t(\boldsymbol{h}_i) = \boldsymbol{h}_i \Theta^t$ and $t(\tilde{\boldsymbol{h}}_i) = \tilde{\boldsymbol{h}}_i \Theta^t$ by means of $\Theta^t \in \mathbb{R}^{r \times r}$ being trainable parameters. Finally, let $\phi : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}$ be the Cosine Distance and let ω the stop-grad operator, which treats the representation of nodes as constant. Then NIFTY learns a representation able to minimize the following loss:

$$L_s = \frac{1}{2(n+m)} \sum_{i=1}^{n+m} \left[\phi(t(\boldsymbol{h}_i), \omega(\tilde{\boldsymbol{h}}_i)) + \phi(t(\tilde{\boldsymbol{h}}_i), \omega(\boldsymbol{h}_i)) \right]. \tag{1}$$

We redirect any interested reader to the original proposal [85] for the theoretical analysis. Apart from learning a representation, in this work, we also want the classifier to be able to learn how to label the nodes of the graph, i.e., $f(v_i, \mathcal{G}^{train}) = \boldsymbol{h}_i \theta^f$ where $\theta^f \in \mathbb{R}^r$ are learnable parameters. For this reason, NIFTY adds to the loss the Binary Cross-Entropy between real and predicted target L_c, obtaining the following total loss

$$(1 - \lambda) L_c + \lambda L_s, \tag{2}$$

where λ is the regularization coefficient that controls the trade-off between L_c (accuracy) and L_s (stability and fairness). NIFTY trains $\Theta^h, \Theta^t, \theta^f$, namely all the learnable parameters, using the ADAM optimizer [111] with a learning rate p_l, weight decay p_w, and p_e epochs.

3. Our proposal

In this work, we empower NIFTY, according to the fairness metrics (DDP, DEO$^+$, and DEO$^-$), using a twofold strategy, i.e., Fair Attributes Preprocessing (Section 3.1) and Biased Edge Dropout (Section 3.2), while minimally impacting the final performance, according to the accuracy metrics (ACC and AUROC).

3.1. Fair attributes preprocessing

The first strategy focuses on the concept of Learning Fair Representation introduced by Zemel et al. [62]. The main idea of the approach is to learn a fair representation rather than simply learning a fair model [62,112,113]. In this work, we will leverage this idea to improve further the NIFTY fairness by preprocessing the node attribute into their unbiased version. In other words, we will try to learn first a way to preprocess the same node attributes into a slightly modified version which is, from one side, as close as possible to the original one and, from the other side, as fair as possible. For this purpose a single fully connected hidden layered autoencoder [112,114–116] is adopted such that

$$[\dot{\boldsymbol{z}}_1, \dots, \dot{\boldsymbol{z}}_{n+m}]^T = \mathrm{ReLU}([\boldsymbol{z}_1, \dots, \boldsymbol{z}_{n+m}]^T W_e) W_d, \tag{3}$$

where $W_e \in \mathbb{R}^{d \times r_h}$ and $W_d \in \mathbb{R}^{r_h \times d}$ are the encoder and decoder weights. These weights are trained using ADAM [111] with learning rate p_l minimizing the following loss function

$$\lambda L_r + (1 - \lambda) L_f, \tag{4}$$

where λ is the regularization coefficient that controls the trade-off between the accuracy of the reconstruction measured with the Means Square Error (usually called MSE) [117] (L_r) and the fairness of the hidden representation ($\mathrm{ReLU}([\boldsymbol{z}_1, \dots, \boldsymbol{z}_{n+m}]^T W_e)$) measured according to the metric of fairness that we want to minimize (DDP, DEO$^+$, and DEO$^-$ of the representation) adopting the approach proposed by [64,113]. In particular, for DDP, we have that L_f can be expressed as

$$\left\| \hat{\mathbb{E}}_{v \in \mathcal{V}_{\mathcal{D}} \cup \mathcal{V}_{\mathcal{U}} | s_v = 0} \mathrm{ReLU}(\boldsymbol{z}_v^T W_e) - \hat{\mathbb{E}}_{u \in \mathcal{V}_{\mathcal{D}} \cup \mathcal{V}_{\mathcal{U}} | s_u = 1} \mathrm{ReLU}(\boldsymbol{z}_u^T W_e) \right\|^2, \tag{5}$$

namely, the average representation for a node with $s = 0$ (e.g., male) must be similar to the one with $s = 1$ (e.g., female), which is the first order convex and differentiable approximation of the DDP [113]. Note that $\hat{\mathbb{E}}$ represents the empirical average computed over the subset of nodes indicated by the subscript. For what concerns the DEO$^\diamond$ with $\diamond \in \{\pm\}$ we have that L_f can be expressed as

$$\left\| \hat{\mathbb{E}}_{v \in \mathcal{V}_{\mathcal{D}} | s_v = 0, y_v = \diamond 1} \mathrm{ReLU}(\boldsymbol{z}_v^T W_e) - \hat{\mathbb{E}}_{u \in \mathcal{V}_{\mathcal{D}} | s_u = 1, y_u = \diamond 1} \mathrm{ReLU}(\boldsymbol{z}_u^T W_e) \right\|^2, \tag{6}$$

namely, similarly to the DDP case, the average representation for a node with $s = 0$ (e.g., male) labelled with $y = \diamond 1$ must be similar to the one with $s = 1$ (e.g., female) labelled with the same $y = \diamond 1$, which is the first order convex and differentiable approximation of the DEO$^\diamond$ [113]. Note that the DDP L_f exploits all the data, even the unlabelled ones, while the DEO$^\diamond$ L_f exploits only the labelled data.

In order to reduce the impact on the computational requirements of this addition in NIFTY, we do not attach this autoencoder directly to the end-to-end training of all parameters (Θ^h, Θ^t, θ^f, W_e, and W_d). We train them separately by minimizing the losses defined in Eqs. (2) and (4).

3.2. Biased edge dropout

The second strategy focuses on slightly, but substantially, modifying the NIFTY edge dropout strategy recalled in Section 2. Specifically, the strategy of NIFTY does not take into account the sensitive attribute of the nodes involved in the dropped edge since dropout, for NIFTY, was a perturbation strategy focused on enforcing stability. In this work, we argue that dropout can also be used to enforce fairness. For this purpose, we try to mitigate the discrimination resulting from a biased topology when we drop the graph edges. More formally, we designed a dropout strategy which enforces the resulting perturbed graph to have a desired homophily rate (i.e., edges that connect nodes with the same values of the sensitive attributes). This idea takes inspiration from several studies [77–80] which show that accurately rewiring the graph edges can effectively lead to trained models with better fairness properties. In particular, we implemented a variation of the FairDrop algorithm [77]. FairDrop concentrates on mitigating the bias introduced by the principle of homophily, which describes the situation where individuals are more likely to interact with their equals. In social networks, for example, this principle translates into the fact that individuals with similar sensitive characteristics (e.g., ethnicity, gender, sexual or political orientation) are more likely to be connected, thus introducing unequal influences and generating groups segregations issues (i.e., "filter bubbles") [118–120]. In our work, we simplify the formulation of FairDrop and introduce a parallel algorithm called Biased Edges Dropout, which uses a new hyperparameter, the Homophilous Rate ρ, that specifies the maximum ratio of allowed homophilous connections.

Let us present in detail our Biased Edges Dropout strategy. Given the adjacency matrix A and the perturbed one $\tilde{A}$ according to the NIFTY strategy (see Section 2), we define

$$\mathcal{E}^{\tilde{A}} = \{(v_i, v_j) : (v_i, v_j) \in \mathcal{E}_{\mathcal{D} \cup \mathcal{U}}, \tilde{a}_{i,j} = a_{i,j} = 1\}, \tag{7}$$

as the set of edges not dropped by NIFTY and

$$\mathcal{E}^{\neg \tilde{A}} = \{(v_i, v_j) : (v_i, v_j) \in \mathcal{E}_{\mathcal{D} \cup \mathcal{U}}, \tilde{a}_{i,j} = 0, a_{i,j} = 1\}, \tag{8}$$

as the set of dropped edges. Based on $\mathcal{E}^{\tilde{A}}$, we compute the percentage of homophilous edges. If this percentage is greater than ρ, then the algorithm starts replacing homophilous connections in $\mathcal{E}^{\tilde{A}}$ with heterophilous ones in $\mathcal{E}^{\neg\tilde{A}}$ until the desired ρ is reached. Of course, reaching ρ is not always possible if the original sub-graph does not have enough heterophilous connections. In this case, we reach the closest possible value to ρ.

3.3. Considered classification scenarios

To test the effectiveness of our proposal, we will consider a series of increasingly challenging scenarios. These scenarios will deal with different levels of knowledge about the entire graph, i.e., how many portions are known and what sub-portion is labelled at the training and forward phases. In particular:

Scenario (1) in the forward phase, we want to classify the unlabelled nodes of the training graph, i.e., $\mathcal{V}_{\mathcal{T}} = \mathcal{V}_{\mathcal{U}}$. This scenario represents the transductive setting.

Scenario (2) when $\mathcal{V}_{\mathcal{U}} = \emptyset$, we classify a single node by adding it to $\mathcal{G}$, i.e., $\mathcal{V}_{\mathcal{T}} \cap \mathcal{V}_{\mathcal{D}} = \emptyset$, $|\mathcal{V}_{\mathcal{T}}| = t = 1$, and all the edges between the node in $\mathcal{V}_{\mathcal{T}}$ and nodes in $\mathcal{V}_{\mathcal{D}}$ are allowed. This scenario represents the semi-inductive setting.

Scenario (3) when $\mathcal{V}_{\mathcal{U}} = \emptyset$, we classify the nodes in an entirely unseen subgraph of $\mathcal{G}$, i.e., $\mathcal{V}_{\mathcal{T}} \cap \mathcal{V}_{\mathcal{D}} = \emptyset$ and no edges between nodes in $\mathcal{V}_{\mathcal{T}}$ and nodes in $\mathcal{V}_{\mathcal{D}}$ are allowed. This scenario represents the more challenging inductive setting.

The scenario obviously impacts how the hyperparameter selection and performance assessment strategies are performed. In particular, it impacts how the training, validation, and test sets are constructed. The splitting is repeated multiple times to ensure statistically meaningful results.

In the results, we reported the average ACC and the AUROC together with the average DDP, DEO$^{\diamond}$ with $\diamond \in \{\pm\}$, and CF on the test sets. ACC and AUROC are the classical metrics, while DDP is computed as follows

$$\left| \hat{\mathbb{E}}_{\substack{\text{Nodes in } \mathcal{V}_{\mathcal{T}} \\ \text{with } s=0}} [\text{Predicted Label} = 1] - \hat{\mathbb{E}}_{\substack{\text{Nodes in } \mathcal{V}_{\mathcal{T}} \\ \text{with } s=1}} [\text{Predicted Label} = 1] \right|, \tag{9}$$

DEO$^{\diamond}$ with $\diamond \in \{\pm 1\}$ are computed as follows

$$\left| \hat{\mathbb{E}}_{\substack{\text{Nodes in } \mathcal{V}_{\mathcal{T}} \\ \text{with } s=0 \text{ and} \\ y=\diamond 1}} [\text{Predicted Label} = \diamond 1] - \hat{\mathbb{E}}_{\substack{\text{Nodes in } \mathcal{V}_{\mathcal{T}} \\ \text{with } s=1 \text{ and} \\ y=\diamond 1}} [\text{Predicted Label} = \diamond 1] \right|, \tag{10}$$

and, finally, CF is computed as follows

$$\hat{\mathbb{E}}_{\substack{\text{Nodes in} \\ \mathcal{V}_{\mathcal{T}}.}} \left| [\text{Node Label} = \text{Predicted Label}] - \left[\text{Node Label} = \begin{matrix} \text{Predicted Label} \\ \text{setting } s = \neg s \end{matrix} \right] \right|, \tag{11}$$

where the Iverson bracket notation has been used.

4. Experimental results

In this section, we compare original NIFTY [85] with our proposal presented in Section 3 on a series of datasets, i.e., German [94], Credit [95], Bail [96], Pokec [97], Facebook [98], and Google Plus [98]. More specifically, in Section 4.1, we will describe the datasets in detail; in Section 4.2, we will report the experimental settings, and finally, Sections 4.3 and 4.4 report the actual results and their discussion respectively.

4.1. Datasets

German [94] and Credit [95] datasets evaluate the risk of bankruptcy of 1.000 and 30.000 samples analysing 29 and 14 financial factors respectively. For these two datasets, we decided to use gender (for German) and age (for Credit, binarized as ≤25 or >25 years of age [85]).

The Bail [96] dataset presents the risk of recidivism of ≈19.000 samples evaluated across 18 attributes, where skin colour and/or sex can be exploited as sensitive attributes. In our experiments, we use the one identifying the colour of the skin.

Since these datasets are originally tabular and not graph datasets, we extracted a topology from the samples' similarities. Following [85], we defined the similarity between two nodes using the Minkowski distance between their attributes. In our graphs, we connect samples belonging to the 80th percentile of the maximum similarity values for the German dataset. We used the 70th percentile for Credit and 60th percentile for Bail datasets.

Finally, in order to include real-world datasets that were born natively as graphs, we exploited the Pokec [97], Facebook [98], and Google Plus [98] datasets.

Pokec represents the most popular social network in Slovakia, and the dataset contains personal information for about 65.000 users. The 276 features are related to both physical (e.g. age, eye and hair colour) and psychological (e.g., hobbies, musical taste, political orientation) attributes. As our task for this dataset, we evaluated the performances in predicting the binary marital status (i.e., engaged or not-engaged) while considering the binary region of provenience (i.e. foreign or native) as the sensitive attribute.

Facebook and Google Plus datasets are two real-world graph datasets broadly used in the fair graph learning literature [121,122]. They consist of 1.045 and 3.601 users, respectively, extracted from the two world-known social networks. The users are described by 574 and 2.533 social features related to occupation, education and other personal information. In general, these two datasets are exploited for evaluating the performances of node-clustering algorithms since the graphs are organized in clusters known as ego-circles [98]. As our downstream task, we want to predict whether a node belongs to one or more ego-circles while imposing fairness with respect to the gender of the observed users.

Fig. 1 depicts the class and sensitive attributes distributions of the considered datasets along with the edges homophily.

4.2. Experimental settings

In order to assess the performance of our proposal, we consider a mixture of technical, i.e. ACC and AUROC (upper is better ↑), and fairness metrics, i.e. DDP, DEO^{+}, DEO^{-}, and CF (lower is better ↓).

In our experimental pipeline, we created 30 different partitions for training, validation, and test sets according to the different scenarios depicted in Section 3.3.

We rely on the grid search hyperparameter selection strategy on the validation splits, optimizing for $r_h \in \{d, d/2, d/4\}$ for Θ^h, $p_l \in \{0.001, 0.0001\}$, $p_w \in \{0.00001, 0.000001\}$, and $p_e = 100$ for Θ^f, $p_d \in \{0.2, 0.5, 0.8\}$, $p_\sigma \in \{0.2, 0.5, 0.8\}$, $\lambda \in \{0.2, 0.5, 0.8\}$, and $\rho \in \{0.2, 0.5, 0.8, 1\}$. Note that the original NIFTY formulation can be recovered from our method by removing Biased Edge Dropout ($\rho = 1$) and Fair Attribute Preprocessing.

Since we search for optimal models according to different metrics, the selection of the optimal hyperparameters is not trivial. For this

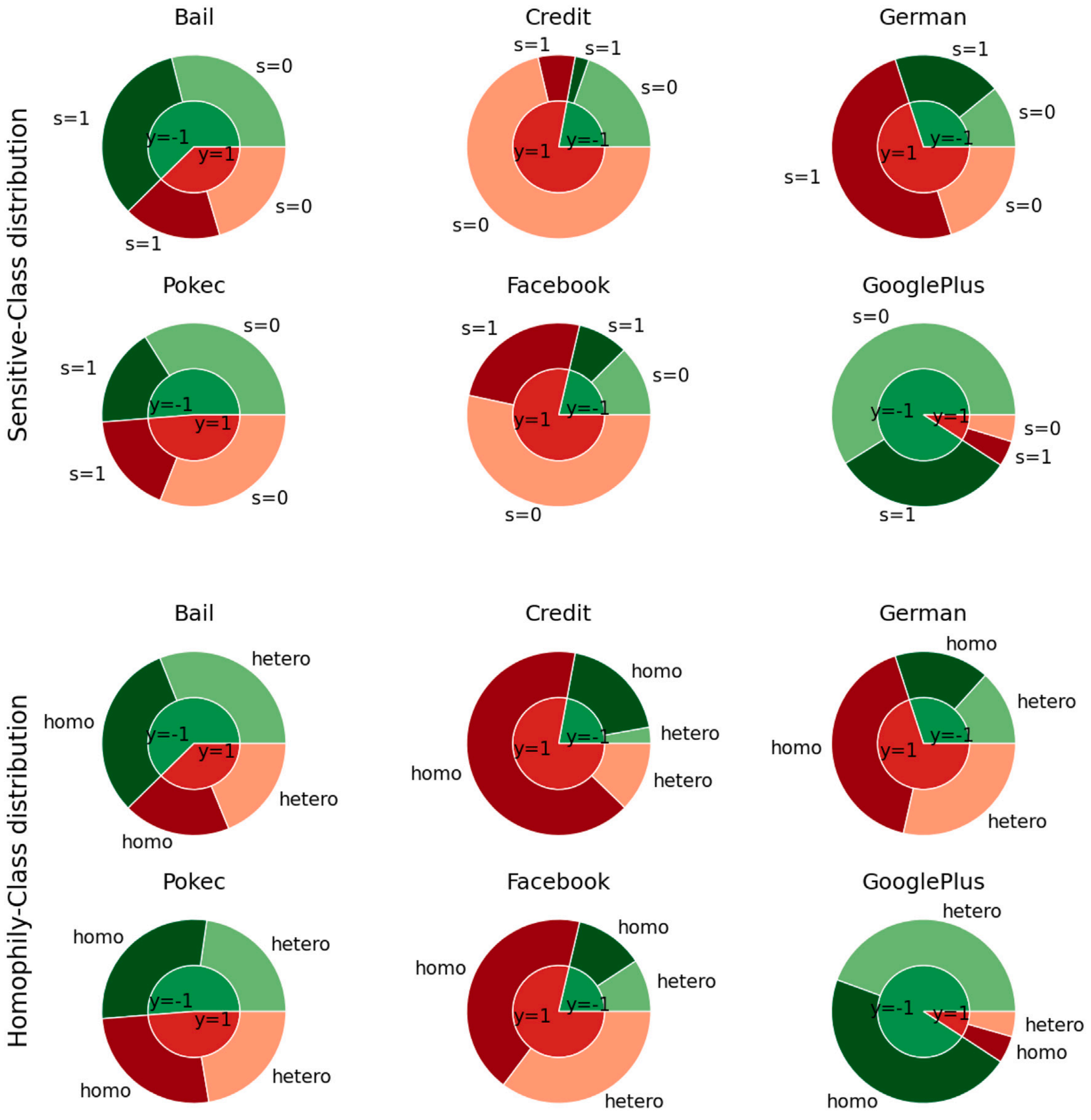


Fig. 1. German [94], Credit [95], Bail [96], Pokec [97], Facebook [98], and Google Plus [98] datasets: class and sensitive attributes distributions along with the edges homophily.

Table 1
Comparison between NIFTY and our proposal on the three scenarios and on the six datasets when the DDP is exploited as the fairness metric and ACC or AUROC is exploited as the technical metrics. Best results are highlighted in bold.

Scenario	Dataset	Algorithm	ACC (↑)	DDP (↓)	AUROC (↑)	DDP (↓)
1	German	NIFTY	0.7 ± 0.03	0.14 ± 0.04	0.63 ± 0.05	0.15 ± 0.04
		Our proposal	0.7 ± 0.03	**0.03 ± 0.03**	**0.64 ± 0.04**	**0.09 ± 0.05**
	Bail	NIFTY	**0.67 ± 0.04**	0.31 ± 0.04	**0.7 ± 0.04**	0.31 ± 0.04
		Our proposal	0.63 ± 0.04	**0.03 ± 0.02**	0.6 ± 0.05	**0.03 ± 0.02**
	Credit	NIFTY	0.77 ± 0.03	0.57 ± 0.09	0.59 ± 0.05	0.57 ± 0.09
		Our proposal	**0.79 ± 0.02**	**0.05 ± 0.03**	**0.6 ± 0.04**	**0.06 ± 0.03**
	Pokec	NIFTY	0.81 ± 0.03	0.81 ± 0.32	0.9 ± 0.02	0.81 ± 0.32
		Our proposal	**0.82 ± 0.03**	**0.31 ± 0.23**	0.9 ± 0.02	**0.31 ± 0.23**
	Google Plus	NIFTY	0.92 ± 0.02	0.17 ± 0.1	0.35 ± 0.07	0.17 ± 0.09
		Our proposal	0.92 ± 0.02	**0.02 ± 0.01**	**0.4 ± 0.12**	**0.03 ± 0.03**
	Facebook	NIFTY	0.79 ± 0.03	0.08 ± 0.06	0.73 ± 0.05	0.09 ± 0.06
		Our proposal	0.79 ± 0.03	**0.02 ± 0.02**	**0.75 ± 0.05**	**0.02 ± 0.02**

(*continued on next page*)

reason, we will follow the same strategy proposed by the authors of [100]: the selected hyperparameters are the ones that minimize the fairness metrics (DDP, DEO$^+$, DEO$^-$, or CF) while keeping at least the τ = 95% of the max value for the technical metrics (ACC or AUROC).

Table 1 (*continued*).

Scenario	Dataset	Algorithm	ACC (↑)	DDP (↓)	AUROC (↑)	DDP (↓)
2	German	NIFTY	**0.7 ± 0.02**	0.18 ± 0.03	**0.65 ± 0.04**	0.19 ± 0.03
		Our proposal	0.69 ± 0.01	**0.05 ± 0.04**	0.63 ± 0.06	**0.08 ± 0.05**
	Bail	NIFTY	**0.67 ± 0.03**	0.32 ± 0.04	**0.7 ± 0.03**	0.32 ± 0.04
		Our proposal	0.63 ± 0.03	**0.03 ± 0.02**	0.59 ± 0.03	**0.04 ± 0.03**
	Credit	NIFTY	0.77 ± 0.03	0.58 ± 0.05	0.58 ± 0.04	0.58 ± 0.05
		Our proposal	**0.8 ± 0.01**	**0.04 ± 0.03**	**0.61 ± 0.04**	**0.07 ± 0.02**
	Pokec	NIFTY	0.8 ± 0.02	0.62 ± 0.17	**0.9 ± 0.01**	0.62 ± 0.17
		Our proposal	0.8 ± 0.02	**0.18 ± 0.1**	0.89 ± 0.02	**0.19 ± 0.13**
	Google Plus	NIFTY	**0.93 ± 0.01**	0.21 ± 0.13	0.38 ± 0.08	0.21 ± 0.13
		Our proposal	0.92 ± 0.04	**0.01 ± 0.01**	**0.39 ± 0.09**	**0.01 ± 0.01**
	Facebook	NIFTY	0.8 ± 0.04	0.1 ± 0.08	0.76 ± 0.03	0.1 ± 0.07
		Our proposal	0.8 ± 0.04	**0.08 ± 0.05**	0.76 ± 0.04	**0.03 ± 0.02**
3	German	NIFTY	**0.7 ± 0.03**	0.25 ± 0.05	**0.63 ± 0.05**	0.25 ± 0.05
		Our proposal	0.69 ± 0.03	**0.08 ± 0.04**	0.62 ± 0.04	**0.18 ± 0.05**
	Bail	NIFTY	**0.67 ± 0.03**	0.34 ± 0.05	**0.69 ± 0.04**	0.34 ± 0.05
		Our proposal	0.63 ± 0.03	**0.03 ± 0.02**	0.6 ± 0.05	**0.03 ± 0.02**
	Credit	NIFTY	0.76 ± 0.03	0.58 ± 0.09	0.59 ± 0.05	0.58 ± 0.09
		Our proposal	**0.79 ± 0.02**	**0.04 ± 0.03**	**0.6 ± 0.04**	**0.07 ± 0.03**
	Pokec	NIFTY	**0.81 ± 0.04**	0.73 ± 0.31	**0.91 ± 0.02**	0.73 ± 0.31
		Our proposal	0.8 ± 0.03	**0.33 ± 0.19**	0.9 ± 0.02	**0.31 ± 0.18**
	Google Plus	NIFTY	0.92 ± 0.02	0.31 ± 0.18	0.35 ± 0.06	0.31 ± 0.18
		Our proposal	0.92 ± 0.02	**0.04 ± 0.03**	**0.39 ± 0.92**	**0.12 ± 0.06**
	Facebook	NIFTY	0.79 ± 0.03	0.13 ± 0.09	0.86 ± 0.03	0.14 ± 0.09
		Our proposal	0.79 ± 0.03	**0.06 ± 0.04**	**0.89 ± 0.03**	**0.06 ± 0.04**

Table 2
Comparison between NIFTY and our proposal on the three scenarios and on the six datasets when the DEO$^+$ is exploited as the fairness metric and ACC or AUROC is exploited as the technical metrics. Best results are highlighted in bold.

Scenario	Dataset	Algorithm	ACC (↑)	DEO$^+$ (↓)	AUROC (↑)	DEO$^+$ (↓)
1	German	NIFTY	0.7 ± 0.03	0.14 ± 0.05	0.63 ± 0.05	0.15 ± 0.05
		Our proposal	0.7 ± 0.03	**0.04 ± 0.03**	**0.64 ± 0.04**	**0.08 ± 0.06**
	Bail	NIFTY	**0.67 ± 0.04**	0.3 ± 0.07	**0.7 ± 0.04**	0.3 ± 0.07
		Our proposal	0.6 ± 0.03	**0.04 ± 0.02**	0.6 ± 0.05	**0.04 ± 0.02**
	Credit	NIFTY	0.77 ± 0.03	0.57 ± 0.12	0.59 ± 0.05	0.57 ± 0.12
		Our proposal	**0.79 ± 0.02**	**0.07 ± 0.05**	**0.6 ± 0.04**	**0.06 ± 0.04**
	Pokec	NIFTY	0.8 ± 0.03	0.62 ± 0.26	0.89 ± 0.03	0.66 ± 0.29
		Our proposal	0.8 ± 0.03	**0.25 ± 0.14**	0.89 ± 0.02	**0.23 ± 0.15**
	Google Plus	NIFTY	0.92 ± 0.02	0.12 ± 0.08	0.35 ± 0.07	0.12 ± 0.08
		Our proposal	0.92 ± 0.02	**0.04 ± 0.03**	**0.39 ± 0.09**	**0.06 ± 0.05**
	Facebook	NIFTY	0.79 ± 0.03	0.09 ± 0.06	0.73 ± 0.06	0.09 ± 0.06
		Our proposal	0.79 ± 0.03	**0.03 ± 0.02**	**0.75 ± 0.05**	**0.04 ± 0.03**
2	German	NIFTY	**0.7 ± 0.02**	0.17 ± 0.04	**0.65 ± 0.04**	0.17 ± 0.05
		Our proposal	0.69 ± 0.02	**0.1 ± 0.01**	0.63 ± 0.06	**0.14 ± 0.04**
	Bail	NIFTY	**0.66 ± 0.03**	0.27 ± 0.09	**0.69 ± 0.04**	0.27 ± 0.09
		Our proposal	0.63 ± 0.03	**0.03 ± 0.02**	0.59 ± 0.03	**0.04 ± 0.03**
	Credit	NIFTY	0.77 ± 0.03	0.56 ± 0.06	0.58 ± 0.04	0.56 ± 0.06
		Our proposal	**0.8 ± 0.01**	**0.05 ± 0.04**	**0.61 ± 0.04**	**0.07 ± 0.02**
	Pokec	NIFTY	0.8 ± 0.01	0.44 ± 0.17	0.9 ± 0.01	0.51 ± 0.2
		Our proposal	0.8 ± 0.02	**0.19 ± 0.15**	**0.91 ± 0.01**	**0.19 ± 0.15**
	Google Plus	NIFTY	0.93 ± 0.01	0.17 ± 0.14	0.38 ± 0.08	0.17 ± 0.14
		Our proposal	0.93 ± 0.01	**0.03 ± 0.03**	**0.44 ± 0.19**	**0.04 ± 0.03**
	Facebook	NIFTY	0.8 ± 0.04	0.09 ± 0.06	0.74 ± 0.03	0.09 ± 0.06
		Our proposal	0.8 ± 0.04	**0.04 ± 0.04**	**0.78 ± 0.03**	**0.04 ± 0.04**
3	German	NIFTY	**0.7 ± 0.03**	0.26 ± 0.07	**0.63 ± 0.05**	0.26 ± 0.07
		Our proposal	0.69 ± 0.03	**0.09 ± 0.06**	0.62 ± 0.05	**0.19 ± 0.06**
	Bail	NIFTY	**0.67 ± 0.03**	0.32 ± 0.08	**0.69 ± 0.04**	0.32 ± 0.08
		Our proposal	0.63 ± 0.03	**0.04 ± 0.03**	0.64 ± 0.04	**0.04 ± 0.03**
	Credit	NIFTY	0.76 ± 0.03	0.58 ± 0.11	0.58 ± 0.05	0.58 ± 0.11
		Our proposal	**0.79 ± 0.02**	**0.08 ± 0.06**	**0.6 ± 0.04**	**0.07 ± 0.04**
	Pokec	NIFTY	0.81 ± 0.04	0.65 ± 0.29	**0.91 ± 0.02**	0.65 ± 0.29
		Our proposal	0.81 ± 0.04	**0.29 ± 0.17**	0.9 ± 0.02	**0.25 ± 0.16**
	Google Plus	NIFTY	0.92 ± 0.02	0.19 ± 0.16	0.35 ± 0.06	0.19 ± 0.16
		Our proposal	0.92 ± 0.02	**0.04 ± 0.03**	**0.39 ± 0.13**	**0.1 ± 0.09**
	Facebook	NIFTY	0.79 ± 0.03	0.15 ± 0.08	0.85 ± 0.03	0.15 ± 0.12
		Our proposal	0.79 ± 0.03	**0.12 ± 0.09**	**0.9 ± 0.03**	**0.12 ± 0.09**

Table 3
Comparison between NIFTY and our proposal on the three scenarios and on the six datasets when the DEO^{-} is exploited as the fairness metric and ACC or AUROC is exploited as the technical metrics. Best results are highlighted in bold.

Scenario	Dataset	Algorithm	ACC (↑)	DEO^{-} (↓)	AUROC (↑)	DEO^{-} (↓)
1	German	NIFTY	0.7 ± 0.03	0.12 ± 0.06	0.63 ± 0.05	0.14 ± 0.06
		Our proposal	0.7 ± 0.03	**0.04 ± 0.03**	**0.64 ± 0.04**	**0.08 ± 0.06**
	Bail	NIFTY	**0.67 ± 0.04**	0.31 ± 0.05	**0.7 ± 0.04**	0.31 ± 0.05
		Our proposal	0.63 ± 0.03	**0.03 ± 0.02**	0.6 ± 0.05	**0.02 ± 0.01**
	Credit	NIFTY	0.77 ± 0.03	0.61 ± 0.2	0.59 ± 0.05	0.61 ± 0.2
		Our proposal	**0.78 ± 0.02**	**0.07 ± 0.05**	**0.6 ± 0.04**	**0.07 ± 0.05**
	Pokec	NIFTY	0.81 ± 0.02	0.8 ± 0.21	**0.91 ± 0.02**	0.8 ± 0.21
		Our proposal	0.81 ± 0.03	**0.28 ± 0.14**	0.9 ± 0.02	**0.25 ± 0.11**
	Google Plus	NIFTY	0.92 ± 0.02	0.18 ± 0.1	0.35 ± 0.07	0.18 ± 0.1
		Our proposal	0.92 ± 0.02	**0.02 ± 0.02**	**0.4 ± 0.12**	**0.03 ± 0.03**
	Facebook	NIFTY	0.79 ± 0.03	0.12 ± 0.1	0.73 ± 0.06	0.13 ± 0.08
		Our proposal	0.79 ± 0.03	**0.05 ± 0.04**	**0.75 ± 0.05**	**0.05 ± 0.04**
2	German	NIFTY	**0.7 ± 0.02**	0.19 ± 0.05	**0.65 ± 0.04**	0.22 ± 0.03
		Our proposal	0.69 ± 0.01	**0.08 ± 0.05**	0.63 ± 0.06	**0.11 ± 0.04**
	Bail	NIFTY	**0.64 ± 0.03**	0.32 ± 0.04	**0.7 ± 0.03**	0.32 ± 0.04
		Our proposal	0.62 ± 0.03	**0.02 ± 0.01**	0.63 ± 0.04	**0.02 ± 0.01**
	Credit	NIFTY	0.77 ± 0.03	0.68 ± 0.24	0.59 ± 0.04	0.68 ± 0.24
		Our proposal	**0.79 ± 0.01**	**0.06 ± 0.04**	**0.6 ± 0.05**	**0.06 ± 0.04**
	Pokec	NIFTY	**0.83 ± 0.02**	0.75 ± 0.23	**0.92 ± 0.01**	0.75 ± 0.23
		Our proposal	0.8 ± 0.01	**0.26 ± 0.09**	0.9 ± 0.01	**0.26 ± 0.07**
	Google Plus	NIFTY	**0.93 ± 0.01**	0.21 ± 0.13	0.38 ± 0.08	0.21 ± 0.13
		Our proposal	0.92 ± 0.04	**0.01 ± 0.01**	**0.39 ± 0.09**	**0.01 ± 0.01**
	Facebook	NIFTY	0.8 ± 0.04	0.2 ± 0.12	**0.75 ± 0.05**	0.22 ± 0.12
		Our proposal	0.8 ± 0.04	**0.08 ± 0.04**	0.74 ± 0.05	**0.07 ± 0.05**
3	German	NIFTY	**0.7 ± 0.03**	0.22 ± 0.07	**0.63 ± 0.05**	0.22 ± 0.07
		Our proposal	0.69 ± 0.03	**0.07 ± 0.06**	0.62 ± 0.04	**0.15 ± 0.08**
	Bail	NIFTY	**0.67 ± 0.03**	0.34 ± 0.06	**0.69 ± 0.04**	0.34 ± 0.06
		Our proposal	0.63 ± 0.03	**0.02 ± 0.02**	0.6 ± 0.05	**0.03 ± 0.02**
	Credit	NIFTY	0.76 ± 0.03	0.61 ± 0.21	0.58 ± 0.05	0.61 ± 0.21
		Our proposal	**0.77 ± 0.03**	**0.06 ± 0.05**	**0.6 ± 0.04**	**0.06 ± 0.05**
	Pokec	NIFTY	**0.83 ± 0.03**	0.68 ± 0.23	**0.92 ± 0.02**	0.68 ± 0.23
		Our proposal	0.8 ± 0.03	**0.27 ± 0.12**	0.91 ± 0.02	**0.26 ± 0.13**
	Google Plus	NIFTY	0.92 ± 0.02	0.31 ± 0.19	0.35 ± 0.06	0.31 ± 0.19
		Our proposal	0.92 ± 0.02	**0.04 ± 0.03**	**0.39 ± 0.13**	**0.13 ± 0.06**
	Facebook	NIFTY	0.79 ± 0.03	0.19 ± 0.12	0.85 ± 0.03	0.19 ± 0.12
		Our proposal	0.79 ± 0.03	**0.08 ± 0.06**	**0.89 ± 0.03**	**0.07 ± 0.05**

The value $\tau = 95\%$ has been selected as a reasonable loss in accuracy to achieve higher fairness as in [100] but we will also analyse the impact of τ on the results later in this section.

In the results, we reported the mean and standard deviation for the different metrics computed on the different test splits.

Experiments have been coded in Python 3.10.11, and are based on PyG[3] 2.1.0 and the PyTorch [123] $1.12.1 + cu116$ frameworks. The code for all the experiments in the paper is available in a git repository.[4]

4.3. Experimental results

Table 1 reports the comparison between NIFTY and our proposal on the three scenarios over the six datasets when fairness is measured through the DDP and ACC or AUROC is exploited as the technical metric. Tables 2, 3, and 4 report the analogous results of Table 1 when fairness is measured according to the DEO^{+}, the DEO^{-}, and the CF metrics respectively.

In order to ease the discussion of the results of Tables 1–4, Fig. 2 graphically outline the same tabular results for the transductive (Fig. 2(a)), semi-inductive (Fig. 2(b)), and inductive (Fig. 2(c)) scenarios respectively.

From Tables 1–4 and Fig. 2, it is possible to observe how our proposal consistently outperforms NIFTY in terms of fairness.

Let us consider the DDP metric in Table 1. Pokec is the dataset where NIFTY results in the highest DDP. In all three scenarios, we can reduce by more than half the DDP value without sacrificing accuracy or AUROC (maximum reduction of 0.01). For the Credit dataset, the improvement in DDP we obtain is of an order of magnitude, again without sacrificing accuracy or AUROC. For Bail, we observe a similar improvement in DDP, with a maximum drop of 5% in accuracy. For the other datasets, we can still observe a considerable reduction in DDP. Similar considerations can be drawn when considering the DEO fairness metric in Tables 2 and 3. The highest reductions in DEO (an order of magnitude) are obtained on the Bail and Credit datasets. Also in this case, Pokec is the dataset with the highest DEO metric, and we can reduce it considerably. Finally, concerning CF in Table 4, we can see improvements of around an order of magnitude for Bail and Credit datasets. For the other datasets, the improvements are still pretty high, with a reduction of more than 50% in CF.

To better understand the sensitivity of the results of Tables 1–4 and Fig. 2 to the hyperparameter choice, Fig. 3 shows the performance in terms of technical (ACC and AUROC) and fairness (DDP, DEO^{+}, DEO^{-}, and CF) metrics varying the tested hyperparameters, underling also the Pareto optimal hyperparameters combinations. We report the results for the Pokec dataset on the most challenging scenario alone for space constraints, but results are analogous also on the other datasets. Fig. 3

[3] https://www.pyg.org/.

[4] https://github.com/danif93/Fair-Graph-Representation-Learning.

Table 4
Comparison between NIFTY and our proposal on the three scenarios and on the six datasets when the CF is exploited as the fairness metric and ACC or AUROC is exploited as the technical metrics. Best results are highlighted in bold.

Scenario	Dataset	Algorithm	ACC (↑)	CF (↓)	AUROC (↑)	CF (↓)
1	German	NIFTY	0.7 ± 0.03	0.09 ± 0.01	**0.63 ± 0.05**	0.1 ± 0.02
		Our proposal	0.7 ± 0.03	**0.02 ± 0.01**	0.62 ± 0.05	**0.08 ± 0.01**
	Bail	NIFTY	**0.67 ± 0.04**	0.31 ± 0.02	**0.7 ± 0.04**	0.31 ± 0.02
		Our proposal	0.63 ± 0.03	**0.03 ± 0.01**	0.6 ± 0.05	**0.04 ± 0.01**
	Credit	NIFTY	0.75 ± 0.03	0.85 ± 0.02	0.59 ± 0.05	0.87 ± 0.02
		Our proposal	**0.79 ± 0.02**	**0.05 ± 0.01**	**0.6 ± 0.04**	**0.19 ± 0.02**
	Pokec	NIFTY	0.8 ± 0.03	0.58 ± 0.05	0.89 ± 0.03	0.58 ± 0.06
		Our proposal	0.8 ± 0.03	**0.16 ± 0.03**	0.89 ± 0.02	**0.15 ± 0.02**
	Google Plus	NIFTY	0.92 ± 0.02	0.13 ± 0.08	0.35 ± 0.07	0.13 ± 0.08
		Our proposal	0.92 ± 0.02	**0.02 ± 0.01**	**0.38 ± 0.1**	**0.07 ± 0.02**
	Facebook	NIFTY	0.79 ± 0.03	0.11 ± 0.04	0.73 ± 0.05	0.12 ± 0.03
		Our proposal	0.79 ± 0.03	**0.06 ± 0.02**	**0.75 ± 0.05**	**0.07 ± 0.01**
2	German	NIFTY	**0.7 ± 0.02**	0.1 ± 0.01	**0.65 ± 0.04**	0.11 ± 0.02
		Our proposal	0.69 ± 0.02	**0.03 ± 0.01**	0.63 ± 0.07	**0.09 ± 0.01**
	Bail	NIFTY	**0.67 ± 0.03**	0.31 ± 0.02	**0.7 ± 0.03**	0.31 ± 0.02
		Our proposal	0.63 ± 0.01	**0.04 ± 0.01**	0.62 ± 0.03	**0.04 ± 0.01**
	Credit	NIFTY	0.75 ± 0.03	0.86 ± 0.02	0.58 ± 0.04	0.86 ± 0.01
		Our proposal	**0.8 ± 0.01**	**0.05 ± 0.01**	**0.61 ± 0.04**	**0.18 ± 0.02**
	Pokec	NIFTY	0.79 ± 0.02	0.59 ± 0.07	0.89 ± 0.01	0.56 ± 0.03
		Our proposal	0.79 ± 0.02	**0.16 ± 0.02**	0.89 ± 0.01	**0.14 ± 0.02**
	Google Plus	NIFTY	0.93 ± 0.01	0.17 ± 0.1	0.38 ± 0.08	0.17 ± 0.1
		Our proposal	0.93 ± 0.01	**0.02 ± 0.01**	**0.44 ± 0.19**	**0.03 ± 0.01**
	Facebook	NIFTY	0.8 ± 0.04	0.11 ± 0.03	0.74 ± 0.03	0.11 ± 0.03
		Our proposal	0.8 ± 0.04	**0.06 ± 0.02**	**0.75 ± 0.03**	**0.07 ± 0.01**
3	German	NIFTY	0.7 ± 0.03	0.13 ± 0.02	**0.63 ± 0.05**	0.13 ± 0.02
		Our proposal	0.7 ± 0.03	**0.03 ± 0.01**	0.62 ± 0.04	**0.06 ± 0.01**
	Bail	NIFTY	**0.67 ± 0.03**	0.33 ± 0.02	**0.69 ± 0.04**	0.33 ± 0.02
		Our proposal	0.62 ± 0.03	**0.04 ± 0.01**	0.6 ± 0.05	**0.04 ± 0.01**
	Credit	NIFTY	0.75 ± 0.03	0.86 ± 0.02	0.59 ± 0.05	0.87 ± 0.02
		Our proposal	**0.79 ± 0.02**	**0.05 ± 0.01**	**0.6 ± 0.04**	**0.19 ± 0.01**
	Pokec	NIFTY	**0.81 ± 0.04**	0.58 ± 0.06	**0.91 ± 0.02**	0.58 ± 0.06
		Our proposal	0.8 ± 0.04	**0.17 ± 0.03**	0.9 ± 0.02	**0.15 ± 0.03**
	Google Plus	NIFTY	**0.92 ± 0.02**	0.26 ± 0.14	0.35 ± 0.06	0.26 ± 0.14
		Our proposal	0.91 ± 0.04	**0.04 ± 0.02**	**0.39 ± 0.13**	**0.11 ± 0.04**
	Facebook	NIFTY	0.79 ± 0.03	0.17 ± 0.06	0.85 ± 0.03	0.17 ± 0.06
		Our proposal	0.79 ± 0.03	**0.08 ± 0.02**	**0.88 ± 0.03**	**0.09 ± 0.02**

shows that hyperparameter choice does not influence the quality of the results since the cloud of points, and the Pareto optimal points, corresponding to all the combination of hyperparameters of our proposal consistently outperform the one of NIFTY.

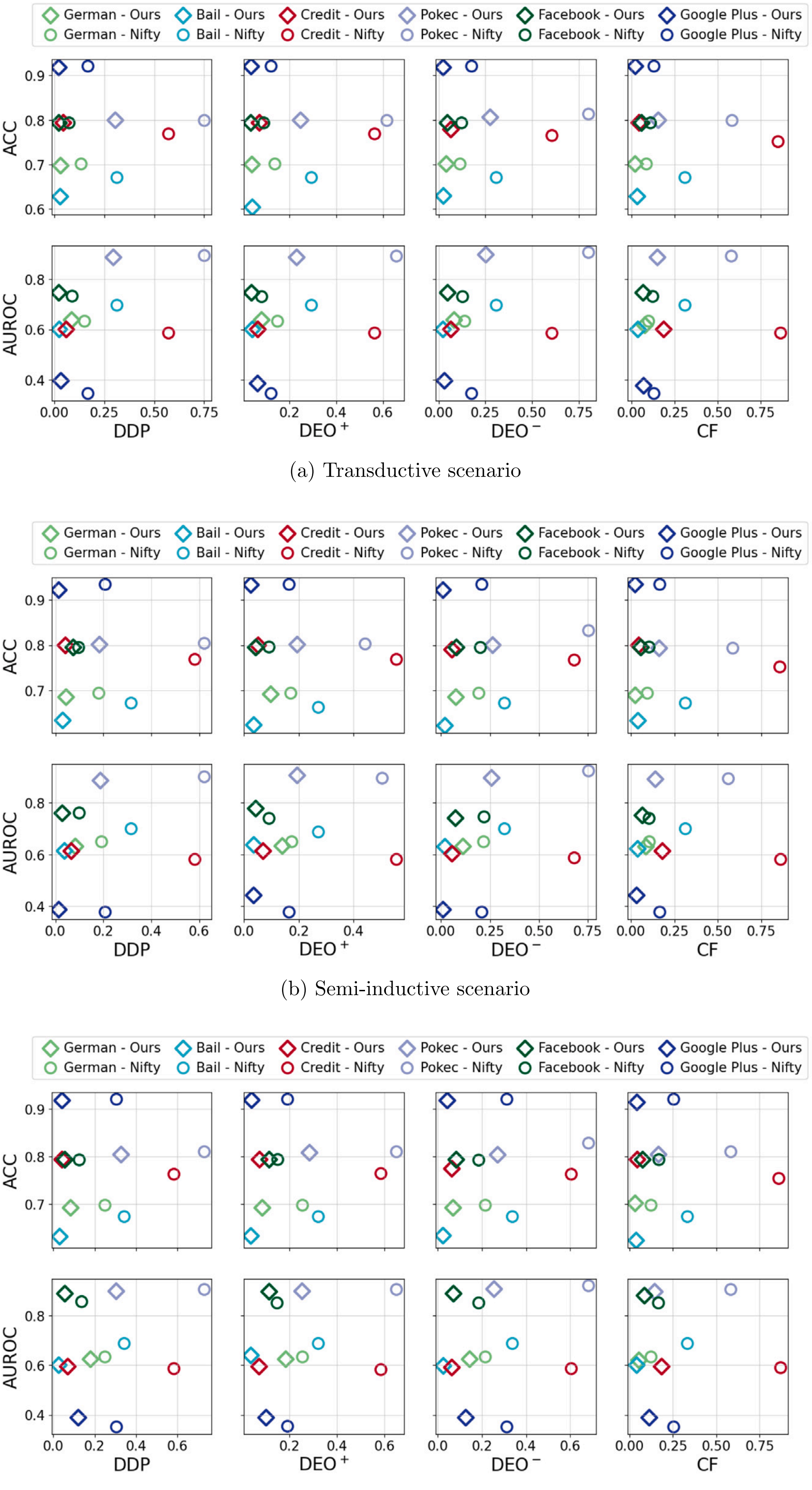
For completeness, Fig. 4 evaluates the impact of τ on the fairness metrics under the same setting of Fig. 3. From Fig. 4 it is possible to see that $\tau = 95\%$ is qualitatively a good threshold, i.e., the one that allows for minimum loss in accuracy while obtaining the minimum discrimination (maximum fairness according to the different metrics). Moreover, when changing τ results change as expected: increasing τ decreases fairness since we tend to care more about accuracy than fairness.

4.4. Discussion

Generally, based on the results of Section 4.3 we can see that our proposal improves the fairness metrics compared to NIFTY in all the cases. While in some cases the loss in performance is negligible, in some other cases, e.g. for the Bail dataset, our proposal is, on one hand, able to improve the fairness of the trained model up to an order of magnitude, while, on the other hand, it sacrifices on average 6% points of accuracy.

This trade-off between fairness and utility is observed in many works [124,125], and it describes the natural tension between the two opposed objectives, where the former aims at removing information linked to discrimination, while the latter tries to just maximize the data prediction accuracy, which is often biased against population subgroups in real-world datasets. Observing the results on the other datasets, our method is still able to achieve a substantial improvement over all the fairness metrics (from ×1.5 to ×10 reduction in discrimination depending on the dataset and the scenario) while achieving comparable utility in terms of technical metrics. Moreover, when dealing with the Credit, Google Plus, and Facebook datasets, our method is also able to consistently achieve better utility besides improving fairness, proving its superiority when compared to NIFTY. While NIFTY already tries to balance utility with reducing bias, the improvements of our proposal can be mainly justified by the fair preprocessing step, which is able to extract a fair data representation that is also tailored for performing well for the required task. In this way, the extracted data embeddings are thus able to counterbalance the tension between maintaining high utility and discarding biased information.

In Fig. 2 we graphically represent the relationship between the predictive performance and the fairness metrics for the three considered scenarios (transductive, semi-inductive and inductive). We can observe that the x-axis spans higher values in the transductive scenario compared to the semi-inductive and inductive ones. When comparing the fairness values of the transductive and the inductive scenario, it is important to consider that GNN rely heavily on the structure of the



(a) Transductive scenario

(b) Semi-inductive scenario

(c) Inductive scenario

Fig. 2. Technical (ACC, AUROC) and fairness (DDP, DEO$^+$, DEO$^-$, CF) metrics for all the tested datasets under the three scenarios. Top-left is better (small un-fairness and high accuracy). Applying the proposed method gives lower discrimination rates while preserving utility.

graph to convey information. In the inductive scenario, there are no edges between the nodes in $\mathcal{V}_{\mathcal{T}}$ and the nodes in $\mathcal{V}_{D}$, while in the transductive scenario the nodes in $\mathcal{V}_{\mathcal{T}} = \mathcal{V}_{\mathcal{U}}$ will be, on average, connected to many nodes in $\mathcal{V}_{D}$, since they are part of the same graph. In the inductive scenario, improving the fairness metrics is less challenging compared to the transductive one because the model can rely on less (biased) topological information since the test samples are composed of isolated sub-graphs. On the other hand, in the transductive scenario the

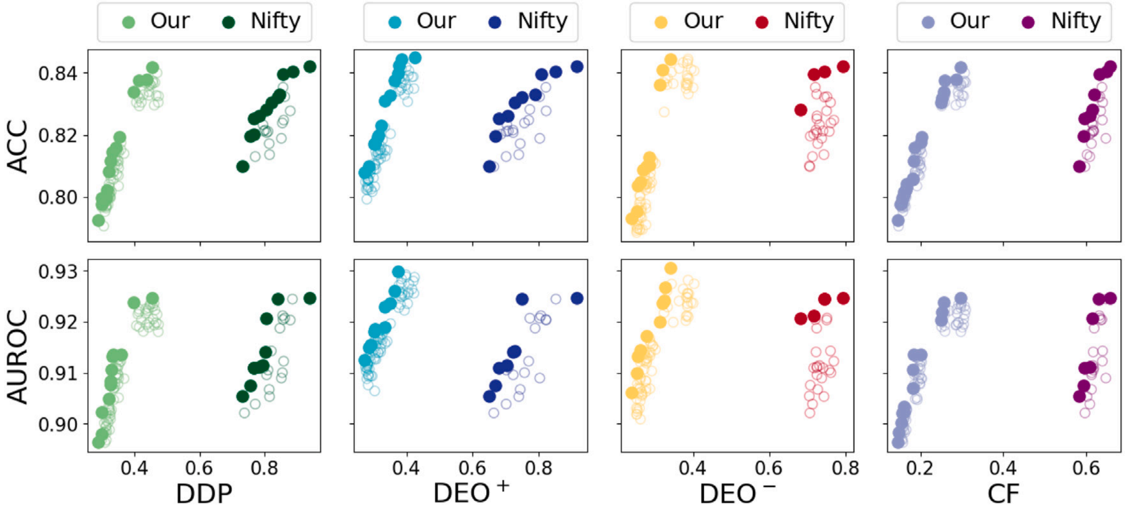


Fig. 3. Hyperparameters distribution according to technical (ACC and AUROC) and fairness (DDP, DEO$^+$, DEO$^-$, and CF) metrics for the test split of the Pokec dataset in the inductive scenario. The Pareto frontier for best utility and fairness (top-left corner) is highlighted in filled circles. Applying the proposed method gives lower discrimination rates while preserving utility.

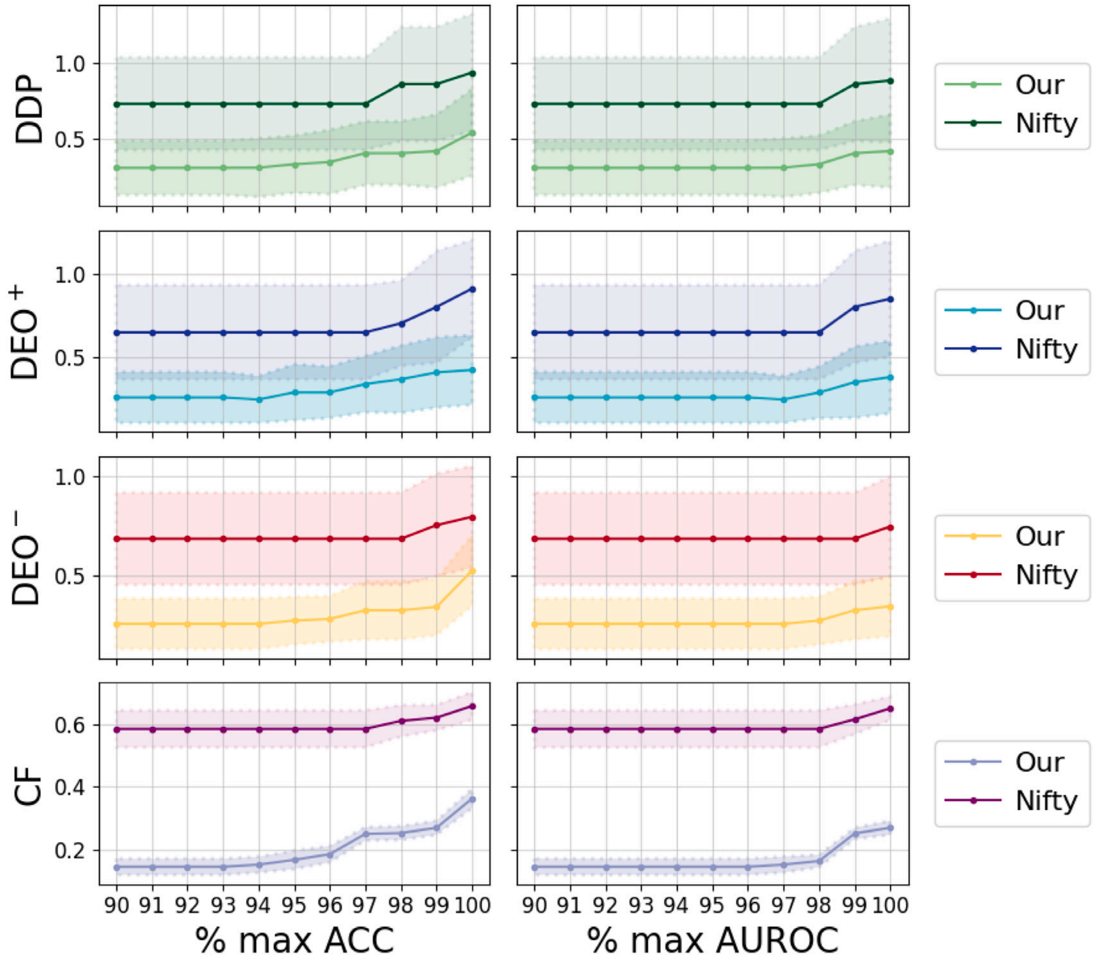


Fig. 4. Fairness metrics (DDP, DEO$^+$, DEO$^-$, CF) on the test split of the Pokec dataset under the inductive setting when $\tau \in [90\%, 100\%]$ for the best-found utility value (ACC, AUROC). Lower values on the y-axis are better. Applying our proposal gives the best results in terms of fairness.

model can exploit sensitive information from the neighbouring training nodes in the GNN's receptive field, that can lead to unfair solutions in case of high homophily on the sensitive attribute. We notice that this behaviour is mostly visible on NIFTY since it does not perform biased edge dropout. Differently, the performance metrics are less impacted by the specific setting [126]. This is particularly evident for some datasets, e.g. Pokec with DEO metric or Google Plus and Facebook for the DEO, DDP and CF metric, where the fairness metric for NIFTY is lower (better) in the second and third settings compared to the first one. However, our proposed method behaves well in all three scenarios, showing similar levels of performance and fairness metrics.

5. Conclusion

Trustworthy Learning algorithms that can be fed directly with complex and large graphs data are nowadays a necessity due to the increasing complexity and amount of data available in real-world applications. In fact, on one hand, machine learning models must meet high technical standards (e.g., high accuracy with limited computational requirements), but, at the same time, they must be sure not to discriminate against subgroups of the population (e.g., based on gender or race). Graph Neural Networks are currently the most effective solution to meet the technical requirements, even if it has been demonstrated that they inherit and amplify the biases contained in the data as a reflection of societal inequities. In fact, when dealing with graph data, these biases can be hidden not only in the node attributes but also in the connections between entities. In this paper, we empowered NIFTY's fairness, one of the most effective Fair Graph Neural Networks, with two new strategies. The first one is a Biased Edge Dropout, namely, we drop graph edges to balance homophilous and heterophilous sensitive connections, mitigating the bias induced by subgroup node cardinality. The second is Fair Attributes Preprocessing, which is the process of learning a fair transformation of the original node attributes. The effectiveness of our proposal has been tested on a series of datasets with increasingly challenging scenarios. These scenarios dealt with different levels of knowledge about the entire graph, i.e., how many portions of the graph are known and what sub-portion is labelled at the training and forward phases. The results have shown promising results, demonstrating the effectiveness of our proposal. Note that the two proposed strategies can be also exploited to make any graph model fairer. This is particularly useful when, differently from the context of NIFTY, the existing legislation does not allow the explicit use of sensitive attributes. In fact, both Biased Edge Dropout and Fair Attributes Preprocessing do not require the sensitive attributes to be explicitly present among the training features but they just need the knowledge of the samples' membership to the sensitive groups. In future work, we plan to improve the computational requirements of the methods including a better hyperparameter search strategy, i.e., with gradient-free methods and to extend the proposed approach to different graph convolutional operators. Moreover, we plan to study the learning of fair graph representations in the incremental and online settings, where both the graph nodes and the supervision arrive over time and periodically re-training the whole model from scratch when new training data is available would be computationally inefficient.

Declaration of competing interest

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

Data availability

We will release the source code and datasets in a GitHub repository after the paper gets accepted.

Acknowledgements

This work is partially supported by the projects: ELSA – European Lighthouse on Secure and Safe AI funded by the European Union under grant agreement No. 101070617; SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU; the PON R&I 2014–2020 project *Smart Waste Treatment* founded by the FSE REAC-EU; the project "iNEST: Interconnected Nord-Est Innovation Ecosystem" funded under the National Recovery and Resilience Plan (NRRP), Mission 4 Component 2 Investment 1.5 - Call for tender No. 3277 of 30 December 2021 of Italian Ministry of University and Research funded by the European Union – NextGenerationEU, project code: ECS00000043, Concession Decree No. 1058 of June 23, 2022, CUP C43C22000340006. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

[1] J. Egger, C. Gsaxner, A. Pepe, K.L. Pomykala, F. Jonske, M. Kurz, J. Li, J. Kleesiek, Medical deep learning-a systematic meta-review, Comput. Methods Programs Biomed. (2022) 106874.

[2] A.M. Froomkin, I. Kerr, J. Pineau, When AIs outperform doctors: confronting the challenges of a tort-induced over-reliance on machine learning, Ariz. L. Rev. 61 (2019) 33.

[3] M. Jarbou, D. Won, J. Gillis-Mattson, R. Romanczyk, Deep learning-based school attendance prediction for autistic students, Sci. Rep. 12 (1) (2022) 1–11.

[4] G. Apruzzese, P. Laskov, E.M. de Oca, W. Mallouli, L.B. Rapa, A.V. Grammatopoulos, F. Di Franco, The role of machine learning in cybersecurity, Digit. Threats: Res. Pract. (2022).

[5] B. Williamson, S. Bayne, S. Shay, The datafication of teaching in Higher Education: critical issues and perspectives, Teach. High. Educ. 25 (4) (2020) 351–365.

[6] U.A. Mejias, N. Couldry, Datafication, Internet Policy Rev. 8 (4) (2019).

[7] M. Pandey, M. Fernandez, F. Gentile, O. Isayev, A. Tropsha, A.C. Stern, A. Cherkasov, The transformational role of GPU computing and deep learning in drug discovery, Nat. Mach. Intell. 4 (3) (2022) 211–221.

[8] D. Bacciu, F. Errica, A. Micheli, M. Podda, A gentle introduction to deep learning for graphs, Neural Netw. 129 (2020) 203–221.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, S.Y. Philip, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2020) 4–24.

[10] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, AI Open 1 (2020) 57–81.

[11] L. Oneto, N. Navarin, B. Biggio, F. Errica, A. Micheli, F. Scarselli, M. Bianchini, L. Demetrio, P. Bongini, A. Tacchella, Towards learning trustworthily, automatically, and with guarantees on graphs: An overview, Neurocomputing (2022).

[12] P. Scherer, M. Trebacz, N. Simidjievski, R. Viñas, Z. Shams, H.A. Terre, M. Jamnik, P. Liò, Unsupervised construction of computational graphs for gene expression data with explicit structural inductive biases, Bioinformatics 38 (5) (2022) 1320–1327.

[13] S. Yassine, S. Kadry, M.A. Sicilia, Detecting communities using social network analysis in online learning environments: Systematic literature review, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 12 (1) (2022) e1431.

[14] A. Sperduti, D. Majidi, A. Starita, Extended cascade-correlation for syntactic and structural pattern recognition, in: Advances in Structural and Syntactical Pattern Recognition, 1996.

[15] A. Sperduti, A. Starita, Supervised neural networks for the classification of structures, IEEE Trans. Neural Netw. 8 (3) (1997) 714–735.

[16] G. Nikolentzos, G. Siglidis, M. Vazirgiannis, Graph kernels: A survey, J. Artificial Intelligence Res. 72 (2021) 943–1027.

[17] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R.P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in: Neural Information Processing Systems, 2015.

[18] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Neural Information Processing Systems, 2016.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.

[20] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, K. Weinberger, Simplifying graph convolutional networks, in: International Conference on Machine Learning, 2019.

[21] S. Luan, M. Zhao, X. Chang, D. Precup, Break the ceiling: Stronger multi-scale deep graph convolutional networks, in: Neural Information Processing Systems, 2019.

[22] E. Rossi, F. Frasca, B. Chamberlain, D. Eynard, M. Bronstein, F. Monti, SIGN: Scalable inception graph neural networks, 2020, arXiv preprint arXiv:2004.11198.

[23] M. Liu, H. Gao, S. Ji, Towards deeper graph neural networks, in: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020.

[24] L. Pasa, N. Navarin, A. Sperduti, Polynomial-based graph convolutional neural networks for graph classification, Mach. Learn. (2022) 1–33.

[25] L. Pasa, N. Navarin, W. Erb, A. Sperduti, Backpropagation-free graph neural networks, in: IEEE International Conference on Data Mining, ICDM, 2022.

[26] S.M. West, M. Whittaker, K. Crawford, Discriminating systems, AI Now (2019).

[27] K. Holstein, J. Wortman Vaughan, H. Daumé III, M. Dudik, H. Wallach, Improving fairness in machine learning systems: What do industry practitioners need? in: CHI Conference on Human Factors in Computing Systems, 2019.

[28] S. Costanza-Chock, Design Justice: Community-Led Practices to Build the Worlds We Need, The MIT Press, 2020.

[29] S. Tolmeijer, M. Kneer, C. Sarasua, M. Christen, A. Bernstein, Implementations in machine ethics: A survey, ACM Comput. Surv. 53 (6) (2020) 1–38.

[30] A.F. Winfield, K. Michael, J. Pitt, V. Evers, Machine ethics: The design and governance of ethical AI and autonomous systems, Proc. IEEE 107 (3) (2019) 509–517.

[31] A.F.T. Winfield, M. Jirotka, Ethical governance is essential to building trust in robotics and artificial intelligence systems, Phil. Trans. R. Soc. A 376 (2133) (2018) 20180085.

[32] P. Boddington, Towards a Code of Ethics for Artificial Intelligence, Springer, 2017.

[33] M. Boden, J. Bryson, D. Caldwell, K. Dautenhahn, et al., Principles of robotics: regulating robots in the real world, Connect. Sci. 29 (2) (2017) 124–129.

[34] J.H. Moor, The nature, importance, and difficulty of machine ethics, IEEE Intell. Syst. 21 (4) (2006) 18–21.

[35] C. Allen, G. Varner, J. Zinser, Prolegomena to any future artificial moral agent, J. Exp. Theor. Artif. Intell. 12 (3) (2000) 251–261.

[36] M. Anderson, S. Anderson, GenEth: a general ethical dilemma analyzer, Paladyn J. Behav. Robot. 9 (1) (2018) 337.

[37] L. Floridi, Establishing the rules for building trustworthy AI, Nat. Mach. Intell. 1 (6) (2019) 261–262.

[38] N.A. Smuha, The EU approach to ethics guidelines for trustworthy artificial intelligence, Comput. Law Rev. Int. 20 (4) (2019) 97–106.

[39] B. Shneiderman, Bridging the gap between ethics and practice: guidelines for reliable, safe, and trustworthy human-centered AI systems, ACM Trans. Interact. Intell. Syst. 10 (4) (2020) 1–31.

[40] S. Thiebes, S. Lins, A. Sunyaev, Trustworthy artificial intelligence, Electron. Mark. 31 (2021) 447–464.

[41] The Verge, Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day, 2023, https://www.theverge.com/2016/3/24/11297050/tay-microsoft-chatbot-racist, accessed: 2023-01-23.

[42] S.U. Noble, Algorithms of Oppression, New York University Press, 2018.

[43] L. Sweeney, Discrimination in online ad delivery, Commun. ACM 56 (5) (2013) 44–54.

[44] P. Sapiezynski, A. Ghosh, L. Kaplan, A. Rieke, A. Mislove, Algorithms that" don't see color" measuring biases in lookalike and special ad audiences, in: AAAI/ACM Conference on AI, Ethics, and Society, 2022.

[45] J. Buolamwini, T. Gebru, Gender shades: Intersectional accuracy disparities in commercial gender classification, in: Conference on Fairness, Accountability and Transparency, 2018.

[46] R. Metz, Portland passes broadest facial recognition ban in the US, 2020, https://edition.cnn.com/2020/09/09/tech/portland-facial-recognition-ban/index.html.

[47] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Comput. Surv. 54 (6) (2021) 1–35.

[48] S. Caton, C. Haas, Fairness in machine learning: A survey, 2020, arXiv preprint arXiv:2010.04053.

[49] S. Verma, J. Rubin, Fairness definitions explained, in: IEEE/ACM International Workshop on Software Fairness, 2018.

[50] A.N. Carey, X. Wu, The statistical fairness field guide: perspectives from social and formal sciences, AI Ethics (2022) 1–23.

[51] L. Oneto, S. Chiappa, Fairness in machine learning, in: Recent Trends in Learning from Data, 2020.

[52] T. Calders, F. Kamiran, M. Pechenizkiy, Building classifiers with independency constraints, in: IEEE International Conference on Data Mining Workshops, 2009.

[53] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, in: Neural Information Processing Systems, 2016.

[54] M.J. Kusner, J. Loftus, C. Russell, R. Silva, Counterfactual fairness, in: Neural Information Processing Systems, 2017.

[55] O.B. Deho, C. Zhan, J. Li, J. Liu, L. Liu, T. Duy Le, How do the existing fairness metrics and unfairness mitigation algorithms contribute to ethical learning analytics? Br. J. Educ. Technol. (2022).

[56] J. Kang, H. Tong, Fair graph mining, in: ACM International Conference on Information & Knowledge Management, 2021.

[57] Y. Wang, Y. Yao, H. Tong, F. Xu, J. Lu, Auditing network embedding: An edge influence based approach, IEEE Trans. Knowl. Data Eng. (2021).

[58] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: International Conference on Machine Learning, 2017.

[59] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, H. Liu, Graph learning: A survey, IEEE Trans. Artif. Intell. 2 (2) (2021) 109–127.

[60] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, Comput. Soc. Netw. 6 (1) (2019) 1–23.

[61] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.

[62] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, C. Dwork, Learning fair representations, in: International Conference on Machine Learning, 2013.

[63] H. Zhao, G. Gordon, Inherent tradeoffs in learning fair representations, in: Advances in Neural Information Processing Systems, 2019.

[64] D. Franco, N. Navarin, M. Donini, D. Anguita, L. Oneto, Deep fair models for complex data: Graphs labeling and explainable face recognition, Neurocomputing 470 (2022) 318–334.

[65] P.K. Lohia, K.N. Ramamurthy, M. Bhide, D. Saha, K.R. Varshney, R. Puri, Bias mitigation post-processing for individual and group fairness, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2019.

[66] M. Wan, D. Zha, N. Liu, N. Zou, In-processing modeling techniques for machine learning fairness: A survey, ACM Trans. Knowl. Discov. Data (2022).

[67] Z. Zeng, R. Islam, K.N. Keya, J. Foulds, Y. Song, S. Pan, Fair representation learning for heterogeneous information networks, in: AAAI Conference on Web and Social Media, 2021.

[68] O.D. Kose, Y. Shen, Fair node representation learning via adaptive data augmentation, 2022, arXiv preprint arXiv:2201.08549.

[69] M. Choudhary, C. Laclau, C. Largeron, A survey on fairness for machine learning on graphs, 2022, arXiv preprint arXiv:2205.05396.

[70] N. Navarin, L. Oneto, M. Donini, Learning deep fair graph neural networks, in: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2020.

[71] Y. Dong, J. Ma, C. Chen, J. Li, Fairness in graph mining: A survey, 2022, arXiv preprint arXiv:2204.09888.

[72] T. Rahman, B. Surma, M. Backes, Y. Zhang, Fairwalk: Towards fair graph embedding, in: International Joint Conference on Artificial Intelligence, 2019.

[73] A. Khajehnejad, M. Khajehnejad, M. Babaei, K.P. Gummadi, A. Weller, B. Mirzasoleiman, Crosswalk: Fairness-enhanced node representation learning, in: AAAI Conference on Artificial Intelligence, 2022.

[74] A. Saxena, G. Fletcher, M. Pechenizkiy, HM-EIICT: Fairness-aware link prediction in complex networks using community information, J. Comb. Optim. 44 (4) (2022) 2853–2870.

[75] S. Tsioutsiouliklis, E. Pitoura, P. Tsaparas, I. Kleftakis, N. Mamoulis, Fairness-aware pagerank, in: ACM Web Conference, 2021.

[76] S. Current, Y. He, S. Gurukar, S. Parthasarathy, Fairmod: Fair link prediction and recommendation via graph modification, 2022, arXiv preprint arXiv:2201.11596.

[77] I. Spinelli, S. Scardapane, A. Hussain, A. Uncini, Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning, IEEE Trans. Artif. Intell. 3 (3) (2021) 344–354.

[78] Z.S. Jalali, W. Wang, M. Kim, H. Raghavan, S. Soundarajan, On the information unfairness of social networks, in: SIAM International Conference on Data Mining, 2020.

[79] P. Li, Y. Wang, H. Zhao, P. Hong, H. Liu, On dyadic fairness: Exploring and mitigating bias in graph connections, in: International Conference on Learning Representations, 2021.

[80] Y. Dong, N. Liu, B. Jalaian, J. Li, Edits: Modeling and mitigating data bias for graph neural networks, in: ACM Web Conference, 2022.

[81] S. Yao, B. Huang, Beyond parity: Fairness objectives for collaborative filtering, in: Advances in Neural Information Processing Systems, 2017.

[82] M. Kleindessner, S. Samadi, P. Awasthi, J. Morgenstern, Guarantees for spectral clustering with fairness constraints, in: International Conference on Machine Learning, 2019.

[83] M. Buyl, T. De Bie, The kl-divergence between a graph model and its fair i-projection as a fairness regularizer, in: Machine Learning and Knowledge Discovery in Databases, 2021.

[84] J. Kang, J. He, R. Maciejewski, H. Tong, Inform: Individual fairness on graph mining, in: ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2020.

[85] C. Agarwal, H. Lakkaraju, M. Zitnik, Towards a unified framework for fair and stable graph representation learning, in: Uncertainty in Artificial Intelligence, 2021.

[86] Y. Dong, J. Kang, H. Tong, J. Li, Individual fairness for graph neural networks: A ranking based approach, in: ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021.

[87] A. Bose, W. Hamilton, Compositional fairness constraints for graph embeddings, in: International Conference on Machine Learning, 2019.

[88] M. Khajehnejad, A.A. Rezaei, M. Babaei, J. Hoffmann, M. Jalili, A. Weller, Adversarial graph embeddings for fair influence maximization over social networks, in: International Joint Conference on Artificial Intelligence, 2020.

[89] C. Wu, F. Wu, X. Wang, Y. Huang, X. Xie, Fairness-aware news recommendation with decomposed adversarial learning, in: AAAI Conference on Artificial Intelligence, 2021.

[90] E. Dai, S. Wang, Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information, in: ACM International Conference on Web Search and Data Mining, 2021.

[91] L. Wu, L. Chen, P. Shao, R. Hong, X. Wang, M. Wang, Learning fair representations for recommendation: A graph-based perspective, in: ACM Web Conference, 2021.

[92] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.

[93] J. Palowitch, B. Perozzi, Debiasing graph representations via metadata-orthogonal training, in: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2020.

[94] D. Dua, C. Graff, UCI machine learning repository, 2017, http://archive.ics.uci.edu/ml.

[95] I.C. Yeh, C. Lien, The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients, Expert Syst. Appl. 36 (2) (2009) 2473–2480.

[96] K.L. Jordan, T.L. Freiburger, The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length, Ethn. Crim. Justice 13 (3) (2015) 179–196.

[97] L. Takac, M. Zabovsky, Data analysis in public social networks, in: International Scientific Conference and International Workshop Present Day Trends of Innovations, 2012.

[98] J. Leskovec, J. Mcauley, Learning to discover social circles in ego networks, in: Neural Information Processing Systems, 2012.

[99] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

[100] M. Donini, L. Oneto, S. Ben-David, J.S. Shawe-Taylor, M. Pontil, Empirical risk minimization under fairness constraints, in: Neural Information Processing Systems, 2018.

[101] A. Micheli, Neural network for graphs: A contextual constructive approach, IEEE Trans. Neural Netw. 20 (3) (2009) 498–511.

[102] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2008) 61–80.

[103] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in: Neural Information Processing Systems, 2017.

[104] C. Morris, M. Ritzert, M. Fey, W.L. Hamilton, J.E. Lenssen, G. Rattan, M. Grohe, Weisfeiler and leman go neural: Higher-order graph neural networks, in: AAAI Conference on Artificial Intelligence, 2019.

[105] Y. Li, D. Tarlow, M. Brockschmidt, R. Zemel, Gated graph sequence neural networks, in: International Conference on Learning Representations, 2016.

[106] J. Gasteiger, A. Bojchevski, S. Günnemann, Predict then propagate: Graph neural networks meet personalized pagerank, in: International Conference on Learning Representations, 2019.

[107] M. Chen, Z. Wei, Z. Huang, B. Ding, Y. Li, Simple and deep graph convolutional networks, in: International Conference on Machine Learning, 2020.

[108] C. Dwork, N. Immorlica, A.T. Kalai, M. Leiserson, Decoupled classifiers for group-fair and efficient machine learning, in: Conference on Fairness, Accountability and Transparency, 2018.

[109] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a" siamese" time delay neural network, in: Neural Information Processing Systems, 1993.

[110] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Innovations in Theoretical Computer Science Conference, 2012.

[111] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, 2015.

[112] C. Louizos, K. Swersky, Y. Li, M. Welling, R. Zemel, The variational fair autoencoder, in: International Conference on Learning Representations, 2016.

[113] L. Oneto, M. Donini, G. Luise, C. Ciliberto, A. Maurer, M. Pontil, Exploiting mmd and sinkhorn divergences for fair and transferable representation learning, in: Neural Information Processing Systems, 2020.

[114] E. Creager, D. Madras, J. Jacobsen, M. Weis, K. Swersky, T. Pitassi, R. Zemel, Flexibly fair representation learning by disentanglement, in: International Conference on Machine Learning, 2019.

[115] A. Amini, A.P. Soleimany, W. Schwarting, S.N. Bhatia, D. Rus, Uncovering and mitigating algorithmic bias through learned latent structure, in: AAAI/ACM Conference on AI, Ethics, and Society, 2019.

[116] D. Moyer, S. Gao, R. Brekelmans, A. Galstyan, G. Ver Steeg, Invariant representations without adversarial training, in: Neural Information Processing Systems, 2018.

[117] M. Naser, A.H. Alavi, Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences, Archit. Struct. Constr. (2021) 1–19.

[118] E. Pariser, The Filter Bubble: What the Internet is Hiding from You, penguin UK, 2011.

[119] T.T. Nguyen, P. Hui, F.M. Harper, L. Terveen, J.A. Konstan, Exploring the filter bubble: the effect of using recommender systems on content diversity, in: International Conference on World Wide Web, 2014.

[120] E. Bozdag, J. Van Den Hoven, Breaking the filter bubble: democracy and design, Ethics Inf. Technol. 17 (2015) 249–265.

[121] Y. Dong, J. Ma, S. Wang, C. Chen, J. Li, Fairness in graph mining: A survey, IEEE Trans. Knowl. Data Eng. (2023).

[122] E. Dai, T. Zhao, H. Zhu, J. Xu, Z. Guo, H. Liu, J. Tang, S. Wang, A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability, 2022, arXiv preprint arXiv:2204.08570.

[123] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. De Vito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Neural Information Processing Systems, 2019.

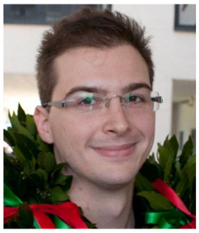
[124] S. Mitchell, E. Potash, S. Barocas, A. D'Amour, K. Lum, Algorithmic fairness: Choices, assumptions, and definitions, Annu. Rev. Stat. Appl. 8 (2021) 141–163.

[125] D. Pessach, E. Shmueli, Algorithmic fairness, 2020, arXiv preprint arXiv:2001.09784.

[126] F. Errica, M. Podda, D. Bacciu, A. Micheli, A fair comparison of graph neural networks for graph classification, in: 8th International Conference on Learning Representations, ICLR, 2020.

Danilo Franco was born and raised in the beautiful province of Genoa, Italy in 1993. He graduated in computer science and obtained a M.Sc. in Data Science and Engineering from the University of Genoa with the master thesis "Algorithmic Fairness: Learning Fair Representation" where he explored and implemented regularization methods for training face-recognition neural networks which aim at inherently satisfying fairness measures. He is currently pursuing a Ph.D. on trustworthiness in AI where he studies methods for horizontally delivering fair robust private, and explainable ML applications.

Vincenzo Stefano D'Amato was born in Genoa, Italy, in 1992. He received his B.Sc. and M.Sc. in Computer Science at the University of Genoa, Italy, respectively, in 2016 and 2019. In 2023, he received his Ph.D. from the same university in Computer Science and System Engineering with the thesis "Deep Multi-Temporal Scale Networks for Human Motion". He has been involved in the FET Proactive H2020 project called EnTimeMent. Currently, he works as postdoctoral researcher at the University of Genoa.



Luca Pasa obtained his master's degree in Computer Science from the University of Padova (2013). In the same university, Luca received his Ph.D. in Mathematical Sciences (curriculum Computer Science) in march 2017 under the supervision of prof. Alessandro Sperduti. From July 2017 to June 2019, he worked as postdoctoral researcher at the Center of Translational Neurophysiology Speech and Communication (CTNSC) (Istituto Italiano di Tecnologia - IIT), under the supervision of Dott. Leonardo Badino. Then, from July 2019 to December 2021 he continued his research activity as postdoctoral researcher at the University of Padova (Department of Mathematics / Human Inspired Technologies Research Center). Since January 2022, he has been assistant professor (RTDa) at the Department of Mathematics (University of Padova). His main research interests lie in the field of Machine Learning, including Deep Learning, Computational Neuroscience, and Automatic Speech Recognition. Currently, his research activity is focused on the application of Deep Learning methods on structured domains (sequences, trees, and graphs). Luca has co-authored several research papers published in international refereed journals and conference proceedings and he has been actively involved in organizing several special sessions at international machine learning conferences. He is a member of the IEEE Task Force on Deep Learning and the Italian Association for Artificial Intelligence (AIxIA).

Nicolò, Navarin is a tenure-track assistant professor in computer science at the Department of Mathematics "Tullio Levi-Civita", University of Padua, Italy. He got his Ph.D. in computer science from the University of Bologna, Italy, in 2014. He has been a visiting Researcher at the University of Freiburg, Germany and at the Università della Svizzera Italiana, Lugano, Switzerland. He has been a research fellow at the University of Nottingham, UK and at the University of Padua. His research interests lie in the field of machine learning, including kernel methods and neural networks for structured data, online and continual learning, and trustworthy ML. Dr. Navarin has been serving as PC member in major machine learning conferences, and he has been actively involved in the organization of several special sessions (ESANN, WCCI, IJCNN) and conferences (INNS Big Data and Deep Learning 2019, International Conference on Process Mining 2020, IEEE Symposium Series in Computational Intelligence 2021, IEEE World Congress on Computational Intelligence 2022). He is an associate editor for the journals Evolving Systems (Springer) and Neurocomputing (Elsevier), and an editorial board member for Intelligenza Artificiale (AIxIA, IOS Press). He is a member of IEEE Computational Intelligence Society, IEEE Task Force on Deep learning, IEEE Task Force on Learning from Structured Data.

Luca Oneto was born in Rapallo, Italy in 1986. He received his B.Sc. and M.Sc. in Electronic Engineering at the University of Genoa, Italy respectively in 2008 and 2010. In 2014 he received his Ph.D. from the same university in the School of Sciences and Technologies for Knowledge and Information Retrieval with the thesis "Learning Based On Empirical Data". In 2017 he obtained the Italian National Scientific Qualification for the role of Associate Professor in Computer Engineering and in 2018 he obtained the one in Computer Science. He worked as Assistant Professor in Computer Engineering at University of Genoa from 2016 to 2019. In 2018 he was co-funder of the spin-off ZenaByte s.r.l. In 2019 he obtained the Italian National Scientific Qualification for the role of Full Professor in Computer Science and Computer Engineering. In 2019 he became Associate Professor in Computer Science at University of Pisa and currently is Associate Professor in Computer Engineering at University of Genoa. He has been involved in several H2020 projects (S2RJU, ICT, DS) and he has been awarded with the Amazon AWS Machine Learning and Somalvico (best Italian young AI researcher) Awards. His first main topic of research is the Statistical Learning Theory with particular focus on the theoretical aspects of the problems of (Semi) Supervised Model Selection and Error Estimation. His second main topic of research is Data Science with particular reference to the problem of Trustworthy AI and the solution of real world problems by exploiting and improving the most recent Learning Algorithms and Theoretical Results in the fields of Machine Learning and Data Mining.