Machine Learning (2023) 112:3051–3076
https://doi.org/10.1007/s10994-023-06342-9

*Fair*SwiRL: fair semi-supervised classification with representation learning

Shuyi Yang[1,4] · Mattia Cerrato[2] · Dino Ienco[3] · Ruggero G. Pensa[1] · Roberto Esposito[1]

Received: 22 November 2022 / Revised: 20 March 2023 / Accepted: 21 April 2023 /
Published online: 27 June 2023

Abstract

Semi-supervised learning has shown its potential in many real-world applications where only few labeled examples are available. However, when some fairness constraints need to be satisfied, semi-supervised classification models often struggle as they are required to cope with the lack of sufficient information for predicting the target variable while forgetting its relationships with any sensitive and potentially discriminatory attribute. To address this issue, we propose a fair semi-supervised representation learning architecture that leads to fair and accurate classification results even in very challenging scenarios with few labeled (but biased) instances. We show experimentally that our model can be easily adopted in very general settings, as the learned representations may be employed to train any supervised classifier. Moreover, when applied to several synthetic and real-world datasets, our method is competitive with state-of-the-art fair semi-supervised approaches.

Keywords Semi-supervised autoencoder · Fairness · Deep neural networks

Editors: Fabio Vitale, Tania Cerquitelli, Marcello Restell, and Charalampos Tsourakakis.

✉ Ruggero G. Pensa
ruggero.pensa@unito.it

Shuyi Yang
shuyi.yang@unito.it

Mattia Cerrato
mcerrato@uni-mainz.de

Dino Ienco
dino.ienco@inrae.fr

Roberto Esposito
roberto.esposito@unito.it

[1] Department of Computer Science, University of Turin, Turin, Italy

[2] Institut für Informatik, Johannes Gutenberg-Universität Mainz, Mainz, Germany

[3] UMR TETIS-LIRMM, INRAE Montpellier, Montpellier, France

[4] Data Science & Artificial Intelligence, Intesa Sanpaolo, Turin, Italy

1 Introduction

In an ideal scenario, modern supervised machine learning algorithms are able to get the most from all available training data instances so to accomplish the task at hand, be it classification, regression or ranking. Unfortunately, in real-world applications, this is almost never the case due to several reasons, among the others, the necessity to access huge amounts of labeled instances to train supervised algorithms. Labels often require cost-intensive collection procedures and huge efforts from human experts, especially in challenging domains such as medical and financial ones. Semi-supervised learning precisely addresses this issue by considering, together with a small amount of labeled information, unlabeled instances during the learning process, leveraging the so-called *smoothness* and *cluster* assumptions: if two data instances are close to each other or belong to the same cluster in the input distribution, then they are likely to belong to the same class (Chapelle et al., 2006; van Engelen & Hoos, 2020). If the few available labels are of good quality, and clusters are well separated, unlabeled instances contribute to improve the accuracy significantly. Nonetheless, the labels might contain biases against certain groups. This might be an effect of historical explicit discriminations which may be reflected in a human expert's beliefs, data scarcity or even biases in the data generation/measuring process itself (Barocas et al., 2019). Beyond ethical issues, fairness in machine learning models is becoming an increasingly pressing concern at a practical level as regulators and the general public become more aware of the potential for automatic discrimination. The EU Commission's AI Legal framework proposal,[1] for instance, would require practitioners to "[...] minimise the risk of unfair biases embedded in the model [...]".

If the lack of labeled training instances and fairness are complex problems individually, avoiding biases in a semi-supervised learning scenario is even more challenging. In a worst-case scenario, the few available labeled instances could be all or almost all associated to unfair sources, thus leading to very biased results or preventing any debiasing process. On the other hand, unlabeled instances do not carry any explicit bias and could be useful for driving the learning algorithm towards a fairer model. Despite its clear potential, fair semi-supervised learning has not been deeply investigated. The few existing approaches are based on preprocessing strategies that seek to extract fair training datasets by leveraging unlabeled instances (Zhang et al., 2022; Chakraborty et al., 2021). However, to the best of our knowledge, no representation learning method specifically designed for semi-supervised learning with fairness constraints has been proposed so far.

Representation learning allows one to automatically construct a new feature space that better captures the different factors of variation behind the data (Bengio et al., 2013). Such new representation can then be used to feed any machine learning algorithms, including supervised and unsupervised ones. Autoencoders are among the most popular representation learning methods and both fair (Madras et al., 2018) and semi-supervised (Gogna & Majumdar, 2016) versions of them have been proposed. In this paper, we propose a fair semi-supervised autoencoder that leads to fair and accurate classification results even in very challenging scenarios with few labeled (but biased) instances. The classic auto-encoding architecture (Hinton & Zemel, 1993) is enhanced with two components. One is trained to classify instances and employs the available labeled training instances. The second is a debiasing component that removes as much information as possible about the sensitive

[1] https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai.

attribute, in an adversarial fashion. Additionally, our model is inductive and, as such, it can be used to classify unseen examples as well. We name our contribution *FairSwiRL*, which stands for **Fair S**emi-supervised classification **wi**th **R**epresentation **L**earning.

Through an extensive experimental validation on synthetic and real world datasets, we show that the representations learned by *FairSwiRL* as the training data for different classifiers leads to reasonably accurate models while respecting the fairness constraint, even when very few labeled examples are available at training time. Moreover, our method compares favorably to other state-of-the-art fair semi-supervised classification approaches. To the best of our knowledge, our contribution is the first one providing a comparison of preprocessing and representation learning approaches in the Semi Supervised Learning (SSL) setting under fairness constraints.

The remainder of the paper is organized as follows: Sect. 2 provides a brief review of the relevant literature from semi-supervised learning and fair representation learning; Sect. 3 formalizes the problem setting; Sect. 4 presents our method *FairSwiRL*; Sect. 5 describes the datasets used in the experiments; Sect. 6 presents and discusses the results of the experiments; Sect. 7 concludes the paper and discusses future work.

2 Related works

In this section, we explore the main results in the semi-supervised learning and fair machine learning literature, with a special focus on representation learning.

2.1 Semi-supervised learning

Semi-supervised learning (SSL) algorithms are aimed at computing classification models by leveraging (a small amount of) labeled and (a vast amount of) unlabeled data (Chapelle et al., 2006). Due to the wide range of real-world applications these methods can fit, SSL has been a hot research topic in machine learning in the last decade (van Engelen & Hoos, 2020). Recent developments in SSL involve the use of deep neural network models through the lens of generative (Springenberg, 2016), consistency-regularization (Rasmus et al., 2015), geometric-based (Hu et al., 2019) and pseudo-labeling (Cheng et al., 2016) methods. The majority of these approaches are devoted to signal data, like images or time series, while only few deep learning methods are proposed for tabular information.

Furthermore, SSL algorithms can be categorized into inductive and transductive methods, depending on whether they are able to build a general model or not for the underlying data. Transductive methods are mostly based on graphs, with (dis)similarity between nodes representing the weight of the graph edges. In these approaches, once the graph has been constructed, an inference method is applied to make predictions on unlabeled nodes (Yamaguchi et al., 2016).

Inductive methods, on the other hand, can build classification models that can predict the class of examples unseen during the training stage (Yang et al., 2021). Since the SSL setting assumes that both labeled and unlabeled data are available at training time, many research works focus on combining supervised and unsupervised paradigms (van Engelen & Hoos, 2020) to obtain the final classification model. Semi-supervised autoencoders (SSAE) have been recently investigated (Gogna & Majumdar, 2016; Le et al., 2018) as a similar proposal in the representation learning scenario. SSAEs combine the benefits of unsupervised learning (autoencoders) with discriminative approaches that exploit the small amount of labels providing supervision. Even though an autoencoder is originally designed to perform an unsupervised reconstruction task, in its semi-supervised version an extra prediction layer is attached to the bottleneck layer to perform class predictions, in a multi-task setting.

2.2 Fair representation learning

Concerns about *fairness* in machine learning have been raised since the 90 s, when Friedman and Nissenbaum (1996) reasoned that automatic decision-making performed by "machines" could pose a concrete risk of discrimination against historically underprivileged groups. In more recent years, various authors have proposed different definitions that deal with the notion of a protected (or underprivileged) group. Individuals belong to a protected group if their innate characteristics have been the subject of systemic, explicit discrimination in the past.

At a basic level, a "fair" machine learning model assigns positive outcomes in a balanced fashion to underprivileged and privileged groups (the fair model is then said to enforce *statistical parity*). We refer the reader to Mehrabi et al. (2021) and Zafar et al. (2017) for in-depth discussions of other actionable fairness definitions and metrics.

Methodologies to constrain statistical learning algorithms for fairness may be divided into two broad classes. *Preprocessing* approaches modify the training data so to balance, for instance, positive outcomes between groups (Kamiran & Calders, 2009); *regularization* approaches, on the other hand, insert a regularization term in the objective function which measures the fairness of the model. Thus, it is possible to learn models which find different trade-offs between utility and fairness depending on the strength of the regularization. The fair representation learning task owes its name to Zemel et al. (2013) which employs probabilistic modeling. Since then, many authors have employed neural networks as the base learning algorithm of choice, pairing them with different debiasing techniques: among others, Madras et al. (2018); Xie et al. (2017); Zhang et al. (2018) employ adversarial training; Oneto et al. (2020) have leveraged different probabilistic divergences which may be employed in representation space such as Gretton et al. Maximum Mean Discrepancy (Gretton et al., 2012); a variational approach has been presented by Louizos et al. (2016) and dubbed the Variational Fair Autoencoder.

Our framework for fair semi-supervised representation learning leverages adversarial learning and is fully described in Sect. 4. In previous literature, several approaches that leverage unlabeled data to obtain fair results [for instance, FESF (Zhang et al., 2022) and FairSSL (Chakraborty et al., 2021)] have employed a preprocessing strategy. In short, these strategies train the model on a "fair subset" of the original data (Zhang et al., 2022), although it is also possible to perform pseudo-labeling over the remaining data (Chakraborty et al., 2021). These techniques bear some resemblance to well-known preprocessing strategies in fully-supervised fair classification (Kamiran & Calders, 2009). As far as fair representation learning algorithms are concerned, Louizos et al. Variational Fair Autoencoder (VFAE) (Louizos et al., 2016) was originally tested in the fully-supervised setting but may be also applied to SSL as long as a classification layer is available. However, it is worthwhile to mention that the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) "fair regularization" term employed in VFAE is only usable for binary-valued sensitive attributes (Xie et al., 2017). Our approach, on the other hand, is to learn an auxiliary classifier which predicts the sensitive attribute. Its output dimension may be adapted depending on how many values the sensitive attribute takes and, as such, it does not suffer from the same limitation as VFAE. We provide an experimental comparison between VFAE and *Fair*`SwiRL` in Sect. 6.

3 Problem setting

In this section, we describe the problem of semi-supervised fair classification. In this scenario, one seeks to learn a classifier by using both labeled instances and unlabeled ones. Moreover, we would also like to satisfy a fairness constraint with respect to a given sensitive attribute, i.e. a feature representing an individual's membership in an historically underprivileged group. The rationale here is to avoid potentially discriminatory decisions by the learned classifier (Barocas et al., 2019).

We denote with $(\mathbf{X}_l, \mathbf{s}_l, \mathbf{y}_l)$ the features, the sensitive attributes, and the target variables of labeled instances, with $(\mathbf{X}_u, \mathbf{s}_u)$ the features and the sensitive attributes of unlabeled instances. In semi-supervised fair classification, we seek to learn a classifier which is able to leverage both $(\mathbf{X}_l, \mathbf{s}_l, \mathbf{y}_l)$ and $(\mathbf{X}_u, \mathbf{s}_u)$ such that the predictions of target variable $\mathbf{y}_t$ computed on an unseen test set $(\mathbf{X}_t, \mathbf{s}_t)$ are accurate and satisfy some fairness constraints. In the following, capital non-bold letters will be used to denote random variables (e.g., X, Y, S will denote the stochastic variables associated with examples, labels and sensitive attributes).

As a fairness constraint, we here consider independence, or *statistical parity* [SP (Castelnovo et al., 2021; Barocas et al., 2019)]. Thus, we require that the probability of assigning a positive outcome to an individual is independent of the sensitive information S. Formally, we require that:

$$P(\hat{Y} = 1 \mid S = 0) = P(\hat{Y} = 1 \mid S = 1), \tag{1}$$

where $\hat{Y}$ is the stochastic variable associated with the prediction of the model. As a way to quantify how far we are from the statistical parity, we consider the statistical absolute difference (SAD) measure, (Bellamy et al., 2018):

$$\text{SAD} = \left| \mathbb{E}[\hat{Y} \mid S = 0] - \mathbb{E}[\hat{Y} \mid S = 1] \right|. \tag{2}$$

The lower the SAD, the better it is, with statistical parity at SAD = 0. We note here that removing the sensitive attributes $\mathbf{s}_l$ and $\mathbf{s}_u$ is usually insufficient to achieve statistical parity as some information about S may be present in the remaining variables $\mathbf{X}_l$ and $\mathbf{X}_u$ or the labels $\mathbf{y}_l$. Thus, *Fair*`SwiRL` seeks to optimize the SAD metric by learning a debiased representation of the original data - i.e. a new representation of the data X in which all information about S has been removed. After the debiasing, any classifier trained on the latent representation will be able to achieve low SAD values without being specifically optimized for this metric.

We now move onto discussing how unlabeled instances can, in principle, be useful in improving the aforementioned debiasing process. We provide a constructive example via an analysis of the toy dataset presented in Table 1.

In the given example, s represents the sensitive attribute, x_1 and x_2 are two independent variables while x_3 is computed from x_1 and x_2 with the formula $x_1 \bigotimes x_2$, where $\bigotimes$ is an AND if $s = 0$ and OR otherwise:

Table 1 Toy Dataset. s represents the sensitive attribute, x_1, x_2 and x_3 are features and y is the target variable

s	x_1	x_2	x_3	y	$\hat{y}_{\mathrm{SSL}}$	$\hat{y}$
0	0	0	0	0	0	0
0	0	1	0	0	0	0
1	1	0	1	1	1	1
1	1	1	1	1	1	1
0	1	0	0	unknown (0)	0	1
0	1	1	1	unknown (1)	1	1
1	0	0	0	unknown (0)	0	0
1	0	1	1	unknown (1)	1	0

The column $\hat{y}_{\mathrm{SSL}}$ reports the predictions made by a semi-supervised algorithm while the column $\hat{y}$ represents the predictions provided by a fair semi-supervised classifier. The first four rows are labeled instances while the last four rows are unlabeled (the number inside the bracket represents the unobserved label)

$$x_1 \bigotimes x_2 = \begin{cases} x_1 \wedge x_2 & \text{if } s = 0 \\ x_1 \vee x_2 & \text{if } s = 1 \end{cases}. \tag{3}$$

It follows that there is a functional relationship between s and x_3, which makes this latter variable a potential source of bias. The target variable y is computed as $y = \mathbb{I}\left[x_1 + x_2 + x_3 > 1\right]$, where $\mathbb{I}[\cdot]$ is the indicator function. We assume that the data generation process described above is unknown and that we are interested in learning a classifier from the data reported in Table 1. Please note that the SAD value computed on s and y is 0.5; the dataset is, thus, unfair (as far as the statistical parity metric is concerned), but we would like to obtain a fair classifier anyway.

Firstly, if we examine the toy dataset under a semi-supervised setting without any constraint on fairness, we would likely consider the distribution of both labeled and unlabeled instances and conclude that $s + x_1 + x_2 + x_3 = 2$ is a good candidate as a separation hyperplane. The predictions induced by this choice of hyperplane are reported in the column $\hat{y}_{\mathrm{SSL}}$ in Table 1. These predictions have 100% accuracy but are as unfair as the original target values y.

By introducing the fairness constraint, we would like to remove information on the sensitive attribute. If we consider the labeled instances only (first four rows), we see that the sensitive attribute s, the attributes x_1 and x_3 and the target variable y are highly (actually perfectly) correlated. To remove the bias introduced by these variables we might be tempted to remove s, x_1, x_3 from the dataset and then train a classifier only on x_2 to predict y. This classifier, however, would be no better than random guessing.

If we repeat the analysis while including the unlabeled instances, we can verify that there is no correlation between s and x_1, or even between s and x_2, while s and x_3 do show some correlations. To debias the dataset, we can now improve on our previous attempt and remove only s and x_3. In this latter case, a classifier should then learn to ignore the x_3 variable, as this is not a good predictor of y. It follows that the only reasonable prediction model available is $\hat{y} = x_1$, as shown in the second-to-last column of the Table 1. This debiased classifier has an accuracy of 75% and a SAD value equal to 0. Hence, the accuracy is decreased with respect to the performance of $\hat{y}_{\mathrm{SSL}}$ but the SAD value is improved. We conclude that employing unlabeled instances during the learning process can dramatically improve a classifier in both accuracy and fairness terms.

To take advantage of this property, we introduce *FairSwiRL*. Our proposal is a semi-supervised representation learning method which is able to leverage the unlabeled examples and obtain a less biased representation of the data. We describe our contribution in detail in the next section.

4 Fair semi-supervised classification with representation learning

In our problem setting, label scarcity is paired with fairness constraints. To face these issues, we design an inductive and fair semi-supervised model which leverages representation learning techniques. We employ an auto-encoding architecture (Hinton & Zemel, 1993) which is able to leverage both labeled $\mathbf{X}_l$ and unlabeled data $\mathbf{X}_u$. This architecture maps the original data $\mathbf{X} = \{\mathbf{X}_l \cup \mathbf{X}_u\}$ into a compact representation $\mathbf{z}$ via a series of fully-connected layers, a process which is commonly referred to as encoding. In the following we will refer to this section of our model as the encoder $E_{\theta_e}(\mathbf{x})$, where θ_e are the learnable parameters for the fully connected layers, and the learned latent representation as $\mathbf{z}$. The dimension of this representation is a hyperparameter for the algorithm and may be set up to be lower than $\mathbf{x}$, therefore compressing information. Another series of fully connected layers, a decoder $D_{\theta_d}(\mathbf{z})$, then maps back the latent representation into an approximation $\hat{\mathbf{x}}$ of the original data. This architecture may be learned via gradient descent over a *reconstruction loss* $\mathcal{L}_{\text{rec}}$ which is defined as follows:

$$\mathcal{L}_{\text{rec}}(E_{\theta_e}, D_{\theta_d}) = \sum_{\mathbf{x}_i \in (\mathbf{X}_u \cup \mathbf{X}_l)} \|\mathbf{x}_i - D_{\theta_d}(E_{\theta_e}(\mathbf{x}_i))\|^2. \tag{4}$$

In the semi-supervised setting there is also the additional opportunity to exploit the limited amount of class information provided by the labeled examples $\mathbf{x}_l \in \mathbf{X}_l$. Exploiting this is paramount to obtain representations that are also useful for classification. Therefore, we employ an auxiliary network $C_{\theta_c}(\mathbf{z}_l)$ and train it on the representations $\mathbf{z}_l = E_{\theta_e}(\mathbf{x}_l)$ for which label data are available. As is commonly done in classification with neural networks, we exploit the cross entropy loss to drive the training of this component of the network:

$$\mathcal{L}_{\text{cla}}(E_{\theta_e}, C_{\theta_c}) = \sum_{\mathbf{x}_i \in \mathbf{X}_l} \left(-\sum_{j=1}^{|\mathcal{Y}|} y_{i,j} \cdot log(C_{\theta_c}(E_{\theta_e}(\mathbf{x}_i)))_j \right), \tag{5}$$

where the notation $y_{i,j}$ assumes the one-hot encoding of the class j for the labeled example $\mathbf{x}_i \in \mathbf{X}_l$, and $\mathcal{Y}$ is the set of possible labels (numbered from 1 to $|\mathcal{Y}|$). Lastly, we employ a component which is able to remove information about the sensitive attribute $\mathbf{s}$ from the obtained representations $\mathbf{z}$. This is possible by training another auxiliary classifier which predicts the sensitive attribute from the representation, which we will refer to in the following as F_{θ_f}. Once again, this may be trained via cross-entropy, albeit over both labeled and unlabeled examples, as we assume that sensitive information is available for all data samples:

$$\mathcal{L}_{\text{fair}}(E_{\theta_e}, F_{\theta_f}) = \sum_{\mathbf{x}_i \in \mathbf{X}_l \cup \mathbf{X}_u} \left(-\sum_{j=1}^{|\mathcal{S}|} s_{i,j} \cdot log(F_{\theta_f}(E_{\theta_e}(\mathbf{x}_i)))_j \right), \tag{6}$$

where $s_{i,j}$ is the jth component of the one-hot-encoded $\mathbf{s}$ vector and $\mathcal{S}$ is the set of possible sensible values. Formally, the overall training objective for our method is as follows:

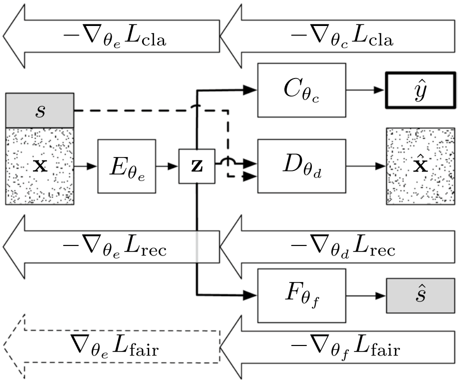

$-\nabla_{\theta_e} L_{\text{cla}}$
$-\nabla_{\theta_c} L_{\text{cla}}$
C_{θ_c}
$\hat{y}$
s
$\mathbf{x}$
E_{θ_e}
$\mathbf{z}$
D_{θ_d}
$\hat{\mathbf{x}}$
$-\nabla_{\theta_e} L_{\text{rec}}$
$-\nabla_{\theta_d} L_{\text{rec}}$
F_{θ_f}
$\hat{s}$
$\nabla_{\theta_e} L_{\text{fair}}$
$-\nabla_{\theta_f} L_{\text{fair}}$

Fig. 1 Fair Semi-supervised with Representation Learning (*Fair*`SwiRL`). The weights of the encoder are updated to reduce the target variable classification loss L_{cla} and the reconstruction loss L_{rec}, but the gradient goes to the opposite direction of minimizing the sensitive attribute classification loss L_{fair}

$$\mathcal{L}_{\text{tot}}(\theta_e, \theta_d, \theta_c, \theta_f) = w_{\text{cla}}\mathcal{L}_{\text{cla}}(\theta_e, \theta_c) + w_{\text{rec}}\mathcal{L}_{\text{rec}}(\theta_e, \theta_d) - w_{\text{fair}}\mathcal{L}_{\text{fair}}(\theta_e, \theta_f), \tag{7}$$

where $w_{\text{fair}}, w_{\text{cla}}, w_{\text{rec}}$ are hyperparameters which may be picked to control the fairness/classification/reconstruction trade-off. The networks are pitted against one another in an *adversarial* fashion. This implies setting up a min-max game where networks $E_{\theta_e}, D_{\theta_d}$ and C_{θ_c} are employed to respectively minimize the reconstruction and classification losses; the network F_{θ_f}, on the other hand, should have maximal loss, i.e., it should be impossible to reconstruct information about the sensitive attribute s from the learned representations z. This leads to the following multi-objective optimization problem:

$$\hat{\theta}_e, \hat{\theta}_d, \hat{\theta}_c, \hat{\theta}_f = \arg\left\{ \min_{\theta_e,\theta_d,\theta_c} \left[w_{\text{cla}}\mathcal{L}_{\text{cla}}(\theta_e, \theta_c) + w_{\text{rec}}\mathcal{L}_{\text{rec}}(\theta_e, \theta_d) - w_{\text{fair}} \min_{\theta_f} \mathcal{L}_{\text{fair}}(\theta_e, \theta_f) \right] \right\}. \tag{8}$$

The equilibrium point in the above problem can be found via *gradient reversal* (Ganin et al., 2016), a procedure where the gradient information from a sub-network is multiplied by -1 when backpropagating into the main architecture. Specifically, we invert the gradient from F_{θ_f} when updating the parameters in our encoder E_{θ_e}. A graphical representation of this procedure may be found in Fig. 1.

In practice, we employ stochastic gradient descent and apply the following parameter updates after each mini-batch:

$$\begin{cases} \theta_c \leftarrow \theta_c - \gamma_c \nabla_{\theta_c} \mathcal{L}_{\text{cla}} \\ \theta_d \leftarrow \theta_d - \gamma_d \nabla_{\theta_d} \mathcal{L}_{\text{rec}} \\ \theta_f \leftarrow \theta_f - \gamma_f \nabla_{\theta_f} \mathcal{L}_{\text{fair}} \\ \theta_e \leftarrow \theta_e - \gamma_{ec} \nabla_{\theta_e} \mathcal{L}_{\text{cla}} - \gamma_{ed} \nabla_{\theta_e} \mathcal{L}_{\text{rec}} + \gamma_{ef} \nabla_{\theta_e} \mathcal{L}_{\text{fair}} \end{cases} \tag{9}$$

In summary, the proposed network (*Fair*`SwiRL`) is a fairness focused extension of the semi-supervised autoencoder. One core property of *Fair*`SwiRL` is that it leverages representation learning to obtain feature vectors which are both useful and fair. The obtained representations may then be used for further downstream tasks with no restriction on the employed model, allowing a practitioner to use the model that best fits the domain knowledge on the task or any business requirements. We show the flexibility of our approach in Sect. 6, where we report experimental results for different classifiers trained on *Fair*

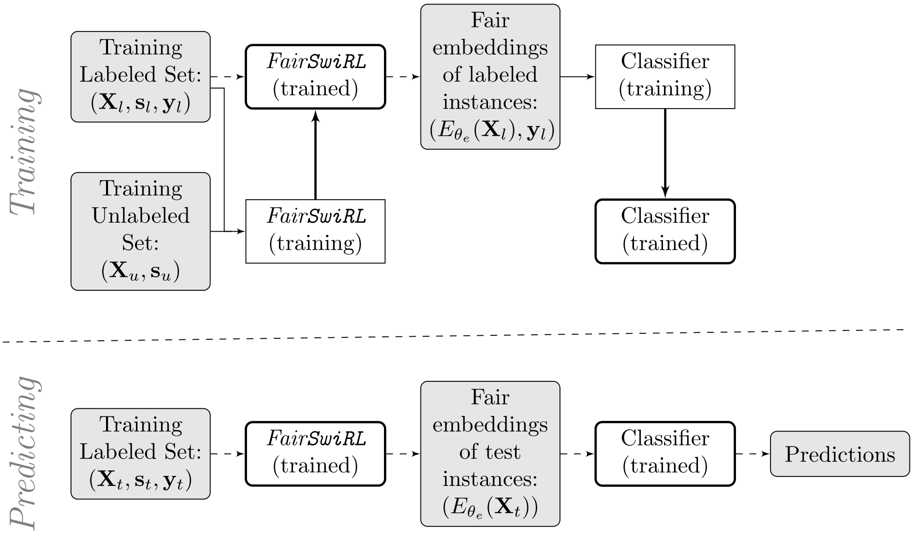

Training
Training Labeled Set: $(\mathbf{X}_l, \mathbf{s}_l, \mathbf{y}_l)$
FairSwiRL (trained)
Fair embeddings of labeled instances: $(E_{\theta_e}(\mathbf{X}_l), \mathbf{y}_l)$
Classifier (training)
Training Unlabeled Set: $(\mathbf{X}_u, \mathbf{s}_u)$
FairSwiRL (training)
Classifier (trained)
Predicting
Training Labeled Set: $(\mathbf{X}_t, \mathbf{s}_t, \mathbf{y}_t)$
FairSwiRL (trained)
Fair embeddings of test instances: $(E_{\theta_e}(\mathbf{X}_t))$
Classifier (trained)
Predictions

Fig. 2 *FairSwiRL* for a classification task. Labeled and unlabeled instances are used to train our *FairSwiRL* model. Once trained, the encoder of the network is used to compute the fair representations of labeled instances. These representations, with the original labels, are then used to train any classifier. In prediction, we apply sequentially the encoder of *FairSwiRL* and the classifier to make predictions on unseen data

Table 2 Datasets used during our experiments

Dataset	Instances	Original features	Post- processed features	Sensitive attribute	Target variable
SYNTHETIC	–	5	5	S	Y
ADULT	48–842	14	107	Sex	Income
BANK	45–211	17	61	Previous campaign	Subscription
CARD	30–000	23	23	Education	Default
COMPAS	6–127	53	18	Ethnicity	Criminal recidivism

SwiRL 's representations (see Fig. 2): Random Forest, K-Neighbors Classifier, Logistic Regression, Support Vector Machines and Neural Network.

5 Datasets

We experiment on one synthetic dataset and four real-world classification datasets (see Table 2 for summary statistics). The four real-world datasets have been extensively employed in papers dealing with fair classification and fair representation learning (Madras et al., 2018; Louizos et al., 2016). Furthermore, we designed a synthetic dataset in which the data generation process is known, providing us with a controlled experimental setup. In this dataset one has full control on the number of instances, the data generation process and the level of correlation (bias) between the sensitive attribute and the target variable.

5.1 Synthetic dataset

Let X_0, X_1, X_2, X_3 be four independent random variables uniformly distributed in the interval $(-1, 1)$. We will draw samples from these random variables to model features of a synthetic dataset that we will be using to study, in a controlled environment, the characteristics of the competing algorithms. The sensible attribute is modeled through an additional variable S that we will sample from a Bernoulli distribution with $p = \frac{1}{2}$. We note that S is independent of $X_0, \ldots, X_3$. However, we also define a "surrogate" sensitive attribute $S' = S + X_0$. S' is functionally related to X_0 and S and therefore a potential source of bias. This setup is similar to the motivating toy dataset introduced in Table 1. To model the target variable, we start by defining an intermediate random variable $Z = X_1 + S' + N$, where N is a noise term which we model using a normal distribution with parameters $\left(0, \frac{1}{2}\right)$. The target random variable Y is then defined as $Y = \mathbb{I}[Z > \mathbb{E}[Z]]$ (where $\mathbb{I}[\cdot]$ is the indicator function). Examples $(\mathbf{x}, y)$ in the synthetic dataset are realizations of the vector $((X_1, X_2, X_3, S', S), Y)$. It is worth noting that the variable X_0 is not directly observed, but is an important factor in the definition of Y. This fact, together with the noise introduced by N, makes it impossible to predict the target variable with perfect accuracy by training on finite realizations of the dataset. Also, since S' is correlated with S, fairness cannot be achieved only by getting rid of the sensitive variable s.

5.2 Real-world datasets

In our experimental study, we will use the following real-world benchmark datasets.

ADULT (also known as Census Income Dataset) (Dua & Graff, 2017) is an extraction of the 1994 US Census database performed by Barry Becker. It contains 14 attributes (numerical and categorical) and 48,842 instances. The target variable indicates whether a person's annual income exceeds \$50 000. The sensitive attribute is the sex.

BANK (Dua & Graff, 2017) contains data related to a phone call marketing campaign of a Portuguese bank. It has 17 attributes (numerical and categorical) and 45 211 instances. The target variable to predict is whether the client will subscribe a term deposit. The sensitive attribute is the outcome of the previous marketing campaign.

CARD (Dua & Graff, 2017) contains the data of credit cards clients in Taiwan. It has 23 attributes (numerical and categorical) and 30 000 instances. The binary target variable indicates the case of default payments. The sensitive attribute is the education.

COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) (Angwin et al., 2016) contains the data of people convicted of felonies in Florida. It has 53 attributes and 7 214 rows. We followed the preprocessing described by Angwin et al. (2016) and reduced the number of instances to 6 172. The binary target variable indicates the recidivism within 2 years. The sensitive attribute is a binary variable indicating whether the person is African-American or not.

6 Experiments

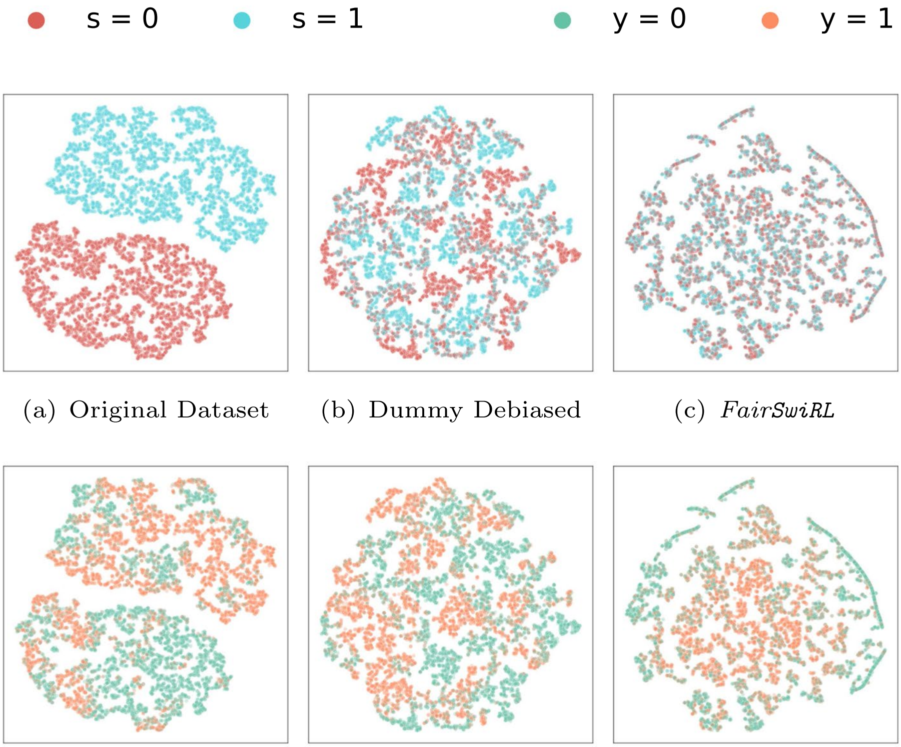
Our experimental efforts are focused on evaluating the representations learned by *Fair-*`SwiRL`. One advantage of fair representation learning is the ability to decouple the decision model from the representation model (McNamara et al., 2017). This is useful in practice as it adds flexibility to the overall methodology making it possible to choose which classifier to employ; furthermore, it is possible to investigate the representations themselves to understand whether sensitive information has been removed. This is common practice in fair representation learning and domain-invariant learning (Zemel et al., 2013; Xie et al., 2017; Ganin et al., 2016). In summary, in this section we aim to answer the following questions:

- *Q1* Is *Fair*`SwiRL` able to learn *fair* representations, i.e. feature vectors in which information about the sensitive attribute s is removed?
- *A1* Yes. Individuals belonging to different groups are mixed together in the representation space learned, and information about the sensitive attribute is therefore unrecoverable. We provide qualitative evidence in the form of a visualization experiment (Sect. 6.1) and quantitative evidence by training various classifiers on the learned representations and observing that they learn non-discriminative decisions (Sect. 6.2).
- *Q2* Are representations learned with *Fair*`SwiRL` *useful*, i.e., is it possible to employ them in classification tasks?
- *A2* Yes. We compute the Matthews Correlation Coefficient for different classifiers and observe good predictive power (Sect. 6.2). Furthermore, we observe that different classifiers, both linear and non-linear, have similar performance when trained on the representations learned by our method.
- *Q3* How does *Fair*`SwiRL` compare to other methods in the fair semi-supervised learning literature?
- *A3* Favorably. When analyzing the classification performances under fairness constraints, it is paramount to employ a trade-off analysis as it is done e.g., in multi-task learning. We show in Sect. 6.3 that *Fair*`SwiRL` paired with a random forest decision model is a good performer among fair semi-supervised competitors on four datasets out of five when assuming a balanced fairness/accuracy trade-off. Additionally, we employ a combined fairness+accuracy metric that weighs fairness as exponentially more important than accuracy. Here, we observe that *Fair*`SwiRL` is able to outperform the other algorithms or remain competitive on most of the datasets. Finally, we test our method in two extremely challenging scenarios with very few labeled instances (Sect. 6.4) and very biased labeled instances (Sect. 6.5). The results of these last experiments are still favorable if compared to competitor methods.

For the sake of reproducibility, all the details about the experiments are given in Appendix 1.

6.1 Visual inspection with t-SNE

We inspect the latent representations provided by *Fair*`SwiRL` through a qualitative assessment with t-SNE (Van der Maaten & Hinton, 2008). In Figs. 3 and 4, we report the 2D visualizations for the synthetic and the adult datasets introduced in Sects. 5.1 and 5.2 as learned by the t-SNE algorithm. Plots on the left are obtained from the original data; the center plots employ the same original data after removing only the sensitive attribute; the plots on the right display the latent representations learned by *Fair*`SwiRL` on $n_l = 100$ labeled instances and $n_u = 10000$ unlabeled instances. We plot 10000 data samples from the test set. Colors in the top row are assigned according to the values of the sensitive



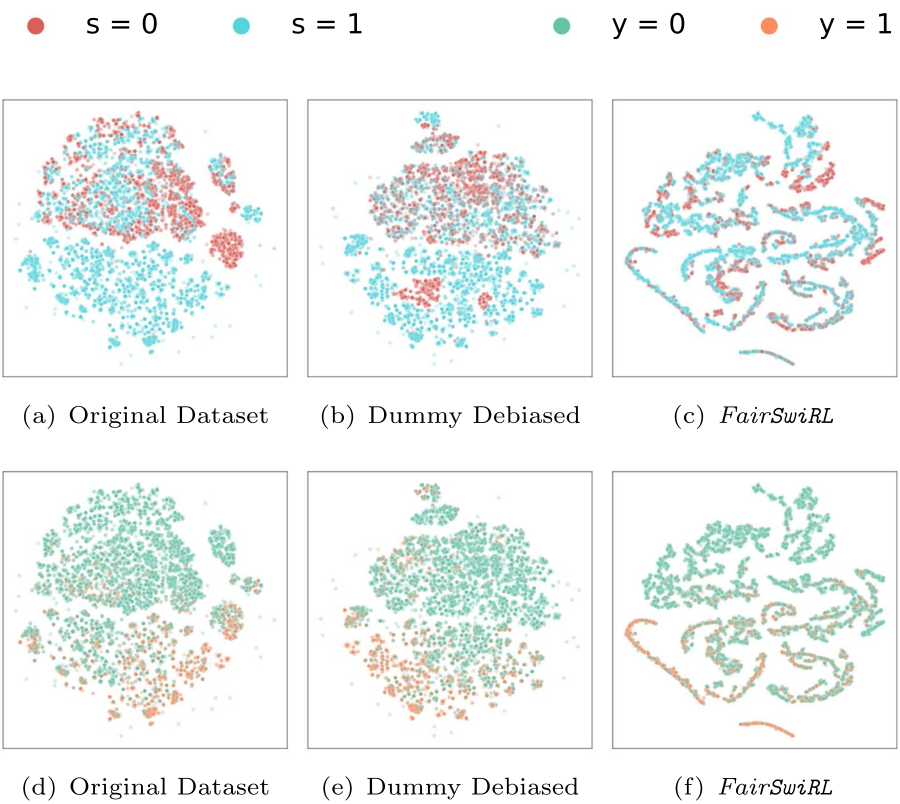
(a) Original Dataset (b) Dummy Debiased (c) *FairSwiRL*

(d) Original Dataset (e) Dummy Debiased (f) *FairSwiRL*

Fig. 3 t-SNE representations (default parameters: components = 2, perplexity = 30, early exaggeration = 12) of the SYNTHETIC dataset. From left to right: representations obtained from the original dataset, dummy debiased and processed by *FairSwiRL*. In the first row the points are colored according to the sensitive attribute while in the second row they are colored according to the target variable. Best viewed in color

attribute; in the bottom row, colors are assigned according to the values of the target attribute.

We can notice that in the plot on the left (original data), the instances with different values for their sensitive attribute are well separated into two clusters. This is the expected behavior, as the sensitive attribute is present in the data and it is used by t-SNE to better separate the points. By removing the sensitive attribute (central column), the fairness situation improves: while still not ideal, points are now harder to classify according to s. However, points with the same sensitive attribute value are clustered in small groups, and a non-linear classifier would recognize these patterns quite easily. The representations learned by *FairSwiRL* (right column) do not show such pattern: the points with the same sensitive attribute value appear to be well-mixed and distributed randomly. On the other hand, if we look at the second row of plots, we note that a similar pattern can be observed for the colors assigned to the target attribute: debiasing via *FairSwiRL* is making it harder to separate examples according to the target variable. It is worth noting, however, that while the colors of the points in the top right plot (attribute s) appear to be truly random, the ones in the bottom right plot (attribute y) do show some clustering patterns, which



(a) Original Dataset (b) Dummy Debiased (c) *FairSwiRL*

(d) Original Dataset (e) Dummy Debiased (f) *FairSwiRL*

Fig. 4 t-SNE representations of the ADULT dataset. From left to right: representations obtained from the original dataset, dummy debiased and processed by *Fair*`SwiRL`. In the first row the points are colored according to the sensitive attribute while in the second row they are colored according to the target variable. Best viewed in color

can may useful for downstream classifiers. The relationship between debiasing (removing sensitive information) and predictive performance is widely studied in the literature (Zafar et al., 2017; McNamara et al., 2017; Zemel et al., 2013), and it is often the case that some correlation between s and y can be observed. Thus, this behavior is also expected. Nonetheless, representations learned by *Fair*`SwiRL` appear to be well-mixed w.r.t. the sensitive attribute but still usable for classification. This phenomenon is quantitatively investigated in the next sections.

The observations we made for the synthetic dataset are valid also for the other datasets under study: their plots show similar patterns and are omitted here due to space constraints.

6.2 *Fair*`SwiRL` in combination with different supervised classifiers

In this experiment, we compare different classifiers in combination with *Fair*`SwiRL`, namely: random forest (*Fair*`SwiRL` +RF), k-nearest neighbors (*Fair*`SwiRL` +KNN), logistic regression (*Fair*`SwiRL` +LR), support vector machines (*Fair*`SwiRL` +SVC) and neural network (*Fair*`SwiRL` +NN).

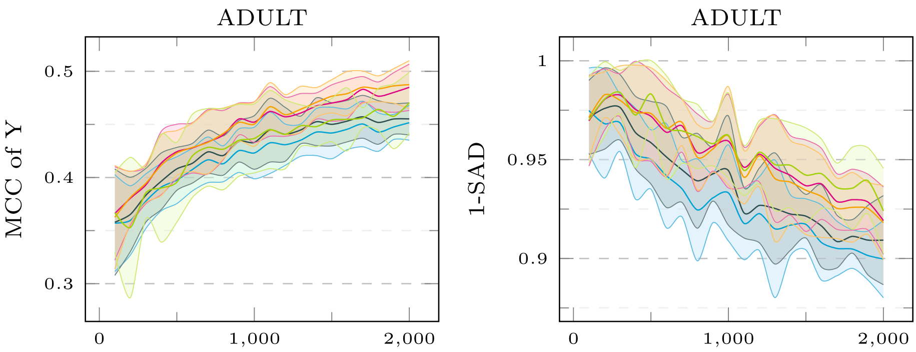


Fig. 5 Performances of *FairSwiRL* for increasing number of labeled instances and in combination with different classifiers: RF (in cyan), KNN (in blue), LR (in red), SVC (in orange), NN (in green). Lines represent the mean of the given metric over 10 repetitions, shaded area correspond to ± one standard deviation. The *x*-axis represents the number of labeled instances used during the training process. Best viewed in color (Color figure online)

We now define the data splits and the evaluation metric we will employ in this section and in the rest of the paper. Let n_l, n_u, n_v, and n_t be the number of labeled, unlabeled, validation and test examples. We start with the following configuration: $n_l = 100$, $n_u = 10000$, $n_v = 100$, $n_t = 10000$ (in case of the COMPAS dataset $n_l = 100$, $n_u = 1900$, $n_v = 100$, $n_t = 1900$). We use the validation examples to find a good configuration of the hyperparameters and then, by using the same hyperparameters, we increase the number of labeled instances n_l from 100 to 2000. For each combination of (n_l, n_u, n_v, n_t) we repeat the experiments ten times by sampling different datasets from the original data, and compute the average performance metrics. We stress that the number of available examples for a given experimental run is computed in absolute terms, not relative. This lets us compare the performance of the methodologies across the same number of test examples, no matter how many labeled examples are available. To measure the fairness level we employ 1-SAD (see Eq. 2) while for the predictive performance we compute the Matthews Correlation Coefficient (MCC (Baldi et al., 2000; Chicco & Jurman, 2020)) defined as: $\text{MCC} = \frac{\text{TP}\cdot\text{TN}-\text{FP}\cdot\text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}}$, where TP, TN, FP, FN represent the entries of the confusion matrix (True Positive, True Negative, False Positive and False Negative).

The behaviors of different combinations of *FairSwiRL* +classifier are similar across the datasets, here we report only the results for the ADULT dataset in Fig. 5. We can notice that the trends of the different classifiers are the same in predicting the target variable and in being fair. These results show that the latent representations induced by *FairSwiRL* can be used by different classifiers and, as the number of labeled examples increases, the performances on the target variable tend to increase. While the 1-SAD value (higher is better) slightly suffers from the bias introduced by the additional examples, we note that it remains very close to optimal values (> 0.9) nonetheless. This behavior is consistent with the motivating example introduced in Sect. 3.

In the next section, we will compare *FairSwiRL* with competing approaches. In order to enable a fair comparison, we do not choose the best performing combination for each dataset. Instead, we choose the worst combination *FairSwiRL* +RF and keep it fixed in all the experiments presented in this work.

6.3 *FairSwiRL* +RF compared to competitors

In this experiment, we test the effectiveness of *FairSwiRL* on different datasets and against different competitors. The experiment setting is the same as in Sect. 6.2, but we choose only one combination *FairSwiRL* +RF (i.e., the worst performing one) as our candidate combination. In addition to *FairSwiRL* +RF, we include the following competitors:

- *PD+RF*–An RF model trained on a dataset processed by a *Perfect Debiasing* method, i.e., the dataset is manipulated to guarantee that any information (direct or derived) on the sensitive attribute is removed. This is possible only in the case of the synthetic dataset (Sect. 5.1) where the data generation process is known. Specifically, we remove S and substitute S' with X_0;
- *FESF*–An implementation of Fairness-Enhanced Sampling Framework (Zhang et al., 2022);
- *FairSSL*–An implementation of the algorithm presented by Chakraborty et al. (2021) with Label Spreading (Zhou et al., 2003) as the pseudo-labeling algorithm.
- *VFAE*–An implementation of the Variational Fair Autoencoder (Louizos et al., 2016) used to get the latent representation on which a random forest is then trained for the classification task, as in *FairSwiRL* +RF.

The results are reported in the left column of Fig. 6. The plots report on the x-axis the performance metric (MCC) and on the y-axis the fairness metric (1-SAD). We vary the number of labeled examples in the dataset and run the experiments ten times for each configuration. Each point in the plot represents one experiment, shapes vary according to the algorithm used and colors vary according to the number of labeled examples in the dataset. The best possible point in each plot is at coordinates (1, 1), but this is usually unattainable. The gray dashed line has slope -1 and, as such, points on that line have the same tradeoff between accuracy and fairness. The lines showed in each plot pass through the point closest to (1, 1) under the L_1 metric. These points are, thus, the best performers under the assumption that fairness and accuracy are equally important.

We can see that, with the exception of the plot concerning the COMPAS dataset, the points (★) representing *FairSwiRL* +RF are always in the upper half of the plots. Higher values of 1-SAD mean that the debiasing component of *FairSwiRL* is working as expected. In the SYNTHETIC dataset we can notice that the points representing *FairSwiRL* +RF are the closest to the ones of the random forest trained on perfectly debiased data (PD+RF), which is theoretically perfect as far as fairness is concerned.

The comparisons with FairSSL (⬢), FESF (⬟) and VFAE (▶) are also favorable. Except for the CARD dataset, *FairSwiRL* lies on the optimal tradeoff line. In CARD, where the best results are attained by FESF, *FairSwiRL* has a better fairness, but the lower MCC leads the FESF model to prevail in terms of the linear trade-off we are assuming here. This is a typical case of accuracy-fairness dilemma: higher 1-SAD implies also lower predictive power when the sensitive attribute and target variable are correlated. On the COMPAS dataset we have a mixed situation, while the best points are attained by *FairSwiRL*, we can see that for some experiments (specifically, those with fewer labeled examples) it attains worse performances than the competitors. Overall, we would not judge this experiment as a clear win for *FairSwiRL*, but we still maintain that it is a competitive approach also in this case. Given the peculiarity of COMPAS, additional experiments on this dataset are presented and discussed in Appendix 2.

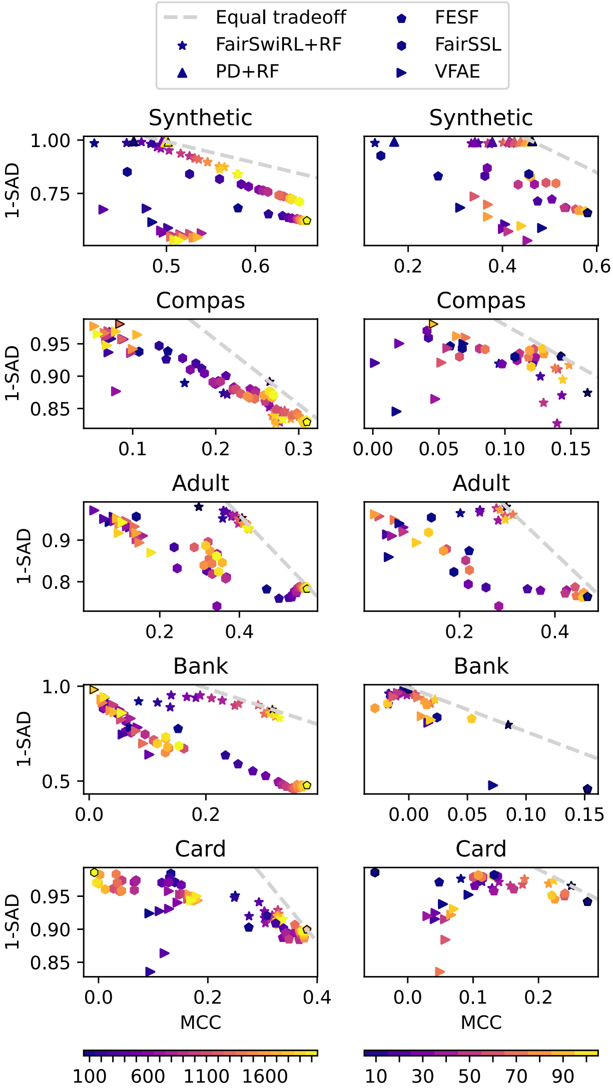


Fig. 6 A comparison of *FairSwiRL* +RF to the other competitors. The plots report on the x-axis the performance metric MCC and on the y-axis the fairness metric 1-SAD (higher is better for both). Each point is the average of 10 repeated runs with the same configuration but different samples. The colors represent the number of labeled instances: in the left column it is in the range 100–2000 while in the right column it is in the range 10–100. Best viewed in color

As far as more general trends are concerned, we observe that more labeled instances (warmer colors in Fig. 6) lead all methodologies to more accurate, but less fair results. This result, in our view, justifies further future employment of semi-supervised techniques in fair classification: a small amount of labeled data does not impact fairness negatively.

Beyond the linear tradeoff discussed above, we also experiment in an hypothetical context in which fairness is paramount and performance may be pursued only when fairness is already guaranteed. To model this situation, we repeated the experiments recording the discounted MCC metric: $\text{DisMCC} = \text{MCC}_y \cdot e^{-\alpha \text{SAD}}$, where MCC_y is the MCC computed on the target variable. It is worth noting that, in this metric, the fairness performances, as measured by the SAD statistic, are weighted exponentially. Figure 7 plots the average rankings of the competing approaches for increasing number of labeled examples. Rankings

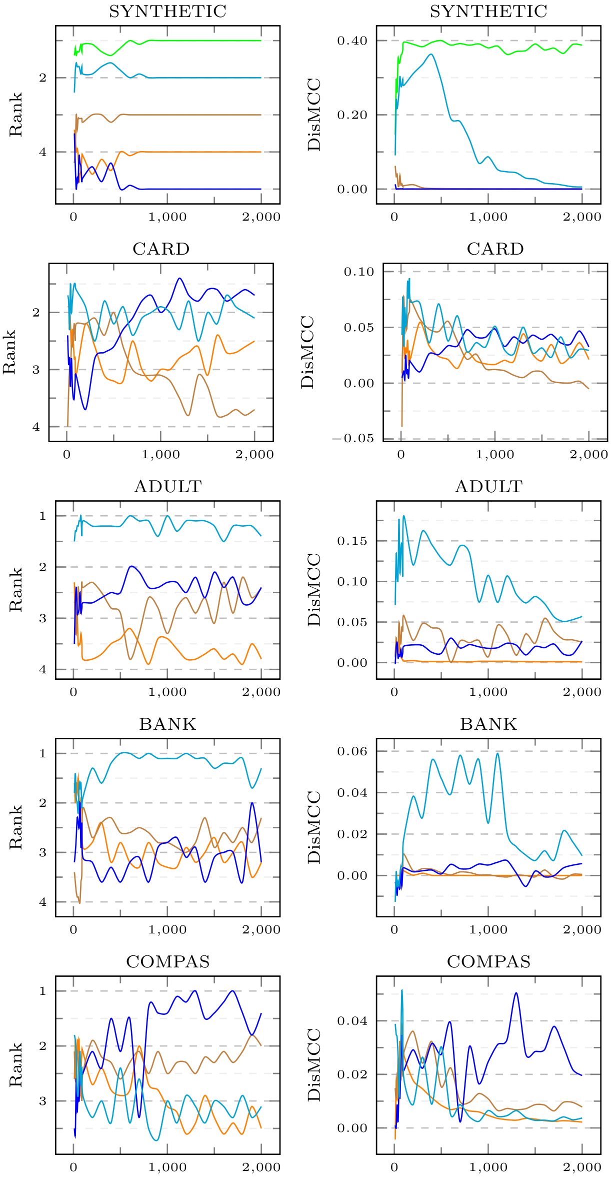


Fig. 7 Performances of *FairSwiRL* and competitors for increasing number (100–2000) of labeled instances: *FairSwiRL* +RF (in cyan), FairSSL (in brown), PD+RF (in green), FESF (in orange), VFAE (in blue). Lines represent the average rank (on the left) and the average DisMCC (on the right) over 10 repetitions. Best viewed in color (Color figure online)

are evaluated according to the value of DisMCC with $\alpha = 30$. We note that lower rankings, which are better, are displayed higher in the picture. The actual values of DisMCC obtained in the corresponding experiment are displayed in the right column (higher values are better). In SYNTHETIC the PD+RF method dominates, as expected, because it represents the theoretical upper-bound, unreachable in a real setting since the data generation process is usually unknown. However, the second best candidate is *FairSwiRL* +RF. In CARD *FairSwiRL* +RF reaches the best performance only sometimes but if compared to VFAE and FairSSL it has a more stable trajectory when the number of labeled instances changes. *Fair*`SwiRL` is overall the strongest performer on both ADULT and BANK. In COMPAS we observe worse performances than the competitors, while the other fair representation learning strategy we tested (VFAE) is the strongest performer. Overall, we would judge that also in a context where the fairness is exponentially weighted *Fair*`SwiRL` +RF performs well on average.

6.4 *Fair*`SwiRL` +RF compared to other competitors when the number of labeled instances is very low

In this experiment the setting is similar to the previous one. The only difference is the number of labeled instances that does not change from 100 to 2000 but from 10 to 100, thus leading to a more challenging scenario with very few labeled instances. As before, we consider *Fair*`SwiRL` +RF as our candidate combination and PD+RF, FESF, FairSSL, VFAE as competitors.

Before looking at the data, it is worth reporting that, given the low number of labeled instances, FESF and FairSSL fail their training procedure in several runs and on different datasets because, at a certain point, the training set becomes empty: FESF involves a down sampling procedure while FairSSL uses *situation testing* which also reduces the number of data points. We still report the average value of successful runs in the plots whenever possible in order to enable a comparison.

The results of this experiment are reported in the right column of Fig. 6. The plot setting is identical to the one reported in Sect. 6.3, the only difference being in the number of labeled examples.

As in the previous case, we can see that, except for the COMPAS dataset, the points (★) representing *Fair*`SwiRL` +RF are always in the upper half of the plots. This means that the debiasing component of *Fair*`SwiRL` is working as expected also when the number of labeled instances is very low. In particular, in COMPAS, *Fair*`SwiRL` +RF seems to have the best classification performances while still remaining near the grey dashed line. In the case of the SYNTHETIC dataset *Fair*`SwiRL` +RF is the only model that has almost the same level of 1-SAD reached by theoretically optimal PD+RF. In ADULT, both *Fair*`SwiRL` +RF and FESF reach the threshold line: the former gives more importance to the fairness, while the latter is more optimized for the classification task. In BANK, every method reaches a good fairness but none of them display solid classification performances. We posit that, given the extremely low number of labeled instances we considered, the classification models learned on this dataset are not too different from random guessing. In CARD, *Fair*`SwiRL` +RF remains very competitive by reaching high level of fairness while maintaining also a good performance on the classification task.

To complete the analysis, we report, in Table 3, the performances of the algorithms in terms of the average rank computed by using the DisMCC metric with $\alpha = 30$. The value of α is arbitrary chosen at the beginning and kept fixed during all experiments. Coherently with the observations made for the plots in the second column of Fig. 6, *Fair*`SwiRL` +RF outperforms the competing methods in SYNTHETIC, ADULT, BANK and CARD. In the remaining dataset (COMPAS), FESF performs better. It is worth pointing out that, while *Fair*`SwiRL` +RF results as the best performer in most datasets, VFAE (the other fair representation learning strategy) behaves poorly in this extreme setting in which only a very low number of labeled instances are available.

Table 3 Average ranks of different methods when the number of labeled instances (n_l) is very low

Dataset	n_l	*FairSwiRL* +RF	FESF	FairSSL	VFAE
SYNTHETIC	20	**1.8**	3.7	3.9	4.4
SYNTHETIC	40	**1.7**	3.9	3.4	4.7
SYNTHETIC	60	**1.7**	4.8	3.1	4.1
SYNTHETIC	80	**1.9**	4.5	3.1	4.4
SYNTHETIC	100	**1.9**	4.1	3.2	4.7
ADULT	20	**1.3**	3.3	2.7	2.7
ADULT	40	**1.2**	2.9	3.0	2.9
ADULT	60	**1.1**	3.5	2.6	2.8
ADULT	80	**1.0**	3.3	2.8	2.9
ADULT	100	**1.1**	3.8	2.4	2.7
BANK	20	**1.4**	2.0	3.6	3.0
BANK	40	2.0	**1.8**	3.9	2.3
BANK	60	**1.6**	1.9	4.0	2.5
BANK	80	**1.6**	1.9	3.6	2.9
BANK	100	**1.9**	2.9	2.1	3.1
CARD	20	**1.8**	1.9	3.4	2.9
CARD	40	**1.5**	2.2	3.0	3.3
CARD	60	**1.8**	2.3	2.6	3.3
CARD	80	**1.5**	2.8	2.2	3.5
CARD	100	**1.6**	3.1	2.2	3.1
COMPAS	20	**1.9**	**1.9**	2.6	3.6
COMPAS	40	2.8	**1.9**	2.1	3.2
COMPAS	60	2.5	**1.9**	2.6	3.0
COMPAS	80	2.7	2.5	**2.3**	2.5
COMPAS	100	2.9	**2.1**	2.5	2.5

Bold values indicate best results

6.5 *FairSwiRL* +RF compared to other competitors when labeled instances are very biased

In this experiment, we assess the behaviors of *FairSwiRL* +RF and competitor methods in an extreme and difficult setting: for each dataset we cherry-pick a set of 100 labeled instances where the SAD value computed on the target variable is exactly 1 (maximum bias). As mentioned in Sect. 6.2, our general setup is to select a fixed number of unlabeled and labeled instances for each experimental run. Therefore, we are able to construct a maximum-bias setup by only selecting instances with positive outcomes ($Y = 1$) for the privileged group ($S = 1$). Symmetrically, we include instances with negative outcomes ($Y = 0$) for the underprivileged group ($S = 0$).

In Table 4 we report the average SAD values (lower is better) computed on the predictions provided by different methods. We also report the values of $\mathbb{E}[\hat{Y} \mid S = 0]$ and $\mathbb{E}[\hat{Y} \mid S = 1]$ within the parentheses.

We can observe that, in this extreme setting, the CARD dataset is problematic for every method: FairSSL failed the training process, *FairSwiRL* +RF and FESF predict $\hat{Y} = 0$ for almost every instance of the test set, and VFAE provide very biased predictions. In

Table 4 Average SAD (lower is better) of different methods when the labeled instances are very biased

	FairSwiRL +RF	FESF	FairSSL	VFAE
SYNTHETIC	**0.022** (0.50, 0.52)	0.40 (0.31, 0.71)	–	0.22 (0.38, 0.60)
ADULT	**0.11** (0.27, 0.38)	0.50 (0.12, 0.62)	0.47 (0.12, 0.59)	0.20 (0.30, 0.50)
BANK	**0.040** (0.018, 0.058)	0.14 (0.022, 0.16)	–	0.077 (0.0081, 0.085)
CARD	0.0023 (0, 0.0023)	0 (0, 0)	–	0.42 (0.00014, 0.42)
COMPAS	0.27 (0.30, 0.57)	**0.094** (0.46, 0.55)	0.11 (0.40, 0.51)	0.14 (0.43, 0.57)

Bold values indicate best results

The two expected values defining the SAD are in brackets

Table 5 Average MCC (higher is better) of different methods when the labeled instances are very biased

	FairSwiRL +RF	FESF	FairSSL	VFAE
SYNTHETIC	0.43	0.60	–	0.42
ADULT	0.33	0.45	0.39	0.075
BANK	0.035	0.071	–	0.056
CARD	−0.0018	0	–	−0.024
COMPAS	0.11	0.14	0.16	0.079

SYNTHETIC, ADULT and BANK datasets, *FairSwiRL* +RF always provides the least biased predictions while other methods fail the training process or give biased results. In COMPAS dataset, FESF is the most unbiased method. This outcome is coherent with the results presented in previous experiments (where *FairSwiRL* +RF was not competitive when measured with DisMCC) because in this setting labeled instances are not only scarce but also very biased.

Having made sure of the fact that the representations learned by *FairSwiRL* are as unbiased as possible also in this extreme setting, let's consider the target variable prediction performance in Table 5. In this table we can observe that the representations learned by *FairSwiRL* , while remaining as much unbiased as possible, still provide useful and not random predictions in almost every dataset.

7 Conclusion

We have proposed a neural network for representation learning that addresses two challenging issues simultaneously: the lack of sufficient labeled examples in the training data, and the presence of sensitive attributes potentially leading to unfair decisions. We have shown that unlabeled examples help the learning algorithm to cope with both problems, leading to fair and accurate semi-supervised classification of unseen examples. The experiments, conducted on synthetic and real-world data, have shown the effectiveness of our approach, even in comparison with state-of-the-art fair semi-supervised methods which employ preprocessing strategies. We have also performed a full comparison with another fair representation learning strategy (VFAE) (Louizos et al., 2016) which had so far never been tested in the SSL setting. *FairSwiRL* displays more stable performances with respect to the competitors, especially when label information is extremely limited (under 100 examples).

Our experiments show that regularization approaches, and fair representation learning in particular, are able to outperform feature preprocessing strategies in the semi-supervised setting and such a result transfers across different tradeoffs for fairness vs. accuracy.

In this paper we have optimized our model only for one particular fairness definition and a single sensitive attribute. A few significantly different fairness definitions have been proposed in literature (Castelnovo et al., 2021; Barocas et al., 2019) and a natural direction for future work is to generalize *FairSwiRL* to satisfy other fairness metrics. We note that in a typical semi-supervised setting the number of labeled instances is very limited. Some of the alternative fairness definitions (e.g., equalized odds (Hardt et al., 2016)) require to estimate the probability distribution of the target variable for each sensitive attribute value. In this scenario, it can be complicated to obtain good estimates of the underlying probability distributions given the paucity of labeled examples in an SSL setting.

Focusing on *FairSwiRL*, one specific challenge is the adaptation of the system to settings where multiple sensitive attributes are involved. In this scenario, the most straightforward approach is the usage of multiple sub-networks, each one predicting the values of a different sensitive attribute. However, finding an equilibrium point between the resulting competing models could be, in practice, quite hard and it is unclear to us if this strategy would be stable enough to be useful in practice.

Despite these difficulties, we believe that efforts to address these two challenges would be well spent, as the resulting system would generalize significantly the methodology presented here, and may foster additional new contributions to the field of fair semi-supervised learning.

Appendix 1: Experiments reproducibility

For the experiments presented above, we have used a machine equipped with 32 CPUs Intel Xeon Processor (Skylake, IBRS) 2099.998 MHz, 256GB RAM and Tesla T4. Experiments have been orchestrated using Weights & Biases (Biewald, 2020). Hyperparameters searches used Bayesian Hyperparameter Optimization.[2] The objective function used for the hyperparameters search is an extended version of Discounted Matthews Correlation Coefficient:

$$\text{eDisMCC} = \text{MCC}_y \cdot e^{-\alpha(\text{SAD}+\text{MCC}_s)}, \tag{10}$$

where MCC_y is the MCC computed on the target variable and MCC_s is the MCC computed on the sensitive attribute by using the latent representation. In the case of FESF and FairSSL (having no latent representation) we set $\text{MCC}_s = 0$.

In optimizing the hyperparameters, we computed MCC_y on the labeled validation set (100 instances) and SAD on half of the unlabeled training set (5000 instances). This allowed us to overcome the problem raised by the paucity of examples in the validation set. We note that this is possible because we do not need y labels for the computation of SAD.

[2] The code used to perform all experiments can be found at: https://github.com/ngshya/fairswirl.

Table 6 List of hyperparameters of debiasing methods

Method	Hyperparameters	Possible values
FESF	k	From 1 to 500
FESF	Base model	LR, RF, SVC, KNN
FairSSL	cr	From 0.1 to 1.0
FairSSL	f	From 0.1 to 1.0
FairSSL	Base model	LR, RF, SVC, KNN
VFAE	D	From 5 to 200
VFAE	α	From 0.1 to 10
VFAE	β	From 0.1 to 10
VFAE	Decoder 1 hidden layer size	From 4 to 32
VFAE	Decoder 2 hidden layer size	From 4 to 32
VFAE	Encoder 1 hidden layer size	From 4 to 32
VFAE	Encoder 2 hidden layer size	From 4 to 32
VFAE	Us hidden layer size	From 4 to 32
VFAE	z1	From 4 to 16
VFAE	z2	From 4 to 16
VFAE	Learning rate	0.01 or 0.001
VFAE	Epochs	From 10 to 200
FairSwiRL	# layers of the encoder	From 1 to 4
FairSwiRL	# layers of the classifier	From 1 to 3
FairSwiRL	# layers of the debiaser	From 1 to 3
FairSwiRL	# neurons per layer	From 2 to 16
FairSwiRL	Learning rate	0.01 or 0.001
FairSwiRL	Epochs	From 10 to 200
FairSwiRL	W_{rec}	From 0.1 to 100
FairSwiRL	W_{cla}	From 0.1 to 100
FairSwiRL	W_{fair}	From 0.1 to 100

It is worth pointing out that the test set is only used during the assessment of the final performances so to allow unbiased estimates of the relevant metrics.

Table 6 reports the hyperparameters subjected to optimization. For t-SNE, we used the default hyperparameters provided by the scikit-learn (Pedregosa et al., 2011) package: perplexity=30, early exaggeration = 12, learning rate = 200 and maximum number of iterations = 1000. In the case of *FairSwiRL* and VFAE we used the Adam optimizer. For FairSSL, we used Label Spreading (Zhou et al., 2003) as pseudo-labeling algorithm. For supervised classifiers (e.g., RF), we used the default hyperparameters provided by the scikit-learn (Pedregosa et al., 2011) package.

Appendix 2: Additional experiments on COMPAS

In this Section we propose a deeper investigation into *FairSwiRL* 's performance on the COMPAS dataset, as this was the most challenging setup for *FairSwiRL* +RF (see Sect. 6). First, we observe that COMPAS has a limited number of labeled *and* unlabeled instances (n_u = 1900). This dataset is by far the smallest one in our experimentation. Having such a small sample is in contrast with the SSL setting (where one assumes

Table 7 Performances (MCC, 1-SAD) of *FairSwiRL* +RF on COMPAS with (opt) and without (not opt) the hyperparameters of RF optimized

Labeled instances	MCC (not opt)	MCC (opt)	1-SAD (not opt)	1-SAD (opt)
10	**0.095**	–	**0.929**	–
50	0.129	**0.180**	0.876	**0.922**
100	0.163	**0.191**	0.889	**0.953**
500	0.265	**0.298**	**0.891**	0.887
1000	0.279	**0.328**	0.846	**0.850**
1500	0.271	**0.327**	0.831	**0.839**
2000	0.281	**0.332**	**0.833**	0.807

Bold values indicate best results

Table 8 Performances (DisMCC) of *FairSwiRL* +RF on COMPAS with (opt) and without (not opt) the hyperparameters of RF optimized

Labeled instances	DisMCC (not opt)	DisMCC (opt)
10	**0.039**	–
50	0.012	**0.054**
100	0.019	**0.070**
500	**0.030**	0.021
1000	0.007	**0.010**
1500	0.003	**0.005**
2000	**0.004**	0.003

Bold values indicate best results

that unlabeled data is plentiful) and with the specific goals of *FairSwiRL*, which aims to perform three different tasks (classification, reconstruction, debiasing). In addition, since optimizing the final performances was not the main goal of this work, we did not perform any hyperparameter optimization in our experiments and this is likely to have also affected the performances. Then, we set out to investigate whether the performances of *FairSwiRL* +RF could be improved by optimizing the hyperparameters of the final classifier (RF). We emphasize, however, that special care must be taken during hyperparameter selection: improving predictive performance (higher MCC) may worsen the fairness of the resulting classifier (lower 1-SAD). This effect was observed, for instance, in Fig. 5. Therefore, we used an optimization strategy similar to the one presented in Appendix 1, where the optimal hyperparameters were defined as the ones obtaining the highest value of a combined fairness/performance metric. The Table 7 reports the MCC and the 1-SAD values in two different scenarios: *FairSwiRL* +RF with default RF hyperparameters ("not opt" columns) and *FairSwiRL* +RF with optimized hyperparameters for RF ("opt").

The hypterparameter search, as it was to be expected, improved the predictive performances in almost all configurations. A little more counterintuitive are, instead, the performances on the 1-SAD metric where there are still cases where the results for the unoptimized version are better. This behavior is confirmed also when we compare DisMCC metric values (see Table 8).

According to these results, hyperparameter optimization for the final classifier in our framework can give some boost in both performance and fairness, but care needs to be taken to avoid worsening the fairness of the classifier.

Nonetheless, this procedure is a downstream classifier hyperparameters optimization: an implicit assumption made here is that the end user of the learned representations is willing to engage in such optimization/debiasing.

Author contributions Conceptualization: SY, MC, DI, RGP, RE; Methodology: SY, MC, DI, RGP, RE; Formal analysis and investigation: SY, MC; Writing - original draft preparation: SY, MC, DI, RGP, RE; Writing - review and editing: SY, MC, DI, RGP, RE; Supervision: RGP, RE. All authors read and approved the final manuscript.

Funding Open access funding provided by Università degli Studi di Torino within the CRUI-CARE Agreement. Not applicable.

Data availability All data are available online and accessible to everyone.

Code availability Source code and scripts used in our experiments are available at https://github.com/ngshya/fairswirl.

Declarations

Conflict of interest Mattia Cerrato, Dino Ienco, Ruggero G. Pensa and Roberto Esposito are members of the Editorial Board. The authors have no further competing interests to declare that are relevant to the content of this article.

References

Angwin, J. , Larson, J. , Mattu, S., Kirchner, L. (2016). *Machine bias.*https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F., & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: An overview. *Bioinformatics, 16*(5), 412–424.

Barocas, S. , Hardt, M., Narayanan, A. (2019). Fairness and machine learning. http://www.fairmlbook.org

Bellamy, R.K.E. , Dey, K. , Hind, M. , Hoffman, S.C. , Houde, S. , Kannan, K., Zhang, Y. (2018). AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *CoRR,*arxiv:1810.01943

Bengio, Y., Courville, A. C., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798–1828.

Biewald, L. (2020). *Experiment tracking with weights and biases.*https://www.wandb.com/ *(Software available from wandb.com)*

Castelnovo, A. , Crupi, R. , Greco, G., Regoli, D. (2021). The zoo of fairness metrics in machine learning. *CoRR,*arxiv:2106.00467

Chakraborty, J. , Tu, H. , Majumder, S., Menzies, T. (2021). Can we achieve fairness using semi-supervised learning? *CoRR,*arxiv:2111.02038

Chapelle, O., Schölkopf, B., & Zien, A. (2006). Introduction to semi-supervised learning. In O. Chapelle, B. Schölkopf, & A. Zien (Eds.), *Semi-supervised learning* (pp. 1–12). Cambridge: The MIT Press.

Cheng, Y. , Zhao, X. , Cai, R. , Li, Z. , Huang, K., Rui, Y. (2016). Semi-supervised multimodal deep learning for RGB-D object recognition. In S. Kambhampati (Ed.), *Proceedings of IJCAI 2016* (pp. 3345–3351). IJCAI/AAAI Press.

Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics, 21*(1), 1–13.

Dua, D., & Graff, C. (2017). *UCI machine learning repository.*http://archive.ics.uci.edu/ml

Friedman, B., & Nissenbaum, H. (1996). Bias in computer systems. *ACM Transactions on Information Systems, 14*(3), 330–347.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., & Lempitsky, V. S. (2016). Domain-adversarial training of neural networks. *The Journal of Machine Learning Research, 17*, 59:1-59:35.

Gogna, A., & Majumdar, A. (2016). Semi supervised autoencoder. In *Proceedings of ICONIP 2016* (pp. 82–89).

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. J. (2012). A kernel two-sample test. *The Journal of Machine Learning Research, 13*, 723–773.

Hardt, M. , Price, E., Srebro, N. (2016). Equality of opportunity in supervised learning. In *Proceedings of NIPS 2016, Dec 5–10, 2016, Barcelona, Spain* (pp. 3315–3323).

Hinton, G.E., & Zemel, R.S. (1993). Autoencoders, minimum description length and helmholtz free energy. In *Proceesings of NIPS 1993* (pp. 3–10). Morgan Kaufmann

Hu, F. , Zhu, Y. , Wu, S. , Wang, L., Tan, T. (2019). Hierarchical graph convolutional networks for semi-supervised node classification. In S. Kraus (Ed.), *Proceedings of IJCAI 2019* (pp. 4532–4539).

Kamiran, F., & Calders, T. (2009). Classifying without discriminating. In *Proceesings of IEEE-IC4 2009.*

Le, L. , Patterson, A., White, M. (2018). Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In *Proceesings of NeurIPS 2018* (pp. 107–117).

Louizos, C. , Swersky, K. , Li, Y. , Welling, M., Zemel, R.S. (2016). The variational fair autoencoder. In *Proceedings of ICLR 2016.*

Madras, D. , Creager, E. , Pitassi, T., Zemel, R.S. (2018). Learning adversarially fair and transferable representations. In *Proceedings of ICML 2018* (pp. 3381–3390).

McNamara, D. , Ong, C.S., Williamson, R.C. (2017). Provably fair representations. *CoRR*, arxiv:1710.04394

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys, 54*(6), 115:1-115:35.

Oneto, L., Donini, M., Luise, G., Ciliberto, C., Maurer, A., & Pontil, M. (2020). Exploiting MMD and sinkhorn divergences for fair and transferable representation learning. In *Proceesings of NeurIPS 2020.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830.

Rasmus, A. , Berglund, M. , Honkala, M. , Valpola, H., Raiko, T. (2015). Semi-supervised learning with ladder networks. *Advances in NIPS 2015* pp. 3546–3554.

Springenberg, J.T. (2016). Unsupervised and semi-supervised learning with categorical generative adversarial networks. In Y. Bengio & Y. LeCun (Eds.), *Proceedings of ICLR 2016.*

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research, 9*(11), 2579–2605.

van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning, 109*(2), 373–440.

Xie, Q. , Dai, Z. , Du, Y. , Hovy, E.H., Neubig, G. (2017). Controllable invariance through adversarial feature learning. In *Proceeding of NIPS 2017* (pp. 585–596).

Yamaguchi, Y. , Faloutsos, C., Kitagawa, H. (2016). CAMLP: confidence-aware modulated label propagation. In S.C. Venkatasubramanian & W.M. Jr. (Eds.), *Proceedings of SIAM SDM 2016* (pp. 513–521). SIAM.

Yang, S., Ienco, D., Esposito, R., & Pensa, R. G. (2021). Esa*: A generic framework for semi-supervised inductive learning. *Neurocomputing, 447*, 102–117.

Zafar, M.B. , Valera, I. , Gomez-Rodriguez, M., Gummadi, K.P. (2017). Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of WWW 2017* (pp. 1171–1180). ACM.

Zemel, R.S. , Wu, Y. , Swersky, K. , Pitassi, T., Dwork, C. (2013). Learning fair representations. In *Proceedings of ICML 2013* (Vol. 28, pp. 325–333).

Zhang, B.H. , Lemoine, B., Mitchell, M. (2018). Mitigating unwanted biases with adversarial learning. In *Proceedings of AIES 2018* (pp. 335–340). ACM.

Zhang, T., Zhu, T., Li, J., Han, M., Zhou, W., & Yu, P. S. (2022). Fairness in semi-supervised learning: Unlabeled data help to reduce discrimination. *Transactions on Knowledge and Data Engineering, 34*(4), 1763–1774.

Zhou, D. , Bousquet, O. , Lal, T.N. , Weston, J., Schölkopf, B. (2003). Learning with local and global consistency. In *Proceesings of NIPS 2003* (pp. 321–328). MIT Press.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.