Machine Learning (2023) 112:2761–2786
https://doi.org/10.1007/s10994-023-06360-7

FAC-fed: Federated adaptation for fairness and concept drift aware stream classification

Maryam Badar[1] · Wolfgang Nejdl[1] · Marco Fisichella[1]

Received: 13 February 2023 / Revised: 9 June 2023 / Accepted: 28 June 2023 /
Published online: 26 July 2023

Abstract

Federated learning is an emerging collaborative learning paradigm of Machine learning involving distributed and heterogeneous clients. Enormous collections of continuously arriving heterogeneous data residing on distributed clients require federated adaptation of efficient mining algorithms to enable fair and high-quality predictions with privacy guarantees and minimal response delay. In this context, we propose a federated adaptation that mitigates discrimination embedded in the streaming data while handling concept drifts (FAC-Fed). We present a novel adaptive data augmentation method that mitigates client-side discrimination embedded in the data during optimization, resulting in an optimized and fair centralized server. Extensive experiments on a set of publicly available streaming and static datasets confirm the effectiveness of the proposed method. To the best of our knowledge, this work is the first attempt towards fairness-aware federated adaptation for stream classification, therefore, to prove the superiority of our proposed method over state-of-the-art, we compare the centralized version of our proposed method with three centralized stream classification baseline models (FABBOO, FAHT, CSMOTE). The experimental results show that our method outperforms the current methods in terms of both discrimination mitigation and predictive performance.

Keywords Federated learning · Fairness · Stream classification · Deep neural network · Hedge backpropagation · Privacy

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

✉ Maryam Badar
badar@l3s.de

Wolfgang Nejdl
nejdl@l3s.de

Marco Fisichella
mfisichella@l3s.de

[1] L3S Research Center, Leibniz University of Hannover, Welfengarten 1, 30167 Hannover, Niedersachsen, Germany

1 Introduction

Many automated decision-making systems have been proposed to supplement humans in critical application areas subject to moral equivalence, including fraud detection, criminal re-conviction assessment, credit risk assessment, disease diagnosis and recruitment (Dobbe et al., 2021). However, the practical application of such Machine Learning (ML) methods has raised many concerns regarding their fairness, auditability, privacy-preservation and transparency (Emelianov et al., 2022).

Due to the escalating interest of the research community in issues of fairness and trustworthiness of learning algorithms, a substantial body of work already exists in this domain (Calders et al., 2009; Chakraborty et al., 2021; Hajian et al., 2015; Iosifidis & Ntoutsi, 2020; Kamiran & Calders, 2009, 2012; Kamiran et al., 2012; Zhang et al., 2019). However, real-world applications such as stock market platforms, e-commerce websites, and telemedicine web platforms rely on real-time distributed data streams. These real-time data streams evolve continuously and the statistical dependencies within the data also change over time (concept drift) (Liu et al., 2017). Concept drift, if not tackled properly, leads to compromised predictive performance of model. Therefore massive collections of streaming data necessitate fair, efficient, and concept drift aware data mining algorithms to generate non-discriminatory and high-quality predictions. Recent years have witnessed a few studies that focus on detecting and mitigating discrimination embedded in the streaming data in a centralized environment (Iosifidis et al., 2019; Iosifidis & Ntoutsi, 2020; Zhang et al., 2019). However, centralized access to large volumes of continuously arriving data is a prerequisite for training such conventional stream learning models. With the ubiquitous use of computing devices, data is growing exponentially and distributively. Collecting such large volumes of heterogeneous data on a centralized server raises many challenging concerns such as limited communication bandwidth, network connectivity issues, and substantial storage costs (Zhang et al., 2020). Furthermore, the contemporary advancements in legal constraints, such as the General Data Protection Regulation (GDPR) (Commission et al., 2016), have made societies more privacy-oriented, rendering data aggregation techniques utterly non-viable (Misselhorn, 2020). For example, automatic diagnosis-based telemedicine web platforms enable monitoring of remote patients' vital signs with real-time data streams. Each patient's local data can be useful for better diagnosis of other patients with similar conditions. However, a patient's diagnostic data cannot be shared with other medical professionals or patients because of privacy concerns (Commission et al., 2016). Under the new normal of such pervasive data privacy concerns and continuously growing decentralized data silos, a viable alternative to traditional online ML methods is to design their federated adaptation. Federated Learning (FL) is an emerging decentralized learning paradigm of ML that provides privacy guarantees by offloading model training to the distributed devices (clients) that own the original data. FL enables a multitude of distributed devices to collaboratively train a single shared ML model by exchanging model parameters without revealing their private information.

A plethora of research in the field of FL systems focuses exclusively on improving server performance. For example, protecting the FL system from adversarial attacks (Mothukuri et al., 2021), adapting the FL system to process non-independent identically distributed (non-IID) data (Ma et al., 2022) and improving the communication costs involved in FL system's optimization (Mills et al., 2019). There are also some works that ensure fairness in FL systems including fairness in client selection procedures (Yang et al., 2021) and incentive distribution (Yu et al., 2020). However, little to no attention has been paid to ensuring fairness in the predictions of FL system while improving/maintaining predictive performance in a stream learning environment.

In this work, we propose federated adaptation for fairness and concept drift aware stream classification. The key contributions of our work are as follows:

- We propose a novel adaptation of Federated learning framework to mitigate discrimination while simultaneously handling concept drifts and improving its predictive performance in a stream learning environment.
- We propose a novel adaptive data augmentation technique for discrimination mitigation.
- In FL setup, data is not available on centralized server, therefore, it can be non-independent and identically distributed(non-IID). We have used real world datasets [Bank (Bache & Lichman, 2013), Default (Bache & Lichman, 2013), Adult Census (Bache & Lichman, 2013), Law School (Wightman, 1998)] and proved that even with non-IID data, FAC-Fed converges within a reasonable number of communication rounds.
- We scrutinize the effectiveness of our proposed model by performing extensive experiments with a range of publicly available datasets including: Bank Marketing (Bache & Lichman, 2013), Default (Bache & Lichman, 2013), Adult Census (Bache & Lichman, 2013), Law School (Wightman, 1998). To the best of our knowledge this is the first attempt towards fairness and concept drift-aware federated adaptation for stream classification, therefore, we ensure the superiority of our proposed framework by comparing the results of centralized version of FAC-Fed with a range of centralized state-of-the-art stream classification baselines: FABBOO (Iosifidis & Ntoutsi, 2020), FAHT (Zhang et al., 2019), CSMOTE (Bernardo et al., 2020).

2 Related work

Our literature work is based on four research domains including: Fairness-aware learning, Fairness-aware stream learning, Federated-Learning, Fairness-aware federated learning.

2.1 Fairness-aware learning

Recently, state-of-the-art ML based methods presented in the literature for identifying and subsequently eliminating bias and discrimination have gained great attention. These techniques can be categorized into pre-processing, in-processing, and post-processing techniques.

2.1.1 Pre-processing techniques

Learner outcomes are significantly influenced by training data. There is a substantial likelihood that the learner will make biased predictions if the training data is biased. The literature contains a number of pre-processing methods that aim to provide solutions to fairness issues by manipulating the training data. The most basic pre-processing techniques include massaging (Kamiran & Calders, 2009), reweighting (Calders et al., 2009), preferential sampling (Kamiran & Calders, 2012), and Synthetic Minority Oversampling Technique (SMOTE) inspired fairness-aware upsampling (Chakraborty et al., 2021). However, completely unbiased training data can sometimes lead to biased predictions of the learner because the pre-processing approaches are not able to account for the bias introduced by the learner itself (Zhang et al., 2018).

2.1.2 In-processing techniques

These techniques tailor the classification model to generate fair outcomes. For example, Zhang et al. (2018) proposed an adversarial network to mitigate bias where the adversary tries to identify the relationship between a sensitive attribute and the predictor's outcome, while the predictor's goal was to optimize performance while deceiving the adversary. Furthermore, Zafar et al. (2019) and Padala and Gujar (2020) incorporated fairness constraints into the learner's objective function to achieve fairness. Another strategy to reduce discrimination based on adaptive reweighting of training instances is introduced by Iosifidis and Ntoutsi (2019).

2.1.3 Post-processing techniques

These methods tweak the classifier decisions to mitigate bias, such as Kamiran et al. (2010) ameliorated discrimination by relabeling leaves of decision tree model. Kamiran et al. (2012) proposed decision theory based solutions for discrimination free classification. Another post-processing method removed discrimination by processing the fair patterns with k-anonymity (Hajian et al., 2015).

2.2 Fairness-aware stream learning

These types of learning techniques provide solutions to fairness issues in a stream learning environment. Iosifidis et al. (2019) proposed a chunk-based pre-processing technique to achieve fairness goals. A decision tree-based technique, FAHT (Fairness Aware Hoeffding Tree) (Zhang et al., 2019), resolved fairness issues in data streams by considering fairness gain along with the information gain in the splitting criterion of the decision tree. FABBOO (Iosifidis & Ntoutsi, 2020) is another decision tree-based method which changed the decision boundaries to achieve fairness. But FABBOO and FAHT have fixed the role of the sensitive group across the whole stream, therefore, they cannot deal with reverse discrimination, i.e. discrimination towards the privileged group.

All of these proposed methods for reducing discrimination in a stationary and non-stationary environment are based on the ML ansatz—the learner has access to complete training data. However, this assumption cannot be generalized to the FL settings.

2.3 Federated learning

Federated Learning (FL) (McMahan et al., 2017) was proposed as a decentralized solution to share clients' model updates in the form of weights or gradients during the optimization process instead of their local data to protect clients' privacy rights. This paradigm of ML brings many challenges, such as privacy leakage, limited communication bandwidth, handling non-IID data among distributed clients, and improving clients' personalization experience. Several research works have been presented to overcome these challenges. For example, Bonawitz et al. (2017), Papernot et al. (2016) have proposed methods to avoid the issue of privacy leakage in FL systems by either encrypting the client's training parameters or by adding differential privacy noise to the exchanged training parameters.

Mills et al. (2019) proposed the distributed form of Adam's optimization algorithm to reduce the number of communication rounds and achieved optimal accuracy in fewer rounds. There are also other research works that deal with the problem of limited communication bandwidth in a federated setup (Abdellatif et al., 2022; Paragliola, 2022).

Zhu et al. (2021) investigated the impact of non-IID data on the classification performance of FL clients and found that accuracy drops significantly with non-IID data. To overcome this problem, several works have been presented (Fisichella et al., 2022; Singh et al., 2023; Wei et al., 2022; Yang et al., 2021; Younis & Fisichella, 2022).

Liu et al. (2021) proposed a method to improve the performance of FL framework for personalization improvement by cooperation of similar clients. A similar federated adaptation for improving personalization experience for clients is proposed by Wu et al. (2021).

2.4 Fairness-aware federated learning

In the current state-of-the-art only few studies have been conducted in this research area. There are some works in the literature that address fairness issues in FL. However, these studies focus exclusively on ensuring fairness in the client selection procedures (Huang et al., 2020; Yang et al., 2021) and incentive distribution (Zeng et al., 2020; Zhang et al., 2020, 2022; Yu et al., 2020). The area of ensuring fairness in the outcomes of a FL framework is still under explored. For example, FairFL (Zhang et al., 2020) provides a deep re-enforcement learning framework to reduce demographic bias (statistical parity) while respecting the clients' privacy constraints. A gradient-based approach is presented by Cui et al. (2021) that provides fairness guarantees along with a consistent Pareto utility distribution across all clients. Agnostic-Fair (Du et al., 2021) is another fairness-aware FL framework which reduces discrimination by adding regularization terms to the learning model that reweights the training samples. All of these works focus solely on mitigating discrimination in a static learning environment.

To the best of our knowledge our work is the first attempt towards federated adaptation of fairness and concept drift-aware stream classification. We propose an in-processing technique to mitigate discrimination by adaptively augmenting each client's local data within a defined window of instances in a streaming environment. Our proposed method is not only able to reduce the biases embedded in the clients' data, but also achieves high balanced accuracy without sharing any sensitive information with the server except the model updates of the clients.

3 Conceptual model

Figure 1 represents the conceptual model underpinning the proposed method. In this model, each client hosts a data stream, a concept drift detector, a local learner (a deep neural network), a discrimination detector and a discrimination mitigation module. In each communication round, every client trains its local learner and tries to mitigate discrimination embedded in the streaming data while simultaneously taking into account the concept drifts in the stream. The updated local learner weights are then shared with the global server. The global server averages the aggregated local learner weights. The updated global learner weights are then shared with selected range of clients in the next communication round.

4 Preliminaries

We first define some notations before illustration of the proposed methodology. Suppose we have n local clients $(C_1, C_2, \dots, C_n)$ in an FL environment and a global server G. Each client has its own local streaming dataset d_k with feature space X and output space Y. Each instance in the streaming dataset d_k of client C_k is defined as $f_j^k = \{x_j, y_j\}$. We consider a

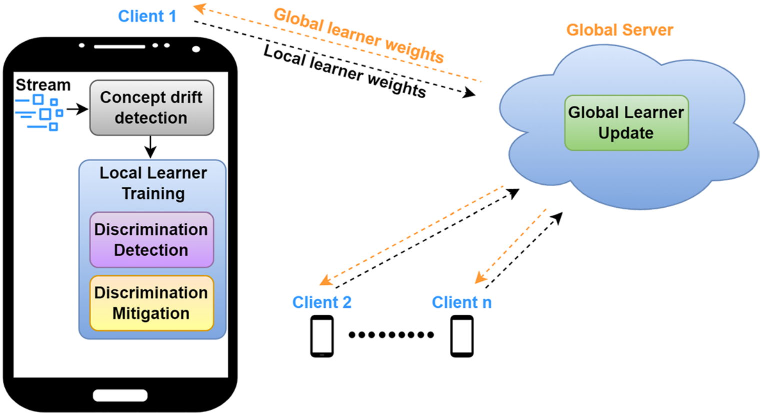

Fig. 1 Conceptual model for federated adaptation of online fairness and concept drift-aware stream classification framework

binary classification problem, i.e., $Y \epsilon \{0, 1\}$ because it is a fundamental and widely applicable problem in many fields where the cost of misclassification is high, such as fraud detection or disease diagnosis. The global server G learns the predictive function between the instances and their respective labels $f(x) = y$ through the collaborative training of the local clients $(C_1, C_2, \dots, C_n)$. The basic steps involved in FL in a streaming environment are listed below:

1. The server G initiates the global model and sends the initial parameters w_g to a random selection of clients.
2. At round l, the client C_k receives the global parameters w_g^{l-1} and uses them to train the local model using its local streaming dataset d_k to achieve the optimal local parameters w_k^l.
3. The server G receives local parameters $w_1^l, w_2^l, \dots, w_n^l$ from clients $(C_1, C_2, \dots, C_n)$ and updates itself using the average of the received parameters using Eq. (1) (McMahan et al., 2017). The server then sends the updated global parameters w_g^l to all the clients.

$$w_g^l = \frac{1}{n} \sum_{j=1}^{n} w_j^l \tag{1}$$

4. Repeat steps 2 and 3 until the end of stream.

We assume that the datasets used to train and test the proposed model have a single sensitive attribute (S) with binary values, where (P) and ($\bar{P}$) represent protected group and non-protected group respectively. For example, if "race" is the sensitive attribute, then the likely protected group (P) could include all instances with the value "black" as the sensitive attribute and the non-protected group ($\bar{P}$) could include all instances with the value "white" as the sensitive attribute. We gauge the discriminating behavior of the proposed method by two notions of fairness. There are many definitions of fairness in the literature (Verma and Rubin, 2018); however, there are no comprehensible criteria in the literature for choosing a particular notion of fairness for a particular problem. In this work, we select two group fairness notions, statistical parity (Stp) and equal opportunity (Eqop) (Verma and Rubin, 2018), to measure discrimination score. Stp ensures that each individual has an equal chance of being assigned to the positive class (y^+), irrespective of its participation in protected or non-protected group, as illustrated in Eq. (2). The positive class is the desired class of the model's objective function.

$$Stp = P(f(x) = y^+ \mid S = \bar{P}) - P(f(x) = y^+ \mid S = P) \tag{2}$$

Eqop ensures that individuals belonging to both the protected and non-protected group get positive outcome (y^+) at equal rates as shown in Eq. (3).

$$Eqop = P(f(x) = y^+ \mid y = y^+, S = \bar{P}) - P(f(x) = y^+ \mid y = y^+, S = P) \tag{3}$$

5 Proposed methodology

The complete visual illustration of the proposed methodology is shown in Fig. 2. The pseudocode of the overall approach for adapting FL framework for concept drift detection and subsequently discrimination mitigation in streaming environment is presented in

Algorithm 1. Each client hosts local streaming data and a local online deep neural network (ODNN) model (Fig. 2A). The global server also has the same ODNN model (Fig. 2G). Section 5.1.1 illustrates the details of the ODNN model used in this work. Every client begins its training by first initializing the ODNN model parameters using the global server's ODNN model parameters. Each client trains its local ODNN model with new incoming instances until the stream ends or until the global server requests the client to share the parameters.

In this setup, for each new instance, the label is predicted by the learner and the evaluation metrics are updated. We assume that the data stream is infinite and non-stationary, i.e., there is a continuous presence of concept drifts which may lead to compromised predictive performance of the learner. Therefore, we employ a concept drift detection mechanism EDDM (Early drift Detection Method) (Baena-Garcıa et al., 2006) (Fig. 2B). Once EDDM detects a concept drift, the sliding window is cleared and a new window of instances is initiated to store the next instances (Fig. 2C).

Using the prequential evaluation strategy, the discriminatory behavior of the model is quantified (Fig. 2D) by one of the aforementioned fairness notions i.e., Stp or Eqop. If the discrimination score (disc: Stp or Eqop) exceeds a user-defined threshold ϵ, the proposed continuous fairness-aware synthetic over-sampling technique (CFSOTE) is employed to mitigate the discrimination (Fig. 2E). CFSOTE uses the variable window of instances maintained by EDDM to mitigate discrimination. Then, the local online learner is trained using the newly synthesized instances. The extensive algorithmic details of CFSOTE are elaborated in Sect. 5.3.1.

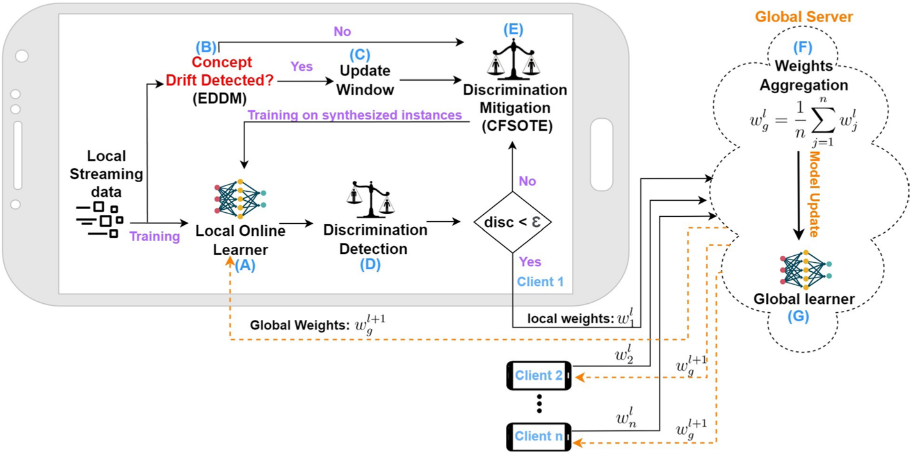

Fig. 2 Federated adaptation of online Fairness and concept drift-aware stream classification framework: **A** Local Online Learner **B** EDDM–Concept Drift Detection **C** Update Window **D** Discrimination Detection **E** CFSOTE–Discrimination Mitigation **F** Global Server Weights Aggregation **G** Update Global Online Learner

Algorithm 1 Discrimination mitigation and concept drift handling procedure with prequential evaluation for each client.

```
Require: global model parameters $w_g^l$, local stream of instances ($S_k$), sliding
    window of instances ($Window_k$) of client ($C_k$), protected attribute (P), non-
    protected attribute ($\bar{P}$), positive label, negative label.
Ensure: Optimized parameters $w_k^{l+1}$ w.r.t. fairness score (disc) and balanced
    accuracy.
 1: while ($has\ next\ instance\ in\ S_k$) do
 2:     $local\_model.initialize(w_g^l)$
 3:     $x_i, y_i \leftarrow next(S_k)$
 4:     while !$global\_server\_parameter\_request$ do
 5:         $class\_weights \leftarrow find\_weights(Window_k)$
 6:         $local\_model.test(x_i)$
 7:         $local\_model.train(x_i, y_i, class\_weights)$
 8:         $eddm.add\_element(y_i)$
 9:         if $eddm.drift\_detected$ then
10:             $eddm.clear(Window_k)$
11:             $update\_sliding\_window(Window_k, x_i, y_i)$
12:         end if
13:         $disc \leftarrow local\_model.disc\_score$
14:         if $disc < 0$ then
15:             $disc \leftarrow -disc$
16:             $swap(P,\ \bar{P})$
17:         end if
18:         if $disc > \varepsilon$ then
19:             $N(C_-, P), N(C_-, \bar{P}), N(C_+, P), N(C_+, \bar{P})$ ←
    $split\_data(Window_k, pos\_label, neg\_label, P, \bar{P})$
20:             $\lambda \leftarrow \lambda_{initial} * (1 + (disc/disc_{tol}))$
21:             if $num\_pred\_pos <= num\_pos\_labels$ then
22:                 $m \leftarrow \lambda * len(N(C_-, \bar{P}))$    ▷ Number of samples to be synthesized
23:                 $X\_syn, Y\_syn \leftarrow$ CFSOTE($N(C_+, P)$, m, $N(C_+, \bar{P})$, k, KNN)
            ▷ $X\_syn$: Synthesized instances, $Y\_syn$: outputs of synthesized instances
24:             else
25:                 $m \leftarrow \lambda * len(N(C_+, P))$
26:                 $X\_syn, Y\_syn \leftarrow$ CFSOTE($N(C_-, \bar{P})$, m, $N(C_-, \bar{P})$, k, KNN)
27:             end if
28:             for $j \leftarrow 0$ to $length(X\_syn)$ do
29:                 $local\_model.train(X\_syn_j, Y\_syn_j)$
30:             end for
31:         end if
32:     end while
33: end while
        ▷ Figure 2, Step A: lines 1 to 7, Step B: lines 8 to 10, Step C: line 11, Step
    D: line 13, Step E: 14 to 31
```

The clients share their respective local learning parameters to the global server as soon as they receive the request to share the parameters from the global server. The server then aggregates and averages (Fig. 2F) the clients' local model parameters. The global ODNN model (Fig. 2G) is updated using the averaged weights of the clients. The detailed methodology is explained in the following subsections.

5.1 Step A: local online learner

Every participating client in the system maintains its own local streaming data and an online deep neural network (ODNN) model (depicted in Fig. 2A). Section 5.1.1 illustrates the details of the ODNN model used in this work. At the start of training, each client initializes its ODNN model parameters using the corresponding parameters from the global server's ODNN model. Subsequently, the client proceeds to train its local ODNN model using newly arrived instances from the data stream, continuing this process until either the stream concludes or the global server requests the client to share its parameters (as outlined in Algorithm 1: lines 1 to 7). The global server periodically requests the clients to share their respective local parameters (Algorithm 1: *global_server_parameter_request*). Each local ODNN model trains using prequential evaluation setup, i.e., test first, then train (Gama, 2010; Zhang et al., 2019) (Algorithm 1: lines 6 to 7). In this configuration, for every incoming instance, the learner makes predictions for its label and updates the evaluation metrics accordingly. Following the prediction phase, the true label of the instance is disclosed to the learner, enabling the model to be updated based on this new information.

5.1.1 Online deep neural network (ODNN) model

Our base model is an online deep neural network inspired by Sahoo et al. (2018). ODNN uses hedge backpropagation to efficiently update the parameters of the DNN in an online environment. The Hedge Backpropagation (HBP) technique extends the backpropagation algorithm to train the DNNs in a streaming environment by utilizing the classifiers of different depths with the Hedge algorithm Freund and Schapire (1997). ODNN initializes with an overcomplete network and automatically adapts the length of the network in an online manner. The network is initialized with maximum L hidden layers, each hidden layer is followed by a softmax classification layer. ODNN works on the principle of online learning with expert advice, where the experts are the DNNs with varying depths. The final prediction of this ODNN model is a weighted combination of classfiers at depth 0, 1, ..., L. The weight of each classifier at depth L ($\alpha^{(l)}$) is learnt during the learning procedure of ODNN model and also shared with the global server. The global server aggregates and averages these weights ($\alpha^{(l)}$) of classifiers along with the weights of the layers of each ODNN model. In the training phase of each ODNN model, we set the binary cross-entropy loss function as the optimization objective. Since most of our datasets are imbalanced, we use the class weighting module when training the ODNN models. When the ratio between positive and negative class is 1:p, we force the ODNN model to give p times more importance to the positive class instances than the negative class instances using the class weighting module.

5.2 Step B–C: drift detection

The Early drift Detection Method (EDDM) (Baena-Garcıa et al., 2006) (Fig. 2B) maintains a sliding window of variable length to store the most recent instances of the data stream, and is able to automatically detect and adjust the size of the window according to the current rate of change. EDDM keeps track of the average distance between two classification errors (e_j), its standard deviation (sd_j), maximum-average error distance (e_max_j), and the maximum standard deviation (sd_max_j). The average error distance at the j^{th} error (e_j) is the average number of examples between two classification errors as presented in Eq. (4) where dis_i is the number of examples between the current error and the previous error, e_{i-1} is the average error distance calculated when the previous error occurred, and n_{ei} is the number of classification errors seen so far. The standard deviation of average error distance (sd_j) is calculated using Eq. (5). In this equation, var_j is the running variance of average error distance. This drift detection method defines the threshold η shown in Eq. (6) to ensure the detection of concept drifts. When left hand side of this relation exceeds the predefined threshold η, EDDM declares that a concept drift has occurred.

$$e_j = \sum_{i=0}^{j} \frac{dis_i - e_{i-1}}{n_{ei}} \tag{4}$$

$$sd_j = \sqrt{\frac{var_j}{n_{ej}}} \quad and \quad var_j = \sum_{i=0}^{j} (dis_i - e_i) * (dis_i - e_{i-1}) \tag{5}$$

$$\frac{e_j + 2 * sd_j}{e\_max_j + 2 * sd\_max_j} < \eta \tag{6}$$

When EDDM identifies a concept drift, it triggers the clearing of the sliding window. Subsequently, a new window is initialized to store the upcoming instances (as illustrated in Algorithm 1: lines 8 to 12) (shown in Fig. 2C).

5.3 Step D–E: discrimination detection and mitigation

By employing the prequential evaluation strategy, the model's discriminatory behavior is measured (depicted in Fig. 2D, as described in Algorithm 1: line 13) using one of the fairness notions mentioned earlier, such as Stp or Eqop.

We hypothesize that discrimination is often deeply rooted in training data, due to the non-trustworthy labelling or the selection bias. Therefore, we propose a data augmentation-based strategy, the Continuous Fairness-aware Synthetic Oversampling Technique (CFSOTE), to mitigate discrimination. This is an adaptation of the Continuous Synthetic Minority Oversampling Technique (CSMOTE) (Bernardo et al., 2020). The proposed method performs data augmentation using the sliding window of instances of each client maintained by the concept drift detector EDDM (Algorithm 1: lines 14 to 27). Then, the local online learner is trained using the newly synthesized instances (X_syn, Y_syn) (Algorithm 1: lines 28 to 29). For data augmentation, we divide each client's training dataset based on the output class (positive class: C_+, negative class: C_-) and the sensitive attribute ($P, \bar{P}$) into four groups: $N(C_-, P)$, $N(C_-, \bar{P})$, $N(C_+, P)$, $N(C_+, \bar{P})$.

Real-world datasets often suffer from the inherent problem of class imbalance. Most fairness-aware learning methods disregard the importance of class imbalance and attempt to mitigate discrimination at the cost of the true-positive rate of the minority class, resulting in poor balanced accuracy. We use class weighting module to address this issue of class imbalance. Furthermore, our discrimination mitigation strategy itself has the ability to improve and maintain balanced accuracy.

For each client, in every communication round, we use prequential evaluation to train the local ODNN model. Through prequential evaluation we keep track of the discrimination score (disc: Stp or Eqop) over the stream. If the discrimination score exceeds user-defined threshold ε, then we up-sample certain groups ($N(C_+, P)$, $N(C_-, \bar{P})$) of data contained in the local sliding window maintained by EDDM to reduce the discrimination embedded in the dataset. The groups are chosen for upsampling based on the number of total positive predictions and the total number of positive labels in the data stream. If the number of positive predictions is less than or equal to the total number of positive labels in the data stream then we upsample the positive protected group $N(C_+, P)$ by a proportion (λ) of negative non-protected group ($N(C_-, \bar{P})$) using CFSOTE. Otherwise, we increase the number of samples in the negative non-protected group $N(C_-, \bar{P})$ by a proportion (λ) of positive protected group ($N(C_+, P)$) using CFSOTE. The algorithmic details of CFSOTE are given in Sect. 5.3.1. The up-sampling proportion λ is calculated through the formula given in Eq. (7), where *disc* is the discrimination score (Stp or Eqop) measured through prequential evaluation of the local ODNN model. $\lambda_{initial}$ and $disc_{tol}$ are hyperparameters. The parameter $disc_{tol}$ controls the effect of *disc* on λ. The higher the value of $disc_{tol}$ the less will be the effect of *disc* on λ and vice versa.

$$\lambda = \lambda_{initial} * (1 + (disc/disc_{tol})) \tag{7}$$

FAC-Fed handles positive discrimination (discrimination towards the protected group) as well as negative discrimination (discrimination towards the non-protected group). To handle negative discrimination, we swap the roles of the protected and non-protected attribute and the rest of the algorithm remains the same.

Algorithm 2 Continuous Synthetic Oversampling Technique (CFSOTE)

```
Input: instances group to be up sampled (instances_pool1), number of samples to be
    synthesized (m: computed in Algorithm 1), instances group from which nearest
    neighbors need to be sought (instances_pool2), number of random samples to
    be selected (z), number of nearest neighbors to be sought (k), KNN procedure
    (KNN)
Output: $m$ newly synthesized samples
$rand\_z\_samples \leftarrow random.sample(instances\_pool1,\ z)$   ▷ Random selection of z samples
$b \leftarrow int(m/z)$   ▷ Number of samples to be synthesized for each randomly selected sample
for $r\_sample_i$ in $rand\_z\_samples$ do
    $nn\_sample \leftarrow KNN.neighbors(r\_sample_i, k, instances\_pool2)$
    while $j < b$ do
        while $l < k$ do
            $nn\_sample_{il} := l^{th}$ nearest neighbor of $r\_sample_i$
            $sample\_new \leftarrow r\_sample_i + (r\_sample_i - nn\_sample_{il}) * R[0, 1]$
        end while
        $X\_syn.append(sample\_new)$
        $Y\_syn.append(sample\_new.label)$
    end while
end for
```

5.3.1 Continuous fairness-aware synthetic oversampling technique (CFSOTE)

CFSOTE is an adaptation of the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002). We propose this algorithm to upsample a selected group ($N(C_+, P)$, $N(C_-, \bar{P})$) from the local sliding window of instances maintained by EDDM. Algorithm 2 explains the procedure we follow for upsampling a selected group. In contrast to the traditional SMOTE algorithm, we do not select all samples of the selected group for up-sampling, but only z samples from the group, where z is a hyperparameter (Algorithm 2: line 1). For each selected sample, we need to generate *m/z* i.e., b (Algorithm 2: line 2) new samples by linear interpolation between the selected sample and its k nearest neighbors, where m is computed in Algorithm 1. k and z are hyperparameters. Nearest neighbors are sought utilizing the K-Nearest Neighbors (KNN) algorithm (Piegl & Tiller, 2002) (Algorithm 2: line 4). KNN calculates the distance between the queried sample and other data samples using the Euclidean distance metric. It then sorts the data samples in ascending order according to their respective distance from the queried sample and returns the first k samples.

The predictions of the classification model should be independent of the sensitive attribute, which eventually leads the model to the ultimate goal of achieving fairness in its decisions. Therefore, we assume that the samples belonging to the positive protected group $N(C_+, P)$ and to the positive non-protected group $N(C_+, \bar{P})$ are in close proximity to each other, with the only difference of the sensitive attribute. Therefore, if we need to up-sample the positive protected group $N(C_+, P)$, we find the nearest neighbors in the search space which includes the positive protected group $N(C_+, P)$ as well as the positive non-protected group $N(C_+, \bar{P})$ (Algorithm 1: instances_pool2 = ($N(C_+, P)$ & $N(C_+, \bar{P})$)). We assign the protected value to the sensitive attributes of the newly synthesized instances. Figure 3 illustrates the CFSOTE method proposed for up-sampling positive protected group ($N(C_+, P)$). However, if we want to up-sample the negative non-protected group $N(C_-, \bar{P})$, we select only the nearest neighbors from the group itself (Algorithm 1: instances_pool2= $N(C_-, \bar{P})$). We do not include the negative protected group $N(C_-, P)$ in this search space because datasets have non-trustworthy labelling therefore, there is a possibility that many samples belonging to the negative protected group $N(C_-, P)$ are biasedly labelled as negatives.

Once the nearest neighbors are sought, we perform linear interpolation between the queried sample and its nearest neighbors to synthesize new samples and assign the protected value to the sensitive attributes of all the newly synthesized instances (Algorithm 2: lines 5 to 8).

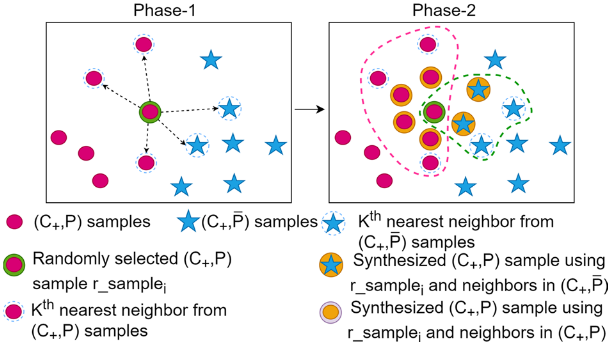
Phase-1
Phase-2
(C+,P) samples
(C+,P̄) samples
Kth nearest neighbor from (C+,P̄) samples
Randomly selected (C+,P) sample r_samplei
Synthesized (C+,P) sample using r_samplei and neighbors in (C+,P̄)
Kth nearest neighbor from (C+,P) samples
Synthesized (C+,P) sample using r_samplei and neighbors in (C+,P)

Fig. 3 An illustration of the Continuous Fairness-aware Synthetic Over Sampling Technique (CFSOTE) for up-sampling $N(C_+, P)$; KNN algorithm finds K nearest neighbors of the randomly selected sample r_sample_i from the groups $N(C_+, P)$ and $N(C_+, \bar{P})$; *m/k* newly synthesized samples with sensitive attribute as P added in the group $N(C_+, P)$

5.4 Step F–G: global server

Upon receiving a request from the global server, the clients promptly share their individual local learning parameters. Subsequently, the server performs parameter aggregation and averaging (as depicted in Fig. 2F) using Eq. (1). The resulting averaged weights of the clients are then used to update the global ODNN model (shown in Fig. 2G). The updated global parameters (w_g^{l+1}) are transmitted to the selected clients for the subsequent communication round.

6 Experimental setup

6.1 Hyperparameters selection

For concept drift detection, we chose the value of η as 0.9 for Eq. (6), as suggested by Baena-Garcıa et al. (2006). For the ODNN model, we performed a grid search and initialized each model with a maximum of $L = 5$ hidden layers and 40 neurons per layer. If we increase these values, the performance of ODNN remains the same; however, the performance degrades if we decrease these numbers. For CFSOTE, we performed a grid search for each dataset and choose the value 5 for k and 5 for z. Since we are upsampling based on a window of instances, k and z are bounded by the current size of the instance group to be upsampled. If we decrease the values of k and z, then the newly synthesized instances will most likely be near duplicates of randomly selected samples; however, if we increase these values, then the performance of the framework will remain comparable. For Eq. (7), we choose the value 0.2 for $disc_{tol}$ and the value 0.05 for $\lambda_{initial}$. These values of $disc_{tol}$ and $\lambda_{initial}$ keep the effect of discrimination score on λ in a moderate range to avoid the undesirable synthesis of large number of instances, which can lead to a high reverse discrimination score.

6.2 Datasets

We evaluate the proposed methodology using a range of real world datasets including Bank Marketing (Bank M.) (Bache & Lichman, 2013), Law School (Law S.) (Wightman, 1998), Default (Bache & Lichman, 2013), and Adult Census (Adult C.) (Bache & Lichman, 2013). These datasets vary in their dimensionality (#Inst.), number of attributes (#Attr.), sensitive attribute (Sen. att.), and imbalance ratio (Im. ratio); the details are presented in Table 1. To adapt the datasets to FL environment, we randomly split each dataset into 3 and 5 clients. Most of the datasets (except Bank M.) used in this work are static datasets, therefore, to ensure reliability we report the results as the average of results obtained by experiments performed on 10 random shuffles of each static dataset. To demonstrate the ability of FAC-Fed to handle non-IID data, we also distribute each dataset among three clients, based on a particular attribute. We choose ‘age’ attribute for splitting Bank M., Default, and Adult C. datasets and ‘income’ attribute for splitting Law S. dataset. These attribute choices are deliberate, as they ensure that each client hosts a distinct data distribution, thus establishing the non-IID nature of the data.

Table 1 Description of datasets

Dataset	#Inst	#Attr.	Sen. att	P	$\bar{P}$	Im. ratio (pos:neg)	Positive label
Bank M. (Bache & Lichman, 2013)	41,188	16	Marital status	Married	Single	1:7.87	Subscription
Law S. (Wightman, 1998)	18,692	11	Gender	Female	Male	1:3.5	Pass bar
Default (Bache and Lichman ,2013)	30,000	23	Gender	Female	Male	1:3.52	Default payment
Adult C. (Bache and Lichman, 2013)	45,175	14	Gender	Female	Male	1:3.0	> 50 K

6.3 Baselines

This section is dedicated to explaining the details of the baseline methods employed for comparison with our proposed approach. To the best of our knowledge, our work is the first attempt towards federated adaptation for fairness and concept drift-aware stream classification. Therefore, we lack fairness aware federated baselines for streaming data to compare our results against. Nonetheless, we have conducted a comparison of the centralized version of our methodology with state-of-the-art centralized stream classification methods. This enables us to assess the performance and efficacy of our approach in a centralized setting.

- *CSMOTE* Bernardo et al. (2020) is not fairness-aware, but it is designed to handle class imbalance in a non-stationary environment by re-sampling the minority class in a defined window of instances.
- *Fairness Aware Hoeffding Tree (FAHT)* Zhang et al. (2019) is a fairness-aware adaptation of Hoeffding tree. It incorporates the fairness gain (Stp score) along with the information gain into the partitioning criteria of the decision tree. This model is not able to deal with class imbalance and concept drifts and is not agnostic with respect to fairness notion; therefore, we report the results only for the case of Stp based optimization.
- *FABBOO* Iosifidis and Ntoutsi (2020) is an online boosting approach that handles class imbalance by monitoring class ratios in an online fashion. It employs boundary adjustment methods to handle discrimination.
- *AC-Fed* is the proposed federated adaptation for concept drift-aware stream classification. This method is incapable to handle fairness issues.
- *FAC-Fed* is the proposed fairness and concept drift-aware federated adaptation for stream classification.

6.4 Evaluation metrics

We evaluate our proposed method for both utility and fairness. Since almost all datasets used in this study are imbalanced therefore we use the evaluation metric *“balanced accuracy”* to measure the utility of the proposed model. We also use *“gmean”* to measure the effectiveness of proposed method. To gauge the discriminatory behavior of FAC-Fed, we use two fairness notions: statistical parity (Stp) and equal opportunity (Eqop). The details of the fairness notions are already explained in Sect. 4.

7 Results and discussion

We perform experiments on a set of real-world datasets. For each dataset, we have presented the results for the random distribution of data among 3 and 5 clients (R3C, R5C). All the evaluation metrics obtained by FAC-Fed with $disc = Stp$ and $disc = Eqop$ are presented in Tables 2, 3, respectively. From Table 2, we can see that FAC-Fed obtained high balanced accuracy and gmean while keeping Stp score between 0.002 and 0.008 for both R3C and R5C data splits of all datasets. Similarly, from Table 3, we can observe that FAC-Fed can achieve high balanced accuracy and gmean while keeping the Eqop score under

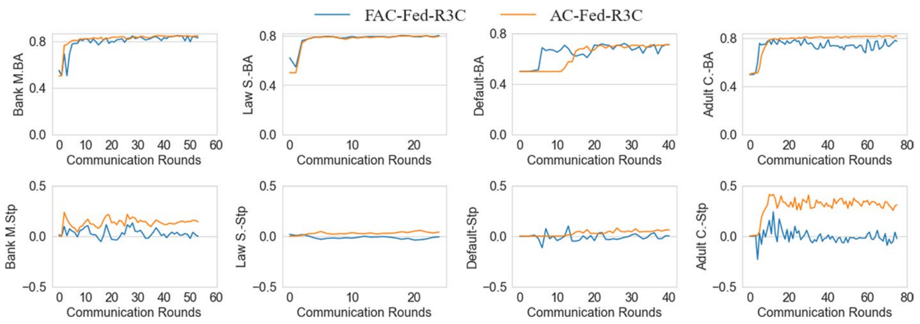

Fig. 4 Comparison of Balanced accuracy (BA) and Statistical parity (Stp) achieved by FAC-Fed and AC-Fed through all communication rounds for Bank M., Law S., Default, and Adult C. datasets with R3C data split

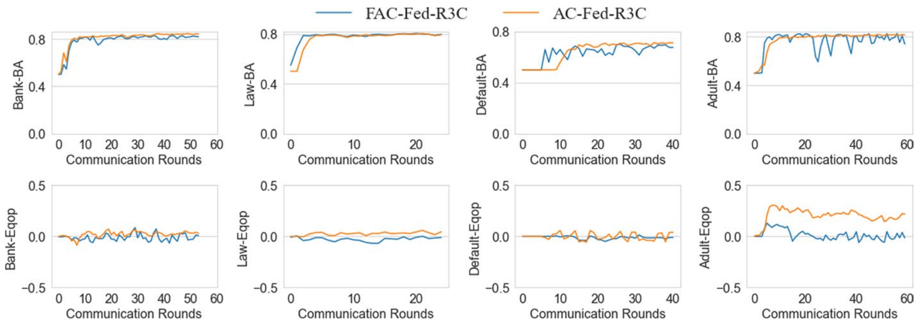

Fig. 5 Comparison of Balanced accuracy (BA) and Equal Opportunity (Eqop) achieved by FAC-Fed and AC-Fed through all communication rounds for Bank M., Law S., Default, and Adult C. datasets with R3C data split

Table 2 Performance measures obtained by proposed method FAC-Fed for Statistical Parity (Stp). Note that RnC implies random split of dataset among *n* clients and Attr3C denotes attribute-based distribution of data among 3 clients

Dataset	R3C			R5C			Attr3C		
	BA	Gmean	Stp	BA	Gmean	Stp	BA	Gmean	Stp
Bank M.	0.8284	0.8282	– 0.0021	0.8253	0.8251	0.0077	0.8383	0.8355	0.0198
Law S.	0.7874	0.7871	– 0.0027	0.7930	0.7884	– 0.0083	0.7972	0.7972	0.0192
Default	0.7114	0.6848	0.0029	0.6643	0.6089	– 0.0089	0.6896	0.6851	– 0.0094
Adult C.	0.7761	0.7757	– 0.0037	0.7748	0.7742	0.0035	0.7580	0.7353	0.0036

0.007 for all datasets and all data splits. From Tables 2, 3, we can deduce that FAC-Fed is agnostic with respect to the notion of fairness used for optimization, since it achieves similar balanced accuracy and gmean while maintaining very low discrimination scores in both cases when we use Stp and Eqop as the optimization criteria. We assess the efficacy of proposed framework against non-IID data by distributing data among three clients based on a

Table 3 Performance measures obtained by proposed method FAC-Fed for Equal Opportunity (Eqop). Note that R*n*C implies random split of dataset among *n* clients and Attr3C denotes attribute-based distribution of data among 3 clients

Dataset	R3C			R5C			Attr3C		
	BA	Gmean	Eqop	BA	Gmean	Eqop	BA	Gmean	Eqop
Bank M.	0.8184	0.8177	0.0073	0.7964	0.7880	– 0.0095	0.8349	0.8318	– 0.004
Law S.	0.7924	0.7918	0.0037	0.8019	0.8014	– 0.0029	0.7872	0.7844	0.0099
Default	0.7017	0.6953	– 0.0042	0.6972	0.6744	– 0.0056	0.6896	0.6851	– 0.0094
Adult C.	0.8133	0.8129	– 0.0036	0.8211	0.821	0.0024	0.7693	0.7524	0.0097

specific attribute. From Tables 2, 3, we observe that FAC-Fed maintains its superior performance in terms of both utility and discrimination mitigation when the data is distributed based on a specific attribute among the clients. This highlights the framework's capability to effectively handle non-IID data.

Figure 4 shows a comparison of the balanced accuracy and Stp score achieved by FAC-Fed and AC-Fed for R3C split of all datasets. From this figure, we can see that FAC-Fed achieves comparable balanced accuracy as AC-Fed for all datasets and maintains it across all communication rounds. Moreover, the Stp score achieved by FAC-Fed is much lower than that of AC-Fed. Similarly, Fig. 5 shows a comparison of the Eqop score and balanced accuracy achieved by FAC-Fed and AC-Fed for R3C split of all datasets. This figure illustrates that FAC-Fed achieves comparable balanced accuracy as AC-Fed for all datasets and maintains it across all communication rounds. However, the Eqop score achieved by FAC-Fed is much less than that of AC-Fed. This proves that the proposed strategy to mitigate discrimination has minimal impact on the utility of the proposed federated framework.

To the best of our knowledge, this is the first attempt towards fairness and concept drift-aware stream classification. Therefore, we compare the performance measures achieved by the centralized version of FAC-Fed with three centralized stream classification models (FABBOO, FAHT, CSMOTE). The results obtained by prequential evaluation

Table 4 Comparison of performance measures obtained by proposed method FAC-Fed and the baseline methods in a centralized environment for statistical parity, with best and second best values shown in bold and italic

Method	Eval. metric	Bank M.	Law S.	Default	Adult C.
FABBOO	BA	0.7849	0.6543	*0.6593*	0.7545
	Gmean	0.7737	0.5852	*0.6124*	0.7479
	Stp	*0.0022*	*0.0046*	*0.0093*	*0.0025*
FAHT	BA	0.6685	0.5433	0.6299	0.7262
	Gmean	0.5968	0.305	0.5399	0.6938
	Stp	0.0257	0.0087	0.0171	0.1637
CSMOTE	BA	**0.8291**	*0.7667*	0.591	*0.7797*
	Gmean	**0.8287**	*0.7666*	0.5503	*0.7791*
	Stp	0.0829	0.0216	0.0246	0.3237
FAC-Fed	BA	*0.8246*	**0.778**	**0.6883**	**0.8194**
	Gmean	*0.8244*	**0.7780**	**0.6730**	**0.8123**
	Stp	– **0.0009**	– **0.0006**	**0.0036**	– **0.0005**

Table 5 Comparison of performance measures obtained by proposed method FAC-Fed and the baseline methods in a centralized environment for statistical parity, with best and second best values shown in bold and italic

Method	Eval. Metrics	Bank M.	Law S.	Default	Adult C.
FABBOO	BA	0.7649	0.6420	*0.6577*	0.7682
	Gmean	0.7452	0.5593	*0.6131*	0.7545
	Eqop	**0.0012**	0.0455	**0.0014**	*0.0186*
CSMOTE	BA	**0.8392**	*0.7672*	0.5927	*0.7758*
	Gmean	**0.8381**	*0.7672*	0.559	*0.7746*
	Eqop	0.0229	0.0219	0.0237	0.1527
FAC-Fed	BA	*0.8298*	**0.8136**	**0.6872**	**0.8199**
	Gmean	*0.8298*	**0.8093**	**0.6605**	**0.8007**
	Eqop	− *0.0021*	− **0.0035**	− *0.0081*	− **0.0005**

of centralized FAC-Fed and the competing baselines with Stp and Eqop as the optimization criteria are shown in Tables 4, 5 respectively. From Table 4 we can see that FAC-Fed achieves the best Stp score, balanced accuracy and gmean for all datasets except for the Bank Marketing dataset. For the Bank Marketing dataset, the centralized version of FAC-Fed follows the performance of CSMOTE in terms of balanced accuracy and gmean with a difference of only 0.45% and 0.43%, respectively. However, the Stp score achieved by FAC-Fed (−0.0009) is much lower compared to that of CSMOTE (0.0829). Similarly, in Table 5, for all datasets with Eqop as the optimization criterion, we can observe that FAC-Fed achieves the best balanced accuracy, gmean, and Eqop score compared to all baselines except the Bank Marketing dataset. For Bank Marketing dataset, FAC-Fed achieves comparable balanced accuracy and gmean as that achieved by CSMOTE. However, the Eqop score of FAC-Fed (−0.0021) is much lower than that of CSMOTE (0.0229). For Bank dataset, FABBOO achieves the best Eqop score (0.0012), FAC-Fed follows it with a close margin, nevertheless, FAC-Fed achieved 6.49% higher balanced accuracy than that achieved by FABBOO. With Default dataset, FABBOO achieves best Eqop score (0.0014) whereas FAC-Fed follows it by a narrow margin (−0.0081), while the balanced accuracy and gmean values are 2.95% and 4.74% higher than those of FABBOO, respectively. The difference between the balanced accuracy and the gmean achieved by FABBOO is large for most datasets, suggesting that FABBOO achieves a lower discrimination score at the expense of either true-positive rate or the true-negative rate. In contrast, FAC-Fed achieves much lower discrimination scores (Stp, Eqop) compared to FABBOO, while the balanced accuracy and gmean reported by FAC-Fed are close.

Figures 6, 7 show a comparison of performance measures obtained by FABBOO and centralized FAC-Fed with prequential evaluation over the entire data stream for all datasets. From these plots we can observe that although the fairness performance of FABBOO and FAC-Fed are quite similar yet FAC-Fed achieves higher balanced accuracy than FABBOO. Results show that FAC-Fed achieves high balanced accuracy, Stp and Eqop even in the centralized environment, although, it is designed for a federated environment. If we compare the results of federated version of FAC-Fed and centralized version of FAC-Fed, we observe that the difference in performance measures is not substantial. For instance, in Table 2, for the Bank M. dataset, the federated FAC-Fed achieved balanced accuracies of 82.84% and 82.51%, as well as Stp scores of −0.0021 and 0.007 for the R3C and R5C

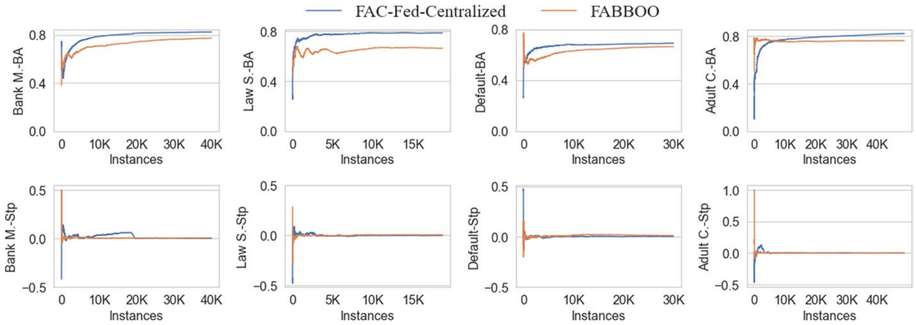

Fig. 6 Comparison of Balanced accuracy (BA) and Statistical parity (Stp) achieved by centralized version of FAC-Fed and FABBOO with prequential evaluation through out the stream for Bank M., Law S., Default, and Adult C. dataset

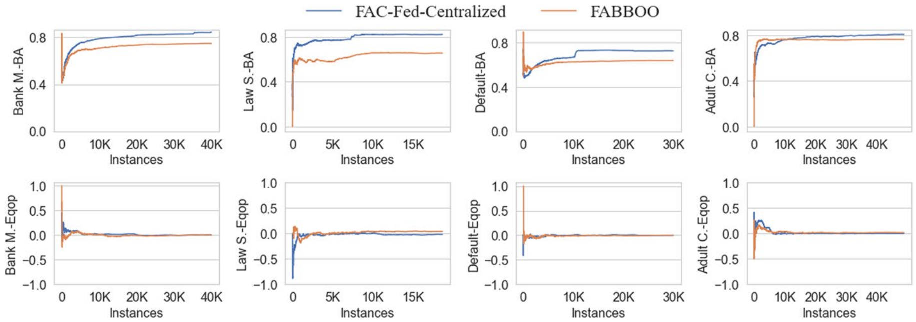

Fig. 7 Comparison of Balanced accuracy (BA) and Equal Opportunity (Eqop) achieved by centralized version of FAC-Fed and FABBOO with prequential evaluation through out the stream for Bank M., Law S., Default, and Adult C. dataset

splits of the dataset, respectively. On the other hand, the centralized version of FAC-Fed achieved a balanced accuracy of 82.46% (Table 4) and an Stp score of 0.0009, which are very close to the results obtained by the federated version of FAC-Fed. A similar trend can be observed for the Adult C., Default, and Law S. datasets, indicating that the proposed methodology is robust and reliable in both federated and centralized environments.

8 Conclusion

To the best of our knowledge, we proposed a pioneering work in the domain of federated stream learning that mitigates the discrimination inherent in the client data while improving the framework's predictive performance (FAC-Fed). The experimental results demonstrate the effectiveness of FAC-Fed in terms of predictive performance and fairness and highlight the following key advantages of the proposed framework:

- FAC-Fed is able to reduce the discrimination score and maintain it over the stream.
- FAC-Fed is agnostic in nature with respect to the fairness notion used during optimization.

- For datasets with severe class imbalance, FAC-Fed is able to ensure significantly better predictive performance while maintaining low discrimination scores.
- FAC-Fed demonstrates consistent predictive and discrimination mitigation performance even with non-IID data.
- Fairness is ensured for each client.
- The proposed framework has the potential to be used as a centralized fairness-aware learning framework as well. For all the datasets, the centralized version of proposed method is able to ensure significantly better predictive performance than the competing baselines while maintaining low discrimination scores.

With the advances in sensor networks, distributed and heterogeneous data sources generate data regularly and dynamically. A possible extension of the proposed work could be to adapt the FAC-Fed to asynchronously train large number of clients with continuously arriving streaming data.

A: Non-IID distribution of data

In a federated setup, non-iid (non-independent and identically distributed) data refers to the scenario where the data distribution across clients is not uniform or homogeneous. Each client may have a different data distribution, different data characteristics, or different data proportions. In Table 6, we present positive class to negative class ratio of data distributed among 3 clients based on a particular attribute. From this table we can observe that for each dataset, each client hosts a different distribution of data, which makes the data non-IID. Also, for each dataset, the characteristics of data also vary, for example for Law S. dataset, client 1 hosts all clients data who have income group '1' or '2', client 2 hosts local data with income group '3', and local data of client 3 comprises all instances with income group '4' or '5'.

In the Results and discussion section (Sect. 7), we demonstrate that the proposed methodology achieves high balanced accuracy and low discrimination score even for non-IID data.

Table 6 Positive class to negative class ratio of local data of 3 clients

Dataset	Client 1 (pos_class: neg_class)	Client 2 (pos_class: neg_class)	Client 3 (pos_class: neg_class)
Bank M	1:8.14	1:4.64	1:8.45
Law S	1:6.45	1:8.13	1:10.92
Default	1:3.37	1:3.93	1:3.22
Adult C	1:1.85	1:17.94	1:2.74

Note that this data distribution is done based on attribute, 'age' attribute for splitting Bank M., Default, and Adult C. datasets and 'income' attribute for splitting Law S. dataset

B: Evaluation metrics

We used balanced accuracy and gmean to meausure the predictive performance of the proposed framework and competing baselines. The mathematical representation of balanced accuracy and gmean are illustrated in Eqs. (10) and (11).

$$Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

$$Specificity = \frac{TN}{TN + FP} \tag{9}$$

$$BalancedAccuracy = \frac{Sensitivity + Specificity}{2} \tag{10}$$

$$Gmean = \sqrt{Sensitivity * Specificity} \tag{11}$$

C: Hyper-parameters selection

We used grid search for each dataset to choose $disc_{tol}$ and $\lambda_{initial}$ to avoid undesirable synthesis of large number of instances, which can lead to a high reverse discrimination. Figures 8, 9, 10, and 11 show effect of different values of $disc_{tol}$ and $\lambda_{initial}$ on balanced accuracy and discrimination score (Eqop) achieved by FAC-Fed for Law School dataset with R3C split. Law School dataset exhibits negative discrimination i.e., discrimination towards the non-protected group.

The proposed methodology helps in improving and maintaining high balanced accuracy. $disc_{tol}$ allows controlled synthesis of new instances to mitigate the discrimination (either positive or negative). Figure 8 shows that if we keep the value of $disc_{tol} = 0.005$ then balanced accuracy takes nearly 15 communication rounds to become stable compared to only nearly 6 communication rounds with $disc_{tol} = 0.2$. This happened because such a low $disc_{tol}$ led to synthesis of insufficient number of instances hence slowed down the process

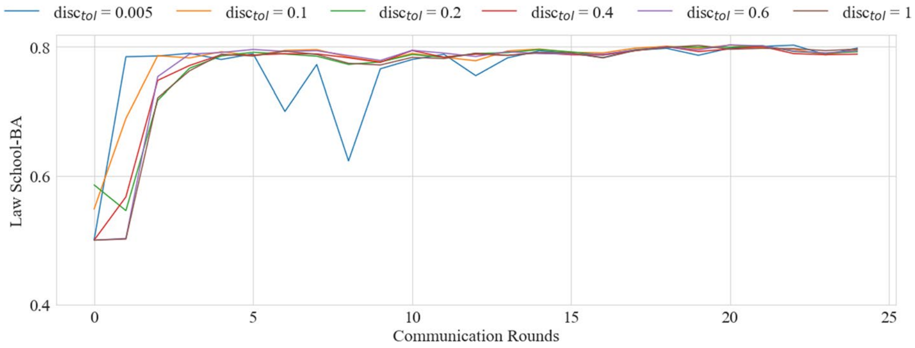

Fig. 8 Balanced Accuracy (BA) achieved by proposed method FAC-Fed for different values of $disc_{tol}$ with $\lambda_{initial} = 0.05$ over all communication rounds for R3C split of Law School dataset. BA achieved for all values of disc$_{tol}$ is nearly similar, therefore, $disc_{tol}$ is chosen based on the Eqop plots

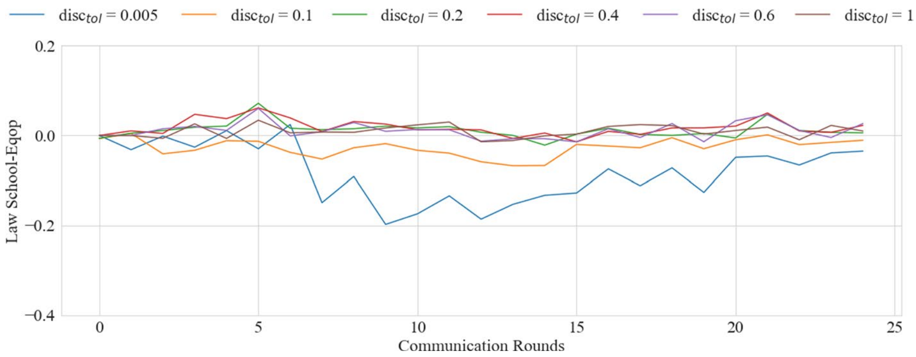

Fig. 9 Equal Opportunity (Eqop) values achieved by proposed method FAC-Fed for different values of $disc_{tol}$ with $\lambda_{initial} = 0.05$ over all communication rounds for R3C split of Law School dataset. We choose $disc_{tol} = 0.2$ because this value is keeping the discrimination score (Eqop) near to '0' compared to Eqop values achieved with all the other values of $disc_{tol}$

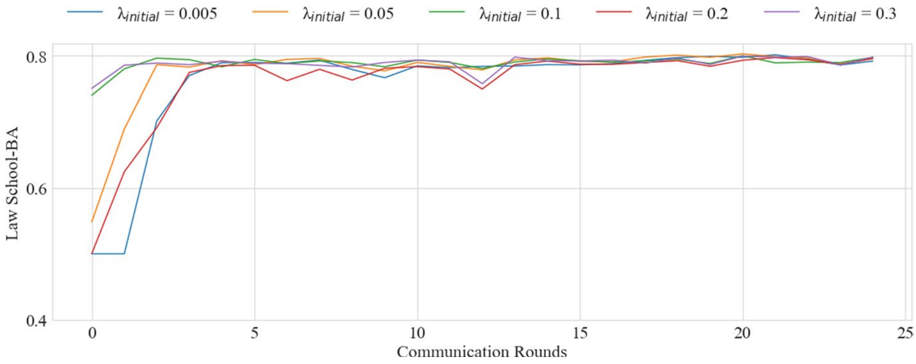

Fig. 10 Balanced Accuracy (BA) achieved by proposed method FAC-Fed for different values of $\lambda_{initial}$ with $disc_{tol} = 0.2$ over all communication rounds for R3C split of Law School dataset. BA achieved for all values of $\lambda_{initial}$ is nearly similar, therefore, $\lambda_{initial}$ is chosen based on the Eqop plots

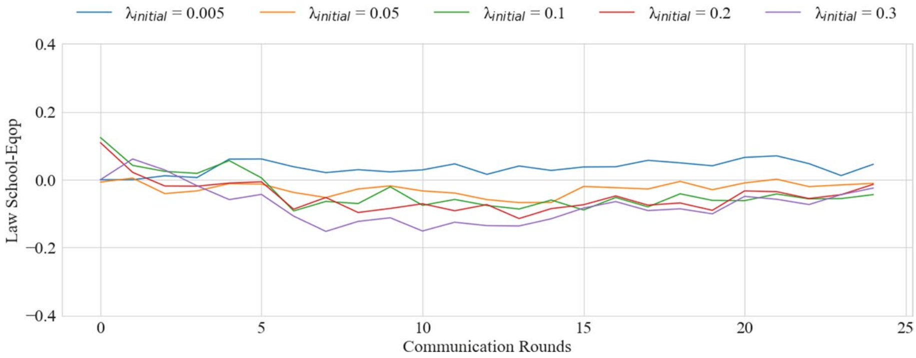

Fig. 11 Equal Opportunity (Eqop) values achieved by proposed method FAC-Fed for different values of $\lambda_{initial}$ with $disc_{tol} = 0.2$ over all communication rounds for R3C split of Law School dataset. We choose $\lambda_{initial} = 0.05$ because this value is keeping the discrimination score (Eqop) near to '0' compared to Eqop values achieved with all the other values of $\lambda_{initial}$

of achieving and maintaining high balanced accuracy. In Fig. 9, it can be seen that this low value of $disc_{tol}$ did not help in mitigating discrimination (Eqop in this case). We choose $disc_{tol} = 0.2$ because this value is keeping the discrimination score (Eqop) near to '0' compared to Eqop values achieved with all the other values of $disc_{tol}$ as shown in Fig. 9.

Figure 10 shows that, changing value of $\lambda_{initial}$ does not significantly change the balanced accuracy. We choose $\lambda_{initial} = 0.05$ because this value is keeping the discrimination score (Eqop) near to '0' compared to Eqop values achieved with all the other values of $\lambda_{initial}$ as shown in Fig. 11.

Author contribution MB was in charge of writing the code, performing the experimental evaluations, writing the first draft, and editing the final draft of the manuscript. MF supervised the development of the research and provided feedback at all the stages of the process till the final draft of the manuscript. WN contributed in proof reading the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. The Lower Saxony Ministry of Science and Culture (Niedersächsische Ministerium für Wissenschaft und Kultur). This research was partially funded by the Lower Saxony Ministry of Science and Culture, Germany (Niedersächsische Ministerium für Wissenschaft und Kultur) and the Federal Ministry of Education and Research (BMBF), Germany under the project LeibnizKILabor with grant No. 01DD20003.

Data availability All the datasets used in this study are publicly available at Bache and Lichman (2013); Wightman (1998).

Code availability The code is available at the following link: https://github.com/badarm/FAC-Fed.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval and Consent to participate This article does not require permission for ethics approval or consent to participation as this work is based on all the publicly available datasets.

Consent for publication The authors of this manuscript consent to its publication.

References

Abdellatif, A. A., Mhaisen, N., Mohamed, A., Erbad, A., Guizani, M., Dawy, Z., & Nasreddine, W. (2022). Communication-efficient hierarchical federated learning for IoT heterogeneous systems with imbalanced data. *Future Generation Computer Systems, 128*, 406–419.

Bache, K., Lichman, M. (2013). Uci machine learning repository.

Baena-Garcıa, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavalda, R., Morales-Bueno, R. (2006). Early drift detection method. In *Fourth International Workshop on Knowledge Discovery from Data Streams* (Vol. 6, pp. 77–86).

Bernardo, A., Gomes, H.M., Montiel, J., Pfahringer, B., Bifet, A., Della Valle, E. (2020). C-smote: Continuous synthetic minority oversampling for evolving data streams. In *IEEE big data* (pp. 483–492). IEEE.

Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *SIGSAC* (pp. 1175–1191).

Calders, T., Kamiran, F., Pechenizkiy, M. (2009). Building classifiers with independency constraints. In *2009 ICDM Workshops* (pp. 13–18). IEEE.

Chakraborty, J., Majumder, S., Menzies, T. (2021). Bias in machine learning software: Why? How? What to do?. In *ESEC/FSE* (pp. 429–440).

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

Cui, S., Pan, W., Liang, J., Zhang, C., & Wang, F. (2021). Addressing algorithmic disparity and performance inconsistency in federated learning. *Advances in Neural Information Processing Systems, 34*, 26091–26102.

Dobbe, R., Gilbert, T. K., & Mintz, Y. (2021). Hard choices in artificial intelligence. *Artificial Intelligence, 300*, 103555.

Du, W., Xu, D., Wu, X., Tong, H. (2021). Fairness-aware agnostic federated learning. In *SDM* (pp. 181–189).

Emelianov, V., Gast, N., Gummadi, K. P., & Loiseau, P. (2022). On fair selection in the presence of implicit and differential variance. *Artificial Intelligence, 302*, 103609.

European Commission. (2016). Reform of EU data protection rules. European Commission.

Fisichella, M., Lax, G., & Russo, A. (2022). Partially-federated learning: A new approach to achieving privacy and effectiveness. *Inf. Sci., 614*, 534–547.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences, 55*(1), 119–139.

Gama, J. (2010). *Knowledge discovery from data streams*. Chapman and Hall/CRC.

Hajian, S., Domingo-Ferrer, J., Monreale, A., Pedreschi, D., & Giannotti, F. (2015). Discrimination-and privacy-aware patterns. *Data Mining and Knowledge Discovery, 29*(6), 1733–1782.

Huang, T., Lin, W., Wu, W., He, L., Li, K., & Zomaya, A. Y. (2020). An efficiency-boosting client selection scheme for federated learning with fairness guarantee. *IEEE Transactions on Parallel and Distributed Systems, 32*(7), 1552–1564.

Iosifidis, V., Ntoutsi, E. (2019). Adafair: Cumulative fairness adaptive boosting. In *CIKM* (pp. 781–790)

Iosifidis, V., Ntoutsi, E. (2020). FABBOO-online fairness-aware learning under class imbalance. In *DS* (pp. 159–174). Springer.

Iosifidis, V., Tran, T. N. H., Ntoutsi, E. (2019). Fairness-enhancing interventions in stream classification. In *DEXA* (pp. 261–276). Springer.

Kamiran, F., Calders, T. (2009). Classifying without discriminating. In *ICCC* (pp. 1–6). IEEE.

Kamiran, F., Calders, T., Pechenizkiy, M. (2010). Discrimination aware decision tree learning. In *ICDM* (pp. 869–874). IEEE.

Kamiran, F., Karim, A., Zhang, X. (2012). Decision theory for discrimination-aware classification. In *ICDM* (pp. 924–929). IEEE.

Kamiran, F., & Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems, 33*(1), 1–33.

Liu, B., Guo, Y., Chen, X. (2021). Pfa: Privacy-preserving federated adaptation for effective model personalization. In *WWW* (pp. 923–934).

Liu, A., Song, Y., Zhang, G., Lu, J. (2017). Regional concept drift detection and density synchronized drift adaptation. In *IJCAI*.

Ma, X., Zhu, J., Lin, Z., Chen, S., & Qin, Y. (2022). A state-of-the-art survey on solving non-iid data in federated learning. *Future Generation Computer Systems, 135*, 244–258.

McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273–1282). PMLR.

Mills, J., Hu, J., & Min, G. (2019). Communication-efficient federated learning for wireless edge intelligence in iot. *IEEE Internet of Things Journal, 7*, 5986–5994.

Misselhorn, C. (2020). Artificial systems with moral capacities? A research design and its implementation in a geriatric care system. *Artificial Intelligence, 278*, 103179.

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., & Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems, 115*, 619–640.

Padala, M., Gujar, S. (2020). Fnnc: Achieving fairness through neural networks. In *IJCAI*.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., Talwar, K. (2016). Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*.

Paragliola, G. (2022). Evaluation of the trade-off between performance and communication costs in federated learning scenario. *Future Generation Computer Systems, 136*, 282–293.

Piegl, L. A., & Tiller, W. (2002). Algorithm for finding all k nearest neighbors. *Computer-Aided Design, 34*(2), 167–172.

Sahoo, D., Pham, Q., Lu, J., Hoi, S. C. H. (2018) Online deep learning: Learning deep neural networks on the fly. 2660–2666

Singh, G., Violi, V., & Fisichella, M. (2023). Federated learning to safeguard patients data: A medical image retrieval case. *Big Data Cogn. Comput., 7*(1), 18.

Verma, S., Rubin, J. (2018). Fairness definitions explained. In *International workshop on software fairness (fairware)* (pp. 1–7). IEEE.

Wei, X., Hou, M., Ren, C., Li, X., & Yue, H. (2022). Mssa-fl: High-performance multi-stage semi-asynchronous federated learning with non-IID data. In G. Memmi, B. Yang, L. Kong, T. Zhang, & M. Qiu (Eds.), *Knowledge science, engineering and management* (pp. 172–187). Cham: Springer.

Wightman, L. F. (1998). *LSAC National Longitudinal Bar Passage Study*. ERIC: LSAC Research Report Series.

Wu, J., Liu, Q., Huang, Z., Ning, Y., Wang, H., Chen, E., Yi, J., Zhou, B. (2021). Hierarchical personalized federated learning for user modeling. In *WWW* (pp. 957–968).

Yang, C., Wang, Q., Xu, M., Chen, Z., Bian, K., Liu, Y., Liu, X. (2021). Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data. In *WWW* (pp. 935–946).

Yang, M., Wang, X., Zhu, H., Wang, H., Qian, H. (2021). Federated learning with class imbalance reduction. In *EUSIPCO* (pp. 2174–2178). IEEE.

Younis, R., & Fisichella, M. (2022). Fly-smote: Re-balancing the non-iid iot edge devices data in federated learning system. *IEEE Access, 10*, 65092–65102.

Yu, H., Liu, Z., Liu, Y., Chen, T., Cong, M., Weng, X., Niyato, D., Yang, Q. (2020). A fairness-aware incentive scheme for federated learning. In *AAAI* (pp. 393–399).

Zafar, M. B., Valera, I., Gomez-Rodriguez, M., & Gummadi, K. P. (2019). Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research, 20*(1), 2737–2778.

Zeng, R., Zhang, S., Wang, J., Chu, X. (2020). Fmore: An incentive scheme of multi-dimensional auction for federated learning in mec. In *ICDCS* (pp. 278–288). IEEE.

Zhang, D. Y., Kou, Z., Wang, D. (2020). Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In *IEEE big data* (pp. 1051–1060).

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

Zhang, W., Ntoutsi, E. (2019). Faht: An adaptive fairness-aware decision tree classifier. In *IJCAI* (pp. 1480–1486).

Zhang, J., Wang, W., Sun, Z., Han, Z. X. Y. (2022). RRCM: A fairness framework for federated learning. *FL-IJCAI'22*

Zhang, L., Wu, Y., Wu, X. (2018). Achieving non-discrimination in prediction. In *IJCAI* (pp. 3097–3103).

Zhang, X., Zhu, X., Wang, J., Yan, H., Chen, H., & Bao, W. (2020). Federated learning with adaptive communication compression under dynamic bandwidth and unreliable networks. *Information Sciences, 540*, 242–262.

Zhu, H., Xu, J., Liu, S., & Jin, Y. (2021). Federated learning on non-IID data: A survey. *Neurocomputing, 465*, 371–390.