Federated Learning Biases in Heterogeneous Edge-Devices - A Case-study

Khotso Selialia
University of Massachusetts, Amherst, USA
kselialia@umass.edu

Yasra Chandio
University of Massachusetts, Amherst, USA
ychandio@umass.edu

Fatima M. Anwar
University of Massachusetts, Amherst, USA
fanwar@umass.edu

ABSTRACT

Critical machine learning applications (medical image guidance, task prediction, anomaly detection) require large amounts of data that could not be sufficiently supplied from a single entity, so multiple edge devices collaboratively train their collected data. But this raises privacy and overhead concerns. Federated learning (FL) can be a promising solution to enable these applications while preserving data privacy and mitigating communication overhead. However, an FL model originating from edge deployments with heterogeneous resources may be biased towards a set of devices. We observe that existing bias mitigation techniques in FL focus mainly on the bias that originates from label heterogeneity (due to the skewed distribution of data). We argue that sample feature heterogeneity due to different feature representations at devices is a major contributor to bias in FL. In this paper, we present an analysis of the bias that arises from **sampling feature heterogeneity**, and analyze the potential of existing performance enhancing techniques (normalization) to overcome bias. Our results demonstrate that normalization techniques do not eliminate bias and motivate the need for dedicated bias mitigation techniques in FL.

ACM Reference Format:
Khotso Selialia, Yasra Chandio, and Fatima M. Anwar. 2022. Federated Learning Biases in Heterogeneous Edge-Devices - A Case-study. In *Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT) (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3560905.3568305

1 INTRODUCTION

Recent developments in Machine Learning (ML) have adopted Federated Learning (FL) to mitigate data privacy issues [13]. FL is a form of distributed computing that sends copies of ML model to the entities where data is generated (known as *client*), performing training iterations locally, and sending the results of the computation to a central server for updating the *global model*. It allows the data to remain in custody of the owner while training model locally on heterogeneous devices and aggregating over a server. However, we argue that the heterogeneity between data collection devices (in terms of sensor specifications) poses the problem of inequity in model performance between clients using a shared global model, often known as **bias**.

We believe that due to these biases, FL models and their decisions could become unfair. Fairness in FL is defined in [7] as uniforming in the accuracy distribution over all clients [14] and maximizing the performance of the worst client [18]. We argue that fairness is impacted by favoritism in the model that is a result of data characteristics inheritable from *heterogeneous sensors*. For example, **label heterogeneity** in data is one of the leading cause of bias in models. It represents skewed distribution of data labels across clients and arises from variations in data collection environment [7]. In this paper, we unveil another data characteristic that induces bias in FL models i.e. **sampling feature heterogeneity**, which refers to the *difference in sample feature vectors* across clients with the same labels [9]. In essence, we observe that unfair models are biased towards clients with favorable characteristics such as large dataset (addressing **label heterogeneity**) and diverse data sets (addressing **sampling feature heterogeneity**). The prior bias mitigation techniques focus mainly on label heterogeneity [9] using different centralized machine learning techniques such as pre-processing [11] and in-processing [12]. Pre-processing with normalization solves the performance degradation problem (not bias) caused by label heterogeneity [9]. This technique enhances the performance of the model making the assumption that all clients classify uniform data (single test data shared across all clients). However, this assumption does not hold for multiple edge devices with variable features data, and hence a subset of clients without diverse data still experience performance loss. In-processing methods work well in mitigating bias because they add a discrimination-aware regularizer to the model optimization formulation [1]. However, these methods are more demanding in terms of resources. For example, on a client with abundant resources, in-processing can help mitigate more bias and move toward a fair model without degrading performance but on the other hand it may cause performance degradation [12] with client that have limited resources. Both of these techniques pose an *accuracy-versus-fariness* and *accuracy -versus-resource trade-off*. Since we are considering distributed ML in context to prevalent edge deployments and their unique characteristics are resource heterogeneity in terms of sensor types and sampling rates as well as limited resources in terms of limited time, compute, power, and bandwidth.

Furthermore, these existing techniques have not explored the effects of **sampling feature heterogeneity** on bias as it is quite challenging to determine key heterogeneous data attributes for commonly used bias mitigation methods [11, 23]. Analyzing data heterogeneity is critical as it leads to unfair decision making in real-world settings where institutions heavily rely on heterogeneous devices to collect private samples such as cancer images to perform cancer diagnosis tasks [15].

To this end, in this paper, we investigate the following research questions: (1) How does sensor heterogeneity (sampling feature

SenSys '22, November 6–9, 2022, Boston, MA, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9886-2/22/11…$15.00
https://doi.org/10.1145/3560905.3568305

heterogeneity) affect the bias in FL models and their applications in settings with resource-constraint edge-devices? Considering the surgical image guidance application, for instance, is it possible for the global model to maintain uniform performance across clients if their training data has non-identical feature representations?; (2) Can existing performance enhancement techniques (*normalization*) for label heterogeneity help mitigate the bias from sampling feature heterogeneity? In enhancing the performance of image guidance in settings with heterogeneous data, can bias mitigation be the bi-product of normalization methods used to enhance the global model's performance label heterogeneity settings?

Contributions. While answering above research questions, we make the following contributions:
(1) We present an empirical study to analyze sampling feature heterogeneity in a real-world privacy-preserving application (surgical task prediction [5]) and a state-of-the-art classification benchmark (CIFAR10 [19]). The analysis include the impacts of feature heterogeneity on the bias (per-client performance) in FL.
(2) We present a detailed analysis on bias mitigation leveraging existing performance enhancing techniques (normalization) while examining *performance-vs-resources* trade-offs. Normalization methods are useful for bias mitigation because they modify raw data by altering the characteristics that lead to the bias [1]. In addition, these methods can be deployed without the dependency on any underlying machine learning algorithm.

2 BACKGROUND

FL uses remote execution in which copies of machine learning algorithms are sent to locations where data are generated (*clients/partitions*), training iterations are performed locally, and the results of the computation are sent to a central server (*aggregator*) in order to update a single *global model*. The FedAvg algorithm [17] is commonly used to update the global model. This algorithm uses model averages of different clients participating in the learning process [17].

2.1 Bias in Federated Learning

Bias is an inequity in classification performance between clients using a shared global model. Data records collected by a distinct client contain characteristics inherent to each client as a result of heterogeneous sensors used to collect data. This heterogeneity could be: **sample feature heterogeneity** where the same label has different feature vectors between partitions [9]; or **label heterogeneity** that results from sensor heterogeneity in terms of sampling rates[7].

Discrimination Index (Φ_α) quantifies the bias of the global ML model toward a particular partition(s) [24] (denoted by Φ_α). Formally, we calculate the discrimination index across partitions i and j as:

$$\Phi_\alpha = L_i(w) - L_j(w) \tag{1}$$

where $L_i(w)$ denotes the model test loss in the classification of test samples related to partition i using the global model (w). Similarly, $L_j(w)$ denotes the loss in classifying all samples related to partition j using the global model (w). Ideally, the ideal discrimination index should be zero (i.e., achieving the same accuracy to classify samples in partitions i and j) [24].

2.2 Normalization in Federated Learning

Data normalization techniques are used to improve the overall performance of the global model in FL settings with label heterogeneity. The commonly used normalization techniques are discussed below.

Batch normalization (BatchNorm) [10] improves the performance of the model by normalizing the input distribution to zero mean and unit variance. This technique estimates the global mean and variance by using data mini-batches since these values are not directly attainable.

Group normalization (GroupNorm) [22] is an alternative to BatchNorm that computes mean and variance by dividing channels of image samples into groups. The computation of per-group mean and variance makes it channel. Other variants of variants of GroupNorm include Layer normalization(LayerNorm) [3] and Instance normalization (InstanceNorm) [9]. LayerNorm computes the mean and variance across all channels for each sample. This is GroupNorm with group size for all channels. InstanceNorm, on the other hand, is GroupNorm with group size of one.

3 METHODOLOGY

In Federated learning, the use of low quality sensors for a subset of clients and high quality HMD sensors for another subset of clients in medical image capture tends to cause the problem of *heterogeneous sample features* across clients. This deficiency is due to noise introduced to the data samples due to the use of low-quality sensors [16]. In addition, utilizing devices with heterogeneous sampling rates to collect data across distinct clients has a high probability of introducing *label heterogeneity* across partitions. Within a fixed time frame, clients with heterogeneous sampling rates collect different numbers of data points of the same label. This difference in sample features across clients (plus label heterogeneity) causes aggregation bias; the aggregator's fusion algorithm to combine client model updates weighs client contributions differently due to the importance of different feature representations [17], leading to a model biased toward supposedly important feature representations.

Therefore, the main goal is to ensure that the global model is unbiased across all partitions. This goal is significant because it will eliminate tangible consequences of biased models in critical applications.

3.1 Procedure

The first step is to partition the dataset into K partitions, each partition containing its local data for training. We vary the total number of samples across clients to simulate label heterogeneity. In addition, we simulate sampling feature heterogeneity by varying sample features across partitions by adding noise of variable levels to samples across partitions. The second step consists of deploying an FL platform to train and obtain a single global model. We use the FLOWER framework [4] as the FL platform. Finally, we analyze the impact of bias as a result of using the global model trained using data collected with heterogeneous devices across partitions.

3.2 Models, Usecase & Datasets

We evaluate image classification applications. In addition, we assess different deep learning model structures and training datasets:

Image classification with ResNet: We used images from the CIFAR10 dataset [19]. The CIFAR10 dataset consists of ten classes with 6000 images per class. These classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The standard train/test split is class-balanced and contains 50000 training images and 10000 test images. We use BatchNorm as the default normalization method and train the model until the error does not change with evenly distributed labels and uniform feature distribution.

Surgical Task Prediction with EfficientNet [5]: We use images from the Cholec80 dataset [20]. Cholec80 contains 80 videos of cholecystectomy surgeries performed by 13 surgeons, complete with phase annotations of the 7 surgical phases for a procedure (25 FPS) defined by a senior surgeon. For all tests using the Cholec80 dataset, 32 videos were used for a train split, 40 videos for a test split, and the remaining 8 videos for a validation split, as in prior work [5]. We use BatchNorm as the default normalization method and train the model until the error does not change with evenly distributed labels and uniform feature distribution.

3.3 Platform

We use the FLOWER framework as the FL platform choice because it is a novel FL framework that supports experimentation with both algorithmic and systems-related challenges in FL [4]. FLOWER enables the training of global models progressively by making clients responsible for generating individual weight-updates for the model based on their local datasets. These updates are then sent to the server, which will aggregate them to produce a better model. The most common aggregation strategy that FLOWER implements is FedAvg (discussed in the previous sections). Finally, the server sends this improved version of the model to each client. A complete cycle of weight updates is called a round.

3.4 Experimental Setup

We performed all experiments using two clients and one server. The first client and server are simulated using desktop devices with Quadro P2200 GPUs on a 12-core AMD Ryzen threadripper pro 3945wx 12-cores × 24 processors. On the other hand, the second client is simulated using a desktop device with GeForce RTX 2080 GPU on Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz processor.

3.5 Data Partitioning

We performe the analysis in different data splits by varying the ratios of labels to each client and the addition of noise to the data samples in a subset of clients. This data partitioning is important because it simulates label and sampling feature heterogeneity that are common in real-wild Federated learning applications.

Sampling rate Partitioning: We create non-identical data partitions (in terms of size) by partitioning according to the Dirichlet distribution $Dir(\alpha)$ [21] where $\alpha > 0$ is a concentration parameter controlling the label distribution identicalness among clients. With $\alpha \rightarrow \infty$, all clients have identical label distributions; with $\alpha \rightarrow 0$, all clients have non-identical label distributions. We vary the values of α in range $[0, 0.2, 0.5, 1, 3, \infty]$ to generate partitions that cover a range of identicalness.

Feature Distribution Partitioning: We synthesize sampling feature heterogeneity by adding noise to data samples held by a subset of partitions. We add this Gaussian noise to both the training and test sets to simulate data samples that adopt features inherent to the devices on each partition.

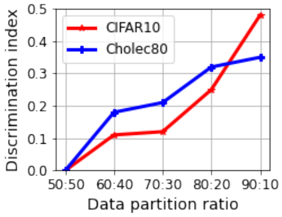
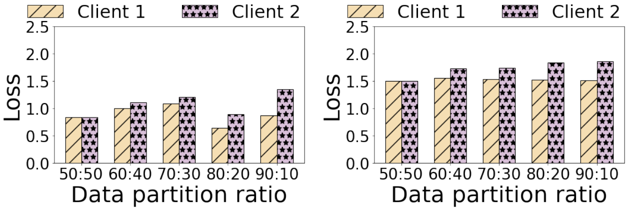
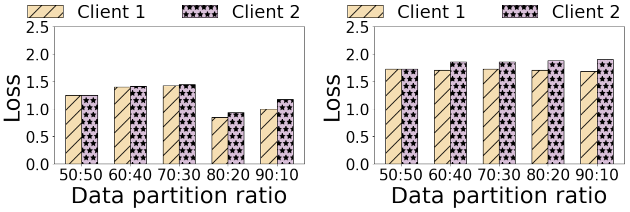


Figure 1: Impact of sensors' sampling heterogeneity rate on the bias (discrimination index). Discrimination index is computed on different levels of label heterogeneity. Batch normalization method is used in each application.



(a) CIFAR10 dataset **(b) Cholec80 dataset**

Figure 2: Impact of label heterogeneity on per-client performance across partitions with default batch normalization technique on our representative benchmarks (a) CIFAR10 dataset. (b) Cholec80 dataset.



(a) CIFAR10 dataset **(b) Cholec80 dataset**

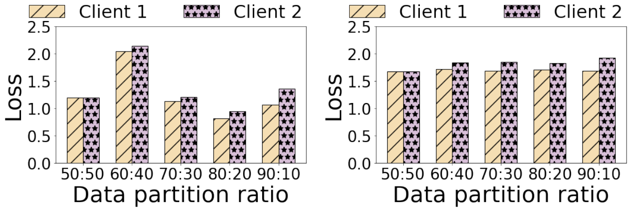
Figure 3: Impact of label heterogeneity on per-client performance across partitions with group normalization technique on our representative benchmarks (a) CIFAR10 dataset. (b) Cholec80 dataset.

4 RESULTS

In this section, we discuss how **sensor heterogeneity** affects the bias of a global model.

4.1 Impact of Label Heterogeneity on Per-user Performance

First, we quantify the impact of **label heterogeneity** with respect to loss on overall model performance and per-client performance. In this section, we are evaluating effects of bias in model performance when deployed across different partitions. All partitions consist of samples with the same features with a varied label distribution across partitions. To carry out this experiment, we vary the number of samples across each partition by changing the sampling rate



(a) CIFAR10 dataset **(b) Cholec80 dataset**

Figure 4: Impact of label heterogeneity on per-client performance across partitions with instance normalization technique on our representative benchmarks (a) CIFAR10 dataset. (b) Cholec80 dataset.

partitioning across partitions. We set the differences in data samples across clients according to the ratios 60 : 40, 70 : 30, 80 : 20 and 90 : 10 between the first and second clients, respectively. For example, the proportion of 60 : 40 means that the first client holds 60 percent of the total data for the FL (sum of data points from all partitions), while the second client only contributes 40 percent of the total data. We vary the ratios to assess how small and large gaps between different data sizes affect the difference in performance between unique partitions. Our base case is where two clients hold the same samples, ideally these two clients should perform the same with zero discrimination index value.

In Figure 1 we see that the discrimination index (bias) increases with the increase in the partitions of the dispropotion client. We can observe that label heterogeneity causes the global model to perform biased when classifying heterogeneous data with batch normalization. However, the increase in the discrimination index is not linear to the disproportion of data samples, which indicates *label heterogeneity*. This discrimination index (bias) ranges from 0 to 0.47 for the application that uses the CIFAR10 datasets. The bias of 0 corresponds to the base case, while 0.47 corresponds to the setting with extreme label heterogeneity with a partition ratio of 90 : 10. Similarly, the performance difference ranges from 0 to 0.36 for the application that uses Cholec80 datasets. This *label heterogeneity* causes the global model to be biased because some clients have under-representation of data and some have over-representation. With over-representation, sensors are likely to collect many data samples representing the diverse class/label distribution within a fixed time frame. This over-representation may be good because it enables the global model to learn features corresponding to diverse classes/labels. For under-representation, sensors may not have enough data samples to represent the diverse class/label distribution. This deficiency causes the model to experience low performance when deployed on under-represented partitions due to a lack of generalization. The class distribution for our two clients is shown in Table 1 for the CIFAR10 dataset and Table 2 for the Cholec80 dataset. We observe that some labels are underrepresented in client A but over-represented in in client B. Taking into account the partitioning ratio 70 : 30, for example, client A possesses 83 percent of data points that belong to the "bird" label, while client B only has 17 percent of samples representing this class. This under-representation may have caused client B to underperform leading to overall performance degradation of the model aiding to bias (increased discrimination index). However, from the overall performance of the model it is hard to determine how much this performance degradation is a result of **label heterogeneity**. To test the impact of overrepresentation and underrepresentation of the client on overall degraded performance, we use normalization as a way to solve this performance degradation and see the behaviour of each client with a varied number of samples. In Figure 2 we can see that the impact of label heterogeneity on per-client performance in classifying the CIFAR10 and Cholec80 datasets, while we use the default batch normalization technique. It is evident that performance degradation across clients is proportional to the number of diverse labels assigned during partitioning (Table 1 and Table 2).

To solve the problem of performance degradation that is caused by **label heterogeneity**, various deep learning normalization techniques such as Group Normalization have been used in [9]. This mini-batch independent techniques improve the performance degradation in settings with heterogeneous label distribution because it eliminates the shortcomings of batch normalization, which suffers in conditions with heterogeneous label distribution. We assess the impact of using these mini-batch independent techniques in improving the per-client performance of the global model across different partitions. This can help us measure how **label heterogeneity** affects bias in the overall global model while using performance enhancing normalization techniques that improve performance under label heterogeneity [9]. We replace batch normalization with mini-batch independent normalization methods including group normalization (*GroupNorm*), instance normalization (*InstanceNorm*), and layer normalization (*LayerNorm*).

Figures 3 and Figure 4 show the impact on the per-client performance across each partition in classifying the CIFAR10 and Cholec80 datasets. Despite using state-of-the-art normalization techniques that improve performance, we observe that the global model still performs differently across different clients because of the **label heterogeneity**. For GroupNorm, this difference in per-user performance (in terms of loss) ranges from 0 to 0.18 for the application using CIFAR10 datasets. On the other hand, the performance difference ranges from 0 to 0.22 for the application with the Cholec80 datasets. For InstanceNorm, the difference in per-user performance ranges from 0 to 0.29 for the application with the CIFAR10 datasets and the difference in performance ranges from 0 to 0.24 for the application with Cholec80 datasets. In general, the difference in performance is directly proportional to the level of label heterogeneity across partitions (label distributions corresponding to each data-split ratio are in tables 1 and 2).

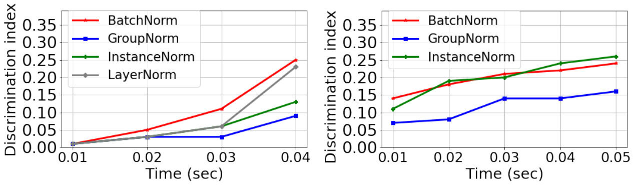
The bias problem discussed above arises because, in FL settings with label heterogeneity and data under-representation hinders the global model from learning diverse local normalization statistics (μ_i and σ_i) across the partitions. The hindrance leads to global models being biased towards over-represented clients because sufficient knowledge about their normalization statistics gets distilled into the global model. From these results, we conclude that batch-independent normalization methods do not lead to uniform per-client performance across partitions with different numbers of samples leading to label heterogeneity.

	Client A					Client B				
Label	50:50	60:40	70:30	80:20	90:10	50:50	60:40	70:30	80:20	90:10
airplane	0.5	0.86	0.99	0.8	0.99	0.5	0.14	0.01	0.2	0.01
automobile	0.5	0.4	0.07	0.8	0.99	0.5	0.6	0.93	0.2	0.01
bird	0.5	0.49	0.83	0.8	0.99	0.5	0.51	0.17	0.2	0.01
cat	0.5	0.97	0.99	0.8	0.01	0.5	0.03	0.01	0.2	0.99
deer	0.5	0.57	0.35	0.8	0.92	0.5	0.43	0.65	0.2	0.08
dog	0.5	0.99	0.75	0.8	0.96	0.5	0.01	0.25	0.2	0.04
frog	0.5	0.11	0.99	0.8	0.95	0.5	0.89	0.01	0.2	0.05
horse	0.5	0.5	0.88	0.8	0.99	0.5	0.5	0.12	0.2	0.01
sheep	0.5	0.67	0.98	0.8	0.99	0.5	0.33	0.02	0.2	0.01
truck	0.5	0.37	0.18	0.8	0.93	0.5	0.63	0.88	0.2	0.07

Table 1: CIFAR10 dataset: The share of image in each partition based on labels. Dirichlet distribution is used to partition the data.

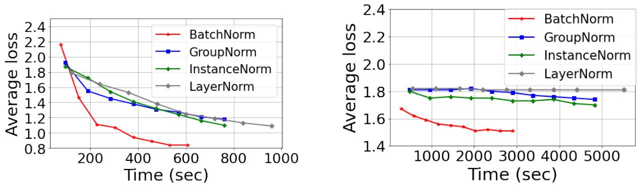
	Client A					Client B				
Label	50:50	60:40	70:30	80:20	90:10	50:50	60:40	70:30	80:20	90:10
CalotTriangleDissection	0.5	0.76	0.78	0.83	0.99	0.5	0.24	0.22	0.17	0.01
CleaningCoagulation	0.5	0.68	0.05	0.24	0.38	0.5	0.32	0.95	0.76	0.62
ClippingCutting	0.5	0.01	0.69	0.99	0.47	0.5	0.99	0.31	0.01	0.53
GallbladderDissection	0.5	0.45	0.84	0.98	0.98	0.5	0.55	0.16	0.02	0.02
GallbladderPackaging	0.5	0.56	0.99	0.3	0.89	0.5	0.44	0.01	0.7	0.11
GallbladderRetraction	0.5	0.54	0.15	0.7	0.9	0.5	0.46	0.85	0.3	0.1
Preparation	0.5	0.75	0.93	0.9	0.99	0.5	0.25	0.07	0.1	0.01

Table 2: Cholec80 dataset: The share of image in each partition based on labels. Dirichlet distribution is used to partition the data.



(a) CIFAR dataset **(b) Cholec80 dataset**

Figure 5: Impacts of distortion level (Gaussian noise) on the bias of the model in (a) CIFAR10 dataset. (b) Cholec80 dataset.

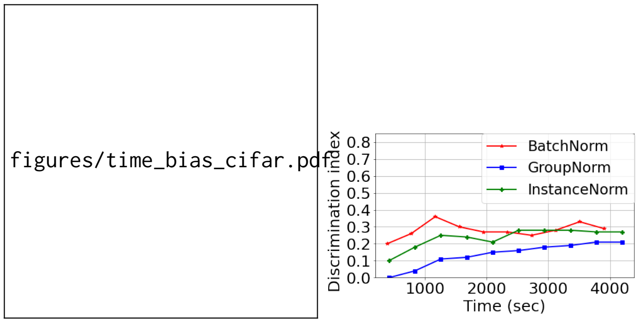


(a) CIFAR10 dataset **(b) Cholec80 dataset**

Figure 6: Impact of training time on the overall performance of the model in (a) CIFAR10 dataset. (b) Cholec80 dataset.

4.2 Impact of Sampling Feature Heterogeneity on Bias

In addition to label heterogeneity, another factor that contributes to the bias in the global model is the existence of **sampling feature heterogeneity** in data samples across partitions as a result of distortions. In addition, different levels of distortion appear in the data as a result of the use of sensors under various conditions. For example, cameras capture images under poor and adequate lighting conditions. These conditions add Gaussian noise with zero-mean and severity level controlled by variance [16]. To simulate the effect of **sampling feature heterogeneity** we add gaussian noise with different variance values to alter the levels of sampling feature heterogeneity. In Figure 5 it is shown that the bias is proportional to the level of **sampling feature heterogeneity** for all the normalization techniques explored in this work. Bias deficiency increases because learning occurs differently across partitions due to different data representations. As a result, the aggregator's fusion algorithms to combine local model updates can weigh contributions differently if feature representations are non-uniform [17]. This difference in weight contributions favors partitions with over-represented data. In summary, we observe that **sampling feature heterogeneity** causes the bias of the global model when deployed across different clients even with different normalization methods that happen to enhance performance. The bias arises from variable local optimization profiles which in turn vary with the number of iterations (time) since each local model's performance improves or degrades over time. This raises the question of how training time affects the overall bias in disproportional clients. In the next set of evaluations, we attempt to answer this question.



(a) CIFAR10 **(b) Cholec80 dataset**

Figure 7: Impact of training time on the bias of the model in (a) CIFAR10 dataset. (b) Cholec80 dataset.

4.3 Training Time Impact on Bias

This section explores the training time and bias trade-offs in FL that quantify how limited FL training time (common in resource-constraint edge devices) affects the bias discussed in the previous sections. To study this trade-off, we measure the bias and training time elapsed after each training round. In Figure 7 the impact of training time on the bias is shown with various normalization techniques. It should be noted that for the Cholec80 dataset, layer normalization was omitted since the model could not learn under heterogeneous settings due to instability in normalization statistics because it assumes all channels have equal contributions in model training [9]). For the CIFAR10 and Cholec80 datasets, models that deploy InstanceNorm and GroupNorm experience an increase in bias as training time increases (green and blue curves). This increase occurs because distinct partitions use non-identical (label and sampling feature heterogeneity). This difference in datasets gives local models room to specialize in their local data, leading to a more biased global model as training time increases. Although models that use InstanceNorm and GroupNorm experience low bias given a limited training time, this comes at the cost of performance degradation (compared to BatchNorm) as shown in Figure 6. This performance degradation occurs because InstanceNorm and GroupNorm calculate the mean and variance for every group of channels, whereas BatchNorm calculates the mean and variance once for the whole batch [22] and the optimization of the model slows.

The performance of different normalization techniques over time is shown in Figure 6 (base cases with homogeneous labels and features). learning instability due to non-identical data across partitions. For BatchNorm, there is a divergence in mini-batch mean and variance across different clients [9] that leads to a biased global model as training time progresses. This mismatch mainly depends on the similarity/difference of mini-batches across clients (not necessarily on time). For LayerNorm, it assumes that all inputs make similar contributions to the final prediction, but this assumption does not hold for some models, such as convolutional neural networks, where the activation of neurons should not be normalized with non-activated neurons [3]. As a result, LayerNorm experiences performance fluctuations in settings with non-identical data across partitions, leading to unstable bias as training time increases.

Our results on the impact of training time on bias suggest that model training time also affects the introduction of bias due to heterogeneous sensors. The inability of resource-constraint edge devices to train large models for long periods can reduce the bias in global models and move them towards fair models but can cause performance loss in short training periods. Furthermore, high-performance applications, such as surgical guidance systems, can be adversely affected by this performance degradation.

5 DISCUSSION

As a result of our work thus far, we discuss future directions for bias mitigation in FL settings with heterogeneous sensors to collect data across partitions.

Multi-modal deep learning Deep learning models that combine information from multiple modes are trained with multi-modal deep learning. This technique has the benefit of improving performance of the underlying application during inference as a result of the robustness gained from learning diverse data representations from multiple modes [22]. Future directions in FL for medical image guidance could involve the development of unbiased global models through multi-modal deep learning. This multi-modal deep learning technique will mainly focus on extracting relevant cross-modal features from medical images and eliminate irrelevant features by employing information bottleneck [25]. Despite its potential in mitigating bias, this approach might face some challenges when deployed in FL. It will be difficult to determine and combine relevant features from different modes without revealing private information about each client.

Data augmentation improves training classifiers' resistance to distortions by enhancing the training dataset though operations such as image flipping and image rotation [18]. Future directions in FL for medical image guidance could be to develop an unbiased global models through training local models that can learn image representations that are robust to the heterogeneity introduced by sensors. Due to the success of deep learning models in combating noise (feature heterogeneity) in FL settings with heterogeneous [18] sensors, this approach may mitigate bias in FL settings with heterogeneous sensors. On the other hand, the challenge with this approach in FL will be the generation of data augmentation samples across partitions without revealing private data about each partition. This will pose a challenge because the generation of augmentation samples at each partition is dependent on the knowledge of the local dataset discrepancies (such as sample count for each class) in comparison with other partitions. This methods violates privacy as private information pertaining to each client needs to be revealed.

6 RELATED WORK

Several works have studied the fundamentals of bias in FL. In a few recent studies, pre-processing [6, 8, 11, 23] has the used as bias mitigation technique. In [1] few of these techniqeus has been surveyed. In this, each client applies the concept of data re-weighing [11] where the local dataset is normalized using weights computed (based on sensitive attributes of the local training dataset) during pre-processing. Each client then uses the normalized dataset for local training to mitigate the bias. Reweighing methods are applied to datasets with easily identifiable sensitive attributes. While pre-processing methods mitigate heterogeneity-related bias, they still have limitations. For example, in case of medical imaging data collected through heterogeneous sensors, the identification of sensitive attributes is problematic.

Others studied data normalization as a possible technique to improve overall FL performance including heterogeneity bias. Author in [9] addresses the problem of heterogeneity in FL using normalization. This work employs Group normalization as an alternative to Batch normalization to avoid the problem of unstable normalization statistics (mean and variance). However, it mainly addresses the problem of performance degradation in FL with heterogeneous settings (it does not solve the problem of bias). Another work related to data normalization is studied in [2] and [15]. In this work, data normalization statistics (with BatchNorm) are not shared with the server for aggregation. This paradigm enhances performance of local models in heterogeneous settings as the local statistics ensure that the intermediate activations are centered to a similar value across clients [2]. However, the problem with this approach is that it leads to model personalization across clients. It becomes difficult for these models to perform well outside of their domain. All the above discussed techniques show weakness in solving bias directly or indirecly. In our work we study bias caused as a result of heterogeneity and how existing techniques impact the tradeoff between accuracy-vs-fairness.

7 CONCLUSION

In this paper, we conducted an empirical study to investigate the effects of sensor heterogeneity in federated learning (FL) bias. We find that textbflabel heterogeneity and **sampling feature heterogeneity** cause bias in FL models. A large part of these biases is attributed to the sampling feature heterogeneity, due to the inherent factors of the heterogeneous device that impact the overall data collection. To perform our analysis, we used existing performance enhancing techniques (normalization) to quantify the bias and evaluate *performance-vs-resource* trade-offs. Our results demonstrated that while these normalization techniques failed to mitigate the bias completely, the bias is proportional to the degree of heterogeneity in the sensor sampling features. With this observation, we make a case for the need of robust mitigation techniques based on complex multi-modal deep learning and data augmentation techniques.

REFERENCES

[1] Annie Abay, Yi Zhou, Nathalie Baracaldo, Shashank Rajamoni, Ebube Chuba, and Heiko Ludwig. Mitigating bias in federated learning. *arXiv preprint arXiv:2012.02447*, 2020.

[2] Mathieu Andreux, Jean Ogier du Terrail, Constance Beguier, and Eric W Tramel. Siloed federated learning for multi-centric histopathology datasets. In *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, pages 129–139. Springer, 2020.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Daniel J Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, et al. Flower: A friendly federated learning framework. 2022.

[5] Mitchell Doughty, Karan Singh, and Nilesh R Ghugre. Surgeonassist-net: Towards context-aware head-mounted display-based augmented reality for surgical guidance. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 667–677. Springer, 2021.

[6] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[7] Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. *arXiv preprint arXiv:2110.00857*, 2021.

[8] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 259–268, 2015.

[9] Kevin Hsieh, Amar Phanishayee, Onur Mutlu, and Phillip Gibbons. The non-iid data quagmire of decentralized machine learning. In *International Conference on Machine Learning*, pages 4387–4398. PMLR, 2020.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[11] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012.

[12] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 35–50. Springer, 2012.

[13] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

[14] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[15] Xiaoxiao Li, Meirui Jiang, Xiaofei Zhang, Michael Kamp, and Qi Dou. Fedbn: Federated learning on non-iid features via local batch normalization. *arXiv preprint arXiv:2102.07623*, 2021.

[16] Zida Liu, Guohao Lan, Jovan Stojkovic, Yunfan Zhang, Carlee Joe-Wong, and Maria Gorlatova. Collabar: Edge-assisted collaborative image recognition for mobile augmented reality. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 301–312. IEEE, 2020.

[17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[18] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625. PMLR, 2019.

[19] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

[20] Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel De Mathelin, and Nicolas Padoy. Endonet: a deep architecture for recognition tasks on laparoscopic videos. *IEEE transactions on medical imaging*, 36(1):86–97, 2016.

[21] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[22] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

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

[24] Daniel Yue Zhang, Ziyi Kou, and Dong Wang. Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1051–1060. IEEE, 2020.