Privacy-enhanced ZKP Framework for Balanced Federated Learning

Stefano Marzo
21264466
stefano.marzo2@mail.dcu.ie
Dublin City University
Dublin, Ireland, IE

Royston Pinto
20210611
royston.pinto2@mail.dcu.ie
Dublin City University
Dublin, Ireland, IE

***Abstract*—Federated learning (FL) is a distributed machine learning approach that enables remote devices i.e. workers to collaborate to compute the fitting of a neural network model without sharing their data. While this method is favorable to ensure data privacy, an imbalanced data distribution can introduce unfairness in the model training, causing discriminatory bias towards certain under-represented groups. In this paper, we show that imbalance federated data decreases indexes of statistical parity difference, equal opportunity difference, and equal odd difference. To address the problem, we propose a FL framework called Z-Fed that 1) balances the training without exchange of privacy protected data using a zero knowledge proof (ZKP) authentication technique, and 2) allows collecting information on data distributions based on one or more categorical features to produce metadata about population proportions. The proposed framework is able to infer the precise data distribution without exchanging knowledge of the data and use them to coordinate a balanced training. Z-Fed aims to mitigate the effect of imbalanced data in FL without using mediators or probabilistic approaches. Compared to a non-balanced framework, Z-Fed, on average, improves the indexes of equal opportunities of 53.54%, the equal odds of 56.41%, and the statistical parity of 46.1% on imbalanced UTK datasets, reducing biased predictions among subgroups. Given the results obtained, Z-Fed can reduce discriminatory behaviors of FL AI and enhance trustworthy federated learning.**

***Index Terms*—Federated learning, distributed databases, zero knowledge proof, imbalance data fairness evaluation.**

I. Introduction

Federated learning (FL) is a machine learning (ML) technique that allows an artificial neural network (ANN) model to be trained through the use of decentralized edge devices i.e. workers that maintain data locally, without sharing the data with the server i.e. the service provider. This method requires a central server to broadcast the ANN model to multiple workers, coordinating transmission and responses. The workers will locally fit the ANN and send the updated weights back to the server. The burgeoning interest in FL, from both research and applicative viewpoints, has led to the development of applications in many fields such as healthcare [1], [2], natural language processing (NLP) [3], and computer vision [4].

FL principles [5] require that the server responsible to orchestrate the learning process does not receive workers' data under any circumstances, allowing a neural model to be trained without compromising data privacy. This follows the GDPR requirements for "data protection by design and by default", Art. 25. Thus, it is possible to overcome problems related to the processing and storing of personal data and obtain accurate predictive models. Moreover, FL enables cross-device online training and scales at almost no additional cost [6]. However, it is necessary to consider that in a FL environment data can be unevenly distributed within the workers, leading to under-representation of one or more specific population subgroups. This can result in unfair prediction, statistical disparity, and inequity.

Zero knowledge proof (ZKP) [7] can be used to prove a statement having no knowledge of the statement itself. It is possible to implement a version of the Schnorr's ZKP authentication protocol based on the elliptic curve discrete logarithm problem [8], [9] that can be used to infer the data distribution of the remote workers without data exchange.

Considering a classical FL approach [5], the service provider has no means to ensure that data have even proportion, and cannot estimate the impact of the data distribution across the whole set of workers on ML predictions. Motivated by experimental results that show unfair treatment for imbalanced data (sections III, VI), the following research questions are investigated:

1) To what extent it is possible to mitigate federated learning bias according to GDPR and EU guidelines for data ethics and trustworthy AI?
2) To what extent can ZKP inferred data about the proportions of population groups generated in a federated learning environment can be used to enhance trustworthiness in FL AI?

By performing ethnicity differentiated evaluations on the imbalanced fitted ANN model, it is possible to observe an average increment of statistical parity difference, equal opportunities difference, and equal odds difference of 404.51%, 63.8%, and 33.9% respectively, compared to the balanced model. Unequal treatments due to imbalances drove to design a reworked self-balancing ZKP FL environment called Z-Fed to support a fair learning process and enhance demographic parity following this technical approach:

1) The federated server generates tokens to authenticate all

the possible workers with the ZKP Schnorr's protocol [8],
2) workers encrypt their feature labels, fit the learning model, and send the update to the server,
3) the server can zero-knowledge prove that workers belong to a certain feature group by retaining the encrypted version of the workers' labels and count individuals, and
4) the server uses a self-balancing queue system to accept updates only in case the clients will not compromise the balance.

In this paper, we define a multi-layer artificial neural network based on the statistical gradient descent (SGD) algorithm for weights update. This model is used for supervised training tasks on the UTK dataset [10] and is trained using face images to predict the age. Using ethnicity, gender, and age features of the UTK dataset, it is possible to artificially create highly imbalanced samples to use for ANN fitting. In addition, we implemented a FL framework and trained the ANN with balanced and imbalanced samples of the dataset, measuring a notable degradation of equity in predictions. Hence, we redesigned the federated server and workers to implement ZKP authentication and to accept only updates that do not perturb the balance.

The main contributions of the present paper are synthesized below.

- Observation that imbalanced datasets lead to equality degradation in FL training.
- Design of a self-balancing ZKP FL framework, *Z-Fed*, implementing zero knowledge authentication to avoid malicious workers to update the model and to mitigate the bias introduced by uneven distribution.
- Implementation of ZKP inference of data distribution and use it to allow data augmentation and rejection of imbalanced updates to counter effect bias.
- Evaluation of Z-Fed based on an SGD ANN. The experimental results can be summarized as follows: the measured scores relative to absolute multi-class (section VI) statistical parity difference (SPD), equal opportunity difference (EPD), and equal odds difference (EOD) are considerably improved in the experiments conducted using self-balancing Z-Fed. Detailed results are available in the evaluation section VI.

II. Contents

In section III, we research 1) the open problems in FL and discuss the applications of such technology in presence of imbalance privacy protected data, 2) the effects of imbalance data in ML and specifically in FL that results in degradation of the model performances, and 3) the scope of application of ZKP in authentication protocols, finding room for integration in a FL environment to enhance model training and reduce bias without compromising privacy. We give an overview of the main scientific findings available and show experimental results that drove the development of the Z-Fed framework.

In section IV we define the requirements to address the research questions stated in I. Additionally, we discuss the main components necessary to enable a ZKP FL framework that can rebalance the scheduling of worker updates and collect data distribution information with the purpose of reducing bias. We also describe the duties of the framework and possible technical improvements.

In section V we describe the technical implementation of the Z-Fed framework used to enhance equality in imbalance FL. We give the specifications of 1) the learning model, 2) the FL settings, 3) the framework initializer settings, 4) the ZKP protocol used for registration and authentication. We discuss the process of initialization of Z-Fed, supported by diagrams to describe a) the ZKP worker registration and authentication protocol figure 2, b) the Z-Fed initialization setup figure 4, c) the framework components and communication figure 3, and d) the workflow of Z-Fed figure 5.

In section VI we describe the method used to evaluate Z-Fed and state how we set up and performed four experiments for measuring equality metrics.

In section VII we gather use cases of discriminatory AI that could be suitable for using Z-Fed and discuss future improvements.

III. Background and Motivations

A. Background

Federated Learning. FL requires distributed workers to train an ANN model using e.g. SGD. A central server can collect fitted model weights using synchronous or asynchronous protocols [5]. Recent advancements in FL led to the design of solutions to enhance the communication efficiency and reduce payloads [11]–[13], enforce privacy during data aggregation [14], and deal with the accuracy reduction due to uneven distribution of data using mediators [15] or probabilistic approaches [16]. As stated in [6], fairness is a major issue in FL. In this regard, [17] presents a framework that uses deep multi agents reinforcement learning to reduce the bias in privacy-sensitive ML applications.

Imbalanced data machine learning. Learning with imbalance distribution is a widely researched topic. Ensemble methods are proposed to reduce bias in imbalanced data learning [18], but they can suffer the presence of outliers typical of FL. The main proposed solutions for dealing with imbalance data are sampling and augmenting data [19]. Oversampling, often implemented by artificially creating minority classes to counter the effect of disproportions [20], shows promising results, but requires access to data, and hence is not suitable for FL environments. Under-sampling is an easy and practical way to achieve balance by excluding samples from majority groups, although it requires a large amount of data to have a sufficient number of samples to use for ML.

Zero knowledge proof. ZKP can be used to enhance data privacy in online communication [7] and can be implemented using iterative [21] or non-iterative methods [8]. Iterative ZKP is unhandy in FL, since it considerably increases the communication overhead. Non-iterative implementations of ZKP are often used for authentication [9] and do not involve exchange of privacy protected data, which make this method

suitable for FL. ZKP authentication allows a server to prove that a client knows certain information without revealing it. This is possible through the use of encrypted tokens, i.e. proofs and signatures, that ensure that only authorized clients can be authenticated.

B. Motivations

Distribution of data coming from IoT and remote devices is likely to be uneven and lead discriminatory predictions. In this paper, we focus on class imbalance, i.e. specific under-represented groups in the data. [6] describes many scenarios that lead to class imbalance, e.g. privileged demographic groups.

To the best of our ability, scientific research presenting fairness measurements against imbalanced UTK datasets [10] could not be found for benchmark. Therefore, an exploratory search was conducted based on the following hypothesis: in a federated environment, an ANN shows bias if the training dataset is class imbalanced. Following, we present the settings and a summary of the main findings that motivate the interest in developing a framework for self-balancing federated learning.

To assess the influence of imbalanced training data in FL, we trained ANNs with an up-sampled UTK dataset and measured fairness metrics afterwards. We used face images to predict four age ranges i.e. *0 to 9, 10 to 19, 20 to 29*, and *30 to 39* considering four ethnic groups i.e. *Asian, Black, Indian, White* and two gender groups i.e. *Female, Male*. Ethnic and gender groups can be considered in all the 8 possible combinations to form subgroups.

Imbalanced datasets. A simple way to create class inequity is to define a privileged (PR) class that is over-represented with respect to the other unprivileged classes. Four different dataset are built, choosing one ethnic group as privileged, with a class proportion distributed as follows: 85% of the samples belong to the privileged group and the remaining 15% of the sample are equally split among the rest of the unprivileged ethnic groups. All the datasets have the gender and the age range features balanced. The datasets described previously will be further identified as ASIAN-PR, BLACK-PR, INDIAN-PR, WHITE-PR. In addition, an ethic-gender class balanced dataset (BAL) is set up for training, evaluation and comparison, additional details are shown in table I.

ML model architecture. In the designed FL environment, all remote devices synchronously receive an ANN predictive model that will be used to propagate their data locally and calculate the updates of weights $\mathbf{w}$ under the orchestration of the central server using the SGD method. For a given update requested at instant t the updated weight structure is:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla L(\mathbf{x}, y, \mathbf{w}_t)$$

where η is the learning rate and L is the loss function of an input x, a label y, and the current weights. The ANN has the following fully-connected layer (FCL) structure: $2304 \times 96 \times 4$ neurons plus one bias neuron per FCL, and uses a sigmoid activation function and a mean square error (MSE) loss function. The ANN model has a total of 884,736 training parameters i.e. weights and achieve an average accuracy of 50.8%, variance 1.07% after fitting 16,000 samples in one epoch with $\eta = 0.025$ on UTK.

TABLE I: Setting of the distributed UTK dataset

Notation	Ethnicity	Gender, Age	#Samples	Train-Test
ASIAN-PR	85% Asian 15% other	balanced	20,000	80%-20%
BLACK-PR	85% Black 15% other	balanced	20,000	80%-20%
INDIAN-PR	85% Indian 15% other	balanced	20,000	80%-20%
WHITE-PR	85% White 15% other	balanced	20,000	80%-20%
BAL	balanced	balanced	20,000	80%-20%

FL settings. Every worker holds a sample of size one, and the FL framework is set up to compute one epoch per training cycle. In these settings, the model performed an average SPD of 1.86%, EPD of 4.6%, and EPD of 1.84% on BAL, and we measure an SPD of 15.02%, EPD of 15.01%, and EPD of 5.17% on ASIAN-PR. In addition, the model shows a negligible difference in average absolute EOD on both BAL and WHITE-PR, while showing a flat slope on BAL and significant growth of inequity in ASIAN-PR during the model update rounds as shown in figure 1. Moreover, we tested the accuracy of the ANN against a specific ethnic group, measuring the variance among subgroups. Considering an accuracy variance of 0.09% on BAL, the ANN shows a subgroups accuracy variance of 3.22% on ASIAN-PR, meaning that it is more likely to have different treatment in case of imbalanced data. Figure 1 shows equality scores of ANN while training.

ZKP settings. A server S i.e. verifier and a client c i.e. prover are such that c can prove to S that a given condition results true, avoiding sharing any information but the fact that the condition is true. The server chooses a password $S_{password}$ and the client chooses a secret c_{secret} e.g. the value of the ethnic group of belonging that does not want to share with S. Based on [8], S and c choose the following public parameters respectively: an elliptic curve S_{curve}, c_{curve} with elliptic curve generator points S_g, c_g, a hash function S_{hash}, c_{hash}, and a relatively big random number S_{salt}, c_{salt}. In addition, ZKP server and client produce random private variables, S_n, c_n respectively, used to compute a specific point on the elliptic curve. Using these settings it is possible to create a signature i.e. token of the form of $token = g \times hash(secret|salt) \bmod n$ which can be shared publicly revealing no information about the secrets. After c sends its signature to S, the latter can subsequently sign the received token, publish the newly signed token, and retain public client parameters, i.e. *registration*, this way the server can prove if a further token comes from the same client that used the same server signature in the past, i.e. *authentication*. The ZKP registration and authentication

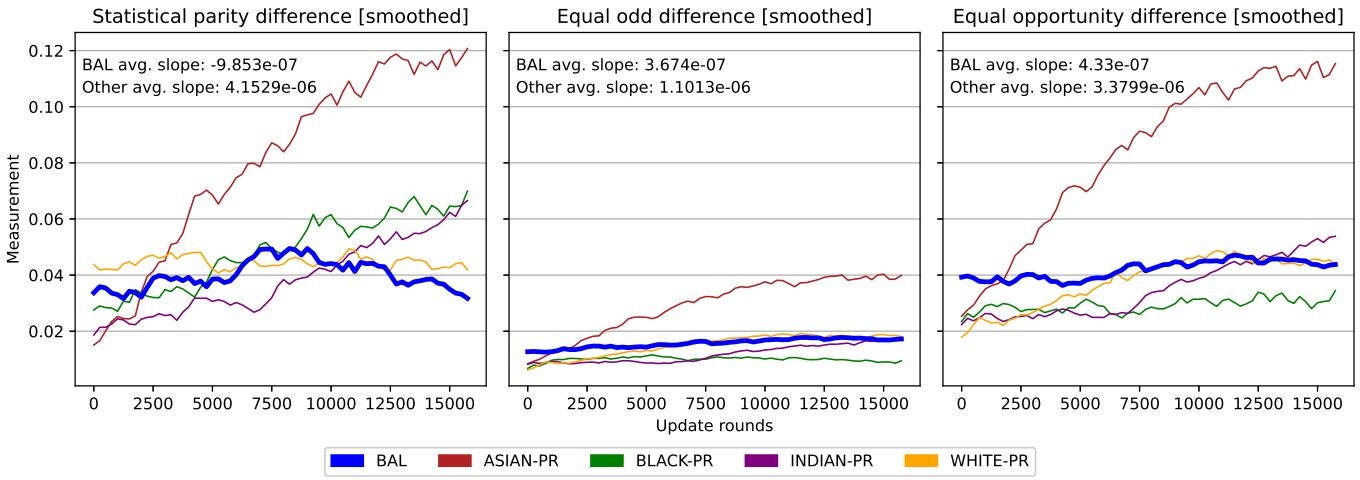

Fig. 1: Measure of equality in terms of SPD, EPD, and EOD on different balanced and imbalanced datasets.

process can be visualized in figure 2.

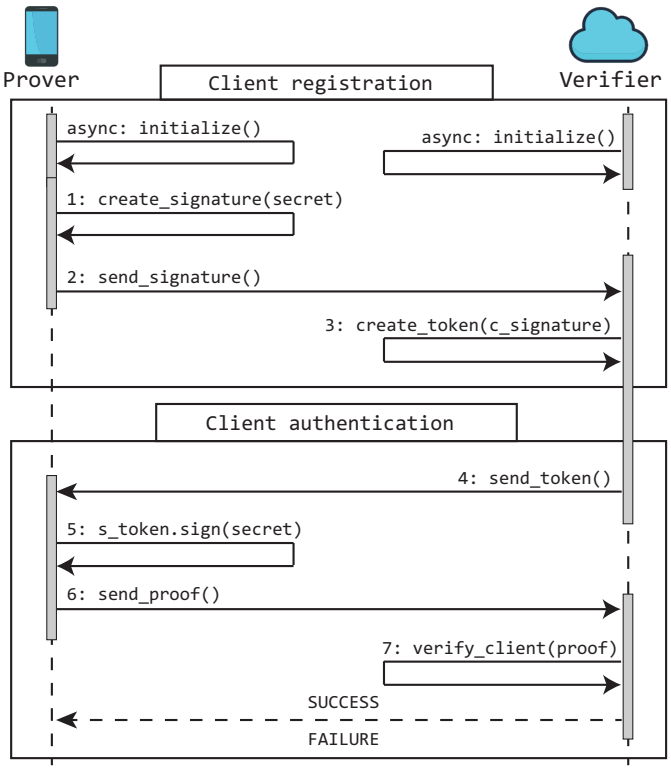

Fig. 2: ZKP registration and authentication process.

To recapitulate, the imbalance class training experiments result in a loss of fairness measured SPD, EOD, and EPD, as well as an increase in accuracy variance among subgroups. Since public exchange of data is not contemplated in FL, the Z-Fed ZKP FL framework for self-balancing ML is proposed to enhance equity while keeping the server completely unaware of workers' data.

IV. Use Cases and Requirements

To further investigate the reasons of the decline in equity, we redesigned the proposed framework and created Z-Fed, a renewed ZKP FL environment exposing the following functionalities:

A ZKP protocol is designed to enable the server to register every possible subgroup within the dataset. To do so, the server must be aware in advance of the possible categorical features that data can possibly have, e.g. values of ethnic group and gender. For this purpose, it is possible to create a number of mock clients, i.e. *client prototypes* equals to the number of subgroups present in the data. Client prototypes are not used for ANN training, but only for creating registration tokens. The server can use them to set up encrypted dictionaries that are used to count the number of sample belonging to specific subgroups.

A service called *framework initializer* is required to generate a private number n that can be transmitted to the workers and produces the client prototypes. In the proposed architecture, the federated framework is initialized starting using the aforementioned service.

While performing FL, the federated server requires the users to ZKP authenticate to prove that 1) they are authorized to contribute training and 2) they are not holding data which is not representative of the population i.e. does not belong to any subgroup. This authentication technique may, to some extent, prevent data poisoning [6], [22] although further investigation is required.

While distributing the model for ML, the federated server is able to count the number of samples used for training, retaining an encrypted representation of the client subgroup categories in its encrypted dictionary. At any given moment, the server can assess whether the distribution of data is even or not.

Before a worker is requested to contribute to the ML process, the server can check if the worker would increment disparity in data distribution, and in this case it will reject

any update from it. When a worker is not able to train the distributed model because of potential imbalance, the server is able to register the workers' identifier to possibly reach it later in case its update would not result in imbalance. To optimize the process, the server retains a priority queue data structure. Moreover, if the dataset is highly imbalanced, the server can augment the training mechanism by requesting multiple epochs of training for under-represented groups.

It is possible to see a diagram of the proposed model in figure 3.

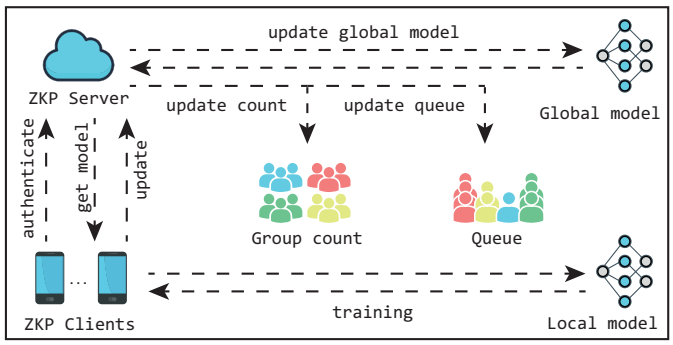

Fig. 3: Z-Fed framework communication diagram.

V. Design of Z-Fed

As aforementioned, FL models trained with imbalance data generate unfair predictions. Based on this experimental evidence, we designed Z-Fed with the purpose to mitigate data imbalance and recover equity.

A. ZKP framework

Requirements. Within a set of individuals, it is possible to define categorical features that describe the population. On the UTK data, we identified two features of interest to keep the model balanced: i.e. *Ethnic group* and *Gender*. The Z-Fed framework must know all the possible values i.e. labels that can be assigned to the identified features. On the UTK dataset, ethnic groups can assume the four values of *Asian, Black, Indian, White*, and gender groups can assume the values of *Female, Male*. This information is stored in a public dictionary data structure denoted with $features$. Information about the structure of the data used for training the model X and the true labels y must be publicly available within the framework.

ZKP Server. The ZKP server must be initialized with a private $password$ to prevent the tokens from being vulnerable by using encryption. The server can use the password to generate ZKP signatures. Having the server countersigning a client signature allows the client to prove that the server is legit.

The server must store a copy of the $features$ data structure. From now on, we will refer to the possible features in the dictionary of the UTK dataset as feature name, e.g. *Ethnicity* and *Gender*, and will refer to the possible values as feature labels, e.g. *Female, Male*, etc.

The ZKP server, during the registration phase, is able to create $tokens$. An authorized ZKP client can send its signature to the server as described in figure 2, and later the server can use the client signature to create a $token$. The registration phase ends when all the possible subgroups combinations, e.g., (*Gender: Female, Ethnicity: Asian*, . . . , *Gender: Male, Ethnicity: White*) have a server-side token representation. The server can authenticate clients by checking if they have a proof that is compatible with any of the tokens, meaning that the client belongs to a specific subgroup.

The ZKP Server requires a data structure to store the ZKP parameters needed for registration and authentication [7], [8], such as the elliptic curve of choice $curve$, the public $Salt$ value, the private number n, the hash function of choice $hash$, and the curve generator point g.

The ZKP Server needs a dictionary structure named $groups$ to count encrypted versions of subgroups. Since every feature label has a token representation on the server, for each of the k feature names on the shared $features$ dictionary, a client must show to have k feature labels compatible with the feature structure in order to authenticate. Once authenticated, the server will receive k count updates indexed with k hexadecimal hash number. The k hashes will be summed and used as a dictionary key to manage the FL server queuing protocol.

ZKP Client. A ZKP client is responsible for representing a specific individual tuple *feature name, feature label* in the distributed dataset. Given the number k of feature names in the features dictionary, every worker will instantiate k ZKP clients. Every ZKP client store the $feature\ name$, the $feature\ label$, the ZK data structure analogue to the one of the ZKP server. ZKP clients can generate a signature encrypted using a password. In this case, the ZKP client password, i.e. $secret$, is the hashed value of the $feature label$ joint with the private number n:

$$secret = hash(feature\ label|n)$$

where | is a string operator, e.g. concatenation. Using the private number n the ZKP server is not able to decode the client token to read the label. Moreover, the ZKP client can create an $encrypted\ label$ using a different method to joint the $feature\ label$ with the number n, e.g.

$$encrypted\ label = hash(n|feature\ label)$$

The client is safe to publicly send the value of $encrypted label$ without revealing the $secret$ or the $feature\ label$. The $encrypted\ label$ value is used to server side count the subgroups.

ZKP framework initializer ZKP authentication is enabled by public data structures, i.e. tokens, that use elliptic curves and generator points to prove that a specific authentication proof come from the same token that was generated from the server during the registration process. Since there is no exchange of private data in the registration phase, a malicious remote client could force the server to register its features even if they do not belong to the $features$ public dictionary. This is possible because the server would receive only an encrypted

version of the label based on the private number n of the ZKP client. In this case, this malicious behavior leads to distortion in the count of groups and results in ineffective control of data balance. Moreover, the framework would suffer higher computational and storage payload by generating multiple authentication tokens for each worker.

In order to counter the effects of unreliable clients, keep the learning environment trustworthy, and reduce the computational resource needed, it is possible to create a very limited number of authentication tokens during an early server setup phase. To do so, we proposed the design of a trusted external service in charge of generating one ZKP client prototype for each of the possible q combination of feature names and values. The proposed framework initializer has the duty of coordinating with the ZKP server to register the q created client prototypes and consequently generate the q authentication tokens allowed. This can be achieved only if all the clients share the same private number n, that allows them to sign the server tokens to generate proofs. For this reason, all the workers must connect to this service and get the value of n prior to authenticate to the ZKP server. While this results in an additional step in the FL process, the framework allows doing this asynchronously and just one time. ZKP authentication approach can be implemented alongside classical authentication methods with no side effects, the transmission of the n parameter can be achieved using a classic RSA [23] approach. The framework setup process is illustrated in picture 4.

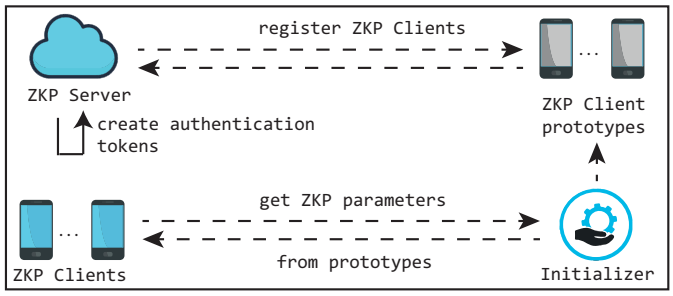

Fig. 4: Z-Fed framework initialization setup.

B. FL framework

Learning model. In the Z-Fed environment, FL is independent of the chosen learning model. We designed a learning model interface to be implemented with no constraints for model selection. Any machine learning model can be used as long as it presents the following APIs: 1) initialize the learning model, setting e.g. the learning rate η, 2) read the values of the trainable parameters e.g. weights together with the configuration settings, 3) fit an input list X given a label list y e.g. propagate the inputs through the neural layers, measure the loss and update the training parameters, 4) load external training parameters received e.g. from remote clients, and 5) produce a list of predictions y_{pred} given a list of inputs X.

Federated server. A self-balancing federated server must be able to discern updates based on the subgroup of belonging of the client. The main assumption for the design of the server, is that there is a high availability of federated data necessary to compute the training. After ZKP authentication, the server estimates whether the count of subgroups would result in imbalance, and, under this circumstance, rejects the update. The server may identify workers with an identification number w_{ID}. This allows the server to organize rejected workers into queues and efficiently select workers for further balanced updates. To simplify, the presented model executes synchronous FL, meaning that the server elaborates updates one at a time. Based on research findings summarized in section III-A, we speculate that there is room for improvement of Z-Fed by integrating aggregations techniques such as FedAvg [13] and perform asynchronous FL. The server retains a dictionary of queues, used to store examples belonging to different subgroups. Since the workers have one or more hexadecimal hashed labels representing the subgroups, it is possible to sum the values to create the index of a hash-table used to access the specific subgroup queue. Moreover, the server can be setup with a gap i.e. a variable used to declare a number of imbalanced updates allowed, and max_gap i.e. a decimal number such that $0 \leq max\_gap \leq 1$ to express the percentage of imbalanced updates accepted. Having the count of subgroups, it is possible to check if an update will keep the model balanced if, for a worker w, having k feature names $name^i$, and i_j feature labels $label^i_j$:

$$avg = average(count(groups[name^i]))$$

$$gap_{current} = groups[name^i][label^i_j] - avg$$

balanced if: $\frac{gap_{current}}{avg} \leq gap$ or $gap_{current} \leq max\_gap$

for each $i = 1, \ldots, k$ for each $j = 1, \ldots, i_j$

Moreover, the federated server can perform an analysis of the population distribution prior to FL in order to estimate the proportions of subgroups and, eventually, augment the data by performing additional training epochs on under-represented workers. To summarize, the federated server is responsible for register workers for training, count labels to keep the subgroups count balanced, register rejected workers in a queue dictionary, keep the updated version of the learning model, request additional learning epochs in case of disparity, and show on request the subgroups count.

Federated worker. The federated worker is responsible for training the distributed model and provide wights updates to the central server. Workers retain a number k of ZKP clients equals to the number of feature names present in $features$. Workers present a data structure to store the parameters required for model training, and a local copy of the learning model. A worker can retain a list of pairs of training features and ground truth, X and y respectively. Additionally, workers retain a dictionary of the secret feature names and labels for subgroup count. A generic Z-Fed worker must load the model received from the server, propagate the model using the X, y pairs, calculate loss and update the wights, send the updates

of weights and subgroups count to the server.

C. Z-Fed framework

The workflow for the initialization, groups count, queue management, data augmentation, and model training of Z-Fed is described as follows:

1) The framework initializer generates a random private number n and uses $features$ to create as many client prototypes as population subgroups. Asynchronously, the server can instantiate the ML model and prepare weights.
2) Once the client prototypes are ready, the framework initializer can request the server to produce the required authentication tokens.
3) Workers are initialized and updated using the client prototypes, from this moment they can retrieve authentication tokens to the server, authenticate and receive the ML model for FL.
4) Optionally, the server can perform a data distribution analysis to consider augmenting the training process by varying the number of epochs for specific underrepresented subgroups.
5) Server authenticates workers and uses the updates to train the global ML model.
6) Rejected workers are organized into a structure of queues to reschedule the training efficiently.

A diagram of the Z-Fed workflow is shown in figure 5.

VI. Evaluation

A. Measure of equity in predictive models

Fairness and equity are general ideas, not restricted to AI. An application that implies decision-making processes can show discriminatory bias towards some specific groups and thus, must be evaluated in terms of fairness. The EU guidelines for trustworthy AI [24] define disparate treatment as a major concern in AI. In the fair credit reporting act (FCRA), fairness regards individual attributes such as gender, race, religion, age, sexual orientation, and more. An unfair or disparate treatment occurs when the outcome of a decision is biased by such factors. While for explainable algorithm it can be easier to identify possible discrimination, this represents a major challenge in FL [17], [25]. There are different possible metrics for measuring fairness in AI. Using the notation described in table II, we describe the metrics of choice for Z-Fed evaluation.

In this paper, we focus mainly on: 1) the difference in rate of favorable outcomes for unprivileged groups with respect to privileged groups i.e. statistical parity difference (SPD) across subgroups, defined as follows:

$$SPD = p(f = True|E \in UPR) - p(f = True|E \in PR)$$

2) the difference in rate of true positive prediction outcomes between privileged and unprivileged groups i.e. the equal odd difference (EOD) across subgroups, defined as follows:

$$EOD = p(TPR|E \in UPR) - p(TPR|E \in PR)$$

TABLE II: Metrics notation

Notation	Description
E	a generic individual of the population
f	a FL prediction of an individual feature
PR	a class of privileged individuals
UPR	a class of unprivileged individuals
TPR	true positive rate
FPR	false positive rate

and 3) the difference of probability to get true positive and false positives between privileged and unprivileged groups, i.e. the equal opportunity index (EPD), defined as follows:

$$EPD = \frac{p(FPR|E \in UPR) - p(FPR|E \in PR) + EOD}{2}$$

All of the evaluation metrics chosen are in the form of a subtraction of probability values and must be interpreted as follows: given a dataset with one privileged class and one unprivileged class, the possible value m of SPD, EPD, or EOD is such that $-1 \leq m \leq 1$. If $m = 0$ it means that no behavioral difference was measured during the predictions of the two classes, hence the predictive model is not discriminatory. If $-1 \leq m < 0$ the predictive model made discriminatory predictions favoring the privileged class, while if $0 < m \leq 1$ the model shows more favorable results in predicting the unprivileged class.

The settings of the experiments performed involve having multiple unprivileged classes and one privileged class. This requires to calculate the SPD, EPD, and EOD metrics one time for each unprivileged class, with respect to the privileged class. Since the purpose of Z-Fed is to mitigate the effect of imbalanced data in a FL environment, we decided to treat both kind of discriminatory behaviors, i.e. favoring privileged groups and favoring unprivileged groups, with the same importance. For a privileged class PR, and l unprivileged classes UPR_i, with $i = 1, \ldots, l$, we calculate the SPD, EPD, and EOD values l times with respect to PR to have a fine-grained measurement of equity. These evaluations can be expensive and difficult to interpret in presence of a high number of different subgroups, considering e.g. the possible combinations of ethnicity, gender, age, etc. For this reason, we consider a more convenient absolute value of equity $|m|$ such that $0 \leq |m| \leq 1$ and present the average of all the values obtained from the l unprivileged subgroups. To summarize, for each measurement m, across l unprivileged groups, the absolute multi-class equity score is:

$$\sum_{i=1}^{l} |m_i|/l$$

In this paper, we refer to the absolute multi-class measurements of statistical parity difference, equal opportunity difference, and equal odds difference, as SPD, EPD, and EOD, respectively.

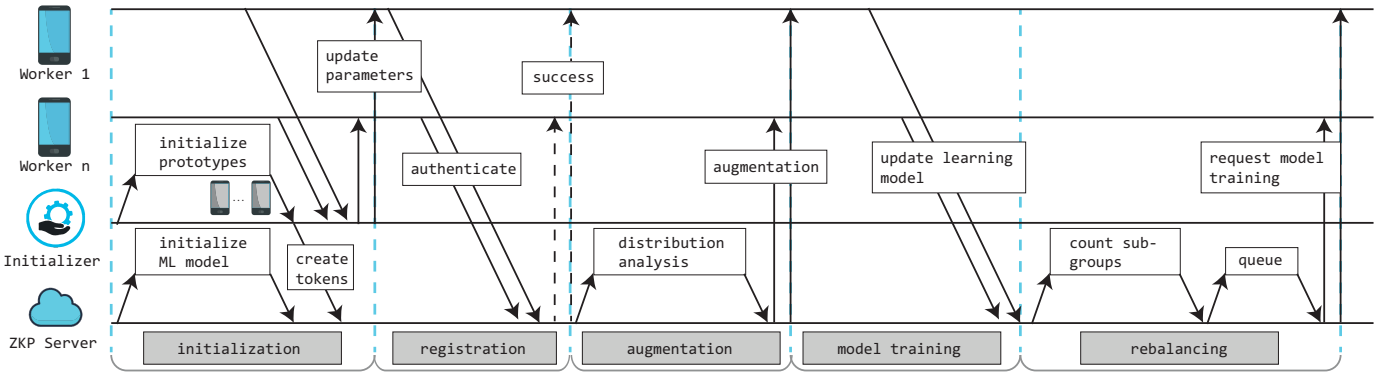

Fig. 5: Z-Fed workflow diagram: initialization, registration, data augmentation with population proportion analysis, federated model training, and rebalance of workers by ZKP count of subgroups.

B. Settings of the experiments

We used the datasets and the results of the experiment described in section III-B as a baseline. In the Z-Fed framework, the support for self-balancing learning can be arbitrarily disabled for testing purposes. We use the imbalanced datasets described in III-B and multiple instances of the same learning model for each experiment, and run training sessions on Z-Fed in order to obtain: 1) the Z-Fed model trained with highly imbalanced classes and self-balancing mode disabled, denoted as *imbalanced*, and 2) the Z-Fed model trained with the same highly class imbalanced dataset and self-balancing mode enabled, denoted as *rebalanced*. The self-balancing Z-Fed is set up to perform multiple training epochs on under-represented groups to counter-effect the fact of having a relatively small number of examples in the dataset. Identically as it was done for the FL experiment in section III-B, we tested the multiple instances of the same learning model on Z-Fed, using the face images as training features and predicting the age ranges. It is important to point out that the features used for creating imbalanced data, i.e. ethnicity, are not training features, means that the influence that can have on predictions is indirect. The age range chosen as feature to predict is, in each dataset, balanced, meaning that for the four age ranges *0-9, 10-19, 20-29, 30-39* have a proportion of 25%$\pm$1% each in every experiment. For this reason, we decided to measure also the accuracy performance of the model to confirm that the Z-Fed framework, aiming to reduce inequity, would not degrade the prediction accuracy. The 8 training models produced in the Z-Fed environment perform an average accuracy of 49.72 % with a variation of 0.52%, that is not distant from the accuracy of 50.8, variation 1.07% measured in the FL environment described in section III-B. This can be interpreted as the fact that Z-Fed improves equity with no significant loss of accuracy. The test sets, used to measure the equity scores in the *imbalanced* Z-Fed experiments, were sampled maintaining the original class proportion of the privileged and unprivileged subgroups. To test the performance of the *rebalanced* experiments we used a class balanced test, this decision is taken to respect the proportions of the balanced training set.

We measure the SPD, EOD, and EPD for each of the four experiments, the results are presented in table III.

By analyzing the proportion of the population groups, Z-Fed is able to request more training epochs to worker belonging to under-represented classes. This results in a bigger number of training updates for the *rebalanced* experiments.

In terms of SPD, Z-Fed successfully reduce the class bias in ASIAN-PR, INDIAN-PR, and WHITE-PR by 79.3%, 77.79%, and -32.95% respectively. Z-Fed produces a small SPD increment of 5.63% in BLACK-PR, meaning that the overall accuracy of the rebalanced learning model has the tendency of favoring either privileged or unprivileged groups in this particular experiment.

The measures of EPD show a considerable improvement in fairness in all the experiments ASIAN-PR, BLACK-PR, INDIAN-PR, and WHITE-PR, with a decrement of opportunity disparity of the 80.8%, 16.89%, 80.14%, and 36.34% respectively. The proportions about true positive results and false positive results in predictions improve considerably with the use of Z-Fed.

The EOD measurements also show a notable improvement in fairness across all the experiments. In ASIAN-PR, BLACK-PR, INDIAN-PR, and WHITE-PR, the odd disparity was reduced by 81.46%, 23.02%, 81.2%, and 39.97% respectively. The true positive rate measurement within privileged and unprivileged groups is considerably improved by the use of Z-Fed.

VII. Conclusions

FL is a promising ML method that ensures data privacy. However, we show how imbalance data lead to disparity in the UTK dataset. The Z-Fed framework proposed is able to mitigate FL bias by reducing disparities according to the EU guidelines for data ethics and trustworthy AI without compromising privacy. We show that ZKP enables to count the number of population samples keeping track of the proportion of subgroups, e.g. ethnicity, gender. Subgroups proportion data can be used to rebalance the FL samples and augment ML data, achieving an increment of fairness in terms of statistical

TABLE III: Z-Fed measurements of SPD, EPD, and EOD

Notation	Training updates	SPD		EPD		EOD	
ASIAN-PR [imbalanced]	16,008	0.1046	(reference)	0.1127	(reference)	0.038	(reference)
ASIAN-PR [rebalanced]	22,737	0.0216	-79.30%	0.0216	-80.80%	0.007	-81.46%
BLACK-PR [imbalanced]	16,008	0.0461	(reference)	0.0587	(reference)	0.0211	(reference)
BLACK-PR [rebalanced]	22,737	0.0488	+5.63%	0.0488	-16.89%	0.0162	-23.02%
INDIAN-PR [imbalanced]	16,008	0.087	(reference)	0.0977	(reference)	0.0344	(reference)
INDIAN-PR [rebalanced]	22,737	0.0194	-77.79%	0.0194	-80.14%	0.0064	-81.20%
WHITE-PR [imbalanced]	16,008	0.0381	(reference)	0.040	(reference)	0.0141	(reference)
WHITE-PR [rebalanced]	22,737	0.0255	-32.95%	0.0255	-36.34%	0.008	-39.97%

parity difference, equal odd difference, and equal opportunity difference. On average, Z-Fed improves the EPD of 53.54%, the EOD of 56.41%, and the SPD of 46.1% on imbalanced UTK datasets. In order to push this research further, it would be necessary to collect more data, and develop a version of Z-Fed capable of dealing with asynchronous updates using aggregation techniques such as FedAvg. The evaluations conducted show that some features may have an indirect influence on the training of a model. In the experiments proposed, the ethnic group is not a training feature, meaning that the ANN is not informed about this value. Nevertheless, the trained model shows bias towards certain groups, denoting that disproportions in the population ethnic groups must be taken into account for fairness. This leads to speculate about all the possible features that are not present in the data that could affect the equity in FL and the paramount importance of designing data models for machine learning.

ACKNOWLEDGEMENTS

We would like to thank our supervisors Dr. Rob Brennan and Dr. Lucy McKenna for the amazing support during the development of this project and for the incredible inspiration sparked by their suggestions and directions. We would like to thank all the DCU School of Computing Academic Body for this intense year of studies and for conveying the knowledge and the means to tackle this work and our feature challenges.

REPOSITORY AND PLAGIARISM STATEMENT

It is possible to find the repository of the project related to this paper at: *https://gitlab.computing.dcu.ie/marzos2/2022-mcm-ZKP-Federated-Framework-For-Balanced-Machine-Learning*
Please find the declaration on plagiarism at: *https://gitlab.computing.dcu.ie/marzos2/2022-mcm-ZKP-Federated-Framework-For-Balanced-Machine-Learning/-/blob/master/docs/documentation/plagiarism.pdf*

REFERENCES

[1] Junghye Lee, Jimeng Sun, Fei Wang, Shuang Wang, Chi-Hyuck Jun, and Xiaoqian Jiang. Privacy-preserving patient similarity learning in a federated environment: Development and analysis. *JMIR Medical Informatics*, 6:e20, 04 2018.

[2] Theodora Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International Journal of Medical Informatics*, 112, 01 2018.

[3] Andrew Hard, Chloé M Kiddon, Daniel Ramage, Francoise Beaufays, Hubert Eichner, Kanishka Rao, Rajiv Mathews, and Sean Augenstein. Federated learning for mobile keyboard prediction, 2018.

[4] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning, 2020.

[5] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. Federated learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(3):1–207, 2019.

[6] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2019.

[7] William J. Buchanan. *11 Zero-knowledge Proof (ZKP) and Privacy Preserving*, pages 337–368. 2017.

[8] Ioannis Chatzigiannakis, Apostolos Pyrgelis, Paul G. Spirakis, and Yannis C. Stamatiou. Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices. In *2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, pages 715–720, 2011.

[9] Adwait Pathak, Tejas Patil, Shubham Pawar, Piyush Raut, and Smita Khairnar. Secure authentication using zero knowledge proof. In *2021 Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–8, 2021.

[10] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *2017 IEEE Conference on*

Computer Vision and Pattern Recognition (CVPR), pages 4352–4360, 2017.

[11] Sumudu Samarakoon, Mehdi Bennis, Walid Saad, and Merouane Debbah. Federated learning for ultra-reliable low-latency v2v communications. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, 2018.

[12] Duo Liu, Chaoshu Yang, Shiming Li, Xianzhang Chen, Jinting Ren, Renping Liu, Moming Duan, Yujuan Tan, and Liang Liang. Fitcnn: A cloud-assisted and low-cost framework for updating cnns on iot devices. *Future Gener. Comput. Syst.*, 91(C):277–289, feb 2019.

[13] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. 2016.

[14] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 1175–1191, New York, NY, USA, 2017. Association for Computing Machinery.

[15] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, and L. Liang. Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. In *2019 IEEE 37th International Conference on Computer Design (ICCD)*, pages 246–254, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society.

[16] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1698–1707, 2020.

[17] 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, 2020.

[18] Patrik Joslin Kenfack, Adil Mehmood Khan, S.M. Ahsan Kazmi, Rasheed Hussain, Alma Oracevic, and Asad Masood Khattak. Impact of model ensemble on the fairness of classifiers in machine learning. In *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*, pages 1–6, 2021.

[19] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[20] Ying Zhou, Jiangang Shu, Xiaoxiong Zhong, Xingsen Huang, Chenguang Luo, and Jianwen Ai. Oversampling algorithm based on reinforcement learning in imbalanced problems. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 01–06, 2020.

[21] Mohammadhafez Bazrafshan and Nikolaos Gatsis. Convergence of the z-bus method for three-phase distribution load-flow with zip loads. *IEEE Transactions on Power Systems*, 33(1):153–165, 2018.

[22] Ronald Doku and Danda B. Rawat. Mitigating data poisoning attacks on a federated learning-edge computing network. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6, 2021.

[23] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, feb 1978.

[24] High-Level Expert Group on AI. Ethics guidelines for trustworthy ai. Report, European Commission, Brussels, April 2019.

[25] Ravikumar Balakrishnan, Mustafa Akdeniz, Sagar Dhakal, and Nageen Himayat. Resource management and fairness for federated learning over wireless edge networks. In *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, 2020.