CausalFedBlock : Decentralized causal federated learning with fair incentivization

Sreya Francis
MILA - Quebec AI Institute
University of Montreal
franciss@mila.quebec

Irina Rish
MILA - Quebec AI Institute
University of Montreal

***Abstract*—Federated learning has become increasingly popular as it facilitates collaborative training among multiple clients under the coordination of a central server, while preserving the client data privacy. In practice, some of the main challenges posed to federated learning is model robustness, out of distribution generalization, fair incentivization and vulnerability to various privacy attacks. We propose a novel blockchain based causal approach to developing a robust federated ecosystem that achieves generalization beyond the limited set of client environments accessible to the server during a specific round of training, with the capability to extrapolate to new and unseen environments in server/client side enhancing fair incentivization for all participants. For real world deployment of federated systems, participant incentivization based on causal/invariant learning as opposed to associational learning methods will prove to be extremely beneficial in terms of fairness, privacy and robustness.**

***Index Terms*—Federated Learning, Causal Machine Learning, Privacy, Blockchain, Incentivization**

I. Motivation

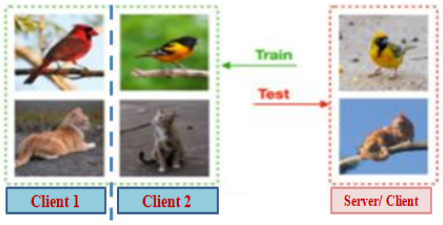

Fig. 1. Client data distribution in usual federated learning setting

Open threats to FL

It is next to impossible to observe client level data to be distributed independent and identically(iid) in a federated learning setting. If the server side environment or client side test environment in which inference is to be done differs from the environment in which the federated learning model was trained, we get poor results owing to the fact that correlations between features and the target are different. To be more specific, the federated model can only map from features to target in one environment resulting in inferring incorrect values of the target for the features in a different server/client environment.

Current associational learning techniques used in federated learning setting aims to exploit correlations between features to gain predictive performance and encodes the invariants of the dataset on which they're trained. Birds and branches are highly correlated due to which if birds dominantly appear on the dataset, we should expect this to be learned. This kind of learning can affect the incentivization mechanisms followed so far for rewarding paticipant/client contributions as the final model performance will highly vary when trained on biased data and tested on unbiased data.

Our goal is to develop a federated model highly confident of predictions outside the client level train datasets with enhanced robustness to distributional shifts away from the training set giving weightage to learning a representation common to all participating clients relying upon features that affect the target in all participating client environments ignoring individual client dataset-specific correlations ensuring fair incentivization for paticipating clients.

II. Background

A. Domain Generalization

Robustness to unseen domains is of utmost importance in federated setting and hence highly dependent on the fields of domain adaptation, domain generalization and invariant learning, all of which aim at enhancing model performance on a test distribution distinct from that of the training distribution. In the field of domain adaptation, the learner exploits the access to labeled data from the training domain and unlabeled data from test domain and performs well on the test domain. Domain generalization methods (15)(3)(10) in supervised learning use labeled data from multiple training domains while not requiring any unlabeled test data and learn models that generalize well to unseen domains. Domain generalization based methods seem to outperform domain adaptation methods in several use cases but have not been explored well in federated setting, which is the objective of this work. In our work, we rely on a recent domain generalization framework called invariant risk minimization (IRM) (15). The IRM uses the following principle to perform well on unseen domains: rely on features whose predictive power is invariant across domains, and ignore features whose predictive power varies across domains.

B. Blockchain based incentivization

The original concept of Federated Learning sought to allow users to keep ownership and privacy of their data during the

model training process by only returning the δ update based on local data and not the local data itself(12)(17). This approach can however lead to privacy breaches by analyzing the δ output of users(19)(8)(7); a proposed solution to these privacy issues is to add noise to the updates at a negligible loss(8)(19)(7). In the absence of blockchain security, distributed learning techniques have been suggested to use homomorphic encryption to protect training data (25)(6); however, Federated Learning's approach to only uploading δ updates ensures privacy, and the blockchain ensures security(12)(8) One implementation of Federated Learning known as *BlockFL* uses blockchain to reward users for their local updates proportional to how many local data points are used(11). The blockchain is meant to enhance privacy and security of local the update δ for the user, and validity of δ to the model(11). In (11), the reward system may not produce a lasting solution since many miners would be paying users "out-of-pocket". These rewards are proportional to the number of data points used for the update, and may be inflated by a malicious node; it is proposed for a miner to analyze computation time of users to combat this, though even that could be misrepresented(11). Though the BlockFL implementation warns that two miners simultaneously creating a block may cause two separate blockchains, using the "Longest Chain" rule will mitigate this (18). *DeepChain* looks into using blockchain for Distributed Deep Learning applications(25) as a means to keep both the data and the model private and secure. DeepChain proposes a incentive-based blockchain mechanism where both *parties* who upload data to the model, and *workers* who process the data and update the model, get paid an amount π_P and π_W based on their *contribution* ω_P and ω_W for parties and worker respectively.(16)proposed solutions to some of the issues faced in the previous approaches to improve privacy, access control and storage.

C. Invariant Risk Minimization

Consider datasets D_e, consisting of features of the feature vector $(x \in \mathcal{X})$ and the label $(y \in \mathcal{Y})$, from multiple training domains $e \in \mathcal{E}_{tr}$. (15) proposes to use these multiple datasets to learn a predictor $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the maximum risk over all the domains $\mathcal{E}$ which implies $\min_f \max_{e \in \mathcal{E}} R_e(f)$ where $R_e(f) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{P}_e}[l(f(\mathbf{x}), y)]$ is the risk under domain e for a convex and differentiable loss function l. The equations is given by

$$\min_{\Phi:\mathcal{X}\to\mathcal{Y}} \sum_{e\in\mathcal{E}_{tr}} R_e(\Phi) + \lambda \left\|\nabla_{w|w=1.0} R_e(w \cdot \Phi)\right\|^2$$

where $\lambda \in [0, \infty)$ is a regularizer balancing between the first term(standard ERM), and the invariance of the predictor $1 \cdot \Phi(x)$. $\Phi : \mathcal{X} \to \mathcal{Y}$ is an invariant predictor, $w = 1.0$ is just a dummy classifier, the gradient norm penalty measures the optimality of the dummy classifier at each domain e.

D. Fairness in Invariant Learning

In light of recent interest in ML community on how robust machine learning methods relate to algorithmic fairness, (2) studied whether algorithms from robust ML can be used to improve the fairness of classifiers that are trained on biased data and tested on unbiased data.In their experiments, they show that IRM(15) achieves better out-of-distribution accuracy and fairness than Empirical Risk Minimization (ERM) methods. (5) proposed another domain-invariant learning framework that incorporates Environment Inference to directly infer partitions that are maximally informative for downstream Invariant Learning with established connection to algorithmic fairness, which enables accuracy improvement and calibration in a fair prediction problem.

E. Federated Invariant Learning

With the above stated benefits of IRM in the field of learning, it is very evident that applying it to a federated setting will enhance robustness, privacy and fairness of the final learned model. (22) proposed an approach for the same in a federated setting and demonstrated the improvement in out of distribution accuracy as well as privacy of the final learned model.

III. Proposed Architecture

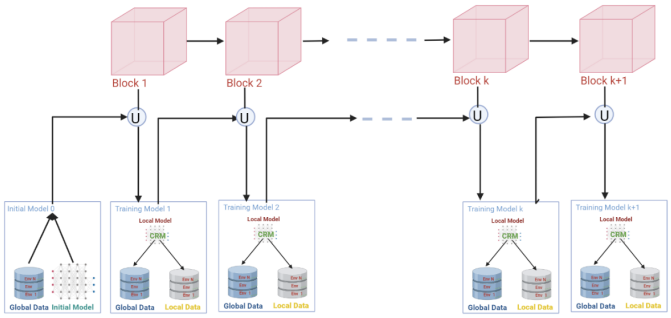

Fig. 2. Model T_{k+1} is calculated from applying the gradient values δ_i in Block $k+1$ to previous model T_k.

CausalFedBlock Architecture

In our proposed architecture, we are using Permissioned Blockchain which gives the owner $\mathcal{O}$ more control over who participates in the training process. This also gives the owner $\mathcal{O}$ full liability of payment for the device and miner work, as opposed to devices D needing to pay for their transactions (13), or miners to reward devices out-of-pocket (11). Although there may be a large number of devices uploading their δ values, the number of miners is restricted to however many the owner $\mathcal{O}$ designates the network needs; in practice the number of miners should be far less than the number of devices since the transactions are infrequent, only arriving once per device per training round at most. In our implementation, we have a model owner $\mathcal{O}$ who defines the initial model, shares the global data and distributes the reward. The *initial model* T_0 has all weights initialized to normally distributed values with mean 0 and variance 1; therefore, we define each weight $w \sim \mathcal{N}(0, 1)$.This initial model, defined as model-0, is seen as a black box to all users except O; we do not allow for any device or miner to view the model.

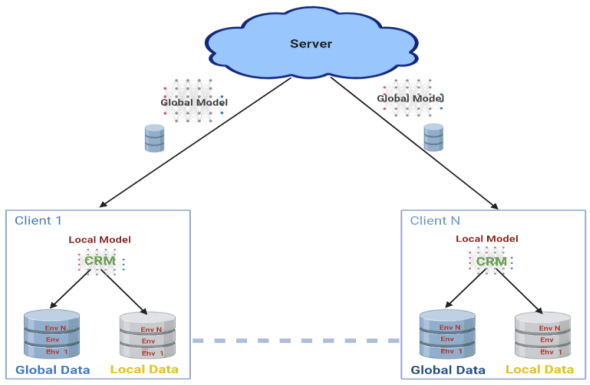

Fig. 3. Causal Federated Learning with Global Data Sharing Strategy
An Overview of Data Sharing Strategy

(22) proposed an approach to apply invaraint risk minimization on a federated setting based on which we built our pipeline for causalfedblock. In (26), it has been shown that globally shared data can reduce EMD(earth mover's distance) between the data distribution on clients and the population distribution which can help in improved test accuracy. As this globally shared data is a separate dataset from that of the client, this approach is not privacy sensitive. To enhance learning of invariant features on each round of training, the model owner shares a small subset of global data containing a distribution over all the classes/enviroments from the server to block 1 during the initialization stage along with model-0. The local model of each client is learned by minimizing the empirical average loss as well as regularizing the model by the gradient norm of the loss for both the shared data from the owner(global environment) and private data from each client(Local Environment). This enhances the learning of invariant features common to both the client and global data environments without losing the privacy of client side data.

Algorithm 1 CausalFedBlock - Data Sharing Strategy

ServerUpdate:

G $\leftarrow$ distribution over all environments owned by model owner

Initialize w_0
Initialize random portion of G as G_0
for each server epoch, t = 1,2,..k **do**
  Select random set of S clients
  Share G_0 and initial model with the selected clients
  for each client $k \in S$ **do**
    $w_{t+1}^k = ClientUpdate(k, w_t)$
  end for
  $w_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
end for

ClientUpdate(w):

$\mathcal{E}_{\text{tr}} \in$ [Client Env] $\bigcup$ [Global Env]
for each local client epoch, t=1,2,..k **do**
  $L_{\text{IRM}}(\Phi, w_t^k) = \sum_{e \in \mathcal{E}_{\text{tr}}} R^e(w \circ \Phi) + \lambda \cdot \mathbb{D}(w, \Phi, e)$
  $w_t^k = w_t^k - \eta \nabla L_{\text{IRM}}(w_t^k)$
end for
return w to server

In order to both keep the model from unaltered visibility from devices while allowing devices to train on the model, we employ homomorphic encryption to the model via the Paillier Cryptosystem which is commonly used in distributed deep learning(25). This symmetric cryptographic scheme has the property such that $E(x+y) = E(x) \cdot E(y)$, where $E(x)$ is the encryption of x. Since for a given weight in our training model we ultimately want to set $w \leftarrow w + \eta \cdot \bar{\delta}$(23), we can achieve this by setting $E(w) \leftarrow E(w) \cdot E(\eta \cdot \bar{\delta})$. To calculate model-1, a miner applies the aggregate of all δ_i updates from the 1st block; we denote the number of updates in block 1 by b_1. We define w_l as being the lth weight in T_0, and $\delta_{i,l}$ as the lth weight of the ith gradient value; the training model will apply the update

$$w_l \leftarrow w_l + \eta \cdot \bar{\delta}_l = w_l + \eta \cdot \frac{1}{b_1} \sum_{i=1,\ldots,b_1} \delta_{i,l}$$

to each $w_l \in T_0$, where η is the *learning rate*(23).

Similarly, to calculate model-$(k+1)$, the owner $\mathcal{O}$ applies the update

$$w_l \leftarrow w_l + \eta \cdot \bar{\delta}_l = w_l + \eta \cdot \frac{1}{b_{k+1}} \sum_{i=1,\ldots,b_{k+1}} \delta_{i,l}$$

to each $w_l \in T_k$. If there are currently K blocks in the blockchain, then model-K is the *Global Model*; this process is shown in Figure2. The consensus protocol we elect to use is that of *Byzantine Fault Tolerance* (BFT)(21)(20). This protocol has a set of *endorsement peers* who each run a transaction, and each output whether a transaction is valid or not. The BFT endorsement protocol requires $3f < n$, where f is the number of faulty nodes and n is the total number of nodes

taking part in BFT(20). In other words, we need more than 2/3 of the endorsement peers to agree on the validity of the transaction in order for consensus to be reached. Once consensus is determined for each transaction, the next Block k is created.

A. System Design

The system design is inspired from (16). The kth model is calculated from applying all of the model updates currently in the blockchain up to the kth block. We define $\mathcal{D}$ to be the set of all devices and $\mathcal{M}$ to be the set of all miners. As in Figure 4, step (1), each device $D_i \in \mathcal{D}$ has a copy of T_k given to it by O, along with the current version v_k; this is defined as *round* k. We walk-through the process of training the next model, validating the transactions and paying the users, referring to Figure 4 and 5.

For each device $D_i \in \mathcal{D}$ upon receiving the kth model T_k, (2) D_i trains the model T_k off-chain using it's local data $\boldsymbol{d_i}$ of size n_i to get an updated model T_k^i. The gradient is then calculated as $\delta_i = T_k^i - T_k$. (3) The values of δ_i, the size of $\boldsymbol{d_i} = n_i$, the device address a_i and the version v_k are set as parameters to the Smart Contract `UploadGradient()` together with a TxID, a timestamp and a digital signature. (4) The device D_i sends this transaction on-chain to its nearest miner M_j. (5a) Each miner $M_j \in \mathcal{M}$ adds the transactions received by the devices to their transaction queue, and simultaneously broadcasts the transactions to other miners; (5b) each miner receives the remaining transactions from other miners and add them to the final queue of transactions to be verified for Block k. (6) Each miner executes each transaction in its queue, which involves a call to `UploadGradient()` where details about each transaction are validated, such as the correct format for δ_i, the correct version v_i, and that the address a_i exists. (7a) Once validated, this transaction is added to the miner's pending queue for the next block; (7b) the device $D_i \in \mathcal{D}$ who submitted a valid transaction is rewarded an amount proportional to the data cost n_i via the submitted address a_i.

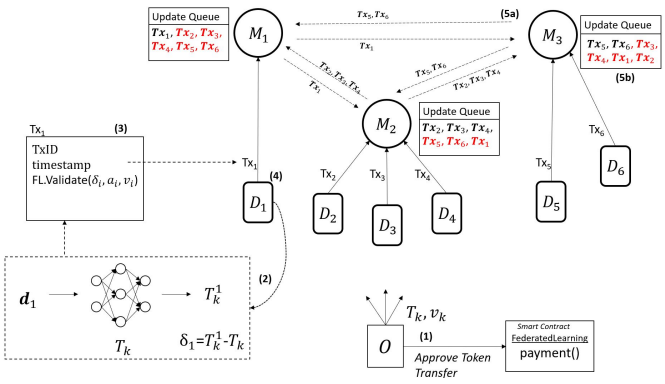

Fig. 4.
CausalFedBlock Workflow

The workflow of round k begins with (1) $\mathcal{O}$ distributing T_k and the version v_k to all devices, and simultaneously approving token transfer to the Payment() Smart Contract; (2) each device D_i uses a local dataset of size n_i and global dataset of size n_g to calculate gradient δ_i(12); (3) the values δ_i, n_i – the number of data points, a_i – the address of D_i, and v_i – the version of the model used, are packed into a Transaction, along with a TxID, timestamp and a Smart Contract function call to `UploadGradient()`; (4) each device D_i sends its transaction to its closest miner M_j; (5a) each miner M_j who received Tx_i validates the transaction and adds it to its queue, while simultaneously broadcasting the received transactions to all other miners and (5b) the miners who receive the remaining transactions have their final queue of transactions for training model T_k. The workflow continues with (6) miner M_j running each transaction in its queue, which involves a call to `UploadGradient()` where the format of δ_i, the correct version number v_i, and the existence of the address a_i are all checked; (7a) the transactions that run without errors are added to the miners queue for the next block and (7b) the devices who sent in the transaction are rewarded an amount of tokens proportional to n_i – the number of datapoints used for training.

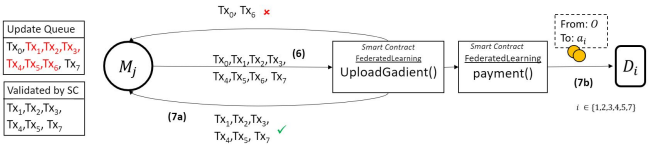

Fig. 5.
CausalFedBlock Incentivization

B. Computation and Storage

A device $D_i \in \mathcal{D}$ must calculate it's update value δ_i locally since we assume we cannot use any miner $M_j \in \mathcal{M}$ to aid in computation since sending local data would break the data privacy benefits of Federated Learning. In order to avoid the need for Mobile Edge Computing (14), it is assumed that a device $D_i \in \mathcal{D}$ has enough processing power and memory to calculate an updated model T_k^i from model T_k; this may not be realistic if the size of T_k becomes much larger than what we had considered. It is not assumed that D_i can store the entire blockchain in-memory, only the block headers are useful to D_i along with the Merkle Path to the reward transaction of D_i.

C. Restrictions for Permissioned Blockchain

This blockchain system is partially private, since all block data needs to be visible to all those that partake in the blockchain, but to nobody outside the system. We define a *member* x of the blockchain as either a device $x \in \mathcal{D}$, a miner $x \in \mathcal{M}$ or the model owner $x = \mathcal{O}$. The only restriction we put is that $\mathcal{O} \cap \mathcal{D} = \emptyset$, meaning $\mathcal{O}$ doesn't take part in the Federated Learning training process and none of the devices take part in the model aggregation process. To

achieve this, we are using a *permissioned blockchain* (4) for this task, and specifically we'll be working with Hyperledger Fabric (4) which has *permissions* which is a ruleset which defines which members have access to which actions. In this ruleset, we define the owner $\mathcal{O}$ as not being able to send any transactions that call the `UploadGradient()` Smart Contract, thus impeding it from contributing its own data to the process. Furthermore, we define a rule to limit a device's only action as sending in a transaction with a call to `UploadGradient()`. At any time, the owner $\mathcal{O}$ can amend this ruleset in order to complete the training process by stopping all devices from uploading gradients. In addition to having a ruleset to restrict/allow actions from members of the network, a permissioned blockchain gives the owner $\mathcal{O}$ full control over who joins the network in the first place, and can revoke access to joined devices at any time.

IV. Results

A. Dataset

Colored MNIST: Unlike the MNIST dataset which consists of digits 0-9 in grayscale, the colored MNIST dataset consists of input images with digits 0-4 colored red and labelled 0 while digits 5-9 are colored green with label with shape of the digit as the causal feature. In our causal federated learning setup, we split the dataset to two environments, each corresponding to a participant/client. We sample 2000 data points per client/server domain. Within the client environments, 80 - 90 % of inputs have their color correlated to the digit whereas within the central server test enviroment has just 10% color-digit correlation which helps in testing the robustness despite the spurious correlation within the inputs.

Chest Xray : To evaluate our approach on a real-world dataset, we make use of Chest X-ray images from different sources/domains: NIH (24), ChexPert (9) and RSNA (1). In this particular case, the inference task is to detect whether or not the input image corresponds to a patient with Pneumonia. In our experiments we included NIH and Chexpert as the 2 client training domains whereas RSNA served as the server side test domain. All the images of class 0 in the client domains are translated vertically downwards in order to create spurious correlation while the server side test domain remains unchanged.

B. Out of Distribution(OOD) Test results

For $n = 10$ clients each with an equal split of the Colored MNIST dataset, and running 15 rounds, we obtain the following accuracy metrics for the Final Model. .

Global Model Version	Accuracy
v.0	0.00949999839067459
v.1	0.1617999851703644
v.2	0.220199978351593
v.3	0.2942999958992004
v.4	0.3209999871253967
v.5	0.357300009727478
v.6	0.3942999792098999
v.7	0.4174999737739563
v.8	0.4260999798774719
v.9	0.4921999788284302
v.10	0.5585999774932861
v.11	0.5636999988555908
v.12	0.567400016784668
v.13	0.5738000059127808
v.14	0.5835000061988831
v.15	0.5926000180244446

TABLE I
RESULTS OF TRAINING 10 DEVICES FOR 15 ROUNDS.

TABLE II
TRAIN RESULTS

Dataset	FedAvg	CausalFed	CausalFedBlock
CMNIST	80.3%	57.42 %	55.32 %
Chest Xray	75.2%	72.7 %	71.2 %

TABLE III
TEST RESULTS

Dataset	FedAvg	CausalFed	CausalFedBlock
CMNIST	11%	60.3 %	**59.83** %
Chest Xray	45.7%	74.8 %	**75.01** %

We observed that when participating clients have out of distribution data in a federated setting, FedAvg does not fare well in the test data set which is drawn from a different distribution though they give highly accurate results on train set from similar distribution. CausalFed and CausalFed-Block perfoms much better on OOD test data for both Colored MNIST and Chest Xray datasets. In our setting, as we are recording all the transactions with a blockchain, it becomes even easier to fairly incentivize the participants for their contributions after the final model update.

V. Conclusion and Future Work

In this work, we addressed the problems of data privacy, out of distribution generalization and fair incentivization in a federated setting using invariant learning and blockchain. An in-depth workflow with detailed architecture and system design was presented on the development of a federated model highly confident of predictions outside the client level train datasets with enhanced robustness to distributional shifts away from the training set giving weightage to learning a representation common to all participating clients relying upon features that affect the target in all participating client environments ignoring individual client dataset-specific correlations ensuring fair incentivization for paticipating clients. Also, our work is the first to blend the fields of blockchain, federated learning

and causal/invariant learning, each of which makes incredible contributions to enhance the other.

For future work, improving model fairness will be of interest. In the Colored MNIST experiments the spurious covariate (colour) is easily inferable from image data using RGB channel information(2). Hence moving forward, we will focus on more biased datasets like the toxic comment detection task to evaluate the generalization and fairness properties of causalfedblock. One way would be to induce positive spurious correlation between the comment being about a particular demographic identity and comment toxicity construct across client domains followed by reversing that correlation strength in the owner/server domain. We also plan to work on medical records wherein attributes like demographic identity play a huge role in biased outcomes. We hope that this work serves as a base framework for similar applications on these datasets.

REFERENCES

[1] Kaggle: Rsna pneumonia detection challenge, 2018.

[2] Robert ADRAGNA, Elliot CREAGER, David MADRAS et Richard ZEMEL : Fairness and robustness in invariant learning: A case study in toxicity classification, 2020.

[3] Mathias Humbert Pascal Berrang-Mario Fritz Michael Backes AHMED SALEM, Yang Zhang : Mlleaks model and data independent membership inference attacks and defenses. 2018.

[4] Christian CACHIN : Architecture of the hyperledger blockchain fabric. *In Workshop on distributed cryptocurrencies and consensus ledgers*, volume 310, 2016.

[5] Elliot CREAGER, Jörn-Henrik JACOBSEN et Richard ZEMEL : Environment inference for invariant learning, 2021.

[6] DECENTRALIZED MACHINE LEARNING : Decentralized machine learning white paper, cited March 2019.

[7] Ristenpart T FREDRIKSON M, Jha S : Model inversion attacks that exploit confidence information and basic countermeasures. pages pp. 1322–1333, 2015.

[8] K.Zhang H. BRENDAN MCMAHAN, Ramage D.Talwar : Learning differentially private language models without losing accuracy. 2017.

[9] Jeremy IRVIN, Pranav RAJPURKAR, Michael KO, Yifan YU, Silviana CIUREA-ILCUS, Chris CHUTE, Henrik MARKLUND, Behzad HAGHGOO, Robyn BALL, Katie SHPANSKAYA, Jayne SEEKINS, David A. MONG, Safwan S. HALABI, Jesse K. SANDBERG, Ricky JONES, David B. LARSON, Curtis P. LANGLOTZ, Bhavik N. PATEL, Matthew P. LUNGREN et Andrew Y. NG : Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison, 2019.

[10] B Schölkopf K ZHANG, M Gong : Multi-source domain adaptation: A causal view. 2015.

[11] Hyesung KIM, Jihong PARK, Mehdi BENNIS et Seong-Lyun KIM : On-device federated learning via blockchain and its latency analysis. *arXiv preprint arXiv:1808.03949*, 2018.

[12] Jakub KONEČNÝ, H Brendan MCMAHAN, Felix X YU, Peter RICHTÁRIK, Ananda Theertha SURESH et Dave BACON : Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[13] A Besir KURTULMUS et Kenny DANIEL : Trustless machine learning contracts; evaluating and exchanging machine learning models on the ethereum blockchain. *arXiv preprint arXiv:1802.10185*, 2018.

[14] Pavel MACH et Zdenek BECVAR : Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3):1628–1656, 2017.

[15] Ishaan Gulrajani David Lopez-Paz MARTIN ARJOVSKY, Léon Bottou : Invariant risk minimization. 2019.

[16] Ismael MARTINEZ, Sreya FRANCIS et Abdelhakim Senhaji HAFID : Record and reward federated learning contributions with blockchain. *In 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 50–57, 2019.

[17] Brendan MCMAHAN et Daniel RAMAGE : Federated learning: Collaborative machine learning without centralized training data. *Google Research Blog*, 2017.

[18] Satoshi NAKAMOTO *et al.* : Bitcoin: A peer-to-peer electronic cash system. 2008.

[19] Moin Nabi ROBIN C. GEYER, Tassilo Klein : Differentially private federated learning: A client level perspective. 2017.

[20] Lakshmi Siva SANKAR, M SINDHU et M SETHUMADHAVAN : Survey of consensus protocols on blockchain applications. *In 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 1–5. IEEE, 2017.

[21] Joao SOUSA, Alysson BESSANI et Marko VUKOLIC : A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. *In 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 51–58. IEEE, 2018.

[22] Irina Rish SREYA FRANCIS, Irene Tenison : Towards causal federated learning for enhanced robustness and privacy, 2021.

[23] TORSTEN HOEFLER TAL BENNUN : Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. 2018.

[24] Xiaosong WANG, Yifan PENG, Le LU, Zhiyong LU, Mohammadhadi BAGHERI et Ronald M. SUMMERS : Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.

[25] J WENG, Jian WENG, J ZHANG, M LI, Y ZHANG et W LUO : Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *Cryptology ePrint Archive, Report 2018/679*, 2018.

[26] Liangzhen Lai Naveen Suda-Damon Civin Vikas Chandra YUE ZHAO, Meng Li : Federated learning with non-iid data. 2018.