A Reinforcement Learning-Based Approach to Graph Discovery in D2D-Enabled Federated Learning

Satyavrat Wagle, Anindya Bijoy Das, David J. Love, and Christopher G. Brinton
Elmore Family School of Electrical and Computer Engineering, Purdue University

GLOBECOM 2023 - 2023 IEEE Global Communications Conference | 979-8-3503-1090-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/GLOBECOM54140.2023.10437633

***Abstract*—Augmenting federated learning (FL) with direct device-to-device (D2D) communications can help improve convergence speed and reduce model bias through rapid local information exchange. However, data privacy concerns, device trust issues, and unreliable wireless channels each pose challenges to determining an effective yet resource efficient D2D structure. In this paper, we develop a decentralized reinforcement learning (RL) methodology for D2D graph discovery that promotes communication of non-sensitive yet impactful data-points over trusted yet reliable links. Each device functions as an RL agent, training a policy to predict the impact of incoming links. Local (device-level) and global rewards are coupled through message passing within and between device clusters. Numerical experiments confirm the advantages offered by our method in terms of convergence speed and straggler resilience across several datasets and FL schemes.**

I. Introduction

Federated Learning (FL) has become a popular approach for global machine learning (ML) model construction across a set of distributed edge devices. The standard operation of FL consists of a coordinating server periodically aggregating models trained locally at the edge devices on their respective local datasets. One of the fundamental challenges in FL is the presence of non-i.i.d data distributions across participating devices, which can slow convergence speed and result in global model bias [1]. These issues are exacerbated when some devices can only communicate their model updates to the server intermittently, e.g., due to poor channel conditions.

A recent trend of work has considered mitigating these issues through augmenting FL with device-to-device (D2D) communications in relevant network settings, e.g., wireless sensor networks [2]. In D2D-enabled FL, short-range information exchange is employed to reduce the tendency of devices to overfit on their locally collected datasets [3]. However, there are two factors which have a strong impact on the efficacy of such procedures: (i) *inter-device trust and privacy concerns*, which may prevent data sharing between specific device pairs, possibly for certain data classes; (ii) *D2D wireless condition variations*, which impacts communication efficiency and can result in intermittent data transmission failures.

In this paper, we ask: *How can we facilitate discovery of an effective D2D structure for FL systems taking these factors into account?* To answer this, we propose a reinforcement learning (RL) [4] methodology for identifying links between devices that maximize a reward measuring FL performance and communication efficiency. In its decentralized form, device-specific policies (i.e., agents) can learn to independently predict these links through low-overhead message passing without complete exposure of local data distributions.

Related work. A few studies have explored bias reduction in FL models through D2D information exchange. For example, they have considered offloading of (i) partial data sets to compensate for heterogeneous computation capabilities across devices [1], (ii) data to devices which are estimated to contribute more to system performance [3], and (iii) unlabelled data for decentralized pseudo-labelling [5]. Our work, by contrast, considers D2D graph discovery to jointly optimize expected learning improvement and communication reliability.

Other works have employed RL to address similar issues. For example, [6], [7] train policies at the server to select devices for aggregation that reduce the bias of the system model. In our work, by contrast, we leverage RL for D2D communication procedures. In addition, all of the above studies assume a centralized decision making system, which exposes additional device information to the network. To the best of our knowledge, a methodology to allow for *device level decision-making* in the presence of inter-device trust constraints and variable communication channels has not been studied.

Summary of Contributions

- We propose a decentralized RL methodology for cooperative discovery of an efficient wireless communication graph for a D2D-enabled FL system (Sec. III). In our scheme, devices act as RL agents, training local policies for link formation and engaging in data/reward sharing.
- Our RL reward modeling and message passing procedure results in a D2D communication structure that (i) encourages reliable communication of impactful information (ii) in the presence of variable network conditions, while (iii) maintaining privacy requirements.
- We evaluate our method by conducting experiments on established datasets and FL schemes (Sec. IV). Our method shows substantial improvement over baselines in terms of convergence speed, reliability of D2D communication, and robustness against the presence of stragglers.

II. System Model and Problem Formulation

In this section, we first detail our models for the wireless network and learning processes. Then, we formulate the D2D exchange graph discovery problem.

A. Network and Learning Models

We consider an FL system over a network of client devices $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_N\}$, hence $|\mathcal{C}| = N$; which are regularly

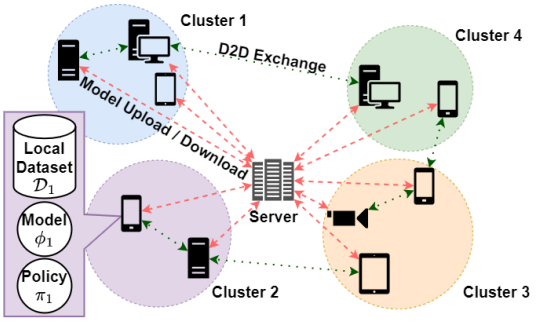

Fig. 1: System model for D2D-enabled FL. In our RL-based approach for graph discovery,where each device acts as a learning agent.

aggregated at server $\mathcal{S}$. Each device $\mathcal{C}_i$ has access to a local dataset $\mathcal{D}_i$ and a local model $\phi_i^t \in \mathbb{R}^p$ where p is the number of parameters, which is updated over the training period $t \in [0, T]$ to minimize a local cost function which is detailed in Section II-A4. Local models $\{\phi_i^t\}_{i\in\mathcal{C}}$ are aggregated every τ_a time steps at server $\mathcal{S}$ to obtain a global model ϕ_G^t, which is broadcast to all devices $\mathcal{C}_i \in \mathcal{C}$ for further local training. Now we discuss the aspects of our model that hold significance in any FL task.

1) Device-to-Device Communication

We assume that D2D communication can be established among the devices in $\mathcal{C}$ in order to exchange a subset of their local data-points with each other prior to the learning task. Recent works such as [8] imply that D2D exchange of a small number of data-points that reduce the non-i.i.d skew in local datasets yields significant performance gains in a learning task. However, predicting such D2D links may not have a straightforward solution additional resource costs required to account for unreliable channels due to network topology. To limit this cost, we allow every receiver $\mathcal{C}_i$ to receive data-points from at most one other remote device $\mathcal{C}_j$, resulting in at most one incoming edge per device over which it receives data.

Now, the signal received at $\mathcal{C}_i$ may suffer from attenuation due to channel conditions between $\mathcal{C}_i$ and $\mathcal{C}_j$, which is observed in the received signal strength (RSS) at $\mathcal{C}_i$. We consider a network architecture similar to [9], which assumes D2D communication conducted using OFDMA. In our scenario, for simplicity, we assume similar noise power σ^2 across all channels and a constant rate of transmission r between devices. Therefore, we can express the probability of unsuccessful transmission to $\mathcal{C}_i$ from $\mathcal{C}_j$ similar to [9] as

$$\mathbf{P}_D(i,j) = 1 - \exp\left(\frac{-(2^r - 1)\cdot\sigma^2}{\mathbf{W}_{i,j}}\right), \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{N\times N}$, such that $\mathbf{W}_{i,j}$ defines the RSS at $\mathcal{C}_i$ when it receives a signal from device $\mathcal{C}_j$. Thus, $\mathbf{P}_D$ is a useful indicator of system performance, which we use to design the reward function, which is detailed in Sec. III-C

2) Clusters of Reliable Devices

A low probability of failure of D2D communication is crucial. Hence, from the perspective of a receiver, it is important to identify links to remote devices which enable this. We therefore partition the devices in $\mathcal{C}$ into K disjoint clusters, where each device $\mathcal{C}_i$ belongs to a cluster k, denoted by $\mathcal{K}_k$, such that devices within a cluster are capable of reliably communicating between themselves. We define a reliable cluster of devices $\mathcal{K}_k$ as one where for all pairs of devices $\mathcal{C}_i, \mathcal{C}_j \in \mathcal{K}_k$; $\mathbf{P}_D(i,j) \leq \alpha_D$; where α_D is a reliability threshold set by the user. Thus, the probability of failure between any two devices within a cluster will always be less than α_D. We can now define two forms of D2D communication as **intra-cluster** and **inter-cluster** communication.

Now, in order to minimize data exchange over unreliable channels (i.e, inter-cluster communication), we define a budget $B(\mathcal{K}_k)$ for each cluster $\mathcal{K}_k$, which limits the total number of data-points **requested** over inter-cluster links. Thus, the number of data-points requested by devices in $\mathcal{K}_k$ from devices which are not in $\mathcal{K}_k$ can be at most $B(\mathcal{K}_k)$. For any $k = 1, 2, \ldots, K$. If $\mathbf{Q}_{j\to i}$ denotes the number of data-points requested by the receiver $\mathcal{C}_i \in \mathcal{K}_k$, we formally define this constraint as

$$\sum_{\mathcal{C}_i\in\mathcal{K}_k}\sum_{\mathcal{C}_j\notin\mathcal{K}_k} \mathbf{Q}_{j\to i} \leq B(\mathcal{K}_k). \tag{2}$$

3) Trust between Devices

In D2D communication, another desired property is protection against privacy breaches. In other words, devices are prohibited from sharing sensitive data with other devices unless the receiver is trusted by the transmitter; for example, a camera equipped device may want to share images other than those of humans for privacy concerns, except with certain trusted devices. We encode this notion of trust in a device specific trust matrix which is denoted by $\mathbf{T}_i \in \mathbb{R}^{N\times L}$, where L is the number of classes in the overall dataset $\mathcal{D} = \bigcup_{\mathcal{C}_i\in\mathcal{C}} \mathcal{D}_i$. The entries of trust matrix $\mathbf{T}_i$ belong to the set $\{0, 1\}$, given by

$$\mathbf{T}_i[j,\ell] = \begin{cases} 1 & \texttt{if } \mathcal{C}_i \texttt{ trusts } \mathcal{C}_j \texttt{ with class } \ell \\ 0 & \texttt{else.} \end{cases} \tag{3}$$

Thus, in our system model, we do not allow a device $\mathcal{C}_i$ to transmit data-points of class ℓ to device $\mathcal{C}_j$ if $\mathbf{T}_i[j,\ell] = 0$. Note that $\mathbf{T}_i[j,\ell] = 1$ does not imply that $\mathbf{T}_j[i,\ell] = 1$.

4) Learning Task

Finally, once intelligent D2D data exchange has been conducted; the local model ϕ_i^t at each device is updated at every time step t to achieve a local learning task, as described in Sec. II-A. We consider a classification task, where each client $\mathcal{C}_i$ has its own local data-set $\mathcal{D}_i$ which consists of tuples $(d,\ell) \in \mathcal{D}_i$ where d is the feature vector for any data-point and ℓ is the corresponding class. The performance of the local model ϕ_i^t depends on a loss function $\mathcal{L}(\phi_i^t, \mathcal{D}_i)$, where

$$\mathcal{L}(\phi_i^t, \mathcal{D}_i) = \sum_{(d,\ell)\in\mathcal{D}_i} \texttt{CELoss}(\phi_i^t, d, \ell) \tag{4}$$

where `CELoss` is the Cross Entropy Loss between the predicted and ground truth classes. Now, in the FL setting, the goal of the system is to learn a global model ϕ_G^* such that

$$\phi_G^* = \arg\min_{\phi\in\mathbb{R}^p} \sum_{i=1}^{|\mathcal{C}|} \mathcal{L}(\phi, \mathcal{D}_i) \tag{5}$$

The optimal global model is expected to perform the classification task with high accuracy across the global data distribution $\mathcal{D} = \bigcup_{i \in \mathcal{C}} \mathcal{D}_i$.

B. Graph Discovery Problem Formulation

As local ϕ_i^t models are updated, they are expected to diverge over the training iterations between aggregation [10], resulting in slow convergence of ϕ_G^*. Studies such as [11] have shown that this effect is more pronounced when the data-sets across devices are non i.i.d. Our aim is to enable faster convergence of ϕ_G^* through cooperative D2D information exchange by improving local data diversity.

Here we define class-distribution vector of local data at client $\mathcal{C}_i$ as $\mathbf{D}_i \in \mathbb{R}^L$, where L is the total number of classes in global dataset $\mathcal{D}$ and ℓ-th entry of $\mathbf{D}_i$ is the number of local data-points of class ℓ available in client $\mathcal{C}_i$. Now, due to the nature of stochastic gradient descent, which is used to optimize local models ϕ_i^t, the number of data-points required to create a noticeable improvement in the local data diversity (hence, in the learning task), must be above a certain threshold. We denote this threshold by $\mathbf{c}_i \in \mathbb{R}^L$, where any entry $\mathbf{c}_i[\ell]$ indicates the threshold for the corresponding class ℓ for client $\mathcal{C}_i$. $\mathbf{c}_i[\ell]$ can be user defined, as different scenarios may limit the number of data-points that can be shared over wireless channels. Now, we take into account the skew of classes across devices by first defining a diversity threshold $\hat{L}$, which is set by the user. We ensure that each device $\mathcal{C}_i$ has at least $\hat{L}$ classes available in their local dataset after D2D exchange by imposing the following constraint:

$$(\sum_{\ell=1}^{L} |\mathbf{D}_i[\ell] \geq \mathbf{c}_i[\ell]|) \geq \hat{L}. \tag{6}$$

Therefore, in short, our goal in this paper is to identify the links over the set of devices $\mathcal{C}$ that improve the diversity metric $\sum_{\ell=1}^{L} |\mathbf{D}_i[\ell] \geq \mathbf{c}_i[\ell]|$ for every $\mathcal{C}_i$, while ensuring that the requirement specified in (6) is fulfilled to create an optimal communication graph. Note that this needs to be optimized subject to the constraints mentioned in Sec. I which include (i) allowing at most one incoming edge for every device $\mathcal{C}_i$, (ii) maintaining a minimum received signal strength (RSS) for every transmission, (iii) not allowing the transmission of a prohibitively large number data-points between two different clusters as shown in (2) and (iv) abiding by the notions of trust defined between the devices in (3).

III. Proposed Methodology

In this section, we discuss our proposed method, which discovers an optimal D2D communication graph over the devices $\mathcal{C}$ to solve the problem established in Sec. II-B. In order to do so, we use a decentralized RL framework, which trains a set of local policies $\pi = \{\pi_i \in \mathbb{R}^N\}_{i \in [1,N]}$ that are used to jointly predict links among devices. These links aim for reliable sharing of salient data-points which reduce the skew of datasets at each client, while obeying system constraints.

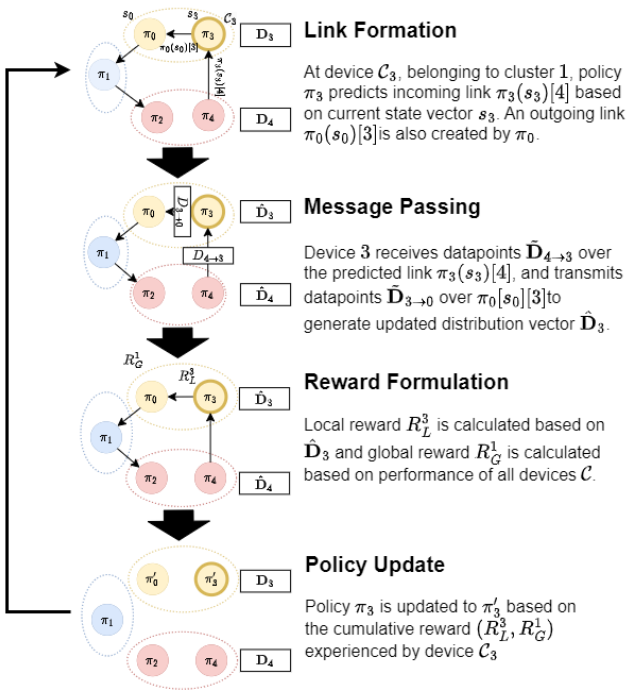

Fig. 2: The intelligent graph discovery process iteratively improves local policies in a decentralized manner by updating them such that information exchange over the predicted links maximize a system-wide performance metric.

We use the Q-Learning framework [4] for training the policy π, where each policy training episode consists of four steps. The first one is **link formation**, where the set of policies π predict a set of links over the graph of devices $\mathcal{C}$. The next one is the **message passing** procedure, which decides the payload of data-points to be transmitted over the predicted edges. After that, the **reward formulation** step calculates the utility of links predicted by π based on a reward function. The final step is the **policy update** which updates π in an iterative manner, based on the experienced rewards and leads to the discovery of an optimal graph. The steps are discussed below in details.

A. Link Formation

In this step, we first define the state at device $\mathcal{C}_i$ as $s_i^t = \{\mathbf{W}_{i,j} : \mathcal{C}_j \in \mathcal{C}\} \in \mathbb{R}^N$ indexed by t, where $\mathbf{W}_{i,j}$ is the RSS at device $\mathcal{C}_i$ while receiving a signal from device $\mathcal{C}_j$ as defined in (1). The set of possible states is given by $\mathcal{M}$, and $|\mathcal{M}| = S$. We also define an experience buffer at each device $\mathcal{C}_i$ as $\psi_i \in \mathcal{R}^{S \times N \times 2}$, which is used to store computed rewards and will be discussed in Sec. III-D. $\psi_i[s, j, 0]$ is the total reward and $\psi_i[s, j, 1]$ counts the frequency, respectively, for all times that π_i selects a link from device $\mathcal{C}_j$ when in state s over the policy training process. Each device $\mathcal{C}_i$ predicts an incoming edge from $\mathcal{C}_j$ using its local policy π_i and s_i^t with probability

$$\pi_i(s_i^t)[j] = \frac{\exp(\frac{\psi_i[t,j,0]}{\psi_i[t,j,1]})}{\sum_{k \in \mathcal{C}} \exp(\frac{\psi_i[t,k,0]}{\psi_i[t,k,1]})} \tag{7}$$

which addresses the exploration-exploitation issue in RL [4]. Once links are predicted for all receiving nodes, information is

exchanged over the edges of the directed graph over $\mathcal{C}$ based on the message passing algorithm, which we describe next.

B. Message Passing

Algorithm 1: D2D Message Passing

1: **Given :** Receiver node $\mathcal{C}_i$, Transmitter node $\mathcal{C}_j$, current state s, policy π
2: Transmitter $\mathcal{C}_j$ calculates the available data for exchange as $\mathbf{V}_{j\to i}$ using (8) and shares it with receiver $\mathcal{C}_i$
3: Receiver $\mathcal{C}_i$ computes the data diversity vector $\mathbf{D}_i$ according to (6)
4: Receiver $\mathcal{C}_i$ calculates the required data vector $\mathbf{Q}_{j\to i}$ using (9)
5: Transmitter $\mathcal{C}_j$ selects local data-points to add to transmission buffer $\mathbf{U}_{j\to i}$ using (10) and transmits them to receiver $\mathcal{C}_i$.

The goal of the message passing algorithm is to select data-points to be transmitted over a link, while maintaining notions of trust between the transmitter and the receiver. We also ensure that the D2D exchange does not result in the transmitter having a more biased local data-set than before. The logic for the message passing algorithm is as follows.

Let $\mathcal{N}_j$ be the set of devices requesting data-points from transmitter $\mathcal{C}_j$ after the link formation step. $\mathcal{C}_j$ first transmits the number of data-points that are available for sharing with all devices $\mathcal{C}_i \in \mathcal{N}_j$ as a vector $\mathbf{V}_{j\to i} \in \mathbb{R}^L$. We calculate $\mathbf{V}_{j\to i}[\ell]$ as follows

$$\mathbf{V}_{j\to i}[\ell] = \mathbb{1}_{\mathbf{T}_j[i,\ell]=1}(\mathbf{D}_j[\ell] - \mathbf{c}_j[\ell]) \tag{8}$$

The above equation ensures that transmitter $\mathcal{C}_j$ only shares those data-points that are allowed by trust matrix $\mathbf{T}_j$.

Upon receiving $\mathbf{V}_{j\to i}$, receiver $\mathcal{C}_i$ forms a requirement vector $\mathbf{Q}_{j\to i} \in \mathbb{R}^L$, where $\mathbf{Q}_{j\to i}[\ell]$ is the number of data-points of class ℓ requested by $\mathcal{C}_i$ from $\mathcal{C}_j$ and is calculated as follows

$$\mathbf{Q}_{j\to i}[\ell] = \begin{cases} \mathbf{V}_{j\to i}[\ell], & \texttt{if } \mathbf{c}_i[\ell] - \mathbf{D}_i[\ell] \geq \mathbf{V}_{j\to i}[\ell] \\ \mathbf{c}_i[\ell] - \mathbf{D}_i[\ell], & \texttt{if } 0 < \mathbf{c}_i[\ell] - \mathbf{D}_i[\ell] < \mathbf{V}_{j\to i}[\ell] \\ 0, & \texttt{if } \mathbf{c}_i[\ell] - \mathbf{D}_i[\ell] \leq 0 \end{cases} \tag{9}$$

The requirement vector $\mathbf{Q}_{j\to i}$ is then shared with transmitter $\mathcal{C}_j$. Based on $\mathbf{Q}_{j\to i}$, $\mathcal{C}_j$ selects data-points of class ℓ from $\mathcal{D}_j$ for all classes $\ell \in \mathcal{L}$ and forms a transmission buffer $\mathbf{U}_{j\to i} \in \mathbb{R}^L$. The transmission buffer $\mathbf{U}_{j\to i} \in \mathbb{R}^L$ contains the number of data-points that $\mathcal{C}_j$ is **actually** able to share. This may differ significantly from $\mathbf{V}_{j\to i}$ due to the different demands $R_{j\to i'}$ made by all $i' \in \mathcal{N}_j$ devices. Note that, if the total demand is higher than what $\mathcal{C}_j$ can afford to transmit, data-points are sent based on relative demand from each receiver $\mathcal{C}_i \in \mathcal{N}_j$. We calculate transmission buffer $\mathbf{U}_{j\to i}$ as follows.

$$\mathbf{U}_{j\to i}[\ell] = \begin{cases} \mathbf{Q}_{j\to i}[\ell], & \texttt{if } \sum\limits_{\mathcal{C}_{i'}\in\mathcal{N}_j} R_{j\to i'}[\ell] \leq \mathbf{D}_j[\ell] - \mathbf{c}_j[\ell] \\ \dfrac{\mathbf{Q}_{j\to i}[\ell]}{\sum\limits_{\mathcal{C}_{i'}\in\mathcal{N}_j} \mathbf{Q}_{j\to i'}[\ell]} \cdot (\mathbf{D}_j[\ell] - \mathbf{c}_j[\ell]), & \texttt{else} \end{cases} \tag{10}$$

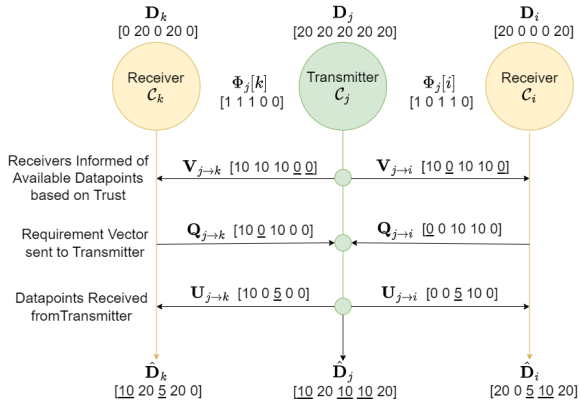

Fig. 3: An example of message passing process for $\mathbf{c}_j[\ell] = 10 \ \forall \ \ell$.

Now, as transmission buffer $\mathbf{U}_{j\to i}$ drops packets with probability $P_D(i,j)$ as per (1), receiver $\mathcal{C}_i$ is receives a buffer $\tilde{\mathbf{D}}_{j\to i}$, such that $\tilde{\mathbf{D}}_{j\to i}[\ell] \leq \mathbf{U}_{j\to i}[\ell] \ \forall \ \ell \in L$ and forms an updated class distribution vector $\hat{\mathbf{D}}_i$ as follows

$$\hat{\mathbf{D}}_i[\ell] = \mathbf{D}_i + \tilde{\mathbf{D}}_{j\to i} - \sum_{k\in\mathcal{N}_i} \tilde{\mathbf{D}}_{i\to k}, j \sim \pi_i(s_i^t) \tag{11}$$

In our simulations, we model the expected number of received data-points $\tilde{\mathbf{D}}_{j\to i}$ as $\tilde{\mathbf{D}}_{j\to i}[\ell] = [1 - P_D(i,j)]\mathbf{U}_{j\to i}[\ell]$.

The message passing algorithm is outlined in Algorithm 1.

Example 1. In Fig. (3), transmitter $\mathcal{C}_j$ shares data-points only from trusted classes with receivers $\mathcal{C}_i$ and $\mathcal{C}_k$, while preserving enough for its own threshold constraints to be satisfied. Consider $\ell = 3$, where the total demand $\mathbf{Q}_{j\to k} + \mathbf{Q}_{j\to i} = 20$, is greater than what is available at $\mathcal{C}_i$ to share, which is $\mathbf{V}_{j\to i}[\ell] = \mathbf{V}_{j\to k}[\ell] = 10$. Here, the demand is split between both, so that $\tilde{\mathbf{D}}_{j\to k}[\ell] = \mathbf{U}_{j\to i}[\ell] = 5$ and $\hat{\mathbf{D}}_j[\ell] = 10$.

Remark 1. Note that for any receiver $\mathcal{C}_i$, the data distribution $\mathbf{D}_i$ is never fully exposed to a transmitter $\mathcal{C}_j$, unless $\mathbf{T}_j[i,k] = 1 \ \forall \ k$ (complete trust). Also, due to (10), $\mathcal{C}_j$ may want to share fewer data-points from a class ℓ with requesting devices, as $\mathcal{C}_j$ must be left with at least $\mathbf{c}_j[\ell]$ data-points after each exchange.

Remark 2. Intuitively, the improved diversity of $\hat{\mathbf{D}}_i$ mitigates the detrimental effect of straggler devices [3] within the system by ensuring that data-points of any class are available at more devices. We explore this effect further in Sec. IV-B.

C. Reward Modelling

Now, we use the updated class distribution vectors $\hat{\mathbf{D}}_i$ to formulate a reward structure for the system. This enables the policies to learn device specific requirements through a local reward, while also optimizing the system-wide metrics via a global reward. Therefore, the cumulative reward experienced by each device $\mathcal{C}_i$ should take into consideration (i) performance of its local policy π_i, (ii) performance of other devices $\{\mathcal{C}_j\}_{j\neq i, j\in\mathcal{C}}$, (iii) reliability of the received signal, as defined in (1) and (iv) the inter-cluster transmission, as defined in (2).

We now briefly discuss the parameters influencing cumulative reward. The local data diversity, as defined in (6), $\hat{\mathbf{D}}_i \ \forall \ i \in [1, N]$ should increase after D2D data exchange to improve

convergence speed. Also, for a predicted link between $\mathcal{C}_i$ and $\mathcal{C}_j$, the probability of failed transmissions $P_D(i,j)$ as defined in (1), should be low in order to reliably receive signals over a selected edge. Also, as defined in (2), the total number of data-points received via inter-cluster exchange must be less than the data budget $B(\mathcal{K}_k)$. Trust concerns are handled by the message passing algorithm in (8).In order to incorporate all of the above metrics, the overall reward must constitute a tradeoff, which is characterized by user defined weights $\alpha_1, \alpha_2, \alpha_3$. The reward consists of two components, a **local reward** r_L^i specific to device $\mathcal{C}_i$, and a **global reward** $r_G^{\mathcal{K}_k}$ specific to cluster $\mathcal{K}_k$.

The local reward is independent of the system performance and captures the performance of the policy π_i for device $\mathcal{C}_i$ in terms of data diversity and reliability. In order to account for the data diversity requirement (6), for a given diversity threshold $\hat{L}$, we first define a score function $f : (\mathbb{R}^L, \mathbb{R}^L) \to \mathbb{R}$, which maps a diversity vector $\mathbf{D}_i$ and a set of threshold values $\mathbf{c}_i = [\mathbf{c}_i[1] \;\; \mathbf{c}_i[2] \dots \mathbf{c}_i[L]]$ as follows

$$f(\mathbf{D}_i, \mathbf{c}_i) = \begin{cases} \sum_{\ell=1}^{L} \mathbb{1}_{\mathbf{D}_i[\ell] \geq \mathbf{c}_i[\ell]} & \texttt{, if } \left(\sum_{\ell=1}^{L} \mathbb{1}_{\mathbf{D}_i[\ell] \geq \mathbf{c}_i[\ell]} \right) \geq \hat{L} \\ 0 & \texttt{, otherwise.} \end{cases}$$

The utility function f ensures that the predicted links satisfy the data diversity requirement in (6), by only returning rewards if the condition is met. Thus, we define the local reward as

$$r_L^i = \underbrace{\alpha_1 \cdot f(\hat{\mathbf{D}}_i, c)}_{\text{Data Diversity}} - \underbrace{\alpha_2 \cdot (P_D(i,j))}_{\text{Reliability Maximization}}, j \sim \pi_i(s_i^t). \quad (12)$$

Next, the global reward $r_G^{\mathcal{K}_k}$ captures the performance of the overall network, ensuring that all devices on average improve while cluster budget constraints are met. To that end, devices share their local r_L^i rewards with other devices in the network. Budget constraints are found by obtaining the number of data-points received over inter-cluster links for device $\mathcal{C}_{i'} \in \mathcal{K}_k$ as $\tilde{Q}_{\mathcal{K}_k} = \sum_{i' \in \mathcal{K}_k} |\mathbf{Q}_{j \to i'}|$ where $j \sim \pi_i(s_i^t)$ and $\mathcal{C}_j \notin \mathcal{K}_k$. We now define the global reward as

$$r_G^{\mathcal{K}_k} = \underbrace{\sum_{i \in C} \frac{r_L^i}{N}}_{\text{System Performance}} + \underbrace{\alpha_3 \cdot (B(\mathcal{K}_k) - \tilde{Q}_{\mathcal{K}_k})}_{\text{Cluster Budget}}; \quad (13)$$

The overall reward for a client $\mathcal{C}_i \in \mathcal{K}_k$ is given by $R_i^{\mathcal{K}_k} = r_L^i + \gamma \cdot r_G^{\mathcal{K}_k}$; where the weighting term γ governs the importance given to the overall performance of the system. If γ is large, the devices tolerate a large reduction in local rewards if the global reward improves as a result, while a small γ results in devices greedily optimizing their local rewards. Next, we discuss how the reward $R_i^{\mathcal{K}_k}$ is used to update local policy π_i.

D. Policy Update

We use a decentralized multi-agent Q-Learning algorithm to update a policy π_i in a state indexed by s, selected an edge from $\mathcal{C}_j$ resulting in a reward $R_i^{\mathcal{K}_k}$ as follows

$$\psi_i[s,j,0] = \psi_i[s,j,0] + R_i^c(t); \quad \psi_i[s,j,1] = \psi_i[s,j,1] + 1.$$

This update increases the probability of a policy predicting links that maximize the experienced rewards, specified in (7).

IV. Simulation Results and Discussion

In this section, we illustrate the advantages of our algorithm against baselines in terms of convergence speed, energy consumption, reliability of D2D communication, consistency in the presence of stragglers and delayed model aggregations.

A. Experimental Setup

We use the RadioML [12], CIFAR10 [13] and FashionMNIST [14] datasets for our evaluations. All datasets are split $80/20$ to obtain training and testing datasets respectively. We consider a network of $N = 25$ devices, and emulate non i.i.d training data across all devices. Each device has 990 samples per device for RadioML, and 1200 for the CIFAR10 and FashionMNIST from 4 different classes. We use a convolutional neural network (CNN) as the FL model for RadioML and CIFAR10, and a fully connected network for FashionMNIST.

B. Results and Discussion

Performance on Various Datasets: Now, we compare our algorithm on various datasets with the following baselines; (i) without data exchange and (ii) graphs generated using the Erdős-Renyi model with uniform edge selection probability (denoted as "uniform") paired with the message passing algorithm (Alg. 1). We show this comparison on RadioML, CIFAR10 and Fashion MNIST in Figs. 4(a), 4(b) and 4(c), respectively. The results illustrate that D2D information exchange using our method improves the FL performance significantly over both of the competing scenarios. We emphasize that our approach finds a desirably structured D2D communication graph, resulting in considerable improvement of the FL performance over the "uniform" case irrespective of overall dataset.

Varying FL Schemes: Next we apply our method to two other FL schemes: FedProx [15] and FedSGD [10], and compare the performance in Fig. 4(d). We observe that our method significantly outperforms both baselines which indicates that it can be applied over different popular FL schemes without sacrificing performance gains. In FedProx, our method complements the proximal dissimilarity term by reducing model bias via D2D exchange. In FedSGD, gradient aggregation is also benefited by our method as the bias in the model gradient is reduced due to data similarity.

Effect of Stragglers on Performance: We now study the performance of our method in the presence of straggler devices [3] in the FL system which do not participate in model aggregation. Thus, as the number of stragglers increases, fewer local models are aggregated. As each model is biased towards non-i.i.d local data, it reduces the accuracy of the global model. In Fig. 4(e), we choose stragglers randomly from the devices and show that our method is more resilient to stragglers than the baselines. It indicates the ability of our method to share data that reduces the bias of the aggregated model as the data exchange allows the system to make up for the bias introduced by stragglers. This shows that our method is inherently robust to node failure and heterogeneous communication capabilities.

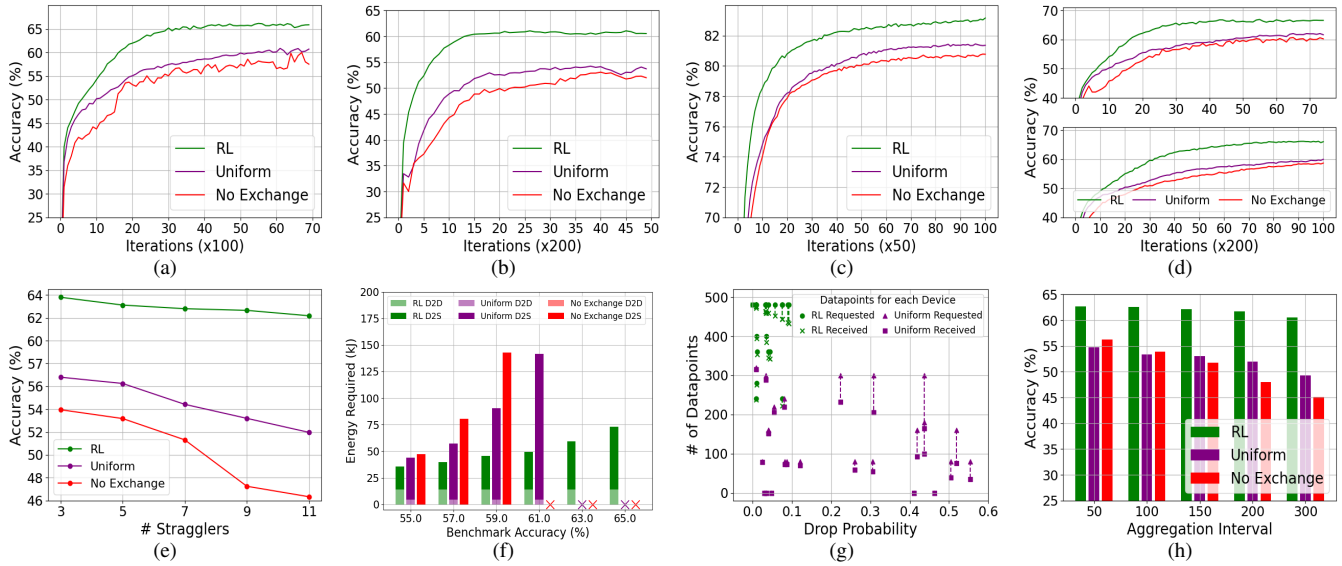

Fig. 4: Simulation results: Our method significantly improves performance over baselines for (a) RadioML ,(b) CIFAR10 and (c) FMNIST. It can be used to augment existing federated learning algorithms such as (d) FedProx (top) and FedSGD (bottom). It retains performance in the presence of stragglers (e) and consumes less energy to reach performance milestones (f). (g) Our method significantly improves the probability of successful D2D transmission. (h) The performance of our method remains relatively consistent over larger global aggregation intervals.

Energy Consumption to reach Benchmarks: Next, we conduct a simulation to compare the energy required by our method to achieve performance benchmarks with baselines. We use the wireless energy consumption model [16] to calculate the energy consumed for D2D and device-to-server (D2S) communication. In this simulation, we assume that the D2S distance is $3\times$ the average D2D distance. Fig. 4(f) shows that our method uses significantly less energy to reach benchmarks as baselines despite the initial overhead due to D2D exchange. Note that in the "uniform" case, suboptimal links result in fewer data-points exchanged, resulting in lower D2D energy, but consequently significant higher D2S energy as a result.

Reliability of D2D Performance: In Fig. 4(g), we study D2D reliability in terms of the probability of successful transmission and the cluster budget. We observe that our method consistently predicts links to reduce inter cluster communication while improving system performance. In practice, it results in a reduction of number of transmitted packets (which is not the case in "uniform" graphs), thus saves additional costs required to ensure successful transmission over unreliable channels.

Change in Aggregation Interval: Next, we observe the effect of various aggregation intervals τ_a, or the frequency of local models synchronization. A low τ_a can result in faster convergence, but involves a larger overhead due to frequent D2S communication required for synchronization. Fig. 4(h) shows that as τ_a becomes larger, our method outperforms the baselines by a considerable margin which indicates its resilience to delays in model aggregation, and a lower local model drift. Thus, a small initial overhead for our method results in significant gains in D2S overhead by retaining similar performance.

References

[1] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, "From federated to fog learning: Distributed machine learning over heterogeneous wireless networks," *IEEE Commun. Mag.*, 2020.

[2] S. Shen, Y. Han, X. Wang, and Y. Wang, "Computation offloading with multiple agents in edge-computing–supported IoT," *ACM Trans. Sen. Netw.*, 2019.

[3] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *IEEE Conf. Comput. Commun. (INFOCOM)*, 2021.

[4] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[5] X. Pei, X. Deng, S. Tian, L. Zhang, and K. Xue, "A knowledge transfer-based semi-supervised federated learning for IoT malware detection," *IEEE Trans. on Dependable Secure Comput.*, 2022.

[6] P. Zhang, C. Wang, C. Jiang, and Z. Han, "Deep reinforcement learning assisted federated learning algorithm for data management of IIoT," *IEEE Trans. Industr. Inform.*, 2021.

[7] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE Conf. Comput. Commun. (INFOCOM)*, 2020.

[8] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *Intl. Conf. on Mach. Learn.*, 2020.

[9] F. P.-C. Lin, S. Hosseinalipour, S. S. Azam, C. G. Brinton, and N. Michelusi, "Semi-decentralized federated learning with cooperative d2d local model aggregations," *IEEE J. Sel. Areas Commun.*, 2021.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Intl. Conf. Artif. Intell. and Stat. (AISTATS)*, 2017.

[11] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Trans. on Neural Networks and Learning Sys.*, 2020.

[12] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engr. App. of Neur. Net.*, 2016.

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Toronto, Ontario, Tech. Rep., 2009.

[14] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017.

[15] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Mach. Learn. and Sys.*, 2020.

[16] L. Xu, C. Jiang, Y. Shen, T. Q. S. Quek, Z. Han, and Y. Ren, "Energy efficient D2D communications: A perspective of mechanism design," *IEEE Trans. Wirel. Commun.*, 2016.