Chapter 3

A Federated Learning Approach with Imperfect Labels in LoRa-based Transportation Systems

This chapter presents a federated learning approach to simultaneously enhance communication and provide robustness against the imperfect labels for the long-range based transportation system. The presented approach performs the selection of inferior participants, followed by a discussion of the data reduction and inclusion mechanisms on each inferior participants using a dataset on the server. The chapter also cover the mathematical formulation for the selection of the optimal fraction of the server dataset and afterwards, the chapter elaborates the experiments.

3.1 Introduction

Intelligent Transportation System (ITS) is an emerging paradigm that integrates sensing, analysis, control, and communication technologies to ensure safer and more informative mobility of the users and goods [58, 59]. ITS finds its application in various domains, including transportation modes recognition, traffic management, speed control, travel time improvement, incident management, and effective implementation of transportation policies [60–62]. The evolution toward developing compact-size and low-cost sensory devices and micro-controller units accelerates the growth of ITS. Various sensors deployed on vehicles and transportation infrastructures sense and collect a vast amount of transportation data. ITS can use the collected dataset to train different learning models to perform automated recognition or prediction. However, data sharing in ITS to remotely train the learning models hampers data privacy and generates

threats.

Federated Learning (FL) is an emerging paradigm that facilitates collaborative learning among participants without compromising data privacy [6, 63–65]. Thus, FL proves to be a key enabler technique to preserve privacy and minimize communication delay using local training of models on the participants and global aggregation of the parameter metrics. However, imperfect labels in the datasets of participants appear to be a challenging problem in FL [66–69]. Such imperfect labels may be generated due to poor annotation mechanisms, labelling through crowd-sourcing, non-expertise volunteers, web-based queries, *etc.* The imperfect labels in the datasets deteriorate the performance of the trained model; thus, reducing the utility of FL.

The transmission of information from the vehicles to the processing unit in ITS is another challenge. It is due to the following reasons, *i.e.,* mobility of vehicle, a considerable distance between vehicles and the processing units, and unstable wireless connectivity. Long-Range Wide Area Network (LoRaWAN) possesses a star-of-stars network topology and supports medium access control to provide low power and long-range communication [70]. LoRaWAN applications grow exponentially in ITS due to a longer communication range, low power consumption, and ease of implementation. Thus, it provides seamless connectivity between the vehicle and the data collection unit, even at a large distance. LoRaWAN architecture comprises the following entities: LoRa Nodes (LNs), LoRa Gateway (LG), Network Server (NS), and an Application Server (AS). A vehicle comprising LN and sensors to sense the vehicle information and transfers it to the LG. LN shares the information with the LG using LoRa communication. Later, the LG transfers data to the AS via NS [71]. Figure 3.1 illustrates an example scenario of FL in ITS using LoRa.

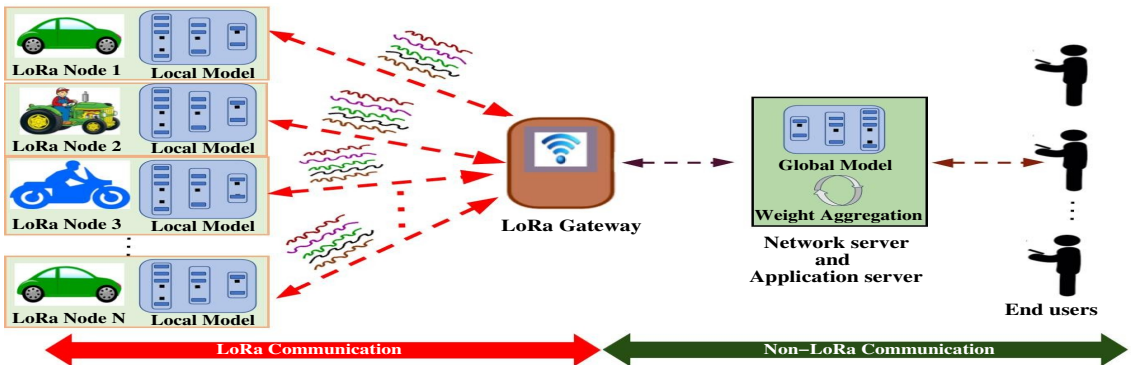


Figure 3.1: Illustration of an example scenario of FL in ITS using LoRa.

This chapter proposes an FL approach that simultaneously provides efficient communication and robustness against the imperfect labels in the datasets of the participant

devices. We assume a perfectly annotated dataset at the server, collected independently and properly annotated by the expertise. We collect the information of vehicles to monitor different parameters, including speed, left turn, right turn, acceleration, deacceleration, and GPS locations. We consider the data transmission from vehicles (participants) to the server using the LoRa. Further, we exploit the perfectly annotated server dataset to improve participants performance with imperfect labels.

Specifically, we explore the following problem in this chapter: *how to develop a federated learning approach that simultaneously provides efficient communication and robustness against imperfect labels in the dataset collected from the vehicles?* To solve the problem, this work proposes a Federated learning approach with imperfect labels for LoRa-based transportation systems, abbreviated as Fed-LoRa. The approach first collects the vehicles information. Fed-LoRa next estimates the class-wise centroids of participants and server. The estimated centroids of participants are compared against the server to determine inferior participants based on similarity scores. The participants with imperfect labels in their local datasets are termed as *inferior participants*. We made the following major contributions in this work:

- **Data collection using LoRa:** The first contribution is to transfer the information from vehicles to the processing unit in a communication efficient manner. Each vehicle has an LN to collect and transfer the information to the LG using LoRa communication. We use LoRa for energy-efficient transmission of limited size data in a long communication range.
- **Estimation of class-wise centroids:** Fed-LoRa estimates the class-wise centroids of the local datasets at participants and server. Initially, the approach generates feature vectors from the raw data on the participants and server. The generated features vector passes to a fully connected layer with a fixed neuron, generating equi-length feature vectors of the same length. The equi-length feature vectors are used to estimate the class-wise centroid vectors.
- **Selection of inferior participants:** The next contribution is to transfer class-wise centroid vectors from participants to the server. Afterwards, the similarity score is estimated between the received and server's centroids. The participants with any of the class similarity scores less than a threshold are inferior.
- **Data reduction and inclusion mechanisms:** The inferior participants demand a fraction of the server dataset to improve their similarity score, thus, performance. We propose novel data reduction and inclusion mechanisms that ensure that the number of instances in the local dataset does not exceed the existing ones. In other words, the data size remains the same or less than ex-

isting. The mechanism facilitates performance improvement without increasing local epochs with instance count.

- **Optimal fraction of server dataset:** We further propose an optimization problem and derive an expression for the optimal fraction of the server dataset transferred to the inferior participants. The received fraction of the server dataset minimizes the impact of imperfect labels in the local dataset of the inferior participants. We also propose an algorithm to demonstrates all the steps required in the Fed-LoRa approach.
- **Experimental validation:** We perform the experimental evaluations to validate the effectiveness of Fed-LoRa on the existing [72] dataset. The results show that the proposed Fed-LoRa provides effective communication and robustness against the imperfect labels in the local dataset of the participants.

The rest of the chapter is formatted as follows. In the next section, we present the preliminaries and network topology of Fed-LoRa. Section 3.3 elaborates Fed-LoRa, which provides efficient communication and robustness against the noisy labels in the local dataset of the participants. Further, we evaluate the proposed approach in Section 3.6. Finally, the chapter concludes with future work in Section 3.7.

3.2 Background and preliminaries

This section describes the preliminaries and overview of this work. Table **??** shows the list of notations used in this work.

3.2.1 Preliminaries

This work considers a scenario of the Fed-LoRa approach, which comprises a set $\mathcal{P}$ of N participants (vehicles), *i.e.,* $\mathcal{P} = \{p_1, p_2, \cdots, p_N\}$. Each participant p_i shares the information to participate in the federation. The participant p_i has its local dataset $\mathcal{D}_i$, where $1 \leq i \leq N$. Similarly, let $\mathcal{D}$ denotes the server dataset having m instances with correctly annotated class labels. This work assumes the server posses dataset, which is free from label imperfections. The participants having imperfect labels in their dataset are termed as inferior participants, and are Q in numbers, where $Q \leq N$. $\mathcal{D}$ consist of m instances, *i.e.,* $\mathcal{D} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{m}$, where $\mathbf{x}_j \in \mathbb{R}^h$ denotes instance j of length h and is associated with a class label $y_j \in \mathbf{y}$. $\mathbf{y} = \{y_1, y_2, \cdots, y_k\}$ is a set of k class labels in $\mathcal{D}$. Similarly, $\mathcal{D}_i = \{\mathbf{X}_i, \mathbf{y}_i\}$ denote the dataset available at participant p_i, which includes the local dataset collected over a fixed period of time, where $1 \leq i \leq N$.

Definition 3.1 (Federated learning) *Federated learning refers to the concept of alleviating the privacy compromise and minimizing the communication overhead while training a shared model without using data from multiple decentralized participants [73]. The participants transfer trained model parameter matrices despite data to the server for aggregation and receive a global parameter matrix for further training.*

Definition 3.2 (Imperfect labels) *In a LoRa-based transportation system, imperfect labels refer to data labels that are incomplete, noisy, inaccurate, or inconsistent. Labels in such systems often correspond to the ground truth of specific attributes or states, such as vehicle positions, transportation modes, or operational statuses. Imperfect labels can arise due to limitations in data collection, transmission errors, environmental interference, or human error during labelling.*

Definition 3.3 (Data reduction) *Data reduction mechanisms in Fed- LoRa aim to optimize the participant's local dataset by removing less relevant feature vectors, thereby reducing the dataset size while maintaining its performance. The key method eliminates feature vectors farthest from the centroid, using Euclidean distance as a metric. This ensures that only the most representative data points remain. The number of removed instances corresponds to the data points required from the server, preserving the dataset's balance. By doing so, Fed-LoRa minimizes the communication overhead of transferring large data fractions while improving class-wise similarity scores.*

Definition 3.4 (Data inclusion mechanisms) *The data inclusion mechanism in Fed-LoRa integrates a fraction of the server dataset into the participant's reduced dataset without increasing its cardinality. This ensures that the number of instances in the local dataset remains constant, preserving its structure while enhancing class-wise similarity. The added data fraction is carefully selected to align with the reduced dataset's properties, maintaining centroid and similarity scores. These mechanisms allow Fed-LoRa to optimize the balance between local data refinement and server data integration, achieving improved model performance with minimal computational and communication overhead.*

3.2.2 Overview of Fed-LoRa approach

We train a shared model on N participants using their local datasets in the considered scenario of monitoring vehicles. It reduces the communication delay and maintains the privacy of the participant dataset in FL. Fed-LoRa comprises N participants and a server, where each participant holds processing and storage devices. A participant p_i

maintains a local model (obtained initially from the server) and periodically updates its locally collected dataset $\mathcal{D}_i$, where $1 \leq i \leq N$. The participant p_i next trains the local model on $\mathcal{D}_i$ for E epochs. The updated weight parameter matrix from p_i is further transferred to the server. The server aggregates the weight parameter matrices from all the participants and sends back the aggregated matrix to them for the next round of the training.

Fed-LoRa provides the mechanisms to handle imperfectly labeled datasets at the participants. Fed-LoRa starts with the estimation of class-wise centroids on all the participants and the server, discussed in Section 3.3.1. The participants next transfer the class-wise centroids to the server, where similarity scores are estimated to determine inferior participants (Section 3.3.2). We later introduce data reduction and insertion mechanisms to improve the similarity score of the inferior participants (Section 3.3.3). We further determine an optimal fraction of the class-wise dataset transmitted from the server to the participants (Section 3.3.4). We finally introduce the mechanisms of local and global (Section 3.3.5) updates to complete the shared model training via FL.

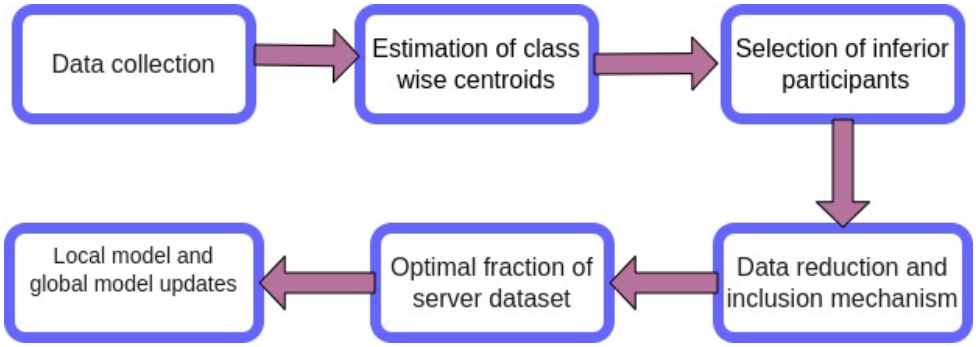


Figure 3.2: An overview of components underlying Fed-LoRa.

3.3 Fed-LoRa: Federated learning approach with imperfect labels using LoRa

This section covers the details of the proposed federated learning approach that provides communication efficiency and robustness against imperfect labels in the dataset, namely Fed-LoRa. We next present an algorithm (Algorithm 4.1) that summarizes different steps involved in Fed-LoRa. Further, we derive an expression for the data fraction demanded by participants from the server to improve the similarity score. Figure 3.4 illustrates the components underlying Fed-LoRa.

3.3.1 Estimation of centroid vector

This sub-section describes the mechanism of estimating centroid vectors of all the classes in the local dataset of a participant and the server. Let m_i^c denotes the instances of class c $(1 \leq c \leq k)$ in the dataset $\mathcal{D}_i^c$ at participant p_i, where $1 \leq i \leq N$. We estimate feature vectors of each instance in $\mathcal{D}_i^c$ despite using raw dataset. We use fully connected layer with the same number of neurons, denoted as n, to generate equal size vectors on all participants and server. Let $\mathbf{f}_{il}^c$ represent the feature vector at p_i of instance l in m_i^c, where $l \in \{1 \leq l \leq m_i^c\}$. We estimate $\mathbf{f}_{il}^c$ using ReLU function as: $\mathbf{f}_{il}^c = \text{ReLU}(\mathbf{w}_{il}^c \mathbf{x}_{il}^c + \mathbf{b}_{il}^c)$, where $\mathbf{w}_{il}^c, \mathbf{x}_{il}^c$, and $\mathbf{b}_{il}^c$ are weight, input, and bias vectors of length n, respectively. Thus, each instance in $\mathcal{D}_i^c$ becomes a vector and we estimate centroid of the vector. The centroid $\mathbf{C}_i^c$ of p_i for class c is estimated as:

$$\mathbf{C}_i^c = \frac{1}{m_i^c}\{\mathbf{f}_{i1}^c \oplus \mathbf{f}_{i2}^c \oplus \cdots \mathbf{f}_{il}^c \oplus \cdots \mathbf{f}_{im_i^c}^c\}, \tag{3.1}$$

where $\oplus$ denotes the element-wise sum operation. We also estimate centroid vector $(\mathbf{C}^c)$ of server for class c on its dataset $\mathcal{D}$ using Equation 3.1. Afterwards, the participants transfer their centroid vectors to the server for further operations.

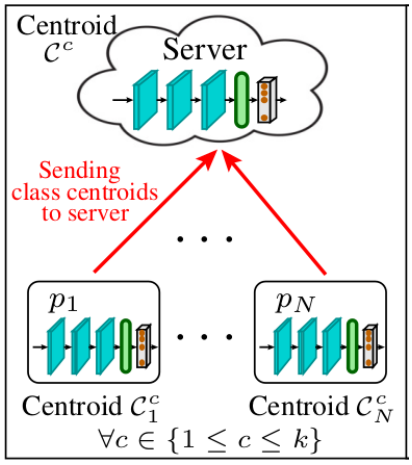


Figure 3.3: An overview of Estimation of centroid vector

3.3.2 Selection of inferior participants

When the server receives the class-wise centroid vectors from all the participants, it estimates the similarity score between the received and own centroid vectors. This work uses cosine similarity to calculate the similarity score because even if the two centroid vectors are far apart by the Euclidean distance, they could still have a smaller

angle between them. Let $\mathcal{S}_i^c(\cdot)$ denotes the similarity score between centroid vectors $\mathbf{C}_i^c$ and $\mathbf{C}^c$ of participant p_i and server for class c, respectively, where $1 \leq c \leq k$ and $1 \leq j \leq N$. $\mathcal{S}_i^c(\cdot)$ is estimated as:

$$\mathcal{S}_i^c(\mathbf{C}_i^c, \mathbf{C}^c) = \frac{\sum_{p=1}^{n} C_{ip}^c . C_p^c}{\sqrt{\sum_{p=1}^{n} (C_{ip}^c)^2} . \sqrt{\sum_{p=1}^{n} (C_p^c)^2}}, \tag{3.2}$$

where C_{ip}^c and C_p^c $(1 \leq p \leq n)$ are the elements of centroid vectors $\mathbf{C}_i^c$ and $\mathbf{C}^c$, respectively. If $\mathcal{S}_i^c(\cdot) \leq 0$ then we replace it with zero (*i.e.,* $\mathcal{S}_i^c(\cdot) = 0$) to avoid negative value.

We compare $\mathcal{S}_i^c$ with a threshold value ϵ_o^c for class c, which depends on the number of instances and level of imperfections in the datasets. We use expected value of similarity $\mathcal{S}_i^c$ for class c of local dataset at p_i to determine ϵ_o^c, is estimated as:

$$\epsilon_o^c = \mathbb{E}[\mathcal{S}_i^c] = \sum_{i=1}^{N} a_i(\mathcal{S}_i^c), \tag{3.3}$$

where a_i is the probability of similarity score of participant p_i $(1 \leq i \leq N)$ and $\sum_{i=1}^{N} a_i = 1$. The participant p_i having $\mathcal{S}_i^c \leq \epsilon_o^c$ for any class c is said to be inferior, as it does not possess sufficient similarity with the server dataset $(\mathcal{D})$. In other words, the participant has data with imperfect labels. From the expression for $\mathcal{S}_i^c$, we can conclude that the inferiority of a participant can be the result of a single class in the local dataset of the participant having similarity $< \epsilon_o^c$. Therefore, while developing a mechanism to handle imperfect labels, class-wise data analysis could be beneficial.

We utilize the perfectly annotated data of the server to handle label imperfections in the participant's dataset for the specific classes. Let Q denotes the number of inferior participants out of N, *i.e.,* $Q < N$. Transmitting the same amount of data from the server to all the Q participants is tedious, as it incurs significant transmission delay. In addition, different participants have unequal data demands. Thus, we determine the optimal fraction of data transmitted from the server to the participant.

3.3.3 Data reduction and inclusion mechanisms

The inferior participants require a fraction of the server dataset to improve their class-wise similarity score, assuming D^c has no imperfect labels. It improves the class-wise performance of the participant. The naive approach is to transfer a maximum fraction of the server dataset to the participants. However, it requires significant communication resources. Fed-LoRa determines the optimal fraction of the sever dataset on the

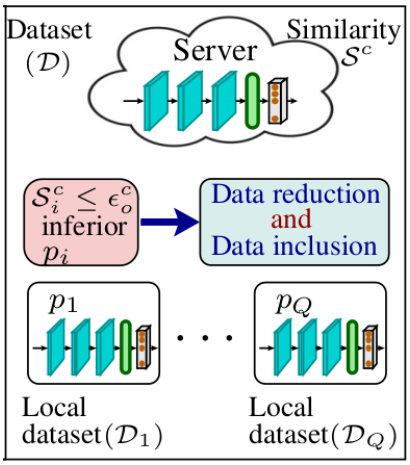


Figure 3.4: An overview of Selection of inferior participants.

participant to improve the similarity score.

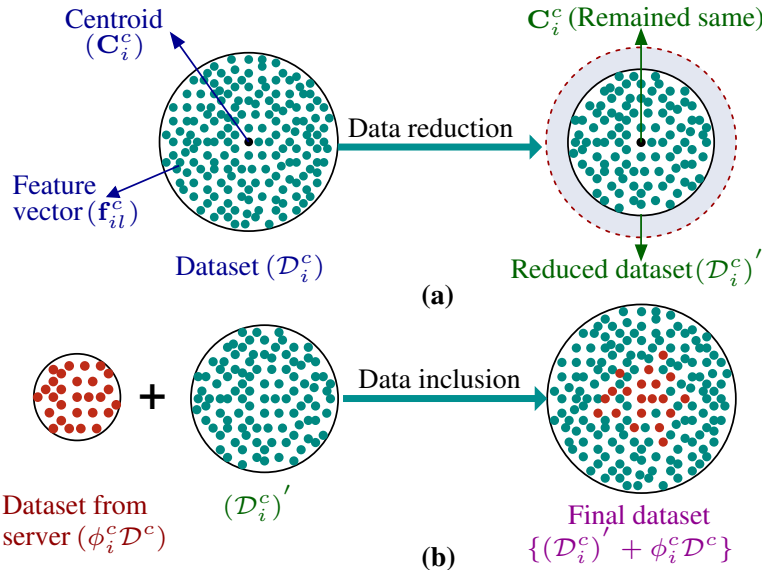


Figure 3.5: Illustration of (a) data reduction and (b) data inclusion mechanisms in Fed-LoRa.

This sub-section introduces the novel mechanisms of *data reduction* and *data inclusion*, which are prerequisites to estimate the optimal data fraction. Figure 5.3.2.2 illustrates an overview of data reduction and data inclusion mechanisms in Fed-LoRa. In data reduction, we eliminate those feature vectors which are at the farthest distance from the centroid. We use Euclidean distance to estimate the distance between centroid and feature vectors. The number of instances eliminated from the local dataset equals the number of instances demanded from the server. The mathematical expression of

the data reduction mechanism is covered in Definition 3.5.

Definition 3.5 (Data reduction) *Let local dataset $\mathcal{D}_i^c$ at participant p_i of class c having centroid $\mathbf{C}_i^c$, where $1 \leq i \leq Q$ and $1 \leq c \leq k$. Let ϕ_i^c is the fraction of server dataset $\mathcal{D}^c$ to be added with $\mathcal{D}_i^c$. Then the data reduction from $\mathcal{D}_i^c$ to $(\mathcal{D}_i^c)'$ must satisfy the property covered in Definition 3.6 and is obtained using following steps:*

1. *Estimate cardinality (number of instances) of fraction ϕ_i^c of $\mathcal{D}^c$ to be added in $\mathcal{D}_i^c$, i.e., represented as $|\phi_i^c.\mathcal{D}^c|$.*
2. *Remove $|\phi_i^c.\mathcal{D}^c|$ entries from $\mathcal{D}_i^c$ having maximum distance from the centroid $\mathbf{C}_i^c$ to obtain $(\mathcal{D}_i^c)'$ such that $\mathbf{C}_i^c(\mathcal{D}_i^c) = \mathbf{C}_i^c((\mathcal{D}_i^c)') \implies \mathcal{S}_i^c(\mathcal{D}_i^c) = \mathcal{S}_i^c((\mathcal{D}_i^c)')$.*

Figure 5.3.2.2(a) demonstrates the data reduction mechanism.

Data inclusion is the next step after data reduction. It ensures that the fraction of data added to the reduced dataset should not increase its number of instances. The mathematical representation of data inclusion is given in Definition 3.6.

Definition 3.6 (Data inclusion) *If ϕ_i^c fraction of server dataset $\mathcal{D}^c$ for class c is added to the local dataset $\mathcal{D}_i^c$ of participant p_i then it must satisfy following condition:*

$$|\mathcal{D}_i^c| = |(\mathcal{D}_i^c)'| + |\phi_i^c.\mathcal{D}^c|, \tag{3.4}$$

where $|\mathcal{D}^c|$ denotes the cardinality or number of instances in dataset $\mathcal{D}^c$ for class c. $(\mathcal{D}_i^c)'$ is the reduced dataset at participant p_i for class c, where $1 \leq i \leq Q$ and $1 \leq c \leq k$. Figure 5.3.2.2(b) depicts the data inclusion ($(\mathcal{D}_i^c)' + \phi_i^c.\mathcal{D}^c$) mechanism of the Fed-LoRa approach.

These specialized mechanisms of data reduction and data inclusion help to simultaneously improve similarity score (or performance) and maintain the same number of instances in the local dataset of the participants. Thus, suppressing the increment in the local epochs for training due to an increase in the number of data instances via additional data insertion.

In this section we further added more explanation about the Optimal fraction of the server dataset demanded by participants without increasing local epochs for training.

The expression for the optimal fraction of the server dataset, ϕ_c^i, ensures that data inclusion does not increase the number of local epochs required for training. In Fed-LoRa, this is achieved by balancing the number of instances removed from the local dataset during data reduction with the number of instances added during data inclusion. Specifically, the fraction ϕ_c^i is defined such that:

$$|D_c^i| = |(D_c^i)'| + |\phi_c^i \cdot D_c| \tag{3.5}$$

Here, $|D_c^i|$ represents the cardinality of the original local dataset, $(D_c^i)'$ is the reduced dataset after data reduction, and $\phi_c^i \cdot D_c$ is the fraction of the server dataset added to the local participant's dataset. This equality ensures that the total number of instances in the participant's dataset remains constant, preventing an increase in local epochs for training. By carefully estimating ϕ_c^i based on the similarity score and centroid distance, Fed-LoRa achieves an optimal trade-off between data inclusion and computational efficiency.

3.3.4 Optimal fraction of server dataset

We define an optimization problem to obtain an optimal fraction of server dataset ϕ_i^c for class c for participant p_i, where $1 \leq i \leq Q$ and $1 \leq c \leq k$. This addition of data improves the similarity score and the performance of the inferior participant. The optimization problem is given as follows:

$$\min_{\phi_i^c} \quad \phi_i^c.\mathcal{D}^c, \tag{3.6a}$$

$$\text{s.t., } \mathbf{c_1}: \ \mathcal{S}_i^c(\mathbf{C}_i^c, \mathbf{C}^c) \geq \epsilon_o^c, \tag{3.6b}$$

$$\mathbf{c_2}: \ 0 < \phi_i^c < 1. \tag{3.6c}$$

• *Objective function:* The objective of the optimization problem in Equation 3.6 is to estimate the optimal value of ϕ_i^c, which improves the performance of the inferior participant p_i.
• *Constraints:* Constraint $\mathbf{c_1}$ indicates that the fraction of data received from the server ensures that the similarly-score of class c for participant p_i exceeds the threshold ϵ_o^c. It removes the inferiority of the participant due to less similar data. Constraint $\mathbf{c_2}$ restricts data fraction (ϕ_i^c) to lies in range $(0, 1)$.
• *Solution:* Solving the problem in Equation 3.6 is difficult due to the following two reasons: (a) the non-linear function in Equation 3.6b restricts the estimation of an exact analytical expression for ϕ_i^c, and (b) the optimal value of ϵ_o^c depends on dataset and level of imperfection. Thus, we can determine the near-optimal ϕ_i^c for each participant using the heuristic-based approach. The relation between ϵ_o^c and ϕ_i^c is given in Theorem 3.1.

3.3.5 Local model and global model updates

The server sends a randomly initialize model on all the N participants in the federation. The participants train the received model on their local datasets for E epochs. This work uses the second-order method for performing local updates on participants, where

we adopt canonical Newton's approach of the form $-\nabla^2(\mathcal{L}_i)^{-1}\nabla\mathcal{L}_i$ [74]. $\nabla^2(\mathcal{L}^i)^{-1}$ and $\nabla\mathcal{L}_i$ are the second-order and first-order gradients of loss function $\mathcal{L}_i$ on participant p_i, where $1 \leq i \leq N$. The canonical approach reduces the convergence time and minimizes the accumulation error. Each participant p_i send its weight parameter matrix to the server for aggregation. Let $\mathbf{W}^{[t]}$ denotes the aggregated weight parameter matrix of the shared model at global iterations t, where $1 \leq t \leq R$. Each participant p_i first computes the local gradients of its loss $\mathcal{L}_i$ for class c as: $\delta_{ci}^{[t]} = \nabla\mathcal{L}_i\big(\mathbf{W}^{[t]}\big)$, where $1 \leq c \leq k$ and $1 \leq i \leq N$. Next, the participant p_i computes the second-order gradient (Hessian matrix) at $\mathbf{W}^{[t]}$ as follows: $q_{ci}^{[t]} = \nabla^2\mathcal{L}_i\big(\mathbf{W}^{[t]}\big)$. Finally, the local model at p_i with η learning rate, is updated as:

$$\mathbf{W}_i^{[t+1]} = \big\{\mathbf{W}_{ci}^{[t+1]}\big\}_{c=1}^{k}, \mathbf{W}_{ci}^{[t+1]} = \mathbf{W}_{ci}^{[t]} - \eta\big(\frac{\delta_{ci}^{[t]}}{q_{ci}^{[t]}}\big). \tag{3.7}$$

When all the N participants finish local training for E epochs, they transfer the updated weight parameter matrices to the server. Next, the server performs weight parameter matrices aggregation to update the global model. Further, the server broadcasts the aggregated weight parameter matrix to the participants for the next round of local training. Assuming that all the participants have the same set of classes, the server can aggregate weight parameter matrices from the participant at global iteration $t+1$ as: $\mathbf{W}^{[t+1]} = \sum_{i=1}^{N} \frac{m_i}{m}\mathbf{W}_i^{[t+1]}$, where $m = m_1 + m_2 + \cdots + m_N$.

Algorithm 4.1 illustrates operations of components involved in Fed-LoRa. Step 2 to 4 covers the estimation of centroid followed by the selection of inferior participants in Steps 5 to 10. Later, data reduction and inclusion are performed on inferior participants, followed by local and global updates, depicted in Step 14 to 20 of Algorithm 1.

The computational efficiency of the proposed Fed-LoRa algorithm is summarized as follows:

1. **Time Complexity:**
   - *Centroid Estimation:* The time complexity for estimating centroids at participants is $O(nk)$, where n is the number of data instances per participant, and k is the number of classes. This step involves generating feature vectors and computing centroids for each class.
   - *Similarity Score Calculation:* The cosine similarity between participant and server centroids has a complexity of $O(kn)$, as it involves operations proportional to the size of the feature vectors.

- *Global Aggregation:* Each participant transfers its local weight matrix to the server. The aggregation at the server is $O(Nk)$, where N is the number of participants.

2. **Data Reduction and Inclusion:** Data reduction and inclusion mechanisms involving Euclidean distance calculations, for instance, elimination, have a complexity of $O(n)$ per class.
3. **Local and Global Model Updates:**
4. Local updates require $O(EL)$, where E is the number of local epochs and L is the size of the local dataset. Global weight aggregation and broadcasting updates have a complexity of $O(Nk)$.
5. **Overall Complexity:** For R global iterations, the total complexity of the Fed-LoRa algorithm is approximately $O(R \cdot N \cdot (k + EL))$, balancing the overhead of local updates, centroid estimations, and global aggregations.

This analysis demonstrates that the Fed-LoRa algorithm achieves computational efficiency that is suitable for federated learning in LoRa-based systems by optimizing communication and computational resources.

Lemma 3.1 *The ratio of increment in $\Delta\mathcal{S}_i^c$ of the inferior participant p_i for class c to the fraction of dataset ϕ_i^c transmitted from the server is constant, i.e., $\left(\frac{\Delta\mathcal{S}_i^c}{\phi_i^c}\right)$ is constant, where $1 \leq i \leq Q$ and $1 \leq c \leq k$. Given: $\Delta\mathcal{S}_i^c = \mathcal{S}_i^c((\mathcal{D}_i^c)' + \phi_i^c\mathcal{D}^c) - \mathcal{S}_i^c(\mathcal{D}_i^c)$.*

Proof: Let $\mathcal{D}_i^c$ and $|\mathcal{D}_i^c|$ denote the local dataset at participant p_i for class c and its cardinality, respectively, where $1 \leq i \leq Q$ and $1 \leq c \leq k$. We can obtain the centroid vector $\mathbf{C}_i^c$ for $|\mathcal{D}_i^c|$ instances using Equation 3.1. Further, we can estimate similarity score ($\mathcal{S}_i^c$) of participant p_i for class c by incorporating Equation 3.2. Let $\mathcal{S}_i^c(\mathcal{D}_i^c)$ denotes the similarity score for dataset $\mathcal{D}_i^c$ for class c on participant p_i. Let ϕ_i^c is the fraction of server dataset for class c that is transmitted to the participant p_i. We can obtain the new dataset at the inferior participant p_i ($1 \leq i \leq Q$), *i.e.*, $\{(\mathcal{D}_i^c)' + \phi_i^c\mathcal{D}^c\}$ using Definition 3.5 and Definition 3.6. The similarity score on new dataset is given as:

$$\mathcal{S}_i^c((\mathcal{D}_i^c)' + \phi_i^c\mathcal{D}^c) = (1-\gamma)\underbrace{\mathcal{S}_i^c((\mathcal{D}_i^c)')}_{\mathbf{T1}} + \gamma\underbrace{\mathcal{S}_i^c(\phi_i^c\mathcal{D}^c)}_{\mathbf{T2}}, \tag{3.8}$$

where γ and $(1-\gamma)$ are the contributions of the term **T2** and **T1**, respectively. We can obtain γ as: $\gamma = \frac{|\phi_i^c.\mathcal{D}^c|}{|(\mathcal{D}_i^c)'|+|\Omega_c^j\mathcal{D}^c|}$. The contribution of term **T1** >> contribution of **T2**. Thus, without loss of generality, we can consider $(1-\gamma) >> \gamma \implies (1-\gamma) \approx 1$. Now, we can rewrite Equation 3.8 as:

Algorithm 3.1: Fed-LoRa Algorithm

Input: Set $\mathcal{P}$ of N participants with their local dataset;
Output: Trained model on all participants;

Server builds a model and broadcasts to participants;
for $i \in \{1, 2, \cdots, N\}$ *and* $c \in \{1, 2, \cdots, k\}$ **do**
    p_i estimate centroid vector $\mathbf{C}_i^c$ using Equation 3.1;
    Server estimate centroid vector $\mathbf{C}^c$;
Server sends the estimated centroid to the participants;
for $i \in \{1, 2, \cdots, N\}$ *and* $c \in \{1, 2, \cdots, k\}$ **do**
    Estimate similarity score $\mathcal{S}_i^c$;
    if $\mathcal{S}_i^c \leq \epsilon_o^c$ **then**
        flag[ ]$\leftarrow$ *append*(p_i);
Determine Q inferior participants: $Q \leftarrow |\text{flag}|$;
for *each inferior participant* $i \in \{1, 2, \cdots, Q\}$ **do**
    Solve optimization problem in Equation 3.6;
    Perform *data reduction* and *data inclusion*;
while $t \leq$ *global_epochs* **do**
    for *each participant* p_i, $i \in \{1, 2, \cdots, N\}$ **do**
        $\mathbf{W}_i^{[t+1]} \leftarrow \left\{\mathbf{W}_{ci}^{[t+1]}\right\}_{c=1}^{k}$ using Equation 3.7;
        Send $\mathbf{W}_i^{[t+1]}$ to the server;
    $\mathbf{W}^{[t+1]} = \sum_{i=1}^{N} \frac{m_i}{m} \mathbf{W}_i^{[t+1]}$;
    Broadcast $\mathbf{W}^{[t+1]}$ to the participants for $t+1$;
    $t \leftarrow t+1$;
return Trained model on each participant;

$$\mathcal{S}_i^c((\mathcal{D}_i^c)' + \phi_i^c \mathcal{D}^c) = \mathcal{S}_i^c((\mathcal{D}_i^c)') + \gamma \mathcal{S}_i^c(\phi_i^c \mathcal{D}^c). \tag{3.9}$$

From Definition 3.6, we have $\mathcal{S}_i^c(\mathcal{D}_i^c) = \mathcal{S}_i^c((\mathcal{D}_i^c)')$; thus, Equation 3.9 can be rewritten as:

$$\begin{aligned} \mathcal{S}_i^c((\mathcal{D}_i^c)' + \phi_i^c \mathcal{D}^c) &= \mathcal{S}_i^c(\mathcal{D}_i^c) + \gamma \mathcal{S}_i^c(\phi_i^c \mathcal{D}^c), \\ \implies \Delta \mathcal{S}_i^c &= \gamma \mathcal{S}_i^c(\phi_i^c \mathcal{D}^c). \end{aligned}$$

$\mathcal{S}_i^c(\phi_i^c \mathcal{D}^c)$ is the similarity score for fraction of server dataset $\mathcal{D}^c$. It implies $\mathcal{S}_i^c(\phi_i^c \mathcal{D}^c) = 1$ and we can obtain:

$$\Delta \mathcal{S}_i^c = \frac{|\phi_i^c \mathcal{D}^c|}{|(\mathcal{D}_i^c)'| + |\phi_i^c \mathcal{D}^c|}.$$

Using Equation 3.4 and replace $|(\mathcal{D}_i^c)'| + |\phi_i^c \mathcal{D}^c|$ with $|\mathcal{D}_i^c|$, we get:

$$\Delta\mathcal{S}_i^c = \frac{|\phi_i^c \mathcal{D}^c|}{|\mathcal{D}_i^c|} \implies \left(\frac{\Delta\mathcal{S}_i^c}{\phi_i^c}\right) = \frac{|\mathcal{D}^c|}{|\mathcal{D}_i^c|}. \tag{3.10}$$

Since $|\mathcal{D}^c|$ and $|\mathcal{D}_i^c|$ are constant values, which implies $\left(\frac{\Delta\mathcal{S}_i^c}{\phi_i^c}\right)$ is constant. □

Theorem 3.1 *For given centroids $\mathbf{C}_i^c$ and $\mathbf{C}^c$ for class c of participant p_i and server, respectively, where $1 \leq i \leq Q$ and $1 \leq c \leq k$. The optimal fraction $(\phi_i^c)^\star$ of dataset transmitted from server to participant p_i is given as:*

$$(\phi_i^c)^\star = \left(\epsilon_o^c - \frac{\sum_{p=1}^n C_{ip}^c . C_p^c}{\sqrt{\sum_{p=1}^n (C_{ip}^c)^2} . \sqrt{\sum_{p=1}^n (C_p^c)^2}}\right) . \frac{|\mathcal{D}_i^c|}{|\mathcal{D}^c|}, \tag{3.11}$$

where ϵ_o^c is the threshold of similarity score.

Proof: Let $\mathbf{C}_i^c$ and $\mathbf{C}^c$ are the centroids for class c of p_i and server on datasets $\mathcal{D}_i^c$ and $\mathcal{D}^c$, respectively. Using Equation 3.2, we can obtain $\mathcal{S}_i^c(\mathbf{C}_i^c, \mathbf{C}^c)$ between $\mathbf{C}_i^c$ and $\mathbf{C}^c$ as:

$$\mathcal{S}_i^c(\mathbf{C}_i^c, \mathbf{C}^c) = \frac{\sum_{p=1}^n C_{ip}^c . C_p^c}{\sqrt{\sum_{p=1}^n (C_{ip}^c)^2} . \sqrt{\sum_{p=1}^n (C_p^c)^2}}. \tag{3.12}$$

Let $(\phi_i^c)^\star$ is the optimal portion of the dataset transmitted from server to participant p_i to improve similarity score. Let $(\mathbf{C}_i^c)'$ is the centroid of the updated dataset at p_i. The similarity score $\mathcal{S}_i^c((\mathbf{C}_i^c)', \mathbf{C}^c)$ is estimated using Equation 3.2. From Constraint $\mathbf{c_1}$ of Equation 3.6b and using limiting condition of equality, we have:

$$\mathcal{S}_i^c((\mathbf{C}_i^c)', \mathbf{C}^c) = \epsilon_o^c. \tag{3.13}$$

Subtracting (3.12) from (3.13), we get the following equation:

$$\begin{aligned} &\mathcal{S}_i^c((\mathbf{C}_i^c)', \mathbf{C}^c) - \mathcal{S}_i^c(\mathbf{C}_i^c, \mathbf{C}^c) \\ &\qquad = \epsilon_o^c - \frac{\sum_{p=1}^n C_{ip}^c . C_p^c}{\sqrt{\sum_{p=1}^n (C_{ip}^c)^2} . \sqrt{\sum_{p=1}^n (C_p^c)^2}}, \\ \implies &\Delta\mathcal{S}_i^c = \epsilon_o^c - \frac{\sum_{p=1}^n C_{ip}^c . C_p^c}{\sqrt{\sum_{p=1}^n (C_{ip}^c)^2} . \sqrt{\sum_{p=1}^n (C_p^c)^2}}. \end{aligned} \tag{3.14}$$

Using Lemma 3.1, we have $\Delta\mathcal{S}_i^c = (\phi_i^c)^\star . \frac{|\mathcal{D}^c|}{|\mathcal{D}_i^c|}$. Substituting value of $\Delta\mathcal{S}_i^c$ in Equation 3.14, we get the following expression:

$$(\phi_i^c)^\star = \Big(\epsilon_o^c - \frac{\sum_{p=1}^{n} C_{ip}^c . C_p^c}{\sqrt{\sum_{p=1}^{n} (C_{ip}^c)^2} \cdot \sqrt{\sum_{p=1}^{n} (C_p^c)^2}}\Big) . \frac{|\mathcal{D}_i^c|}{|\mathcal{D}^c|}.$$

Hence proved. □

3.4 Scalability for larger network configurations.

The proposed solutions are inherently scalable due to their decentralized nature, as federated learning enables training across multiple devices without requiring centralized data aggregation. This approach mitigates bottlenecks in centralized processing and data storage, facilitating scalability to larger network configurations. To address challenges in dense networks, such as increased computational and communication demands and data quality management, the proposed solutions incorporate the following strategies:

1. **Efficient Communication Networks:** Utilizing LoRaWAN, known for its low cost, long-range, and scalability, ensures effective communication in IoT applications, even within highly dense networks. For future enhancements, integration with 5G can further improve scalability by reducing latency and increasing bandwidth.
2. **Optimization Techniques:** Dynamic annotation requests and efficient dataset fraction calculation optimize resource usage and data distribution, ensuring efficient scaling without compromising performance.
3. **Distributed Computation:** Methods like local centroid calculation and anomaly detection distribute the computational load, reducing the strain on individual nodes and enhancing scalability.

With these strategies, the proposed solutions can effectively scale to larger network configurations while maintaining performance, data privacy, and cost-efficiency in dense IoT environments.

3.5 Adaptability to other communication protocols and network configurations

The proposed solutions demonstrate adaptability to various communication protocols and network configurations, as outlined below:

1. **Cellular Networks (e.g., 4G, 5G):**

- Provide high bandwidth and low latency, making them suitable for mobile applications with high mobility requirements.
- Integration with the proposed system ensures adaptability in scenarios requiring real-time data transmission.
- However, these networks have higher energy consumption than low-power protocols like LoRa.

2. **Wi-Fi (wireless fidelity):**
   - Suitable for high-speed data transfer within a limited range (typically within 100 meters).
   - Lower latency than cellular networks but higher energy consumption, making it ideal for indoor IoT applications or stationary setups.
3. **Bluetooth and Bluetooth Low Energy (BLE):**
   - Designed for short-range communication, BLE is optimized for low data rates, making it well-suited for IoT devices with limited power and processing resources.
   - Its integration supports low-power applications such as wearable devices and health monitoring systems.
4. **Zigbee:**
   - Zigbee-based networks can utilize clustering techniques to aggregate sensor data, reducing the dataset size and transmission overhead (e.g., averaging temperature data from sensors in the same area).
   - Offers low power consumption, low data rates, and mesh networking capabilities, enabling communication over a larger range via intermediate nodes.
   - Ideal for home automation and sensor networks where energy efficiency and range are critical.
5. **Satellite Communication:**
   - Suitable for remote and wide-area coverage, such as rural, maritime, or isolated regions.
   - While high latency and low bandwidth are constraints, the protocol remains essential for scenarios with limited terrestrial network availability.

By exploring and integrating these communication protocols, the proposed solutions ensure adaptability to diverse network configurations, facilitating practical implementation in various IoT applications.

3.6 Performance evaluation

This section evaluates Fed-LoRa on Sussex-Huawei Locomotion (SHL) dataset [72], and the deep learning model discussed in [60] using different performance metrics. We first describe the dataset followed by performance metrics to ensure proper reproducibility. This section later demonstrates the results on different parameters discussed in Section 3.3.

3.6.1 Experimental setup

3.6.1.1 Existing Datasets

We use the publicly available SHL dataset given in the challenge of 2018 [72] during the experiment. The dataset comprises values of different smartphone sensors, including accelerometer, gyroscope, magnetometer, orientation, linear-acceleration, *etc.* These sensory values were labeled with different locomotion modes, including walk, run, car, bus, and so on. We consider the sensory instances of six locomotion modes during experimental evaluation, *i.e.,* walk, run, bike, car, bus, and train. We further consider the values of the accelerometer, gyroscope, and magnetometer. We have use python programming language in the results simulation phase. We have a tensorflow-federated library to produce results in this work. We have also collected data from our campus auditorium. Most of the presented results in the chapter follow a random distribution of the imperfect labels to create a resemblance with the real-world scenario. We denote classes in the SHL dataset as $\{\mathbf{a_1}, \cdots, \mathbf{a_6}\}$.

3.6.1.2 Metrics

This work uses standard classification metrics, including F_1 score, and accuracy, to measure the performance of the trained shared model using the Fed-LoRa approach. Let the SHL dataset incorporates a set of k classes. Let TP_j, TN_j, FP_j, and FN_j represent true positive, true negative, false positive, and false negative for class j, where $1 \leq j \leq k$. F_1 score and accuracy can be estimated using following expression: $F_1\ score = \frac{1}{|k|}\sum_{i=1}^{|k|}\frac{2\times TP_j}{2.TP_j+FP_j+FN_j}$ and $Accuracy = \frac{1}{|k|}\sum_{i=1}^{|k|}\frac{TP_j}{TP_j+TN_j+FP_j+FN_j}$, respectively. Further, we derive following two performance metrics using F_1 score and accuracy: *Local performance:* This metric determines the performance of participant p_i on its local dataset $\mathcal{D}_i$. *Global performance:* It estimates the performance (F_1 score or accuracy) of the participants on server dataset $\mathcal{D}$.

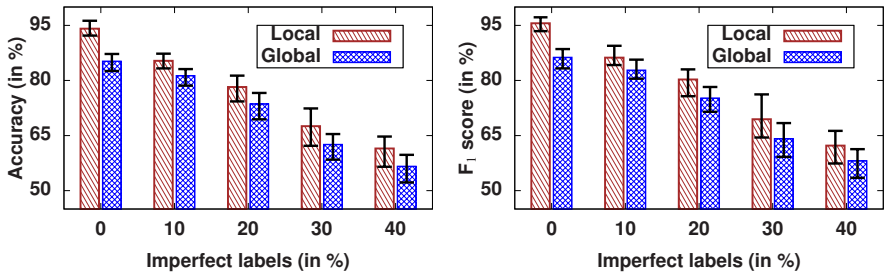
3.6.2 Evaluation results

This section describes the experimental evaluation to illustrate the impact of parameters discussed in Section 3.3, including imperfect labels, similarity threshold, *etc.*

3.6.2.1 Impact of imperfect labels

This experiment aims to demonstrate the impact of imperfect labels in following three aspects: a) accuracy and F_1 score, b) class-wise similarity score, and c) local epochs and global iterations.

a) **Accuracy and F_1 score:** Figure 3.6(a) illustrates the impact of increasing concentration of imperfect labels on local and global accuracy achieved by the trained model on the participants. It depicts the performance deterioration due to the increase in the imperfect labels, which is obvious. However, we observe a rapid decrement in accuracy when the concentration of imperfect labels reaches beyond 20%. In addition, when the concentration is 40%, the accuracy drops suddenly. It indicates that the data of certain classes have nearly all its labels incorrect at 40% concentration due to random insertion of imperfect labels. Thus, we restrict our experimental evaluation to a maximum 40% concentration. We can make a similar observation for the F_1 score, as shown in Figure 3.6(b). It illustrates the F1 score is higher than the accuracy on the SHL dataset. It is because the labels are not uniformly distributed among all instances in the SHL dataset.



(a) Accuracy. (b) F_1 score.

Figure 3.6: Impact of imperfect labels (in %) on local and global accuracy and F_1 score on the SHL dataset.

b) **Class-wise similarity score and performance:** The increasing concentration of imperfect labels decreases the similarity score and deteriorates the performance of all classes of the SHL dataset, as shown in Table 3.1. The results illustrate the impact of imperfect labels on class-wise accuracy, and the F_1 score forms a symmetrical pattern. The class labels $\mathbf{a_1} - \mathbf{a_3}$ suffer from significant performance

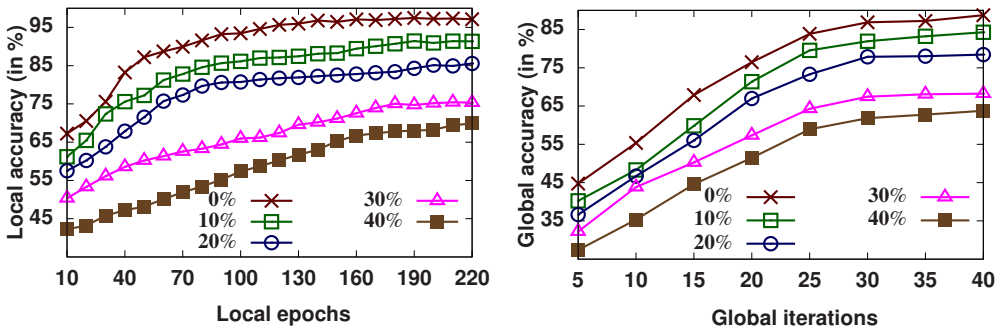
compromise, whereas the decrement in the performance of $\mathbf{a_4} - \mathbf{a_6}$ is graceful. It is because the instances with class labels $\mathbf{a_1} - \mathbf{a_3}$ are much lower in number than $\mathbf{a_4} - \mathbf{a_6}$. Next, the similarity scores without imperfect labels are more than 0.9 for all six classes, *i.e.,* $\mathbf{a_1} - \mathbf{a_6}$. It decreases gradually with the increase in the imperfect labels in the local dataset of the participants. The similarity score of class $\mathbf{a_4}$ shows a rapid decrement with imperfection, *i.e.,* 0.91 to 0.30, when imperfection increases from 0% to 40%. It is because the class labels $\mathbf{a_4}$ overlap with $\mathbf{a_3}$ and $\mathbf{a_5}$ at higher imperfect labels.

Table 3.1: Impact of imperfect labels (in %) on class-wise similarity score and performance (local accuracy and local F_1 score). Accuracy and F_1 score shows the variation of $\pm 2\%$ and similarity score varies by ± 0.03 in the demonstrated results.

Imperfect labels	**Metrics**	**Classes**					
		$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{a_5}$	$\mathbf{a_6}$
0%	Accuracy	81%	83%	66%	96%	92%	97%
	F_1 score	82%	85%	67%	97%	94%	97%
	Similarity	0.96	0.99	1.00	0.91	0.95	0.98
10%	Accuracy	51%	56%	38%	94%	93%	96%
	F_1 score	53%	57%	41%	95%	95%	97%
	Similarity	0.92	0.95	0.94	0.84	0.96	0.93
20%	Accuracy	37%	40%	27%	92%	89%	91%
	F_1 score	38%	42%	29%	93%	89%	92%
	Similarity	0.84	0.93	0.87	0.71	0.94	0.91
30%	Accuracy	21%	29%	17%	89%	82%	83%
	F_1 score	23%	31%	19%	90%	84%	84%
	Similarity	0.76	0.88	0.81	0.52	0.92	0.86
40%	Accuracy	16%	21%	12%	85%	78%	81%
	F_1 score	17%	24%	14%	87%	80%	82%
	Similarity	0.65	0.84	0.73	0.30	0.84	0.78

c) **Local epochs and global iterations:** Figure 3.7 illustrates the variations in local epochs and global iterations to train the local models on the participants using the SHL dataset with different concentrations of imperfections (0% to 40%). We can observe from the result that the imperfect labels hamper the training performance, where decrement is higher beyond 20% imperfections. The training accuracy curve starts flattening after 150 epochs, and we observe minimal/ no improvement in accuracy beyond 160 local epochs. We use $E = 160$ during the entire experimental analysis. Similarly, the model training converges after global iterations 25 and achieves global training accuracy of around 88%. We set $R = 25$ for all the experiments in this work. The convergence of the local model

on the SHL dataset with different concentrations (10% to 40%) of imperfect labels demands a higher number of epochs for convergence while achieving much lower accuracy.



(a) Local epochs. (b) Global iterations.

Figure 3.7: Impact of imperfect labels (in %) on local epochs and global iterations for training shared model in FL.

3.6.2.2 Impact of similarity threshold (ϵ_o^c)

The objective of this experiment is to study the impact of similarity threshold ϵ_o^c on accuracy and similarity score with different concentrations of imperfect labels, *i.e.,* 10%, 20%, 30% and 40%. In this experiment, we try to obtain the optimal value of ϵ_o^c for the given accuracy requirement in the Fed-LoRa approach.

a) **Similarity threshold (ϵ_o^c) *vs.* local accuracy:** Table 3.2 illustrates the impact of ϵ_o^c on the local accuracy of the participants with different concentrations of imperfect labels. We can observe from the result that even at $\epsilon_o^c = 0.4$, the accuracy on the dataset with 40% imperfect labels increases. It violates the Constraint $\mathbf{c_1}$ of the optimization problem given Equation 3.6, which compels to demand additional data from the server. The received fraction of the server dataset improves the accuracy. The violation of Constraint $\mathbf{c_1}$ can be easily observed in the last row of Table 3.1, where we can obtain the value of similarly-score for class $\mathbf{a_4} = 0.30 < \epsilon_o^c = 0.4$. Similarly, Table 3.2 demonstrates the improvement in accuracy on the datasets with 30%, 20%, and 10% imperfect labels start at $\epsilon_o^c = 0.6$, $\epsilon_o^c = 0.8$, and $\epsilon_o^c = 0.9$, respectively.

b) **Similarity threshold (ϵ_o^c) *vs.* delay:**
We estimate the delay ratio to analyze the impact of ϵ_o^c on the data transmission delay before FL operations in Fed-LoRa. Delay ratio $= \left(\frac{\text{delay with } x\% \text{ imperfect labels}}{\text{delay with no imperfect labels}} - 1\right)$, where $x \in \{10, 20, 30, 40\}$. Figure 3.8 illustrates the delay ratio increases

Table 3.2: Impact of ϵ_o^c on local accuracy ($\pm 2\%$) of participant on different concentration of imperfect labels in Fed-LoRa.

Similarity threshold (ϵ_o^c)	**Imperfect labels (in %)**			
	10%	20%	30%	40%
	Local Acc.	Local Acc.	Local Acc.	Local Acc.
$\epsilon_o^c = 0.3$	85.35%	78.19%	67.55%	61.47%
$\epsilon_o^c = 0.4$	85.35%	78.19%	67.55%	65.21%
$\epsilon_o^c = 0.5$	85.35%	78.19%	67.55%	69.73%
$\epsilon_o^c = 0.6$	85.35%	78.19%	74.41%	73.58%
$\epsilon_o^c = 0.7$	85.35%	78.19%	79.32%	76.28%
$\epsilon_o^c = 0.8$	85.35%	86.43%	83.32%	81.74%
$\epsilon_o^c = 0.9$	92.63%	89.37%	87.56%	85.37%

with increasing similarity score for different concentrations of imperfect labels ($10\% - 40\%$). The delay ratio remains steady upto $\epsilon_o^c = 0.3$ for 40% imperfect labels and increases thereafter because it demands a fraction of the server dataset to improve similarity score. Similarly, we obtain steady values for 30%, 20%, and 10% imperfect labels upto $\epsilon_o^c = 0.52$, $\epsilon_o^c = 0.71$, and $\epsilon_o^c = 0.84$, respectively.

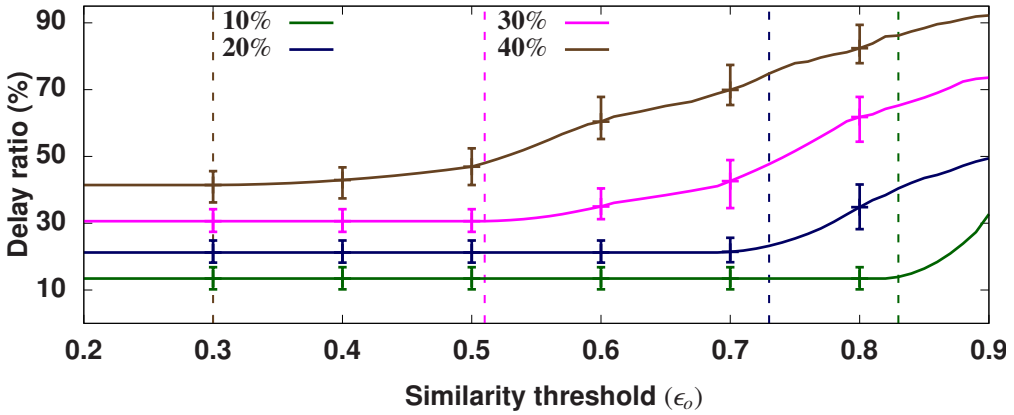


Figure 3.8: Impact of similarity threshold (ϵ_o^c) on data transmission delay prior to the FL operation in Fed-LoRa.

From the above results, we obtain that the delay and accuracy increase with the similarity score (ϵ_o^c) at different concentrations of imperfect labels. Thus, optimal selection of ϵ_o^c is crucial for achieving desired accuracy and comparable delay. This work obtains $\epsilon_o^c = 0.75$ as an optimal value that reaches an accuracy $\geq 80\%$ on all concentrations of imperfect labels with low delay.

c) **Optimal similarity threshold:** In this experiment, we study the impact of optimal $\epsilon_o^c = 0.75$ on class-wise accuracy and similarity score. We do not depict

the results for 10% imperfect labels as it holds a similarity score > 0.75 for all its classes. Thus, no improvement is observed at the optimal value of ϵ_o^c. Table 3.3 illustrates the classes having similarity score < 0.75, demand fraction of server dataset and improve their accuracy and similarity score (updated as 0.75). When we insert the fraction of the server dataset against one or more classes in the local dataset of the participant, it improves the performance and similarity score of the remaining classes. Because it reduces the instances that can be misclassified and provides better distinguishable characteristics to the built classifier of the local model at the participant.

Table 3.3: Impact of fixed ϵ_o^c on class-wise similarity and local accuracy on different concentrations of imperfection in Fed-LoRa.

Imperfect labels	**Metrics**	**Classes**					
		$\mathbf{a_1}$	$\mathbf{a_2}$	$\mathbf{a_3}$	$\mathbf{a_4}$	$\mathbf{a_5}$	$\mathbf{a_6}$
20%	Accuracy	41%	47%	49%	95%	92%	93%
	Similarity	0.88	0.95	0.91	0.75	0.95	0.93
30%	Accuracy	26%	41%	30%	90%	85%	87%
	Similarity	0.79	0.90	0.84	0.75	0.93	0.88
40%	Accuracy	23%	35%	27%	88%	84%	85%
	Similarity	0.75	0.86	0.75	0.75	0.85	0.81

3.7 Conclusions and future work

In this chapter, we presented a federated learning approach that can handle imperfect labels in the local dataset and provide energy-efficient Long-Range communication, namely Fed-LoRa. Apart from the existing work, we exploited the perfectly annotated server dataset to improve the performance of inferior participants with imperfect labels. We proposed an algorithm that summarized all the steps involved in Fed-LoRa. We also estimated an expression for the optimal fraction of the server dataset demanded by participants without increasing local epochs for training. Finally, the experimental results depicted the proposed approach effectively performed on a publicly available dataset of locomotion modes. In the future, we plan to develop low cost and low energy consuming prototypes for efficient locomotion mode recognition. We will also develop mechanisms to handle different issues, such as imperfect instances in the training data from ITS.