PRIVACY-PRESERVING DISTRIBUTED EXPECTATION MAXIMIZATION FOR GAUSSIAN MIXTURE MODEL USING SUBSPACE PERTURBATION

Qiongxiu Li[⋆] *Jaron Skovsted Gundersen*[†] *Katrine Tjell*[†]
Rafal Wisniewski[†] Mads Græsbøll Christensen[⋆]

[⋆] Audio Analysis Lab, CREATE, Aalborg University, Denmark, {qili, mgc}@create.aau.dk
[†] Department of Electronic Systems, Aalborg University, Denmark, {jaron, kst, raf}@es.aau.dk

ABSTRACT

Privacy has become a major concern in machine learning. In fact, the federated learning is motivated by the privacy concern as it does not allow to transmit the private data but only intermediate updates. However, federated learning does not always guarantee privacy-preservation as the intermediate updates may also reveal sensitive information. In this paper, we give an explicit information-theoretical analysis of a federated expectation maximization algorithm for Gaussian mixture model and prove that the intermediate updates can cause severe privacy leakage. To address the privacy issue, we propose a fully decentralized privacy-preserving solution, which is able to securely compute the updates in each maximization step. Additionally, we consider two different types of security attacks: the honest-but-curious and eavesdropping adversary models. Numerical validation shows that the proposed approach has superior performance compared to the existing approach in terms of both the accuracy and privacy level.

***Index Terms*—** Federated learning, differential privacy, secure multiparty computation, information-theoretic, privacy-accuracy.

1. INTRODUCTION

Distributed algorithms are widely used because of their advantageous distribution of computational burden, flexibility, and scalability of the system, and resilience because of the elimination of the potential single point of failure. Moreover, the increasing focus on individual's right to data privacy leads to even more motivation for the distributed setup. An evident example is when data is collected by multiple nodes/parties/agents and computations involve the combined dataset (to yield more accurate results). Consider for instance hospitals across cities and countries collecting data on diseases. Because of strict data protection laws, it is non-trivial for hospitals to share data with other hospitals especially if data is to travel across borders. However, performing statistical analysis on the combined dataset from all hospitals would in many cases yield more accurate results.

In this paper, we seek to circumvent these privacy issues that occur when multiple data owners engage in joint computations. More specifically, we aim to fit a Gaussian mixture model (GMM) based on a dataset that is distributed among a set of nodes. Hence, the nodes want to fit a GMM for the combined dataset (since using as much data as possible increases the accuracy of the model) without having to disclose their individual datasets. For training the model we consider the Expectation Maximization (EM) algorithm [1] because of its wide applicability. Thus, there are a substantial amount of methods that are based on the EM algorithm (see for instance [2–4]) where most would immediately be privacy-preserving by using a privacy-preserving distributed EM algorithm.

The concept of *federated learning* [5] is widely used as a method to train a global model over multiple nodes without sharing each node's local data directly. It does so by each node training a local model and subsequently all local models are combined to a global model. In [6] it is shown how an adversary could attack this approach to learn private data, meaning that even though the nodes does not share directly their local data, information about this private data is leaked anyway. Therefore, we follow up on this result and use an information theoretic approach to prove that a federated version of the EM algorithm is indeed not privacy-preserving and subsequently we propose a solution to circumvent the situation.

There is a substantial amount of work dealing with privacy in machine learning methods, see for instance [7–14]. Of particular interest to our work is [15–21] that also considers privacy of distributed EM. Among them, the privacy of the proposed solution in [21] is based on two party cryptographic computations that tend to be computationally heavy and time consuming. The remaining approaches can be broadly classified into three types. (1) Homomorphic encryption based approaches: the proposed solution in [16–18] is based on homomorphic encryption which is known to be very computationally demanding and consequently time consuming. A distributed approach was proposed in [16] to reduce the overhead in computations. However, there is still a need for a lightweight solution. (2) Differentially private approaches: the works [19, 20] propose differential private EM algorithms, however these approaches are not proposed in a distributed/decentralized setting. Additionally, there is an inevitable trade-off between privacy and accuracy of the output in differentially private approaches. (3) Secure summation approach: the main idea of the work [15] is to apply a secure summation protocol in each M step for protecting privacy. The drawback is that it requires to first detect a so-called Hamiltonian cycle. In addition, this algorithm is very vulnerable to security attacks since it depends on a single node to keep the encryption seed.

In order to address the limitations of the above-existing approaches, in this paper, we propose a new privacy-preserving distributed EM algorithm for GMM using subspace perturbation [22, 23], a privacy-preserving technique based on distributed convex optimization. By inspecting the updating functions of EM algorithm, we observe that the M step is the cause of privacy concern as it requires information exchange between different nodes. To address it, we first formulate the required updates in M step as a distributed convex optimization problem and then exploit the subspace perturbation technique to securely compute these updates. We conduct numerical experiments to validate that the proposed approach achieves a higher privacy level compared to existing approaches without compromising the estimation accuracy.

2. PRELIMINARIES AND PROBLEM SETUP

The following notation is used in this paper. Lowercase letters (x) denote scalars, lowercase boldface letters ($\boldsymbol{x}$) vectors, uppercase boldface letters ($\boldsymbol{X}$) matrices, and calligraphic letters ($\mathcal{X}$) sets. $\boldsymbol{x}_i$ denotes the i-th entry of the vector $\boldsymbol{x}$, and $\boldsymbol{X}_{ij}$ denotes the (i,j)-th entry of the matrix $\boldsymbol{X}$. We use an uppercase overlined letter ($\bar{X}$) to denote a random variable. With a slight abuse of notation we use ($\bar{X}$) for all cases when the realization is a scalar (x), vector ($\boldsymbol{x}$), or matrix ($\boldsymbol{X}$) (can be distinguished from the context).

2.1. Fundamentals of GMM and EM over networks

A network can be modelled as a graph (undirected) $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{1, \ldots, n\}$ denotes the set of n nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of m edges. Note that node i and j can communicate with each other only if there is an edge connecting them, i.e., $(i,j) \in \mathcal{E}$. Denote $\mathcal{V}_i = \{j | (i,j) \in \mathcal{E}\}$ as the set of neighboring nodes of node i. We now introduce fundamentals about the GMM. Consider the scenario where n nodes each has its own data $\boldsymbol{x}_i$ and these nodes would like to collaborate to learn a GMM based on the full dataset $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. We start by introducing the GMM and explain how to fit it to the given dataset. Assume there are in total c Gaussian models and denote $\mathcal{C} = \{1, \ldots, c\}$. Specifically, the GMM density is given by

$$p(\boldsymbol{x}) = \sum_{j \in \mathcal{C}} \beta_j p(\boldsymbol{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j), \tag{1}$$

where $p(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ is the pdf for a Gaussian distribution with mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$ and the β_j's are so-called mixture coefficients. The task is then to estimate β_j, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ for all $j \in \mathcal{C}$, which can be done using the EM algorithm and the available data. The EM algorithm is iterative and for each iteration $t \in \mathcal{T}$ where $\mathcal{T} = \{0, 1, \ldots, T\}$, the following steps are taken for all $j \in \mathcal{C}$:

E-step:

$$P(\boldsymbol{x}_i | \mathcal{N}_j^t) = \frac{p(\boldsymbol{x}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \beta_j^t}{\sum_{k=1}^c p(\boldsymbol{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \beta_k^t}, \tag{2}$$

M-step:

$$\begin{aligned} \beta_j^{t+1} &= \frac{\sum_{i=1}^n P(\boldsymbol{x}_i | \mathcal{N}_j^t)}{n} \\ \boldsymbol{\mu}_j^{t+1} &= \frac{\sum_{i=1}^n P(\boldsymbol{x}_i | \mathcal{N}_j^t) \boldsymbol{x}_i}{\sum_{i=1}^n P(\boldsymbol{x}_i | \mathcal{N}_j^t)} \\ \boldsymbol{\Sigma}_j^{t+1} &= \frac{\sum_{i=1}^n P(\boldsymbol{x}_i | \mathcal{N}_j^t) (\boldsymbol{x}_i - \boldsymbol{\mu}_j^t)(\boldsymbol{x}_i - \boldsymbol{\mu}_j^t)^\top}{\sum_{i=1}^n P(\boldsymbol{x}_i | \mathcal{N}_j^t)}, \end{aligned} \tag{3}$$

where $P(\boldsymbol{x}_i | \mathcal{N}_j^t)$ denotes the conditional probability that data $\boldsymbol{x}_i$ belongs to Gaussian model j. In order to compute the updates in (3) in a distributed manner, data aggregation over the network is required. One straightforward way is for each node to publish its individual dataset to all neighboring nodes. However, directly publishing/sharing the individual data would violate the privacy.

2.2. Definition of private data and adversary models

We define the private data to be the individual data held by each node as it often contains sensitive information. For example, a person's health condition, like whether he/she has Parkinson's disease or not, can be revealed by voice data [24]. Hence, each node i would like to prevent that information regarding its own data $\boldsymbol{x}_i$ is revealed to other nodes during the computation.

We also need to define what the private data should be protected against. We consider two well-known adversary models here: the eavesdropping adversary and the passive (also called honest-but-curious) adversary. The eavesdropping adversary works by listening to the messages transmitted along the communication channels. The passive adversary model controls a number of so-called corrupt nodes who are assumed to follow the algorithm instructions but can collect information together. It will then use the collected information to infer the private data of the non-corrupt nodes, which we will refer to as honest nodes.

2.3. Privacy-preserving distributed EM algorithm

Putting things together, a privacy-preserving distributed EM algorithm for GMM should satisfy the following two requirements:

- Individual privacy: throughout the algorithm execution, each honest node's private data should be protected from being revealed to the adversaries. We quantify the individual privacy using mutual information $I(\bar{X}; \bar{Y})$, which measures the mutual dependence between two random variables $\bar{X}, \bar{Y}$. We have that $0 \leq I(\bar{X}; \bar{Y})$, with equality if and only if $\bar{X}$ is independent of $\bar{Y}$. On the other hand, if $\bar{Y}$ uniquely determines $\bar{X}$ we have $I(\bar{X}; \bar{Y}) = I(\bar{X}; \bar{X})$ which is maximal.
- Output correctness: the fitted model, i.e., the estimated parameters of GMM, should be the same as using non-privacy-preserving counterparts. Namely, the performance of the GMM should not be compromised by considering privacy.

3. FEDERATED EM AND ITS PRIVACY LEAKAGE

In what follows, we will use EM algorithm for GMM as an example to show that federated learning does not always guarantee privacy.

3.1. Federated EM algorithm for GMM

Federated learning aims to learn the model under the constraint that the data is stored and processed locally, with only intermediate updates being communicated periodically with a central server. Therefore, in the context of EM algorithm, instead of directly sharing the data $\boldsymbol{x}_i$, each node i shares the following intermediate updates only:

$$\begin{aligned} a_{ij}^t &= P(\boldsymbol{x}_i | \mathcal{N}_j^t) \\ \boldsymbol{b}_{ij}^t &= P(\boldsymbol{x}_i | \mathcal{N}_j^t) \boldsymbol{x}_i \\ \boldsymbol{C}_{ij}^t &= P(\boldsymbol{x}_i | \mathcal{N}_j^t)(\boldsymbol{x}_i - \boldsymbol{\mu}_j^t)(\boldsymbol{x}_i - \boldsymbol{\mu}_j^t)^\top \end{aligned} \tag{4}$$

Hence, all the above updates can also be computed locally at node i. After receiving these intermediate updates from all nodes, the server will then aggregate all local updates and compute the global update $\beta_j^{t+1}, \boldsymbol{\mu}_j^{t+1}, \Sigma_j^{t+1}$ required in the M-step through the following way:

$$\begin{aligned} \beta_j^{t+1} &= \frac{\sum_{i=1}^n a_{ij}^t}{n} = \frac{a_j^t}{n} \\ \boldsymbol{\mu}_j^{t+1} &= \frac{\sum_{i=1}^n \boldsymbol{b}_{ij}^t}{\sum_{i=1}^n a_{ij}^t} = \frac{\boldsymbol{b}_j^t}{a_j^t} \\ \boldsymbol{\Sigma}_j^{t+1} &= \frac{\sum_{i=1}^n \boldsymbol{C}_{ij}^t}{\sum_{i=1}^n a_{ij}^t} = \frac{\boldsymbol{C}_j^t}{a_j^t}. \end{aligned} \tag{5}$$

Thereafter the server sends the global update $\beta_j^{t+1}, \boldsymbol{\mu}_j^{t+1}, \boldsymbol{\Sigma}_j^{t+1}$ back to each and every node.

3.2. Privacy leakage in Federated EM algorithm

We note that with the above federate EM algorithm, even though each node does not share the private date directly, it is still revealed to the server. This is because with the intermediate updates a_{ij}^t and $\boldsymbol{b}_{ij}^t$, the server is able to determine the private data $\boldsymbol{x}_i$ of each node i since $\boldsymbol{b}_{ij}^t = a_{ij}^t \boldsymbol{x}_i$. That is, at each iteration the server has the following mutual information

$$I(\bar{X}_i; \bar{A}_{ij}^t, \bar{B}_{ij}^t) = I(\bar{X}_i, \bar{X}_i), \tag{6}$$

which is maximal. This means that every node's private data $\boldsymbol{x}_i$ is completely revealed to the server. Hence, in this context federated EM algorithm is not privacy-preserving at all.

4. PROPOSED APPROACH

We now proceed to introduce the proposed approach which addresses the privacy issue raised in federated EM algorithm. As shown in the previous section, the exchange of intermediate updates $a_{ij}^t, \boldsymbol{b}_{ij}^t, \boldsymbol{C}_{ij}^t$ will reveal all private date of each node. We observe that it is sufficient to use the average updates $\frac{1}{n}a_j^t$, $\frac{1}{n}\boldsymbol{b}_j^t$, and $\frac{1}{n}\boldsymbol{C}_j^t$, to compute all global updates $\beta_j^{t+1}, \boldsymbol{\mu}_j^{t+1}, \boldsymbol{\Sigma}_j^{t+1}$ in (5). Therefore, we propose to apply the subspace perturbation technique [22, 23] to securely compute these average updates without revealing each node's intermediate updates. In what follows we will first introduce fundamentals of subspace perturbation and then explain details of the proposed approach.

4.1. Problem formulation using distributed convex optimization

Assume n nodes each has private data $\boldsymbol{s}_i$ and let $\boldsymbol{y}_i$ be the so-called optimization variable, and stacking them together we have $\boldsymbol{s}, \boldsymbol{y} \in \mathbb{R}^n$. The average consensus can be formulated as a distributed convex optimization problem given by

$$\begin{aligned} \min_{\boldsymbol{y}_i} \quad & \sum_{i\in\mathcal{V}} \frac{1}{2}\|\boldsymbol{y}_i - \boldsymbol{s}_i\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{y}_i = \boldsymbol{y}_j, \forall (i,j) \in \mathcal{E}, \end{aligned} \tag{7}$$

where the optimum solution is $\forall i \in \mathcal{V} : \boldsymbol{y}_i^* = n^{-1}\sum_{i\in\mathcal{V}} \boldsymbol{s}_i$. With PDMM [25], for each $e_l = (i,j) \in \mathcal{E}$ it defines two dual variables: $\boldsymbol{\lambda}_l = \boldsymbol{\lambda}_{i|j}, \boldsymbol{\lambda}_{l+m} = \boldsymbol{\lambda}_{j|i}$. The local updating functions are given by

$$\boldsymbol{y}_i^{(t+1)} = \frac{\boldsymbol{s}_i + \sum_{j\in N_i}\left(c\boldsymbol{y}_j^{(t)} - \boldsymbol{B}_{i|j}\boldsymbol{\lambda}_{j|i}^{(t)}\right)}{1 + cd_i}, \tag{8}$$

$$\forall j \in \mathcal{V}_i : \boldsymbol{\lambda}_{i|j}^{(t+1)} = \lambda_{j|i}^{(t)} + c\boldsymbol{B}_{i|j}\left(\boldsymbol{y}_i^{(t+1)} - \boldsymbol{y}_j^{(t)}\right), \tag{9}$$

where c is a constant for controlling the convergence rate, $d_i = |\mathcal{V}_i|$ is the number of neighboring nodes of node i. The matrix $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ is related to the graph incidence matrix: $\boldsymbol{B}_{li} = \boldsymbol{B}_{i|j}$, $\boldsymbol{B}_{lj} = \boldsymbol{B}_{j|i}$ where $\boldsymbol{B}_{i|j} = 1$ if $i > j$ and $\boldsymbol{B}_{i|j} = -1$ if $i < j$.

4.2. Subspace perturbation and the proposed approach

It is shown that the dual variable $\boldsymbol{\lambda}^{(t)}$ will only converge in a subspace, denoted as H, determined by the graph incidence matrix. Denote Π_H as the orthogonal projection into H. We have $\boldsymbol{\lambda}^{(t)} = (\Pi_H)\boldsymbol{\lambda}^{(t)} + (\boldsymbol{I} - \Pi_H)\boldsymbol{\lambda}^{(t)}$ where the former will converge to a fixed point while the latter will not. The main idea of subspace perturbation is to exploit the non-convergent component of dual variable, i.e., $(\boldsymbol{I} - \Pi_{H)}\boldsymbol{\lambda}^{(t)}$ as subspace noise for guaranteeing the privacy. This can be achieved by initializing $\boldsymbol{\lambda}^{(0)}$ with sufficiently large variance for protecting the private data $\boldsymbol{s}_i$ from being revealed to others. Additionally, the accuracy is guaranteed as the subspace noise has no effects on the convergence of the optimization variable. Details of privacy-preserving distributed average consensus using subspace perturbation is summarized in Algorithm 1. As mentioned before, the main idea of the proposed approach is to use subspace perturbation to securely compute the average results $\frac{1}{n}a_j^t$, $\frac{1}{n}\boldsymbol{b}_j^t$, and $\frac{1}{n}\boldsymbol{C}_j^t$ at each M step. Overall, we summarize details of the proposed approach in Algorithm 2.

4.2.1. Performance analysis

Since the applied subspace perturbation output accurate average result as the non-privacy-preserving counterparts, it follows that the proposed approach also satisfies the output correctness requirement as the updates computed in M step is kept accurate.

As for the individual privacy requirement, we will discuss it based on the adversary types. When dealing with the passive adversary, the subspace perturbation approach guarantees the privacy of each honest node as long as it has one honest neighboring node, i.e., $\mathcal{V}_i \cap \mathcal{V}_h \neq \emptyset$ where $\mathcal{V}_h$ denotes the set of honest nodes. In this situation, it is shown that the subspace perturbation protocol only reveals the sum of the honest nodes' inputs, assuming the honest nodes are connected after removing all corrupted nodes (see Proposition 3 in [26]). That is, for each honest node $i \in \mathcal{V}_h$ its individual privacy at each iteration t is given by

$$I(\bar{X}_i; \{\sum_{j\in\mathcal{V}_h} \bar{A}_{jk}^t, \sum_{j\in\mathcal{V}_h} \bar{B}_{jk}^t, \sum_{j\in\mathcal{V}_h} \bar{C}_{jk}^t\}_{k\in\mathcal{C}}), \tag{10}$$

which is a significant improvement compared to the (6) of the federated EM approach. As for the eavesdropping adversary, it requires no channel encryption except for the initialization step for transmitting $\boldsymbol{\lambda}^{(0)}$.

Algorithm 1 Privacy-preserving distributed average consensus using subspace perturbation [23]

1: Every node $i \in \mathcal{V}$ initializes $\boldsymbol{y}_i^{(0)}$ arbitrarily and $\{\boldsymbol{\lambda}_{j|i}^{(0)}\}_{j\in\mathcal{V}_i}$ based on the desired privacy level.
2: Node i sends $\boldsymbol{\lambda}_{i|j}^{(0)}$ to its neighbor $j \in \mathcal{V}_i$ through securely encrypted channels [27].
3: **while** $\|\boldsymbol{y}^{(t)} - \boldsymbol{y}^*\|_2 <$ threshold **do**
4: Randomly activate a node with uniform probability, say node i, updates $\boldsymbol{y}_i^{(t+1)}$ using (8).
5: Node i broadcasts $\boldsymbol{y}_i^{(t+1)}$ to its neighbors $j \in \mathcal{V}_i$.
6: All neighboring nodes $j \in \mathcal{V}_i$ update $\boldsymbol{\lambda}_{j|i}^{(t+1)}$ using (9).
7: **end while**

Algorithm 2 Proposed privacy-preserving distributed EM algorithm for GMM using subspace perturbation

1: Initialize $\{\beta_j^0, \boldsymbol{\mu}_j^0, \boldsymbol{\Sigma}_j^0\}_{j\in\mathcal{C}}$
2: **while** iteration $t \in 0, 1, \ldots, T$ **do**
3: Each node i first computes (2) and then updates $a_{ij}^t, \boldsymbol{b}_{ij}^t, \boldsymbol{C}_{ij}^t$ using (4).
4: Apply Algorithm 1 to securely compute the average results $\frac{1}{n}a_j^t$, $\frac{1}{n}\boldsymbol{b}_j^t$ and $\frac{1}{n}\boldsymbol{C}_j^t$ over the whole network.
5: Each node i updates $\beta_j^{t+1}, \boldsymbol{\mu}_j^{t+1}, \boldsymbol{\Sigma}_j^{t+1}$ using (5).
6: **end while**

4.2.2. Comparison with the existing approach

We remark that, among the existing approaches introduced in the introduction, the secure summation approach [15] is particularly comparable to our proposed solution. Because it is also a fully decentralized protocol that aims to achieve privacy without compromising the accuracy of the output (unlike the differentially private approaches). In addition, the privacy-preserving tool is not based on computationally complex functions such as homomorphic encryption. Its main idea is to apply a secure summation protocol at each M step to first compute the sums, i.e., $\forall j \in \mathcal{C} : \sum_{i=1}^n a_{ij}^t, \sum_{i=1}^n \boldsymbol{b}_{ij}^t, \sum_{i=1}^n \boldsymbol{C}_{ij}^t$ and then updates $\beta_j^{t+1}, \boldsymbol{\mu}_j^{t+1}, \boldsymbol{\Sigma}_j^{t+1}$ using (5). However, we remark the proposed approach achieves a higher privacy level than [15] when considering the passive adversary.

To exemplify this claim, we consider the graph in Fig. 1. The secure summation protocol works by first detecting a Hamiltonian cycle in the graph, which is a path in the graph visiting all the nodes once, and where the start and end node is the same. Take the process of computing the sum $\sum_{i=1}^n a_{ij}^t$ as an example, node 1 chooses a random number r and then sends $a_{1j}^t + r$ to node 2. Node 2 adds a_{2j}^t

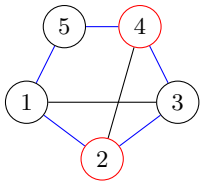

Fig. 1: Example with cycle. The blue edges form a Hamiltonian cycle. The red nodes are corrupt nodes.

and send this to node 3, repeat until node 5 sends $\sum_{i=1}^{5} a_{ij}^t + r$ back to node 1 which subtracts r and broadcast $\sum_{i=1}^{5} a_{ij}^t$ to all nodes. Since the passive adversary is able to collect information from corrupted node 2 and 4, it has the following information denoted as

$$\mathcal{V}_{aj}^t = \{\bar{A}_{1j}^t + \bar{R}, \bar{A}_{2j}^t, \bar{A}_{1j}^t + \bar{A}_{2j}^t + \bar{A}_{3j}^t + \bar{R}, \bar{A}_{4j}^t, \sum_{i=1}^{5} A_{ij}^t\}.$$

Remark that $\bar{A}_{3j}^t = (\bar{A}_{1j}^t + \bar{A}_{2j}^t + \bar{A}_{3j}^t + \bar{R}) - (\bar{A}_{1j}^t + \bar{R}) - \bar{A}_{2j}^t$ Hence $\bar{A}_{3j}^t$ can be uniquely determined using the information in $\mathcal{V}_{aj}^t$.

Similarly, the adversary can collect information $\mathcal{V}_{bj}^t$ and $\mathcal{V}_{cj}^t$ when computing $\sum_{i=1}^{n} \boldsymbol{b}_{ij}^t$ and $\sum_{i=1}^{n} \boldsymbol{C}_{ij}^t$, respectively. Overall, we conclude that at each iteration all information collected by the passive adversary is given by $\mathcal{V}^t = \{\mathcal{V}_{aj}^t, \mathcal{V}_{bj}^t, \mathcal{V}_{cj}^t\}_{j\in\mathcal{C}}$. Therefore, for honest node 3, its individual privacy is given by

$$I(\bar{X}_3; \{\bar{A}_{3k}^t, \bar{B}_{3k}^t, \bar{C}_{1k}^t\}_{k\in\mathcal{C}}) = I(\bar{X}_3; \bar{X}_3) \tag{11}$$

As for honest nodes $i = 1, 5$, we have

$$I(\bar{X}_i; \mathcal{V}^t) \geq I(\bar{X}_i; \{\sum_{j=1,5} \bar{A}_{jk}^t, \sum_{j=1,5} \bar{B}_{jk}^t, \sum_{j=1,5} \bar{C}_{jk}^t\}_{k\in\mathcal{C}}). \tag{12}$$

As for the proposed approach, inserts $\mathcal{V}_h = \{1, 3, 5\}$ in (10) would yield the individual privacy of honest node $1, 3, 5$, which is less than (11), (12), thereby proving our claim that the proposed approach achieves a higher privacy level than the existing approach [15].

5. NUMERICAL RESULTS

In this section, we demonstrate numerical results to validate the comparisons shown in the above section.

5.1. Output correctness

We first simulated a geometrical graph with $n = 80$ by allowing every two nodes to communicate if and only if their distance is within a radius of $\sqrt{2\log n/n}$, thereby ensuring a connected graph at a high probability $1 - 1/n^2$ [28]. We use the Parkinsons dataset [29] from the UCI repository [30]. This dataset contains voice measurements from 31 people and 23 of them are with Parkinson's disease. There are 195 instances in total and each has 22 features. The reason for choosing this dataset is that the involved biomedical voice measurements are highly sensitive information. In the experiment, to reduce the dimensionality of the features we apply principle component analysis to the dataset and choose the first two principle components for GMM fitting. As shown in Fig. 2, we see that the proposed approach estimates the parameters for GMMs identically to the existing approach [15] which has perfect output correctness. Hence, the output correctness of the proposed approach is guaranteed.

5.2. Individual privacy

Previously, we have proved that the federated EM algorithm is not privacy-preserving and the privacy level of the proposed approach is higher than the existing approach [15] considering the passive adversary. To validate these results, we use the sample case shown in Fig. 1. More specifically, we would like to show the individual privacy of honest node 1 given corruptions in the neighborhood using these three algorithms. In order to demonstrate an accurate mutual information estimation, it requires to gather a massive amount of data for each node as statistical analysis needs to be performed. As a consequence, the above deployed Parkinson's dataset is too small to conduct mutual information estimation. To address it, we first generate synthetic data by assuming all the private data $\bar{X}_1, \ldots, \bar{X}_5$ are Gaussian distributed with zero mean and unit variance. Additionally, all $\bar{A}_{ij}$'s are assumed uniformly distributed and their sum is normalized to one. After that, we perform 10^4 times Monte Carlo simulations and then deploy npeet [31] to estimate the normalized mutual information of honest node 1. The results are shown in Fig. 3. We can see that as expected, federated EM algorithm reveals all private information and among all approaches the proposed one reveals the minimum amount of information.

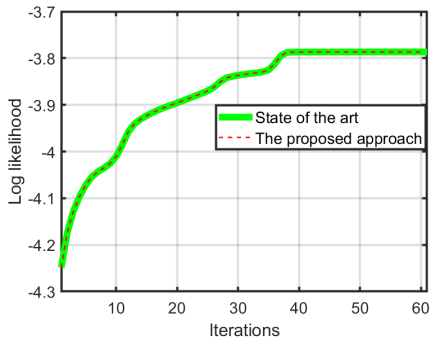

Fig. 2: Log likelihood as a function of the iteration number of the proposed approach and the existing approach.

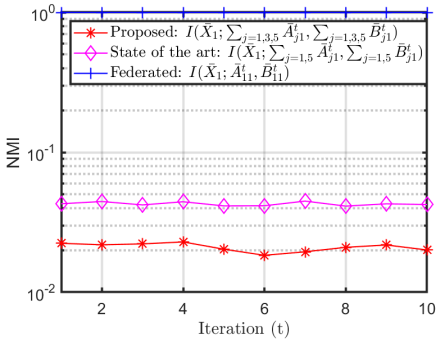

Fig. 3: Normalized mutual information (NMI), i.e., individual privacy of honest node 1 in terms of iteration number using the Federated, the existing and the proposed approach.

6. CONCLUSION

In this paper, we consider the problem of privacy-preserving distributed EM for GMM. We first gave explicit information-theoretical privacy analysis to prove that a federated EM algorithm does not guarantee privacy. After that, we proposed a lightweight privacy-preserving distributed EM algorithm for GMM which has no privacy-accuracy trade-off. Moreover, it offers stronger privacy guarantee when dealing with a number of corrupt nodes than the state-of-the-art algorithm. Numerical simulations were conducted to validate the above claims and demonstrate the superior performances of the proposed approach.

7. REFERENCES

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[2] T. Mensink, W. Zajdel, and B. Kröse, "Distributed EM Learning for Appearance Based Multi-Camera Tracking," in *IEEE/ACM Int. Conf. on Dist. Smart Cameras (ICDSC '07)*, pp. 178–185, 2007.

[3] R. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," *IEEE Trans. Signal Process.,*, vol. 51, pp. 2245 – 2253, 2003.

[4] K. Bhaduri and A. N. Srivastava, "A local scalable distributed expectation maximization algorithm for large peer-to-peer networks," in *IEEE Int. Conf. Data Mining*, pp. 31–40, 2009.

[5] J. Xu, B.S. Glicksberg, C. Su, and et al., "Federated learning for healthcare informatics," *J Healthc Inform Res*, 2020.

[6] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proc. ACM CCS, pp. 603–618, 2017*.

[7] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security Privacy*, vol. 17, no. 2, pp. 49–58, 2019.

[8] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Conf. Comm., Contr., and Computing (Allerton)*, pp. 909–910, 2015.

[9] S. Zapechnikov, "Privacy-preserving machine learning as a tool for secure personalized information services," *Procedia Computer Science*, vol. 169, pp. 393 – 399, 2020.

[10] E. Hesamifard, H. Takabi, M. Ghasemi, and C. Jones, "Privacy-preserving machine learning in cloud," in *Proc. ACM CCS*, New York, NY, USA, CCSW '17, pp. 39–43, 2017, Association for Computing Machinery.

[11] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *IEEE SP*, pp. 19–38, 2017.

[12] S. Jha, L. Kruger, and P. McDaniel, "Privacy preserving clustering," in *Springer ESORICS*, 09, vol. 3679, pp. 397–417, 2005.

[13] D. Ramage S. Hampson H. B. McMahan, E. Moore and B. Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics*, pp. 1273–1282,2017.

[14] M. Abadi et al., "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, pp. 308–318, 2016.

[15] X. Lin, C. Clifton, and M. Zhu, "Privacy-preserving clustering with distributed EM mixture modeling," *Springer KAIS*, vol. 8, no. 1, pp. 68–81, 2005.

[16] A. Alabdulatif, I. Khalil, A. Y. Zomaya, Z. Tari, and X. Yi, "Fully homomorphic based privacy-preserving distributed expectation maximization on cloud," *IEEE TPDS*, vol. 31, no. 11, pp. 2668–2681, 2020.

[17] B. Yang, I. Sato, and H. Nakagawa, "Privacy-preserving EM algorithm for clustering on social network," in *Springer AKDD*, pp. 542–553, 2012.

[18] S. X. Lee, K. L. Leemaqz, and G. J. McLachlan, "Ppem: Privacy-preserving EM learning for mixture models," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 24, 2019.

[19] M. Park, J. Foulds, K. Choudhary, and M. Welling, "DP-EM: Differentially Private Expectation Maximization," in *Proceedings of the 20th AISTATS*, vol. 54, pp. 896–904, 2017.

[20] D. Wang, J. Ding, Z. Xie, M. Pan, and J. Xu, "Differentially private (gradient) expectation maximization algorithm with statistical guarantees," *arXiv preprint arXiv:2104.00245*, 2021.

[21] T. D. Luong and T. B. Ho, "Privacy preserving em-based clustering," in *2009 IEEE-RIVF Int. Conf. Computing and Communication Technologies*, pp. 1–7, 2019.

[22] Q. Li, R. Heusdens and M. G. Christensen, "Convex optimisation-based privacy-preserving distributed average consensus in wireless sensor networks," in *Proc. Int. Conf. Acoust., Speech, Signal Process., pp. 5895-5899*, 2020.

[23] Q. Li, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation: A general framework," in *IEEE Trans. Signal Process., vol. 68, pp. 5983 - 5996, 2020*.

[24] A. H. Poorjam, Y. P. Raykov, R. Badawy, J. R. Jensen, M. G. Christensen and M.A. Little, "Quality control of voice recordings in remote parkinson's disease monitoring using the infinite hidden markov model," in *Proc. Int. Conf. Acoust., Speech, Signal Process., pp. 805-809*, 2019.

[25] G. Zhang and R. Heusdens, "Distributed optimization using the primal-dual method of multipliers," *IEEE Trans. Signal Process., vol. 4, no. 1, pp. 173–187*, 2018.

[26] Q. Li, J. S. Gundersen, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed processing: Metrics, bounds, and algorithms," in *IEEE Trans. Inf. Forensics Security.*, 2021.

[27] D. Dolev, C. Dwork, O. Waarts, M. Yung, "Perfectly secure message transmission," *J. Assoc. Comput. Mach., vol. 40, no. 1, pp. 17-47,*, 1993.

[28] J. Dall and M. Christensen, "Random geometric graphs," *Physical review E, vol. 66, no. 1, pp. 016121*, 2002.

[29] M. Little, P. McSharry, S. Roberts, D. Costello, and I. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," *Nature Precedings*, pp. 1–1, 2007.

[30] D. Dua and C. Graff, "UCI machine learning repository," 2017.

[31] G. Ver Steeg, "Non-parametric entropy estimation toolbox (npeet)," *https://github.com/gregversteeg/NPEET*, 2000.