Privacy-Preserving Distributed Clustering for Electrical Load Profiling

Mengshuo Jia, Graduate Student Member, IEEE, Yi Wang, Member, IEEE, Chen Shen, Senior Member, IEEE, and Gabriela Hug, Senior Member, IEEE

Abstract—Electrical load profiling supports retailers in identifying consumer categories for customizing tariff design. However, each retailer only has access to the data of the customers it serves. Centralized joint clustering on retailers’ union load dataset either enables the identification of more types of users that allows to design more customized retail plans, or informs whether each retailer already has a sufficiently broad customer base. However, the centralized clustering requires access to the confidential data of retailers. This may cause privacy issues among retailers, because retailers can not or do not want to share their confidential information with others. To tackle this issue, we propose a privacy-preserving distributed clustering framework by developing a privacy-preserving accelerated average consensus (PP-AAC) algorithm. Using the proposed framework, we modify several commonly used clustering methods, including k-means, fuzzy C-means, and Gaussian mixture model, to provide privacy-preserving distributed clustering methods. In this way, the clustering on retailers’ union dataset can be achieved only by local calculations and information sharing between neighboring retailers without sacrificing privacy. The correctness, privacy-preserving property, time-saving feature, and robustness to random communication failures of the proposed methods are verified using a real-world Irish residential dataset.

Index Terms—Load pattern recognition, residential load profiling, clustering, privacy-preserving, distributed, consensus.

I. INTRODUCTION

A MASSIVE number of fine-grained electricity consumption data are being collected by smart meters. Identifying the load patterns from these smart meter data, i.e., residential load profiling, supports retailers and distribution system operators (DSO) in having a better understanding of the consumption behavior of consumers.

Technically, residential load profiling aims to capture different types of customers and behaviors. Particularly for retailers, knowing the types of consumers is important, as this is the prerequisite to designing customized retail plans [1]. Retailers can improve their commercial attractiveness by formulating different competitive retail plans for different classes of consumers [2]. Therefore, knowing more and particularly more diverse user types has strong practical significance for retailers. However, each retailer only has access to the data of the consumers it serves (the so-called “horizontally partitioned mode” of data [3]). Hence, it is very likely that a retailer is only aware of a few consumer types. For example, emerging retailers who have just joined the retail market might have little knowledge of user types; existing retailers who adopt fixed retail strategies might only attract specified classes of consumers as well. Joint clustering on the union dataset of multiple retailers allows identifying all customer types in the joint dataset.

Overall, the amount of information provided by joint clustering constitutes an upper bound of what a retailer can obtain using only its own dataset. Therefore, the practical significance of realizing joint clustering is actually twofold: (1) joint clustering either helps retailers identify more types of users, or (2) tells retailers whether their own user types are diverse enough to capture all the types in the union dataset. The former could help retailers design more diversified retail plans and the latter enables retailers to reach an informed decision whether they should stop seeking the cooperation that aims to identify more user types. Both are useful for retailers in terms of practical purposes.

Nevertheless, joint clustering requires retailers to share data with others. Because these data are confidential information, retailers are prohibited to directly or indirectly share this information. The former refers to directly sharing the raw data, and the latter refers to sharing statistics information, e.g., retailers’ numbers of consumers in any category. Thus, a privacy-preserving distributed clustering scheme is required, where retailers can possibly cooperate with others to jointly achieve the clustering results on their union consumption dataset via local calculation and communication. During the cooperation, the confidential information of each retailer, e.g., the raw residential load data or the number of consumers in a category, can not be deduced by others. Note that in this article, we choose the term “privacy” to represent the confidentiality of retailers. The specific definition of “privacy” will be further given in the next section.

So far, various clustering algorithms have been applied for load profiling, such as hierarchical clustering using different linkages [4], CFSDP [5], k-means [6], fuzzy C-means algorithm (FCA) [7], Gaussian mixture model (GMM) [8], self organizing map [9], etc. However, to the best of our
knowledge, there is no relevant research on privacy-preserving distributed clustering for load profiling.

To bridge this gap, this article proposes a privacy-preserving distributed clustering framework for load profiling. This framework can be used to transform three commonly used clustering methods, i.e., k-means, FCA, and GMM, into distributed clustering algorithms for the purpose of privacy-preserving load profiling. There are four reasons why we chose k-means, fuzzy C-means, and Gaussian mixture model (GMM): (1) they are commonly used algorithms for electrical load clustering [10]–[12]; (2) they include both the ‘hard’ and ‘soft’ clustering methods, that is, k-means is a ‘hard’ clustering method that delivers deterministic clustering results [13] whereas FCA and GMM are ‘soft’ clustering methods that provide an extent or a probability measure to describe the belonging of samples to clusters. Such methods can be leveraged to evaluate overlapping clusters or uncertain cluster memberships [14]; (3) they are distance-matrix-free techniques, which means that they do not require a complete communication network, i.e., communication links among neighboring retailers are sufficient; (4) they have commonalities regarding their implementation, which will be further discussed in Section II.

In fact, many works about privacy-preserving clustering for horizontally partitioned data have been conducted in different fields such as marketing and medicine [15]. Among them, the cryptography-based methods are most commonly used. These methods use secure multiparty computation [16], [17], homomorphic encryption technique [18], [19], or the combination of both [20] to turn the clustering methods into the privacy-preserving k-means [18], [20], the privacy-preserving FCA [16], or the privacy-preserving GMM [17], [19]. However, the methods using secure multiparty computation are extremely computationally expensive [21]. Besides, the overheads of encryption in the homomorphic encryption technique also limit the scope of the corresponding clustering methods [22] and result in time-consuming computations [23]. To reduce overheads, secret sharing can be adopted to design the privacy-preserving k-means clustering [21], [24]. However, these secret-sharing-based methods, including the aforementioned cryptography-based methods, are not fully distributed algorithms, because each party (the data owners, like the retailers in this article) either has to interact with a data center [16], [18], [24], or has to communicate with all the other parties [20], [21], or has to share its information along a pre-selected information transmission path [17], [19], [22]. These algorithms have the following drawback: the existence of a data center or a preset information sharing path greatly increases the risk of a single point or single line failure.

The proposed privacy-preserving distributed clustering framework aims to solve the above issues. We first perform commonality analysis of the traditional k-means, FCA, and GMM given the horizontally partitioned load data. In all of these methods, the problem reduces to how to calculate the summation of retailers’ high-dimensional private data in a fully distributed and privacy-preserving manner. To solve this problem, we employ the average consensus (AC) method because this algorithm is robust to communication failures. However, the slow rate of its convergence towards the average is the major deficiency of this algorithm [25]. Besides, the AC algorithm will reveal the private information available to the retailers during the interaction between neighbors. Therefore, we first introduce an accelerated AC (AAC) algorithm to significantly improve the rate of convergence without sacrificing the simplicity of the original AC algorithm [25]. Then, we adapt the AAC algorithm to provide a privacy-preserving version by leveraging the exponentially decaying disturbance with zero-sum property proposed in [26]. The convergence of the proposed privacy-preserving AAC (PP-AAC) algorithm is also proved. Meanwhile, adding noise is a very efficient way to realize solid privacy protection. Therefore, the proposed PP-AAC algorithm is still computation-friendly for high-dimensional load data. We further develop the privacy-preserving distributed clustering framework based on the proposed PP-AAC algorithm. This framework can convert the traditional k-means, FCA, and GMM into fully distributed privacy-preserving clustering methods, where each retailer only needs to communicate with its surrounding neighbors to obtain the exact load pattern identification results of all the consumers. Finally, we provide the privacy and complexity analyses of the proposed framework.

This article makes the following contributions.

- Propose a PP-AAC algorithm for retailers to compute the summation of their private data in an accelerated and privacy-preserving manner.
- Propose a privacy-preserving distributed clustering framework for retailers, which can convert k-means, FCA, and GMM methods into their privacy-preserving distributed versions with high accuracy, satisfactory efficiency, and strong robustness to communication failures.

To the best of our knowledge, this is the first privacy-preserving distributed clustering framework for electrical load profiling.

It should be emphasized that the focus of this article lies on designing tools for retailers to realize joint clustering on their union datasets in a distributed manner with the protection of confidential information. We further verify that these tools have high accuracy, high efficiency, and strong robustness. Identifying under what conditions, for which retailers and for what sizes and types of datasets joint clustering is indeed beneficial is beyond the scope of this article. But as discussed earlier, the result of joint clustering has practical relevance for a wide range of retailers and applications.

The rest of this article is organized as follows. Section II analyzes the commonality of k-means, FCA, and GMM. The PP-AAC algorithm is proposed in Section III. Section IV develops the privacy-preserving distributed clustering framework for the three clustering methods. Case studies are provided in Section V, and Section VI concludes this article.
II. PROBLEM FORMULATION

This section first briefly reviews the standard clustering methods: k-means [27], FCA [28], and GMM [29], and then gives the commonality analyses of them. Before that, we assume that the union dataset consists of $N$ observations. These observations are distributed among $M$ retailers, where retailer $i$ has $N_i$ consumers, i.e., $N_i$ observations. Besides, the centroid of cluster $k$, described by $\mu_k$, is considered as the $k$-th load pattern of the union dataset.

A. K-Means

K-means partitions $N$ observations into $K$ clusters by minimizing the within-cluster variances as follows:

$$\min f = \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{n \in C_k} \| y_{i,n} - \mu_k \|^2$$

where $y_{i,n}$ is the $n$-th observation of retailer $i$. $C_k$ represents the index set of the observations belonging to cluster $k$.

Although finding the solution is NP-hard, Lloyd’s algorithm guarantees to find a local minimum in a few iterations [27]. First, $K$ initial cluster centroids are arbitrarily and randomly assigned. Then, in each iteration, the cluster index of $y_{i,n}$ is computed by

$$c_{i,n} : = \arg \min_k \| y_{i,n} - \mu_k \|$$

(1)

and the centroid of cluster $k$ is updated by

$$\mu_k = \frac{\sum_{i=1}^{M} s_{k,i}}{\sum_{i=1}^{M} z_{k,i}}$$

(2)

$$s_{k,i} = \sum_{n=1}^{N_i} I(c_{i,n} = k) y_{i,n}$$

(3)

$$z_{k,i} = \sum_{n=1}^{N_i} I(c_{i,n} = k)$$

(4)

sequentially. These two steps are repeated until convergence is achieved. Note that $I(a = b)$ equals 1 if $a = b$ and 0 otherwise.

B. FCA

FCA is the best-known method for fuzzy clustering with the objective function given as follows:

$$\min f = \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{n=1}^{N_i} \rho_{k,i,n}^m \| y_{i,n} - \mu_k \|^2$$

where $m$ is the fuzziness index and $\rho_{k,i,n}^m$ is the degree to which $y_{i,n}$ belongs to $C_k$. The following iterative procedure solves this problem: the degree to which the observation belongs to cluster $k$ is first calculated by

$$\rho_{k,i,n}^m = \frac{\| y_{i,n} - \mu_k \|^{-2/(m-1)}}{\sum_{j=1}^{K} \| y_{i,n} - \mu_j \|^{-2/(m-1)}}$$

(5)

Then, the centroid of cluster $k$ is updated by

$$\mu_k = \frac{\sum_{i=1}^{M} s_{k,i}}{\sum_{i=1}^{M} z_{k,i}}$$

(6)

Different from k-means, where each observation either belongs to a cluster or not, FCA assigns degrees for each observation to be in every cluster, i.e., FCA is a type of soft clustering.

C. GMM

As a convex combination of $K$ Gaussian components $N_k$ with weight $o_k$ and covariance $\Sigma_k$, GMM is given by

$$g(y_{i,n}) = \sum_{k=1}^{K} o_k N_k(y_{i,n}|\mu_k, \Sigma_k)$$

(9)

where each Gaussian component represents a cluster.

To divide the union dataset into $K$ clusters by GMM, one should train GMM by leveraging the maximum likelihood estimation, which is given as follows:

$$\max f = \prod_{k=1}^{K} \prod_{i=1}^{M} \prod_{n=1}^{N_i} \frac{1}{\sqrt{2\pi} \det \Sigma_k} \exp \left( -\frac{1}{2} (y_{i,n} - \mu_k)^T \Sigma_k^{-1} (y_{i,n} - \mu_k) \right)$$

s.t. $0 \leq o_k \leq 1$, $\sum_{k=1}^{K} o_k = 1$

The most commonly used maximum likelihood estimation method is the expectation-maximization (EM) algorithm [29], which can be summarized as two iterative steps: the E-step and the M-step. The E-step, as given in

$$Q_{k,i,n} = \frac{o_k N_k(y_{i,n}|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} o_j N_j(y_{i,n}|\mu_j, \Sigma_j)}$$

(10)

computes the probability that an observation belongs to cluster $k$. The M-step updates the parameters in (9) according to

$$o_k = \frac{1}{N} \sum_{i=1}^{M} z_{k,i}$$

(11)

$$\mu_k = \frac{\sum_{i=1}^{M} s_{k,i}}{\sum_{i=1}^{M} z_{k,i}}$$

(12)

$$\Sigma_k = \frac{\sum_{i=1}^{M} h_{k,i}}{\sum_{i=1}^{M} z_{k,i}}$$

(13)

$$s_{k,i} = \sum_{n=1}^{N_i} Q_{k,i,n} y_{i,n}$$

(14)

$$z_{k,i} = \sum_{n=1}^{N_i} Q_{k,i,n}$$

(15)

$$h_{k,i} = \sum_{n=1}^{N_i} Q_{k,i,n} (y_{i,n} - \mu_k)^T (y_{i,n} - \mu_k)$$

(16)

After convergence, the final parameters of (9) are reached. The final probability that an observation belongs to cluster $k$
can be obtained by substituting the final parameters into (10). Same as FCA, GMM is also a soft clustering method.

D. Commonality Analysis

The clustering of k-means, FCA, and GMM have two points in common, which are listed in Remark 1 and 2.

Remark 1: The clustering processes of k-means, FCA, and GMM can all be summarized in two parts: the local calculation part and the global calculation part, where the local one can be performed by each retailer, and the global one is essentially the summation of each retailer’s local calculation results.

In fact, each retailer can directly perform the first steps of the three algorithms via its own data, i.e., the calculation in (1), (5) or (10). Then, retailer i is able to compute the following local results \( L_{k,i} \):

\[
L_{k,i} = \begin{cases} 
  s_{k,i} \text{ in (3)}, & \text{for k-means} \\
  z_{k,i} \text{ in (4)}, & \text{for FCA} \\
  s_{k,i} \text{ in (7)}, & \text{for GMM} \\
  z_{k,i} \text{ in (8)}, & \text{for GMM} \\
  s_{k,i} \text{ in (14)}, & \text{for GMM} \\
  z_{k,i} \text{ in (15)}, & \text{for GMM} \\
  h_{k,i} \text{ in (16)}, & \text{for GMM}
\end{cases}
\]

(17)

depending on the algorithm used. Once each retailer obtains the local results, the global summation of those local results from all retailers is required to continue the clustering method. For example, k-means algorithm needs to sum the local results \( s_{k,i} \) and \( z_{k,i} \) of all retailers respectively to update the centroid of cluster \( k \) in (2). Let \( G_k \) be the global summation result, then we have:

\[
G_k = \begin{cases} 
  \sum_{i=1}^{M} s_{k,i}, \sum_{i=1}^{M} z_{k,i} \text{ in (2), for k-means} \\
  \sum_{i=1}^{M} s_{k,i}, \sum_{i=1}^{M} z_{k,i} \text{ in (6), for FCA} \\
  \sum_{i=1}^{M} s_{k,i}, \sum_{i=1}^{M} z_{k,i}, \sum_{i=1}^{M} h_{k,i} \text{ in (11)-(13), for GMM}
\end{cases}
\]

(18)

Therefore, the relationship between the local and the global calculation parts can be generalized to:

\[
G_k = \sum_{i=1}^{M} L_{k,i}
\]

(19)

where \( L_{k,i} \) can be calculated by each retailer locally using (17), while the computation of \( G_k \) needs cooperations among all retailers. Once \( G_k \) in (18) is obtained, the second steps of the three algorithms can be carried out and the iterative procedure continues.

Remark 2: Each retailer’s local calculation results from k-means, FCA, and GMM contain private information, so that retailer i will refuse to share its \( L_{k,i} \) with others.

In fact, if retailer i shares its \( L_{k,i} \) (\( \forall k \)) with retailer j, the latter can derive the following private information of retailer i:

1) The Number of Retailer i’s Consumers: Once retailer j has received \( z_{k,i} \) (\( \forall k \)) in (4), (8) or (15), it can compute the number via:

\[
N_i = \sum_{k=1}^{K} z_{k,i}
\]

2) The Proportion or Number of Retailer i’s Consumers Belonging to Cluster k: Once retailer j receives \( N_i \), it will also obtain the proportion of retailer i’s consumers belonging to cluster k by:

\[
r_{k,i} = \frac{z_{k,i}}{N_i}
\]

Particularly, retailer j can directly know the specific number of retailer i’s consumers belonging to cluster k by receiving \( z_{k,i} \) in (4).

3) Retailer i’s Local Load Pattern of Cluster k: Once retailer j has received \( s_{k,i} \) in (3), (7) or (14), along with \( z_{k,i} \) in hand, retailer j can compute the local centroid of retailer i in cluster k by:

\[
\mu_{k,i} = \frac{s_{k,i}}{z_{k,i}}
\]

which will reveal the approximate load pattern of retailer i. For example, we choose \( z_{k,i} \) in (3) and \( z_{k,i} \) in (4), then \( \mu_{k,i} \) is essentially the mean of retailer i’s observations belonging to cluster k, which can be considered as its approximate load pattern in cluster k. The approximation lies in the fact that \( s_{k,i} \) and \( z_{k,i} \) are calculated using the global centroid in (2) in the last iteration, not the local centroid of retailer i in the last iteration; otherwise it will be the exact load pattern based on retailer i’s dataset.

Definition 1: In this article, we define the “privacy” of retailer i (\( \forall i \)) as the confidential information set \( P_i = \{y_{i,n}, N_i, r_{k,i}, \mu_{k,i} | n = 1, \ldots, N_i, k = 1, \ldots, K \} \).

Clearly, retailer i will not let retailer j obtain \( P_i \). As a result, directly sharing \( L_{k,i} \) will be refused by retailer i, impeding the implementation of the key summation in (18) for the three algorithms. Therefore, a privacy-preserving distributed summation algorithm to compute (18) is required.

III. PP-AAC Algorithm

To achieve a distributed summation algorithm, this section first introduces an AAC algorithm with a fast convergence rate [25]. After that, we further improve the AAC algorithm by leveraging an exponentially decaying disturbance with zero-sum property to propose a PP-AAC algorithm. Finally, the convergence of the proposed algorithm is proved.

A. AAC Algorithm

The AAC algorithm is graph-theory-based. Therefore, we consider a graph consisting of the M nodes and \( n_l \) edges. Each node represents a retailer, and the edge between each pair of nodes means that there is bidirectional noise-free communication between two retailers. This graph is publicly known by all retailers. Denote the node set by \( \mathcal{V} \) and the edge set by \( \mathcal{E} \). The neighborhood of retailer i is represented by the set \( \Omega_i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}\} \). The degree of retailer i is denoted by \( d_i \), which is defined as the cardinality of \( \Omega_i \). Let \( W \in \mathbb{R}^{M \times M} \) be the Metropolis weight matrix with elements as follows [30]:

\[
W_{ij} = \begin{cases} 
  \frac{1}{1 + \max_{d_i,d_j}} & \text{if } j \in \Omega_i \\
  1 - \sum_{k \in \Omega_i} W_{i,k} & \text{if } i = j \\
  0 & \text{otherwise}
\end{cases}
\]

(20)


In the AAC algorithm, each retailer has a state value that will be updated through iterations. Let \( x_i \) be the state of retailer \( i \) in the AAC algorithm, then the state update equation of the AAC algorithm in the \( t \)-th iteration is given by

\[
x_i(t + 1) = \alpha x_i(t) + (1 - \alpha)x_i^m(t + 1)
\]

which is a convex combination of the value from the original AC algorithm and the predictor given respectively by

\[
x_i^w(t + 1) = W_{i,j}x_j(t) + \sum_{j \in \Omega} W_{i,j}x_j(t)
\]

\[
x_i^p(t + 1) = 2 \cdot x_i^m(t + 1) - x_i(t)
\]

The matrix form of the update is given as follows:

\[
W^* \triangleq (1 + \alpha)W - \alpha I
\]

\[
X(t + 1) = W^*X(t)
\]

where \( X(t) = [x_1(t), \ldots, x_M(t)]^T \), and \( I \in \mathbb{R}^{M \times M} \) is the identity matrix. We call \( W^* \in \mathbb{R}^{M \times M} \) the accelerated Metropolis weight matrix. In this way, \( x_i \) will converge to the mean of all retailers’ initial standardized state values

\[
\lim_{t \to \infty} x_i(t) = \frac{1}{M} \sum_{i=1}^{M} x_i(0)
\]

with the fastest asymptotic worst-case convergence rate if the weight coefficient \( \alpha \) equals the optimal value [25]:

\[
\alpha = \frac{\lambda_M + \lambda_2}{2 - \lambda_M - \lambda_2}
\]

where \( \lambda_M \) is the smallest eigenvalue of \( W \), and \( \lambda_2 \) is the second largest eigenvalue of \( W \). Since the graph is publicly known by all retailers, each retailer can easily compute \( W \) using (20). Then, \( W^* \) can be obtained by all retailers using (24).

It should be emphasized that, \( x_i(t) \) could also be a vector, e.g., \( x_{ij}(t) \in \mathbb{R}^{1 \times K} \). In this case, \( X(t) = [x_{ij}(t)] \in \mathbb{R}^{K \times K} \) and the update in equation (25) is still valid. The converged results are:

\[
\lim_{t \to \infty} x_i(t) = \left[ \frac{1}{M} \sum_{i=1}^{M} L_{1,i}, \ldots, \frac{1}{M} \sum_{i=1}^{M} L_{K,i} \right]
\]

In the following, if necessary, we will use \( x_i(t) \) to represent the form of a vector.

Besides, we should also note that, the AAC algorithm is fully distributed, i.e., each retailer only needs to communicate with its neighbors. Besides, after convergence, retailers can obtain the summation of their initial standardized state values by multiplying the mean in (26) by \( M \). Thus, let \( x_i(0) \) be equal to \( L_{k,i} \), then each retailer can obtain \( G_k \) in (18) in a fully distributed manner using the AAC algorithm. However, in the first iteration, retailer \( i \) will send \( x_i(0) = L_{k,i} \) to its neighbors, which directly reveals the private information of retailer \( i \).

### B. PP-AAC Algorithm

To facilitate the AAC algorithm with privacy-perserving characteristics, we utilize the exponentially decaying disturbance with zero-sum property from [26] to mask the interactive state values among neighbors during the AAC iterations, so that each retailer cannot derive private information of the others.

The proposed PP-AAC algorithm is defined by

\[
x_i(t + 1) = W_{i,j}^*x_j^w(t) + \sum_{j \in \Omega} W_{i,j}^*x_j^p(t)
\]

where \( x_i^w(t) \) is the state value masked by the disturbance \( \theta_i(t) \) as follows:

\[
x_i^w(t) = x_i(t) + \theta_i(t)
\]

\[
\theta_i(t) = \delta_i(t) - \delta_i(t - 1)
\]

The noise \( \delta_i(t) \) is randomly selected from \([-\frac{\sigma}{2} \beta^t, \frac{\sigma}{2} \beta^{t+1}]\) by retailer \( i \), where \( \sigma > 0, \beta \in [0,1) \), and \( \delta(t < 0) = 0 \). This design leads to the two features of \( \theta_i(t) \), which will be used for the following proof of Theorem 1:

- The noise \( \delta_i(t) \) is exponentially decaying as \( \beta \in [0,1) \) and \( t \) grows with the number of iterations. So \( \theta_i(t) \) is also exponentially decaying.
- The disturbance \( \theta_i(t) \) has zero-sum property, which means that if we sum up \( \theta_i(t) (\forall i) \) from \( t = 0 \) to infinity, the result will be 0, i.e.,

\[
\sum_{i=1}^{M} \sum_{t=0}^{\infty} \theta_i(t) = \sum_{i=1}^{M} \lim_{t \to \infty} \delta_i(t) = 0
\]

The proof is given in Appendix A.

**Theorem 1:** The proposed PP-AAC algorithm in (28) will make each retailer’s state value converge to the average of all retailers’ initial state values, i.e., (26) still holds.

**Proof:** See Appendix B.

Remark 3: Theoretically, the summation of noise reaches zero when \( t \) reaches infinity. However, as this summation is exponentially decaying as \( t \) grows, it will not take too many iterations until the summation of noise becomes negligible. E.g., for \( \delta = 2 \) and \( \beta = 0.4 \), when \( t \) reaches 15, the maximal value of \( |\sum_{i=1}^{M} \sum_{t=0}^{15} \theta_i(t)| \) is less than \( 4.30 \times 10^{-6} \), which is negligible already.

### IV. PRIVACY-PRESERVING DISTRIBUTED CLUSTERING FRAMEWORK

This section describes the privacy-preserving distributed clustering framework for k-means, FCA, and GMM incorporating the proposed PP-AAC algorithm. In addition, we provide the privacy and complexity analyses of the proposed framework.

#### A. Clustering Framework

The idea of the clustering framework is that independent retailer first performs its local calculation according to (17); then each retailer sets its local result as the initial state of the proposed PP-AAC algorithm; after convergence, each retailer obtains the global summation of all the local results in (18); finally, using the global summations, each retailer can perform the rest of the clustering method to update the global information, e.g., the centroids of all clusters. The detailed clustering framework is demonstrated in Algorithm 1.
Algorithm 1: The Clustering Framework

Input: Standardized $y_{1,n}$ ($n = 1, \ldots, N_i$) of retailer $i$ ($\forall i$).
Input: Arbitrarily and publicly assign $K$ centroids $\mu_k$.
Output: The load pattern $\mu_k$ ($\forall k$) of the union dataset

while convergence criterion of clustering is not met do
  Retailer $i$ ($\forall i$) calculates $L_{k,j}$ ($\forall k$) in (17);
  Retailer $i$ ($\forall i$) sets $x_i(t) = [L_{1,i}, \ldots, L_{K,i}]$;
  $t = 0$;
  while average consensus is not achieved do
    Retailer $i$ ($\forall i$) randomly selects $\delta_i(t)$ by rule;
    Retailer $i$ ($\forall i$) masks its $x_i(t)$ by (29);
    Retailer $i$ ($\forall i$) computes its $x_i(t+1)$ by (28);
    $t = t + 1$;
  end
  Retailer $i$ ($\forall i$) obtains $[G_{1,i}, \ldots, G_{K,i}]$ by $M \times x_i(t)$;
  Retailer $i$ ($\forall i$) updates global cluster information;
  - K-means: updates $\mu_k$ ($\forall k$) by (2);
  - FCA: updates $\mu_k$ ($\forall k$) by (6);
  - GMM: updates $\omega_k, \mu_k, \Sigma_k$ ($\forall k$) by (11)-(13);
end
Retailer $i$ ($\forall i$) gets the load patterns of the union dataset;
- K-means: gets the final $\mu_k$ ($\forall k$) in (2);
- FCA: gets the final $\mu_k$ ($\forall k$) in (6);
- GMM: gets the final $\mu_k$ ($\forall k$) in (12);

Note that the proposed clustering framework is not applicable to connected-based clustering such as hierarchical clusterings using different linkage criteria [10], nor to density-based clustering such as DBSCAN [1] or CFSFDP [5]. The reasons are twofold:

1) Essentially, the proposed framework is only designed for clustering methods that require the summation of retailers’ data, as given in (19). However, hierarchical clustering algorithms and DBSCAN do not require this summation. Instead, they require a distance matrix among retailers’ data. Computing the distance matrix among retailers’ data is not mathematically equivalent to calculating the summation of retailers’ data.

2) Computing the distance matrix among retailers’ data requires the communication between any two retailers, i.e., the communication network needs to be complete. The proposed framework, however, only embeds neighboring communication.

B. Privacy Analysis

As aforementioned, the AAC algorithm will directly reveal the initial value $x_i(0)$ in the first iteration. On the contrary, in the first iteration of the proposed PP-AAC algorithm, retailer $i$ ($\forall i$) receives $x_j^T(t)$ ($\forall j \in \Omega_i$) instead of $x_j^T(0)$. Since $x_j^T(0)$ is masked using an independent disturbance $\theta_j(0)$ by retailer $j$, retailer $i$ cannot derive the original value of $x_j^T(0)$ from $x_j^T(t)$, thus retailer $i$ will not know the private $L_{k,j}$ of its neighbors, protecting the private information $P_j$ of retailer $j$. In the remaining iterations, the process of adding disturbance continues; meanwhile, $x_i^T(t)$ begins to converge to the mean value in (26) and moves away from its initial value, which further masks the true initial value. Quantitative illustrations will be shown in the next section.

In addition, we should note that if $j \in \Omega_i$, then retailer $i$ can receive all the information that retailer $j$ has received, including retailer $j$’s information, then retailer $i$ can deduce retailer $j$’s initial value even if the disturbance is introduced [26]. Therefore, the authors in [26] and [31] both consider it necessary to assume that retailer $i$ cannot receive all the information that retailer $j$ has. The assumption is also adopted in this article. Since $W$ is publicly known by all retailers, retailer $j$ can tell that whether $\Omega_j$ is a subset of its neighbor $\Omega_i$. If such a situation occurs, retailer $j$ can refuse to communicate with retailer $i$. Therefore, the assumption will hold in practice.

C. Complexity Analysis

For the distributed framework, we investigate each retailer’s computation and communication overhead.

The proposed clustering framework not only keeps all the multiplication calculations in the original clustering methods, but also introduces new multiplication calculations by integrating the proposed PP-AAC algorithm. The multiplication calculations in the original clustering methods are divided by retailers according to their number of observations, i.e., if the computation overhead of the original clustering method is $O(\phi)$, then the overhead of retailer $i$ is $O(\phi N_i/N)$. Moreover, in each iteration of the PP-AAC algorithm, although the disturbance can be queried from the preset lookup table, retailer $i$ ($\forall i$) still needs to compute $W_i(t) x_i^T(t)$ and $W_j(t) x_j^T(t)$ ($\forall j \in \Omega_i$), which requires $d_i = d_i + 1$ multiplications. Let $T_c$ denote the iteration number of the selected clustering method, and $T_o$ represent the iteration number of the proposed AAC algorithm, then the computation overhead of retailer $i$ is $O(\phi N_i/N + d_i T_o T_c)$. Take k-means for example, where $\phi = N K T_c$, then retailer $i$’s overhead is $O(N K T_o + d_i T_o T_c)$. Please note that $d_i T_o \ll N K$, because the number of retailers in a DN is small, and the proposed PP-AAC algorithm’s convergence is accelerated, thus $T_o$ is generally also small. However, $N$ is thousands and $K \geq 2$. Moreover, we know that $N_i \ll N$. Therefore, the computation overhead of retailer $i$ is significantly smaller than that of the centralized k-means. Detailed illustrations are shown in the next section.

Besides, in each iteration of the proposed AAC algorithm, the communication number of retailer $i$ is $d_i$ [32]. Therefore, the communication overhead of retailer $i$ is $O(d_i T_o T_c)$.

D. Advantages in Real-World Scenarios

The advantages of our approach in real-world scenarios are summarized as follows:

1) The proposed distributed approach supports retailers achieve joint clustering on their union datasets, a means that protects confidential information of these retailers. The amount of information provided by joint clustering constitutes an upper bound of what a retailer can obtain using only its own dataset. Therefore, as already mentioned earlier, our approach could either help retailers identify more types of users and customize retail plans,
or inform retailers whether their own customer base is diverse enough to represent all the types present in the union dataset.

2) The proposed approach protects retailers’ confidentiality from leakage. That said, during the communication, the confidential information of each retailer, e.g., the raw residential load data or the number of consumers of a category, cannot be deduced by others. This caters to the realistic requirements of retailers.

3) The proposed approach only requires neighboring communication and is highly robust to multi-communication-failure. Briefly, using our approach, retailers do not need to communicate with data centers, nor communicate with all the other retailers, nor share their information along a pre-selected transmission path. Therefore, our approach is applicable in realistic communication network settings.

4) The proposed approach can generate the same clustering results as the centralized joint clustering does. However, our approach is more efficient than the centralized one. This is because the calculation burden of clustering is distributed to different retailers in our approach. Meanwhile, the way we ensure confidentiality is adding noise, which has high efficiency.

V. CASE STUDY

A. Data Description and Experiment Setup

We utilize the smart meter data from Ireland for verification, which contains 509660 half-hourly daily electrical consumption observations of 5000 consumers [33]. The representative load profile (RLP) of each consumer is obtained via the method presented in [2]. The initial centroids for all clustering methods are randomly chosen. The algorithm was implemented in Python3 and simulations were ran on an i5-7267U 3.1 GHz processor with 8 GB RAM. It should be emphasized that all the methods were coded in serial structures in order to measure the computational time of each retailer separately.

Regarding the convergence criterion, for the PP-AAC algorithm, the convergence criterion is given as:

\[ |x_i(t + 1) - x_i(t)| < \sigma, \quad i = 1, \ldots, M \]

where \( t \) represents the iteration of the PP-AAC algorithm, and \( \sigma = 10^{-5} \) in this article. Besides, the convergence criterion for the three clustering methods are the same, that is, for each retailer (or the data center in the centralized clustering methods):

\[ |\mu_k(l + 1) - \mu_k(l)|^T < \xi, \quad k = 1, \ldots, K \]

where \( l \) represents the iteration of the clustering process, and \( \xi = 10^{-5} \) in this article.

Note that there are outer and inner iterations in the proposed framework, i.e., iterations of the PP-AAC algorithm and iterations of the clustering methods themselves. Since we will frequently discuss the iterations of the PP-AAC algorithm but barely discuss iterations of clustering methods, in the following simulations, we use “iterations” to refer to the iterations in the PP-AAC algorithm. For distinction, we particularly use “iterations of the clustering process” to reflect the iterations of the clustering methods themselves.

As the focus of this article is on designing tools for privacy-preserving joint clustering, the scope of the case studies should focus on evaluating the correctness, privacy-preserving property, time-saving feature, and robustness to communication failures of the proposed tools. Hence, showing whether or not and under what circumstances joint clustering always performs better than individual clustering is not the focus of these studies.

B. Correctness, Efficiency, and Privacy-Preserving Feature of the PP-AAC Algorithm

To verify the correctness and efficiency of the proposed PP-AAC algorithm, we compare it with three algorithms: the original AC algorithm in [30], the AAC algorithm proposed in [25] and the PP-AC algorithm proposed in [26]. We assume that there are 10 retailers in a DN. Their communication topology is shown in Fig. 1, where each retailer only communicates with its one-hop neighbors.

We use the four algorithms to compute the summation of the observations from each retailer’s first consumer. We then illustrate the average error of all retailers relative to the accurate summation result. The errors of the four algorithms for each iteration are shown in Fig. 2. It can be observed that the average error of the proposed PP-AAC algorithm converges to 0, indicating the correctness of this method. In addition, the proposed algorithm has the same convergence rate as the AAC algorithm. The PP-AC algorithm also has the same convergence rate as the AC algorithm. Please note that the proposed algorithm converges faster than both the AC and the PP-AC algorithm, indicating the efficiency of the
proposed algorithm. Therefore, the correctness and efficiency of the proposed PP-AAC algorithm are verified.

Compared to the AC algorithm and the AAC algorithm, the proposed algorithm also has the privacy-preserving feature. To illustrate this feature, we provide the value that retailer 1 shares with its neighbors during the above summation calculation at each iteration. The shared values of the four algorithms are shown in Fig. 3.

These shared values all converge to the real average value, but we should note that retailer 1 shares its real initial value with its neighbors in the first iteration when performing the AC and the AAC algorithm, which directly reveals the private information of retailer 1. However, after introducing the disturbance for masking, the proposed algorithm enables retailer 1 to share its masked initial value to its neighbors, which is far away from the real one as indicated by the black arrow. Thus, the proposed algorithm protects the privacy of retailer 1. Moreover, the proposed algorithm still converges faster than the PP-AC algorithm, even if they both start from the same masked initial point.

C. Correctness of the Proposed Clustering Framework

In the following simulations, we assume that there are 10 retailers in a DN, whose communication topology is shown in Fig. 1. Each retailer has access to 100 consumers. We can employ the proposed clustering framework to obtain privacy-preserving distributed k-means, FCA, and GMM clustering methods. Then, we use them for load pattern identification on the distributed datasets. As benchmarks, we also use the centralized k-means, FCA, and GMM for load pattern identification on the corresponding union dataset.

To verify the correctness of the clustering framework, in Fig. 4, we use the Silhouette coefficient index (SCI) [34] to evaluate the above distributed and centralized algorithms for a different numbers of clusters. The SCI of data point $n$ is defined as follows:

$$SCI(n) = \frac{b(n) - a(n)}{\max\{a(n), b(n)\}}$$

where $a(n)$ is the average distance between data point $n$ and other data points in the same cluster, and $b(n)$ is

$$b(n) = \min\{B(n, k), k = 1, \ldots, K, k \neq k_n\}$$

where $B(n, k)$ is the average distance between data point $n$ and data points in any other cluster except for cluster $k_n$, which is the cluster that data point $n$ belongs to. Note that, for FCA clustering, we choose

$$k_n = \arg \max_k \rho_{m}^{kn}$$

while for GMM clustering, we choose

$$k_n = \arg \max_k Q_{kn}$$

Finally, we use the mean SCI of all data points for illustration. The mean SCIs of different methods are illustrated in Fig. 4. Note that the abbreviation ‘PPD’ in Fig. 4 represents ‘privacy-preserving distributed’. This figure clearly shows that the SCI results of the proposed privacy-preserving distributed algorithms are identical to those of the centralized algorithms. This means that the clustering results on the distributed datasets using the proposed clustering framework, are exactly the same as those on the union dataset computed via the centralized methods, indicating the correctness of the proposed clustering framework.

Furthermore, compared to the centroids computed by the centralized clustering methods, we also study the relative errors of the PPD clustering methods. The relative error is defined as the maximal relative error of centroids computed by each PPD clustering method. The benchmarks are the centroids calculated by the corresponding centralized clustering methods. The relative error of each PPD clustering method is provided in Table I. We conclude that: the errors of the proposed PPD methods are negligibly small thereby demonstrating the correctness.

D. Learning More User Types by Joint Clustering With Our Clustering Framework

In this subsection, we illustrate one of the applications of practical significance of the joint clustering achieved by the proposed clustering framework, namely as it allows retailers to identify more user types. First, we again assume that there are 10 retailers in a DN, whose communication topology is shown in Fig. 1. Each retailer has access to 100 consumers. Next, we choose k-means for demonstration as it is a hard clustering method, which is very convenient for illustration. We use the most common way, i.e., the elbow method based on the sum of squared errors (also known as the objective function

![Fig. 3. Interactive information shared during the iterative process.](image)

![Fig. 4. The indicators results of different clustering methods and their respective privacy-preserving distributed versions.](image)
of k-means), to find the optimal number of clusters [35, 36]. From this, we find that the optimal cluster number of the union dataset (1000 RLPs, i.e., 1000 consumers) is \( K = 6 \), while that of the dataset of retailer 1 (100 RLPs) is \( K = 2 \). After that, we perform the centralized k-means on retailer 1’s dataset, and the results are shown in Fig. 5(a). Besides, we also perform the proposed privacy-preserving distributed k-means and the centralized k-means on the union dataset. The results are demonstrated in Fig. 5(b). The number of RLPs in each cluster is listed in the sub-figure’s title. Meanwhile, the RLPs and the load patterns of retailer 1 are highlighted in Fig. 5(b) as well. Note that, the Y-axis of Fig. 5 is the value of the RLPs, while the X-axis denotes the 48 time instances of the RLPs.

First, from Fig. 5(b), we can observe that the centroids of the proposed algorithm are coincident with the centroids of the centralized k-means. Second, the two load patterns of retailer 1’s dataset in Fig. 5(a), approximately match the 2nd and the 3rd load patterns of the union dataset in Fig. 5(b). However, retailer 1 missed the remaining four categories of consumers. Certainly, if retailer 1 only uses its own two load patterns for tariff design, its products will be difficult to attract the 608 consumers in the remaining clusters. On the contrary, by the proposed clustering framework, each retailer can use the six load patterns of all consumers for tariff design to attract all of them. Therefore, the joint clustering on retailers’ union dataset could help them capture more consumer patterns in this case, while the proposed method can enable retailers to achieve the accurate joint clustering in a distributed and privacy-preserving manner.

We would like to emphasize that the above case reflects the situation of emerging retailers with few consumers that would like to expand their customer base. The emerging retailers not only exist in mature retail markets, but are likely to be more common in the emerging retail markets. Besides, emerging retailers with a few consumer types, like retailer 1 in the above case, would have a strong motivation for capturing more user types by joint clustering. This illustrates the practical significance of joint clustering in terms of revealing more user patterns.

Of course, this test case cannot and does not want to prove that joint clustering can always capture more user types in other cases. It is true that in other cases, the information provided by individual clustering may approach the information provided by joint clustering, particularly in case of retailers whose customer bases are diverse. It does not matter, however, because helping retailers identify more user types is just one aspect of joint clustering that has practical significance, as discussed earlier in this article.

### E. Efficiency Comparisons Related to the Proposed Clustering Framework

For efficiency comparison, we first compare the centralized clustering method and the corresponding privacy-preserving distributed version enabled by the proposed clustering framework. Specifically, we assume that there are again the 10 retailers, each with 100 consumers and with the communication topology shown in Fig. 1. Note that for the distributed methods, the retailers’ computational times are different. Thus the maximum computational time of all retailers is chosen to represent the time of the distributed methods. The computational time and iteration numbers (I-Ns) of the clustering process are given in Table II. From this table, it is obvious that the iteration numbers of the clustering process in the corresponding centralized and distributed clustering methods are the same, but the computational times of the corresponding methods differ by an order of magnitude: the time consumed by each retailer in distributed clustering is significantly less than that of the centralized clustering, indicating the high efficiency of the proposed clustering framework.

For further efficiency comparison, we extended two cryptography-based privacy-preserving k-means clustering methods from other fields and applied them to the considered problem. One is a homomorphic-encryption-based privacy-preserving (HEPP) k-means from the field of participatory sensing [18], the other one is a secure-multiparty-computation-based privacy-preserving (SMCPP) k-means from the field of data mining [15]. These two methods are suitable for multi-party environments. Most importantly, these two methods are capable of continuing the computations even if communications failures occur. Note that the original HEPP k-means not only protects the raw data of each party but also prevents parties from knowing the centroids. To make the HEPP k-means suitable for electrical load profiling, we omit the operations pertaining to the protection of the centroids. Hence, the HEPP k-means used in this article is more efficient than the original one.

For the experiments, we assume again that there are 10 retailers in a DN. Each of them has access to 100 consumers. The communication topology for the proposed PPD k-means is given in Fig. 1, while the communication topologies for the HEPP k-means and SMCPP k-means are shown in Fig. 6.

---

**Fig. 5.** The clustering results on (a) the dataset of retailer 1 (100 RLPs) and (b) the union dataset (1000 RLPs). The RLPs and clustering centers of retailer 1 are also illustrated in (b).
We use different topologies as the methods have different needs regarding communication topology. The proposed PPD k-means requires iterative communication among neighboring retailers. The SMCPP k-means requires sequential communication among all retailers and the communication graph needs to be complete, i.e., every retailer has to communicate with every other one. The HEPP k-means needs sequential communication between retailers and a data analyst. That said, except for communicating with every other retailer, each retailer also needs to exchange information with the data analyst.

The efficiency comparison is given in Table III. The computational time of each method is the maximum time consumed by any retailer. Although the proposed PPD k-means requires iterative communication among retailers, its computational time is still much lower than the computational time of the other two methods. This is because adding noise is a more efficient way than the homomorphic encryption and secure multiparty computation when protecting privacy.

Please also note that the above computational time does not contain communication time. However, this time is probably negligible. In k-means for example, each retailer shares its masked value to its neighbors, which consists of the masked $s_{k,i} \in \mathbb{R}^{48 \times 1}$ and $z_{k,i} \in \mathbb{R}^{1 \times 1}$ for $k = 1, \ldots, 6$. Thus each retailer actually shares 294 floating-point numbers with its neighbors, i.e., 1.15 kbytes. We know that $N = 1000$, $N_i = 100$, $K = 6$ and $T_c = 7$. Meanwhile, $T_a = 27$ as shown in Fig. 2, and the degree of the retailer that consumes the most time is 5 (retailer 1), which is also the maximum degree among the retailers. According to the communication overhead analysis in Section IV-C, the maximum total amount of upstream data of all the retailers will be $1.15 \times 5 \times 27 \times 7 = 1086.75$ kbytes $\approx 1.06$ Mbytes. Since the global average broadband Internet speed is 11.03 Mbps, the actual maximum communication time for retailers will not exceed 0.1 seconds. This cost will be greatly reduced in Europe as it has the world’s highest concentration of countries with the fastest Internet, e.g., Sweden’s average speed is 55.18 Mbps [37].

### F. Robustness Comparisons Related to the Proposed Clustering Framework

The robustness comparisons among the proposed PPD k-means, the HEPP k-means and the SMCPP k-means are carried out for cases with communication failures. Again, we assume that there are 10 retailers in a DN, and each retailer has access to 100 consumers. Besides, as before the communication topologies for the three methods are given in Fig. 1 and Fig. 6, respectively.

To simulate the communication failures, we randomly selected $n_f$ communication lines and disable them during the interactions among retailers (and the data analyst). Specifically, for the proposed PPD k-means, we disconnect $n_f$ random communication lines in each iteration of the embedded PP-AAC method. Note that the number of failed lines is the same in each iteration, yet the lines that fail vary over the iterations. For the HEPP k-means, $n_f$ random communication lines are disconnected during the sequential interactions between retailers and the data analyst. For the SMCPP k-means, we disable $n_f$ random communication lines during the sequential interactions among retailers. The relative error of each evaluated method is listed in Table IV given different $n_f$. It is obvious that, the relative errors of the proposed PPD k-means are negligible despite nearly 30% of communication lines out of operation at each iteration (5 faulted communication lines out of 17 lines). The HEPP k-means does not converge even in case with only one communication line failing. Although the SMCPP k-means converges given one communication failure, its relative error exceeds 30%. As $n_f$ increases, the SMCPP k-means does not converge any more.

In fact, both the HEPP k-means and SMCPP k-means approaches are dependent on sequential communication. That said, the necessary information must be transmitted along a fixed route. Once communication failures occur, retailers or the data analyst will lose part of the necessary information. Therefore, the clustering results will be inaccurate or the clustering process does not converge. However, when communication failures occur in the proposed PPD k-means, retailers can accordingly adjust the Metropolis weights of the remaining neighbors and utilize the remaining information to achieve consensus. For an intuitive illustration, we directly use the PP-AAC algorithm to compute the summation of observations of 10 retailers under $n_f$ random communication failures. We still use the communication topology given in Fig. 1. The evolution of the average error relative to the true summation value is illustrated in Fig. 7.

![Data analyst](image1.png)

![Data analyst](image2.png)

Fig. 6. (a) communication topology for the HEPP k-means; (b) communication topology for the SMCPP k-means.

| $n_f$ | PPD k-means | HEPP k-means | SMCPP k-means |
|------|-------------|--------------|---------------|
| $n_f = 1$ | $0.64 \times 10^{-7}$ | no convergence | 32.85% |
| $n_f = 2$ | $0.77 \times 10^{-7}$ | no convergence | no convergence |
| $n_f = 3$ | $0.81 \times 10^{-7}$ | no convergence | no convergence |
| $n_f = 4$ | $0.74 \times 10^{-7}$ | no convergence | no convergence |
| $n_f = 5$ | $0.92 \times 10^{-7}$ | no convergence | no convergence |
iteration. Note that the convergence rate for the considered cases decreases slightly when the number of communication failures increases.

The computational time under communication failures is given in Table V. For the proposed PPD k-means, the time increases with the number of communication failures. But the increment is small and the efficiency is still satisfactory. For the other two methods, only the SMCPP k-means converges given one communication failure while its computational time is much higher than the computational time of the proposed PPD k-means.

G. Discussions on the Parameters and the Communication Topologies

The noise boundary coefficient δ and β, the weight coefficient α, and the communication topologies may affect the convergence rate of the PP-AAC algorithm and the overall computational time of the proposed framework. In the following, we will discuss in detail the influence of these parameters.

1) Noise Boundary Coefficients δ and β: We use the PP-AAC algorithm to compute the summation of observations of 10 retailers under different boundary coefficients. The communication topology of the retailers is shown in Fig. 1 and each of the retailers has access to 100 consumers. In Fig. 8, we illustrate the evolution of the average error relative to the true summation value. Obviously, the boundary coefficients have influence on the errors during early iterations. This is because a larger boundary coefficient may produce a larger random noise in the early stage. However, due to the exponential decrease of the noise boundary, the added random noise declines to zero rapidly. Hence, the error curves under different coefficients are close after about 20 iterations.

The computational time of the proposed clustering framework using different boundary coefficients is listed in Table VI and Table VII, respectively. The computational time only slightly increases with the increase in the values of boundary coefficients.

2) Weight Coefficient α: We assume that there are 10 retailers, whose communication topology is shown in Fig. 1. Each of the retailers has access to 100 consumers. In this case, the optimal value for α is 0.41 according to (27). For comparison, we use the PP-AAC algorithm to compute the summation of observations of retailers under different α. The evolution of the average error relative to the true summation value is shown in Fig. 9. Clearly, the error curve using the optimal α performs the best. Besides, since α = 0.35 and α = 0.45 are closer to the optimal α, their curves have a similar convergence rate as the optimal one. When α reaches 0.5, its corresponding convergence is slower.

The computational time of the proposed clustering framework using different α is listed in Table VIII. The computational time is the lowest when α is optimal, but is only slightly
different for $\alpha = 0.35$ and $\alpha = 0.45$. The computational time significantly increases when $\alpha = 0.5$.

3) Communication Topology: Different communication topologies are studied taking into account two aspects: different numbers of retailers (i.e., different $M$), and different connections among a fixed number of retailers. For the number of retailers, we would like to emphasize the following two points. First, the number of retailers in a market will not be very large and the retailers belonging to the same company can be treated as one. Second, it is envisioned that the proposed framework is used by only a part of the retailers in a market. Using this framework, they can cooperate without sharing any confidential information and benefit from the union clustering, which can help them improve their position in the competition with the rest of the retailers in this market. Hence, we choose $M = 10$ and $M = 20$ for the simulations to illustrate the performance of the algorithm. For each case, we define three types of connections: original connections, less connections, and more connections. Detailed topologies are shown in Fig. 10.

We use the PP-AAC algorithm to compute the summation of observations of 10 and 20 retailers under different topologies. Then we show the iterative evolution of the average error relative to the true summation value in Fig. 11.

As can be observed, for the same type of connections, the increase in $M$ leads to slower convergence. Besides, for the same number of retailers, in general, the fewer the connections, the slower the convergence. Nevertheless, all these cases can still achieve convergence within 60 iterations.

We further use the proposed framework to perform privacy-preserving distributed clustering given the communication topologies aforementioned. We use the same dataset for different cases to guarantee the same iteration number of the clustering process. Hence for $M = 10$, each retailer has access to 100 consumers, while for $M = 20$, each retailer has 50 consumers. The average computational time of the retailers is listed in Table IX. When $M = 10$, the case of more connections requires the lowest computational time, while the case of fewer connections leads to the highest computational time. The case of $M = 20$ has the same trend as $M = 10$. However, we should emphasize that, although for the same type of connections, a larger $M$ leads to slower convergence, the time consumed by 20 retailers is less than that consumed by 10 retailers. This is because, for the case of 20 retailers, each retailer only has 50 consumers, while for the case of 10 retailers, each of them has 100 consumers. So the local calculation burden of each retailer when $M = 20$ is only half of that cost when $M = 10$. Since the local calculation burden constitutes most of the overall clustering burden, the computational time when $M = 20$ is therefore less than the time consumed when $M = 10$.

Finally, we consider a case consisting of 20 retailers. Each retailer has 100 consumers. This case has a different union dataset as the one used in Table IX. Accordingly, required iteration number of the clustering process differs. As given in Table X, the required iteration number of the clustering process for PPD $k$-means for the different types of connections is 17 whereas the iteration number of the clustering process in the PPD $k$-means for the cases in Table IX is 7. Naturally, the
**TABLE X**

| Topology       | Time (in Seconds) | I-Ns of the Clustering Process |
|----------------|-------------------|-------------------------------|
| Centralized    | 1.114             | 17                            |
| \( M = 20, \) less | 0.114             | 17                            |
| \( M = 20, \) original | 0.081             | 17                            |
| \( M = 20, \) more | 0.071             | 17                            |

increase in number of retailers and iterations of the clustering process increases the required computational time. It should be pointed out that in all cases the computational time of the PPD k-means is significantly lower than the one of the centralized approach. In other words, the proposed framework is still efficient even though considering the case of more retailers with more consumers under fewer connections.

**H. Discussions on Different Settings Related to Retailers**

We have conducted additional case studies to evaluate the correctness, convergence, and efficiency of our approach varying the dataset size, number of retailers and allocations of consumers to retailers.

1) **Various Dataset Sizes:** We consider 10 retailers, whose communication topology is shown in Fig. 1. We carry out a number of simulations which differ by the allocated number of consumers to each retailer, specifically, we assign 50, 100, 150 and 200 consumers.

Note that the dataset size does not affect the convergence rate of the PP-AAC algorithm. This is because the inputs of the PP-AAC algorithm are the statistics information of consumers. The dimensions of the statistics information will not change with the dataset size. In fact, the factors that influence the convergence rate of the PP-AAC algorithm have already been discussed in Section V-G. Nevertheless, the dataset size may affect the correctness and efficiency of the overall framework, as the burden as well as the results of local calculations differ, and the iteration number of the clustering process changes. Therefore, the maximal computational time of the PPD clustering methods as well as their relative errors are listed in Table XI. For comparison, we also provide the computational time of the centralized clustering methods in Table XI indicated by “CK”, “CF” and “CG.” Besides, “PK”, “PF”, and “PG” represent the proposed k-means, FCA, and GMM respectively.

From this table, we conclude that: (1) compared to the centralized clustering methods, the proposed PPD methods in this study are much more efficient no matter how large the dataset is; (2) under any dataset, the errors of the proposed PPD methods are negligible thereby demonstrating their correctness. It should be emphasized that, different datasets may lead to different iteration numbers of the clustering process. There is no clear relationship between the dataset size and the iteration number of the clustering process. For example, the iteration numbers of the clustering process in FCA and GMM under the dataset of 2000 consumers are smaller than the corresponding iteration numbers under the dataset of 1500 consumers, which causes lower computational times in both the centralized and distributed methods. Hence, the computational time of the same PPD clustering method may not necessarily increase with the increase of the dataset size.

2) **Different Numbers of Retailers:** Different numbers of retailers may affect the correctness, convergence, and efficiency of our approach. Because we have already discussed how different numbers of retailers influence the convergence and efficiency of the proposed approach in Section V-G, here merely supplement the evaluations regarding the correctness of the framework.

We consider two cases: \( M = 10 \) and \( M = 20 \). For each case, we define three types of connections: original connections, less connections, and more connections. Detailed topologies are shown in Fig. 10. For \( M = 10 \), each retailer has access to 100 consumers. For \( M = 20 \), each retailer has access to 50 consumers. The relative errors of the PPD clustering methods are given in Table XII. Clearly, all the relative errors are less than \( 10^{-7} \), which demonstrates the correctness of the proposed framework under different numbers of retailers. In fact, the errors are determined by the convergence criterion of the PP-AAC algorithm, as given in Section V-A. That is the reason why the errors are similar. Although the errors are stable, the iteration numbers differ given different numbers of retailers. For the illustrations of iteration numbers as well as computational times under this case, please refer to Section V-G.

3) **Different Patterns for Allocation of Consumers to Retailers:** We first choose a dataset consisting of 1000 consumers. Then we split this dataset into 10 subsets and assign each of these subsets to a retailer. We do this four times to create four different allocations. The distributions for the different
allocations are shown in Fig. 12 indicating the number of consumers assigned to each of the 10 retailers in each of these allocations. Allocation 1 is the most imbalanced allocation, and allocation 4 is the balanced one. Besides, the communication topology we used for the 10 retailers is the one shown in Fig. 1.

Allocations of consumers will not affect the convergence rate of the PP-AAC algorithm, yet may influence the correctness of the proposed framework. The reasons are the same as discussed at the beginning of Section V-H.1. Therefore, the maximal computational time of the PPD clustering methods as well as their relative errors are listed in Table XIII. For comparison, we also provide the computational time of the centralized clustering methods in Table XIII. For comparison, we also provide the computational time of the centralized clustering methods in Table XIII. Note that, as the allocation becomes imbalanced, the local calculation burden of the retailer who has the most consumers also increases. This further increases the computational time of the PPD clustering methods. Nevertheless, the computational times of the PPD clustering methods in allocation 1 are still an order of magnitude smaller than the computational times of the centralized methods. Besides, under any allocation, the errors of the proposed PPD methods are negligible thereby demonstrating their correctness.

VI. CONCLUSION

In this article, we propose a privacy-preserving distributed clustering framework, which can directly modify the traditional k-means, FCA, and GMM clustering methods and provide privacy-preserving distributed variants. To achieve this, we first performed commonality analysis of the three clustering methods, and pointed out that the key of the clustering framework lies in calculating the summation of the retailers’ private information in a fully distributed and privacy-preserving way. Then we developed a PP-AAC algorithm with proven convergence to achieve the summation. Finally, we presented the privacy-preserving distributed clustering framework based on the proposed algorithm with theoretical privacy and complexity analyses.

The proposed PP-AAC algorithm converges faster than the privacy-preserving AC algorithm and the original AC algorithm. Besides, compared to the original AC algorithm and AAC algorithm, the proposed algorithm is privacy-preserving by introducing the exponentially decaying disturbance with zero-sum property into the shared information. Further, the proposed clustering framework can enable each retailer to obtain the exact residential load pattern identification of all consumers in their union dataset instead of only its own consumers. Thus, this framework could either help retailers design more diversified retail plans, or enable retailers to reach an informed decision whether they should stop seeking the cooperation that aims to identify more user types. Furthermore, the clustering framework not only protects every retailer’s confidentiality, but also greatly reduces the computation overhead of each retailer compared to the centralized method. Moreover, the clustering framework has strong robustness to random communication failures.

Note that there are still some open questions worth further investigations. The presented PP-AAC algorithm requires noise-free communication between retailers. If there constantly is undesired noise in the communication, the results of the PP-AAC algorithm would randomly walk and cannot converge to the accurate average consensus. Hence, how to achieve a noise-resisting, privacy-preserving, and accelerated average consensus algorithm deserves further investigations. Besides, this article mainly focuses on the privacy issues of retailers. The privacy concerns of consumers also deserve attention. For example, users’ load data might have different levels of importance and sensitivity. Accordingly, a privacy-preserving framework in which customers with higher sensitivity levels can receive more protection is worth of further investigations as well.

APPENDIX A

PROOF OF THE ZERO-SUM PROPERTY

First, since $\delta(t < 0) = 0$, we have $\theta_i(0) = \delta_i(0) - \delta_i(-1) = \delta_i(0)$. Thus,

$$
\sum_{t=0}^{\infty} \theta_i(t) = \delta_i(0) + \delta_i(1) - \delta_i(0) + \delta_i(2) - \delta_i(1)
$$

$$
+ \cdots + \lim_{t \to \infty} \delta_i(t) - \lim_{t \to \infty} \delta_i(t - 1)
$$

$$
= \delta_i(0) - \delta_i(0) + \delta_i(1) - \delta_i(1) + \delta_i(2) - \delta_i(2)
$$

$$
+ \cdots + \lim_{t \to \infty} \delta_i(t - 1) - \lim_{t \to \infty} \delta_i(t - 1)
$$

$$
+ \lim_{t \to \infty} \delta_i(t)
$$

$$
= \lim_{t \to \infty} \delta_i(t)
$$
Since $\delta(t)$ is randomly selected from $[-\frac{\sigma}{2} \beta^{t+1}, \frac{\sigma}{2} \beta^{t+1}]$, so
\[
\lim_{t \to \infty} |\delta(t)| \leq \frac{\sigma}{2} \lim_{t \to \infty} \beta^{t+1}
\] (31)
where $\delta > 0$ and $0 \leq \beta < 1$. Note that for $\forall \xi > 0$, as long as $t > \frac{\ln \xi}{\ln \beta}$, then we have
\[
t \ln \beta < \ln \xi \Leftrightarrow \beta^t < \xi
\]
This means that when $t$ approaches infinity, $\beta^t < \xi$ will hold sooner or later for $\forall \xi > 0$. Therefore,
\[
\lim_{t \to \infty} \beta^{t+1} = 0
\] (32)
Combining (31) and (32) yields
\[
\lim_{t \to \infty} |\delta(t)| = 0 \Rightarrow \lim_{t \to \infty} \delta(t) = 0
\]
Thus,
\[
\sum_{i=0}^{\infty} \theta_i(t) = \lim_{t \to \infty} \delta_i(t) = 0
\] (33)
which proves
\[
\sum_{i=1}^{M} \sum_{j=0}^{\infty} \theta_i(t) = 0
\]

**APPENDIX B**

**PROOF OF THEOREM 1**

First, we need to prove that $W^*$ is doubly stochastic, i.e., that
\[
1^T W^* = 1^T, \quad W^* 1 = 1
\] (34)
holds. Define $1 \in \mathbb{R}^{M \times 1}$ as a vector of all ones, then we have:
\[
1^T W^* = 1^T W + \alpha (1^T W - 1^T I)
\]
Since $W$ is a doubly stochastic matrix proved in [30], the following holds:
\[
1^T W = 1^T, \quad W 1 = 1
\]
Substitute $1^T W = 1^T$ into $1^T W^*$, we obtain
\[
1^T W^* = 1^T + \alpha (1^T - 1^T) = 1^T
\]
Similarly, we can obtain $W^* 1 = 1$ with the property that $W 1 = 1$.

Second, define $X^+(t) = [x_1^+(t), \ldots, x_M^+(t)]^T$, $X(t+1) = [x_1(t+1), \ldots, x_M(t+1)]^T$ and $\theta(t) = [\theta_1^+(t), \ldots, \theta_M^+(t)]^T$. Then we have the matrix form of the proposed PP-AAC algorithm:
\[
X(t+1) = W^* X^+(t)
\] (35)
\[
X^+(t) = X(t) + \theta(t)
\] (36)
In the linear dynamic system in (35), as long as $W^*$ is doubly stochastic, with the two aforementioned features of $\theta(t)$, the authors in [26] proved that (37) and (38) hold:
\[
\lim_{t \to \infty} \sum_{i=1}^{M} x_i(t) = \sum_{i=1}^{M} x_i(0)
\] (37)
\[
\lim_{t \to \infty} \max_{i} x_i(t) - \min_{i} x_i(t) = 0
\] (38)
Combining (37) and (38) yields (26).

**REFERENCES**

[1] J. Yang, J. Zhao, F. Wen, and Z. Dong, “A model of customizing electricity retail prices based on load profile clustering analysis,” *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3374–3386, Apr. 2019.

[2] M. Sun, I. Konstantelos, and G. Sirbac, “Cvine copula mixture model for clustering of residential electrical load pattern data,” *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2382–2393, May 2017.

[3] X. Lin, C. Clifton, and M. Zhu, “Privacy-preserving clustering with distributed em mixture modeling,” *Knowl. Inf. Syst.*, vol. 8, no. 1, pp. 68–81, 2005.

[4] G. J. Tsekouras, N. D. Hatzigirou, and E. N. Dialynas, “Two-stage pattern recognition of load curves for classification of electricity customers,” *IEEE Trans. Power Syst.*, vol. 22, no. 3, pp. 1120–1128, Aug. 2007.

[5] Y. Wang, Q. Chen, C. Kang, and Q. Xia, “Clustering of electricity consumption behavior dynamics toward big data applications,” *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2437–2447, Sep. 2016.

[6] G. Chicco, R. Napoli, and F. Piglione, “Comparisons among clustering techniques for electricity customer classification,” *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 933–940, May 2006.

[7] Z. Marques, A. de Almeida, A. M. de Deus, A. R. G. da Silva Paulo, and W. da Silva Lima, “A comparative analysis of neural and fuzzy cluster techniques applied to the characterization of electric load in substations,” in *Proc. IEEE/PES Trans. Distrib. Conf. Expo.*, Nov. 2004, pp. 908–913.

[8] B. Stephen, A. J. Mutanen, S. Galloway, G. Burt, and P. Järventausata, “Enhanced load profiling for residential network customers,” *IEEE Trans. Power Del.*, vol. 29, no. 1, pp. 88–96, Feb. 2014.

[9] T. Räisänen, D. Voukantissi, H. Niska, K. Karatzas, and M. Kolehmainen, “Data-based method for creating electricity use load profiles using large amount of customer-specific hourly measured electricity use data,” *Appl. Energy*, vol. 87, no. 11, pp. 3538–3545, 2010.

[10] G. Chicco, “Overview and performance assessment of the clustering methods for electrical load pattern grouping,” *Energy*, vol. 42, no. 1, pp. 68–80, 2012.

[11] S. Haben, C. Singleton, and P. Grindrod, “Analysis and clustering of residential customers energy behavioral demand using smart meter data,” *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 136–144, Jan. 2016.

[12] K. Li, Z. Ma, D. Robinson, and J. Ma, “Identification of typical building daily electricity usage profiles using Gaussian mixture model-based clustering and hierarchical clustering,” *Appl. Energy*, vol. 231, pp. 331–342, Dec. 2018.

[13] R. Li, Z. Wang, C. Gu, F. Li, and H. Wu, “A novel time-use tariff design based on Gaussian mixture model,” *Appl. Energy*, vol. 162, pp. 1530–1536, Jan. 2016.

[14] G. Peters, F. Crespo, P. Lingras, and R. Weber, “Soft clustering—Fuzzy and rough approaches and their extensions and derivatives,” *Int. J. Approx. Reason.*, vol. 54, no. 2, pp. 307–322, 2013.

[15] S. Samet, A. Amir, and L. Orozco-Barbosa, “Privacy preserving k-means clustering in multi-party environment,” in *Proc. SECURIT, 2007*, pp. 381–385.

[16] V. Banikandan, V. Porkodi, A. S. Mohammed, and M. Sivaram, “Privacy preserving data mining using threshold based fuzzy cmeans clustering,” *ICTACT J. Soft Comput.*, vol. 9, no. 1, pp. 1–7, 2018.

[17] C. Clifton, M. Kantarcioğlu, J. Vaidya, X. Lin, and M. Y. Zhu, “Tools for privacy preserving distributed data mining,” *ACM SIGKDD Explor. Newsletter*, vol. 4, no. 2, pp. 28–34, 2002.

[18] K. Xing, C. Hu, J. Yu, X. Cheng, and F. Zhang, “Mutual privacy preserving k-means clustering in social participatory sensing,” *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 2066–2076, Aug. 2017.

[19] K. L. Leematz, S. X. Lee, and G. J. McLachlan, “Corruption-resistant privacy preserving distributed EM algorithm for model-based clustering,” in *Proc. IEEE Trustcom/BigDataSE/ICCESS*, 2017, pp. 1082–1089.

[20] T. Su, F. Bao, J. Zhou, T. Takagi, and K. Sakurai, “Privacy-preserving two-party k-means clustering via secure approximation,” in *Proc. 21st Int. Conf. Adv. Inf. Netw. Appl. Workshops (AINAW)*, vol. 1, May 2007, pp. 385–391.

[21] S. Patel, S. Garasia, and D. Jinwala, “An efficient approach for privacy preserving distributed k-means clustering based on Shamir’s secret sharing scheme,” in *Proc. IFIP Int. Conf. Trust Manag.*, 2012, pp. 129–141.

[22] Z. Gheidi and Y. Challal, “Efficient and privacy-preserving k-means clustering for big data mining,” in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 791–798.
[23] F. Meskine and S. N. Bahloul, “Privacy preserving k-means clustering: A survey research,” Int. Arab J. Inf. Technol., vol. 9, no. 2, pp. 194–200, 2012.

[24] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, “Efficient privacy preserving k-means clustering,” in Proc. Pac.–Asia Workshop Intell. Security Informat., 2010, pp. 154–166.

[25] T. C. Aysal, B. N. Oreskikh, and M. J. Coates, “Accelerated distributed average consensus via localized node state prediction,” IEEE Trans. Signal Process., vol. 57, no. 4, pp. 1563–1576, Apr. 2009.

[26] J. He, L. Cai, P. Cheng, J. Pan, and L. Shi, “Consensus-based data-privacy preserving data aggregation,” IEEE Trans. Autom. Control, vol. 64, no. 12, pp. 5322–5329, Dec. 2019.

[27] S. Lloyd, “Least squares quantization in PCM,” IEEE Trans. Inf. Theory, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.

[28] K.-L. Wu, “Analysis of parameter selections for fuzzy c-means,” Pattern Recognit., vol. 45, no. 1, pp. 407–415, 2012.

[29] R. Singh, B. C. Pal, and R. A. Jabe, “Statistical representation of distribution system loads using Gaussian mixture model,” IEEE Trans. Power Syst., vol. 25, no. 1, pp. 29–37, Feb. 2010.

[30] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in Proc. 4th Int. Symp. Inf. Process. Sensor Netw. (IPSN), Apr. 2005, pp. 63–70.

[31] Y. Mo and R. M. Murray, “Privacy preserving average consensus,” IEEE Trans. Autom. Control, vol. 62, no. 2, pp. 753–765, Feb. 2017.

[32] Y. Mo and B. Sinopoli, “Communication complexity and energy efficient consensus algorithm,” IFAC Proc. Vol., vol. 43, no. 19, pp. 209–214, 2010.

[33] Irish Social Science Data Archive. (2012). Available: http://www.ucd.ie/isssda/data/commissionforenergyregulation/

[34] R. Lletí, M. Ortiz, L. Sarabia, and M. Sánchez, “Selecting variables for k-means cluster analysis by using a genetic algorithm that optimises the silhouettes,” Analytica Chimica Acta, vol. 515, no. 1, pp. 87–100, 2004.

[35] T. Zufferey, A. Ulbig, S. Koch, and G. Hug, “Forecasting of smart meter time series based on neural networks,” in Proc. Int. Workshop Data Anal. Renew. Energy Integr., 2016, pp. 10–21.

[36] P. Bholowalia and A. Kumar, “EBK-means: A clustering technique based on elbow method and k-means in WSN,” Int. J. Comput. Appl., vol. 105, no. 9, pp. 1–8, 2014.

[37] S. Lai, “Countries with the fastest Internet in the world 2019,” ATLAS and BOOTS, Feb. 2019. [Online]. Available: https://www.atlasandboots.com/remote-work/countries-with-the-fastest-internet-in-the-world/

Yi Wang (Member, IEEE) received the B.E. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1993 and 1998, respectively. From 1998 to 2001, he was a Postdoctoral Research Fellow with the Department of Electrical Engineering and Computer Science, University of Missouri Rolla, Rolla, MO, USA. From 2001 to 2002, he was a Senior Application Developer with ISO New England, Inc., Holyoke, MA, USA. Since 2009, he has been a Professor with the Department of Electrical Engineering, Tsinghua University. He is currently the Associate Director of the State Key Laboratory of Power System and Generation Equipment and the Director of the Research Center of Cloud Simulation and Intelligent Decision making with the Sichuan Energy Internet Research Institute, Tsinghua University. He has authored/coauthored more than 180 technical papers and one book, and holds 32 issued patents. His research interests include power system analysis and control, renewable energy generation, and smart grids.

Chen Shen (Senior Member, IEEE) received the B.E. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1993 and 1998, respectively. From 1998 to 2001, he was a Postdoctoral Research Fellow with the Department of Electrical Engineering and Computer Science, University of Missouri Rolla, Rolla, MO, USA. From 2001 to 2002, he was a Senior Application Developer with ISO New England, Inc., Holyoke, MA, USA. Since 2009, he has been a Professor with the Department of Electrical Engineering, Tsinghua University. He is currently the Associate Director of the State Key Laboratory of Power System and Generation Equipment and the Director of the Research Center of Cloud Simulation and Intelligent Decision making with the Sichuan Energy Internet Research Institute, Tsinghua University. He has authored/coauthored more than 180 technical papers and one book, and holds 32 issued patents. His research interests include power system analysis and control, renewable energy generation, and smart grids.

Gabriela Hug (Senior Member, IEEE) was born in Baden, Switzerland. She received the M.Sc. degree in electrical engineering and the Ph.D. degree from the Swiss Federal Institute of Technology, Zürich, Switzerland, in 2004 and 2008, respectively. After the Ph.D. degree, she worked with the Special Studies Group, Hydro One, Toronto, ON, Canada, and from 2009 to 2015, she was an Assistant Professor with Carnegie Mellon University, Pittsburgh, PA, USA. She is currently a Professor with the Power Systems Laboratory, ETH Zürich, Zürich, Switzerland. Her research is dedicated to control and optimization of electric power systems.

Mengshuo Jia (Graduate Student Member, IEEE) received the B.E. degree in electrical engineering from North China Electric Power University, Baoding, China, in 2016. He is currently pursuing the Ph.D. degree in electrical engineering with Tsinghua University, Beijing, China. He is also a Visiting Student Researcher with the Swiss Federal Institute of Technology, Zürich, Switzerland. His research interests include data analytics and privacy protection in power systems.