Abstract—In a multi-regional interconnected grid, the probabilistic load flow (PLF) of any region cannot be calculated individually but should consider the uncertainties introduced in other areas. Accordingly, the topologies, loads, and generations of every region are needed. Although the renewable generation data could be assumed as publicly known, some regional independent system operators (ISOs) still would not share important parameters with others. This motivates the development of a privacy-preserving distributed (PPD) PLF method. The challenge is to identify the mapping between the regional flows and uncertain power injections across regions without full information about the entire grid. The main idea of this paper is to respectively calculate the coefficient matrix and constant vector of the mapping: for the former, a PPD accelerated projection-based consensus algorithm is proposed; for the latter, a privacy-preserving accelerated average consensus algorithm is leveraged. Consequently, a PLF method is derived for each ISO to analytically obtain its regional joint PLF in a distributed way without sharing parameters — the key contribution of this paper. Experiments on the 118- and 1354-bus systems demonstrate that this method can generate the same results as the corresponding centralized method, and has satisfactory accuracy compared with frequently used PLF methods.

Index Terms—Distributed calculation, gaussian mixture model, joint probability distribution, privacy, probabilistic load flow.

NOMENCLATURE

| Abbreviation | Description |
|--------------|-------------|
| AAC | Accelerated average consensus. |
| AC-MC | AC-based Monte Carlo. |
| APC | Accelerated projection-based consensus. |
| CDF | Cumulative distribution function. |
| DC-MC | DC-based Monte Carlo. |
| DLPF | Decoupled linearized power flow. |
| GMM | Gaussian mixture model. |
| ISO | Independent system operator. |
| JSD | Jensen–Shannon divergence. |

PDF | Probability distribution function. |
PEM | Points estimation method. |
PLF | Probabilistic load flow. |
PPD | Privacy-preserving distributed. |

Function
- \( f(\cdot) \): Joint probability distribution of random power injections.
- \( g(\cdot) \): Joint probability distribution of nodal states.
- \( N_{k}(\cdot) \): \( k \)-th Gaussian distribution function.

Variable
- \( P_{W} \): Random nodal active power injections.
- \( Q_{W} \): Random nodal reactive power injections.
- \( V \): Vector of nodal voltage magnitudes.
- \( V_{n}, V_{m} \): Magnitude of the voltage at bus \( n \) and bus \( m \).
- \( \theta \): Vector of nodal voltage angles.
- \( \theta_{nm} \): Phase angle differences between the voltages at bus \( n \) and bus \( m \).
- \( x \): Vector of nodal states.
- \( x_{i} \): Vector of nodal states of region \( i \).
- \( X \): Solution of the augmented linear function.
- \( X_{i}(t) \): ISO \( i \)'s solution in \( t \)-th iteration of the APC algorithm.
- \( \hat{X}(t) \): Estimation of the solution \( X \) at iteration \( t \).
- \( y_{i}(t) \): Updated vector of ISO \( i \) in \( t \)-th iteration of the AAC algorithm.
- \( y_{ni}(t) \): \( n \)-th element of \( y_{i}(t) \).
- \( \delta_{i}(t) \): Updated vector of ISO \( i \) plus random noise in the \( t \)-th iteration of the AAC algorithm.
- \( \Delta_i \): Random noise of ISO \( i \) in the \( t \)-th iteration of the AAC algorithm.

Parameter
- \( A \): Nodal admittance matrix.
A_i \quad \text{Nodal admittance matrix related to the buses within region } i.

Φ_i \quad A^T_i (A_i A^T_i)^{-1} A_i.

Γ_i \quad \text{Projection matrix onto the nullspace of } A_i.

G \quad \text{Conductance matrix in } A.

B \quad \text{Susceptance matrix in } A.

b \quad \text{Nodal power injections.}

b(k) \quad k\text{-th observations of nodal power injections.}

B \quad \text{Matrix composed of the observations of } b.

B_i \quad \text{Matrix composed of the observations of } b \text{ related to the buses within region } i.

\bar{P}_S \quad \text{Deterministic nodal active power injections of } PV \text{ buses.}

\bar{P}_L \quad \text{Deterministic nodal active power injections of } PQ \text{ buses.}

\bar{Q}_L \quad \text{Deterministic nodal reactive power injections of } PQ \text{ buses.}

ρ_W \quad \text{Deterministic nodal active power injections contributed by } V_θ \text{ and } PV \text{ buses.}

σ_W \quad \text{Deterministic nodal reactive power injections contributed by } V_θ \text{ and } PV \text{ buses.}

P_ℓ \quad \text{Vector composed of each ISO’s nodal active power injection.}

\bar{P}_i \quad i\text{-th element of } P_ℓ.

\bar{P}_i^* \quad \text{Value randomly generated by ISO } i \text{ to represent } \bar{P}_i.

P_ℓ(k) \quad k\text{-th observation of } P_ℓ.

P_{W}(k) \quad k\text{-th observation of } P_{Wi}.

Q_{W}(k) \quad k\text{-th observation of } Q_{Wi}.

\Pi \quad \text{Matrix composed of observations of } P_W, Q_W, \text{ and } P_ℓ.

\alpha, \beta \quad \text{Coefficient matrix that maps random nodal power injections to nodal states.}

\alpha_i, \beta_i \quad \text{Coefficient matrix that maps random nodal power injections to nodal states of region } i.

γ \quad \text{Constant vector, i.e., contributions made by deterministic nodal power injections to nodal states.}

γ_i \quad \text{Constant vector, i.e., contributions made by deterministic nodal power injections to nodal states of region } i.

γ^{'n} \quad n\text{-th element of } γ.

γ^{'n} \quad n\text{-th element of } γ'.

ε \quad \text{Coefficient matrix in the augmented linear function.}

ε_i \quad \text{Element in row } n \text{ and column } i \text{ in } ε.

ψ_n \quad \text{Mean of } ε_i P_ℓ, \quad n \in Θ_i, \quad i = 1, \ldots, H.

Λ \quad \text{Matrix composed of the coefficient matrices and constant vector in the augmented linear function.}

ω_k \quad \text{Weight of the } k\text{-th Gaussian distribution function.}

μ_k \quad \text{Mean vector of the } k\text{-th Gaussian distribution in } f(P_W, Q_W).

Σ_k \quad \text{Covariance matrix of the } k\text{-th Gaussian distribution in } f(P_W, Q_W).

λ_k \quad \text{Mean vector of the } k\text{-th Gaussian distribution in } g(x_i).

Δ_k \quad \text{Covariance matrix of the } k\text{-th Gaussian distribution in } g(x_i).

ϕ, η \quad \text{Update parameters of the APC algorithm.}

Υ \quad \text{Mean of } Φ_i, \quad i = 1, \ldots, H.

\nu_{max} \quad \text{Maximal eigenvalue of } Υ.

\nu_{min} \quad \text{Minimal eigenvalue of } Υ.

r \quad \text{Convergence rate of the APC algorithm.}

L_i \quad \text{General form of the information of region } i.

G \quad \text{General form of the average of } L_i, \quad i = 1, \ldots, H.

d_i \quad \text{Degree of ISO } i, \text{i.e., the number of ISO } i\text{'s neighbors.}

W \quad \text{Metropolis weight matrix.}

W_{ij} \quad \text{Element in the row } i \text{ and column } j \text{ in the Metropolis weight matrix.}

W^* \quad \text{Accelerated Metropolis weight matrix.}

W^*_ij \quad \text{Element in the row } i \text{ and column } j \text{ in the Accelerated Metropolis weight matrix.}

\epsilon_{min} \quad \text{Minimal eigenvalue of } W.

\epsilon^2 \quad \text{Second largest eigenvalue of } W.

\varepsilon \quad \text{Acceleration parameter of } W^*.

I \quad H\text{-dimensional identity matrix.}

γ, ζ \quad \text{Boundary coefficients of random noise.}


I. INTRODUCTION

THE GROWING penetration of renewable generation leads to an increase in uncertainties in power system operation. Probabilistic load flow (PLF) methods can effectively evaluate the underlying operational risks [1]. However, in a multi-regional interconnected power system, multiple independent system operators (ISOs) perform regional operation [2]. Given the multi-regional interconnections and output coupling of renewable energy generation, the PLF of any region cannot be calculated individually but should take into account the uncertainties introduced in the other regions. Knowledge of the topology and parameters of the entire grid therefore is a prerequisite for the regional PLF calculation.

Nevertheless, each regional ISO only has access to the parameters within its area which in fact is a common situation. Although power grids are interconnected, the management of regional power grids is still conducted in an independent and closed way by the regional ISOs [3]. For instance, NYISO and ISO-NE do not have the information, e.g., the network topology, of each other [4], [5]. Besides, ISOs would consider their loads and network parameters as confidential information of individual areas [5]. Due to the requirements of privacy [6], [7], decision independence of each area [7], and political/technical reasons [8], collecting the aforementioned confidential information of ISOs may not be appropriate [6] or is unlikely to be implemented [7], [8]. As indicated in the PJM’s Operating Agreement, ISOs are prohibited from disclosing commercially sensitive information to others in general cases [9]. In this case, a distributed approach can provide the means for ISOs to obtain the desired regional PLFs without sharing their system data.

To date, a number of methods have been proposed for centralized PLF calculation but rarely for distributed PLF calculation. The centralized PLF methods can be divided into three categories: numerical, approximate, and analytical methods. Numerical methods rely on massive-scenario load flow calculation to extract the corresponding statistics. Although load flow model selection in numerical methods is usually AC-based, linear models are also used to reduce the computational burden [10].
The most common numerical method is Monte Carlo simulation [10]–[12]. To improve the efficiency of numerical methods, importance sampling [13], Latin hypercube sampling [14], and simple random sampling [15] have been adopted. Approximate methods on the other hand aim to estimate the statistics of load flows using limited samples from known locations. Selecting appropriate samples that keep sufficient uncertain information of random power injections determines the success of these methods [16]. Generally, approximate methods are also AC-based. The most frequently used approximate method is the point estimation method, including the two-point [17], multi-point [18], and fast [19] schemes. As the unscented transformation can improve performance when propagating the mean and covariance information through nonlinear operations [20], it has been widely used to develop new approximate methods [21], [22]. Lastly, analytical methods aim to convert the probability distribution of the random power injections into PLF using functional relationships. However, as the AC model is highly nonlinear with implicit solutions, the explicit relationship may be inaccessible. Thus, analytical methods are mainly based on approximate load flow models, such as linear [22], generalized polynomial chaos [23], and low-rank approximation models [24]. Among them, the linear model is the most common and serves as foundation for various analytical methods, including the convolution method [22], cumulant method with series expansion [25], and a method based on Gaussian mixture model (GMM) [26].

In the aforementioned centralized methods for PLF calculation, regional ISOs should share their parameters with the other ISOs to form the model of the entire grid. To avoid the need for this complete model, a distributed calculation strategy can be adopted. Some works have been conducted on distributed and deterministic load flow calculation, e.g., [3], [27] and [28]. In this paper, we take a further step by proposing a privacy-preserving distributed PLF method that focuses on strictly protecting the confidential information of ISOs under a satisfactory PLF accuracy. The distributed calculation only requires communication between adjacent ISOs, omitting centralized data collection. Moreover, privacy preservation is achieved as follows: 1) each ISO only needs its own parameters for calculation; 2) no ISO can deduce the parameters of others using the communicated information; and 3) no ISO can acquire the PLF of other regions. Note that the target of the privacy-preserving distributed PLF method does not lie on improving the accuracy of PLF, but on strictly protecting the confidential information of ISOs under a satisfactory PLF accuracy.

For proposing the privacy-preserving distributed PLF method, we choose a specific PLF algorithm. Specifically, we first select the state-independent voltage-angle decoupled linearized power flow (DLPF) model [29] as the load flow model, whose performance has been verified in [30], [31]. Then, to consider the correlations among renewable energy sources and variable load flows, we use GMM as the probability model to accurately represent multi-dimensional random variables subject to arbitrary distributions [32]. Further, the GMM-based PLF method presented in [26] is also adopted. Next, we combine the DLPF model and the GMM-based PLF method to obtain a base PLF method, which allows to analytically convert the probability distribution of the random injections into the joint PLF. The key to this base PLF method lies in finding the functional relationship between load flow in each region and random injected power over the entire grid.

We further modify the base PLF method to obtain its privacy-preserving distributed version. To this end, we first reformulate the centralized PLF calculation into a distributed framework for multiple ISOs. The distributed framework for each ISO consists of two parts: 1) calculation of the coefficient matrix in the functional relationship between its regional load flow and random injected power over the whole grid, and 2) calculation of the constant vector in the same relationship. For the first part, we propose a privacy-preserving distributed accelerated projection-based consensus (APC) algorithm. This algorithm can enable each ISO to obtain the coefficient matrix through local calculations and privacy-preserving neighboring communication. For the second part, we leverage the privacy-preserving accelerated average consensus (AAC) algorithm in [33] for each ISO to obtain the corresponding constant vector in a distributed and privacy-preserving manner. Based on these two algorithms, we derive the proposed privacy-preserving distributed PLF method. To the best of our knowledge, this is the first privacy-preserving distributed method for PLF calculation.

The contributions of this paper can be summarized as follows:
- We derive a distributed PLF framework for multiple ISOs. This framework unveils the requirements for a privacy-preserving distributed PLF method.
- We propose a privacy-preserving distributed PLF method. This method enables every ISO to analytically obtain only its own regional joint PLF via a fully distributed manner without revealing its parameters to other ISOs. Meanwhile, this method includes a novel privacy-preserving distributed APC algorithm.

It should be emphasized that, this paper does not make any policy recommendations. Instead, we just realize that under current regulations, information barriers may exist among regional ISOs, as discussed earlier. Therefore, this paper only focuses on the ISOs who currently do not have the necessary information about other areas. Most importantly, these ISOs cannot or do not want to share their information with others. Accordingly, we have designed a privacy-preserving distributed PLF method for such situations.

The rest of this paper is organized as follows. In Section II, we revisit the centralized PLF framework, which is then reformulated into a distributed version. In Section III, the privacy-preserving distributed APC algorithm is developed. Section IV describes the privacy-preserving AAC algorithm. Based on these two algorithms, the proposed privacy-preserving distributed PLF method is derived in Section V. Case studies are performed in Section VI. Section VII concludes this paper.

II. PROBLEM FORMULATION

In this section, we first revisit the centralized GMM-based PLF method with the DLPF model. Then, the privacy issues of the centralized PLF framework are revealed. To mitigate these issues, we formulate the distributed PLF framework.
A. Centralized PLF Framework

As the load flow model of the centralized PLF, the DLPP model assumes that $\cos \theta_{nm} \approx 1$ and neglects $(V_n - 1)^2$ and $(V_m - 1)$, because they are two orders of magnitude smaller than $V_n$ and $V_m$, respectively. Under these assumptions, the basic formulation of the DLPP model can be represented by [29]

$$Ax = b,$$  \hspace{1cm} (1)

where

$$A = \begin{bmatrix} B_{SS} - B_{SE} - G_{SE} \\ B_{LE} - B_{EE} - G_{EE} \\ B'_{WS} - B'_{WE} - G_{WE} \\ G_{SE} G_{EE} - B_{EE} \\ G_{WS} G_{WE} - B_{WE} \end{bmatrix} \in \mathbb{R}^{N \times N},$$  \hspace{1cm} (2)

$$x = \begin{bmatrix} \theta_S^T \\ \theta_W^T \\ V_L^T \\ V_W^T \end{bmatrix} \in \mathbb{R}^N, \hspace{1cm} (3)$$

and

$$b = \begin{bmatrix} P_S \\ P_L \\ Q_L \\ Q_W \end{bmatrix} + \begin{bmatrix} B'_{SR} G_{SR} - G_{SS} \\ B'_L G_{ER} - G_{LL} \\ B'_{WR} G_{WR} - G_{WS} \\ G_{ER} B_{RR} - B_{LL} \\ G_{WR} B_{WR} - B_{WS} \end{bmatrix} \begin{bmatrix} \theta_R \\ V_R \\ V_S \end{bmatrix}$$

$$= \begin{bmatrix} P_S^T P_L^T P_W^T + P_W^T Q_L^T Q_W^T + \sigma_W^T \end{bmatrix} \in \mathbb{R}^N. \hspace{1cm} (4)$$

In the above equations, $A$ and $B$ are the conductance and susceptance matrices, where superscript $t$ represents the matrix without shunt elements. $V$ and $\theta$ are the vectors of nodal voltage magnitudes and phase angles, respectively. For simplicity, we use ‘voltage’ to refer to $V$ and ‘angle’ to refer to $\theta$ in the following text. Subscripts $S$, $L$, and $W$ correspond to the $V_b$, $P_V$, and $P_Q$ buses, respectively, while $W$ corresponds to $M$ buses with random injected power. For example, $P_S$ consists of the given active power injections of $P_V$ buses, while the given reactive power injections of $P_Q$ buses are included in $Q_L$. For more details, please refer to [29].

Substituting (4) into (1), we can obtain $x$ as a linear function of random injected power values $P_W$ and $Q_W$: 

$$x = \alpha P_W + \beta Q_W + \gamma,$$  \hspace{1cm} (5)

where

$$\gamma = A^{-1} \begin{bmatrix} P_S^T P_L^T P_W Q_L^T Q_W^T + \sigma_W^T \end{bmatrix} \in \mathbb{R}^N.$$

and $\alpha \in \mathbb{R}^{N \times M}$ and $\beta \in \mathbb{R}^{N \times M}$ consist of the elements in $A^{-1}$ corresponding to $P_W$ and $Q_W$. After obtaining the functional relationship in (5), we can analytically compute the joint PLF of each region using the GMM-based PLF method in [26]. First, let $x_i \in \mathbb{R}^N$ be the vector that consists of the states of region $i$. Then, we have

$$x_i = \alpha_i P_W + \beta_i Q_W + \gamma_i,$$  \hspace{1cm} (7)

where $\alpha_i \in \mathbb{R}^{N \times M}$ is the submatrix of $\alpha$ whose rows correspond to the states of region $i$, and $\beta_i$ is analogously defined. In addition, $\gamma_i$ is a subvector of $\gamma$, whose elements also correspond to the states of region $i$. Note that $\alpha_i$ and $\beta_i$ are the coefficient matrices of the functional relationship in (7), while $\gamma_i$ is the constant vector.

Second, denote the GMM-based joint probability distribution of $P_W$ and $Q_W$ as

$$f(P_W, Q_W) = \sum_{k=1}^{K} w_k N_k(P_W, Q_W|\mu_k, \Sigma_k),$$

where $N_k(\cdot)$ is the $k$-th $2M$-dimensional Gaussian distribution with mean $\mu_k$ and covariance $\Sigma_k$. The weighting coefficient of $N_k(\cdot)$ is $w_k$.

Finally, using the functional relationship in (7) and the parameters in (8), the joint probability distribution of $x_i$ can be expressed as [26]

$$g(x_i) = \sum_{k=1}^{K} w_k N_k(x_i|\lambda_{ki}, \Delta_{ki}),$$

where

$$\lambda_{ki} = [\alpha_i^T \beta_i^T]^T \mu_k + \gamma_i$$

$$\Delta_{ki} = [\alpha_i^T \beta_i^T]^T \Sigma_k [\alpha_i \beta_i^T].$$  \hspace{1cm} (11)

Note that $g(x_i)$ is the joint probability distribution of all the states in region $i$ (i.e., joint PLF of this region).

B. Privacy Issues of Centralized PLF Framework

Based on the above PLF formulation, we know that once the functional relationship in (7) and joint probability distribution in (8) are known, the joint PLF in (9) can be derived directly.

Establishing the joint probability distribution in (8) requires historical data of the random injected power for training. In this paper, we assume that these data are publicly available, like in the case of electricity metadata generated from European renewable sources available at Eurostat. Meanwhile, the reactive power can be calculated from the active power by assuming a constant power factor [24]. Therefore, each ISO can directly obtain $f(P_W, Q_W)$ using a method such as the expectation–maximization algorithm for training [32].

Identifying the functional relationship in (7) requires the model of the entire grid (i.e., complete $A$ and $b$), because $\alpha_i$ and $\beta_i$ consist of the elements in $A^{-1}$. In addition, each element of $\gamma_i$ is the inner product of $b$ and a row in $A^{-1}$. However, each ISO does not have complete information on $A$ and $b$ but only accesses the following information:

- parameters of transmission lines within its region
- parameters of tie-lines linked to its region
- load and generation information within its region
- states of buses within its region
- states of ends of tie-lines linked to its region

Using the available information, each ISO can only form submatrices of $A$ and $b$, that is, ISO $i$ can only form $A_i \in \mathbb{R}^{N_i \times N}$ and $b_i \in \mathbb{R}^{N_i}$, where $A_i$ consists of the $N_i$ conductance and susceptance rows related to the buses within region $i$, and $b_i$ consists of $N_i$ power injection values related to the same buses. If we consider $H$ ISOs, the relationships among the above
submatrices are expressed as

\[ A_i = \bigcup_{i=1}^{H} A_i = \emptyset, \quad b_i = b, \quad \bigcap_{i=1}^{H} b_i = \emptyset. \]

In the centralized PLF framework, \( A_i \) and \( b_i \) of each ISO are collected to form the complete \( A \) and \( b \). This information sharing leads to privacy issues and may be refused by ISOs.

For ISO \( i \) to calculate its regional joint PLF while preserving privacy, we need to answer the following question: if ISO \( i \) knows only \( A_i \) and \( b_i \), how can it obtain only \( \alpha_i, \beta_i \), and \( \gamma_i \)? Next, we will devise a distributed PLF framework to answer this question.

C. Distributed Framework for Coefficient Matrix Calculation

Before devising the distributed framework, we introduce a vector \( P_t \) to ensure that each ISO cannot obtain the PLF of other regions. First, ISO \( i \) chooses an element \( \tilde{P}_i \) from \( \tilde{P}_L \) in (4), where \( \tilde{P}_i \in \mathbb{R} \) is a nodal active power injection within region \( i \) only available to ISO \( i \). Next, we use these \( H \) elements to form \( P_t \):

\[ P_t = [\tilde{P}_1, \ldots, \tilde{P}_H]^T \in \tilde{P}_L^T. \]  

After that, we reformulate (5) into an augmented form by introducing \( P_t \):

\[ x = \alpha P_W + \beta Q_W + \epsilon P_t + \gamma', \]  

where

\[ \gamma = \epsilon P_t + \gamma' \]  

and \( \epsilon \in \mathbb{R}^{N \times H} \) consists of the elements of \( A^{-1} \) corresponding to \( P_t \). Note that Section IV will further discuss why introducing \( P_t \) can prevent ISOs from obtaining the PLF of other regions.

For ISO \( i \) to obtain coefficient matrices \( \alpha_i \) and \( \beta_i \), we define \( \Lambda \) as follows:

\[ \Lambda = [\alpha \beta \epsilon \gamma'] \in \mathbb{R}^{N \times M}, \]  

where

\[ \tilde{M} = 2M + H + 1. \]

ISO \( i \) should extract \( \alpha_i \) and \( \beta_i \) from \( \Lambda \). To compute \( \Lambda \), the ISOs need to choose \( \tilde{M} \) publicly known observations of \( P_W \) and \( Q_W \), where \( P_W(k) \in \mathbb{R}^M \) is the \( k \)-th observation of \( P_W \). Besides, the ISOs also need to generate \( \tilde{M} \) artificial and publicly known data segments of \( P_t \), where the \( k \)-th data segment is represented by \( P_t(k) \). Substituting \( P_W(k), Q_W(k), \) and \( P_t(k) \) into \( b \) generates \( b(k) \). Then, the following equation holds:

\[ \Lambda \Pi = A^{-1} B, \]

where

\[ \Pi = \begin{bmatrix} P_W(1) \cdots P_W(\tilde{M}) \\ Q_W(1) \cdots Q_W(\tilde{M}) \\ P_t(1) \cdots P_t(\tilde{M}) \\ 1 \cdots 1 \end{bmatrix} \in \mathbb{R}^{\tilde{M} \times \tilde{M}}, \]  

\[ B = \begin{bmatrix} b(1) \cdots b(\tilde{M}) \end{bmatrix} \in \mathbb{R}^{N \times \tilde{M}}. \]

Given that \( \Pi \) is available to all ISOs, if each ISO has the results of the right-hand side of (17), it could then compute \( \Lambda \). Computing the right-hand side of (17) is essentially calculating \( \mathcal{X} \in \mathbb{R}^{N \times M} \) in

\[ A \mathcal{X} = B. \]

However, similar to \( A \) and \( b \), ISO \( i \) can only form a submatrix of \( B \), i.e., \( B_i \in \mathbb{R}^{N_i \times M} \), which consists of \( N_i \) rows of injected power values related to the buses within region \( i \). Therefore, ISO \( i \) mathematically faces the problem of acquiring \( \mathcal{X} \) in

\[ \left[ A_t^T \cdots A_H^T \right]^T \mathcal{X} = \left[ B_t^T \cdots B_H^T \right]^T. \]

Remark 1: ISO \( i \) calculating coefficient matrices \( \alpha_i \) and \( \beta_i \) is essentially solving (21) when only \( A_i \) and \( B_i \) are available. After solving (21), ISO \( i \) can then obtain \( \Lambda \) by (17) and further extract \( \alpha_i \) and \( \beta_i \) from \( \Lambda \). Thus, a privacy-preserving distributed PLF method should guarantee that every ISO solves (21) in a privacy-preserving and fully distributed manner.

D. Distributed Framework for Constant Vector Calculation

To allow ISO \( i \) to obtain the constant vector \( \gamma_i \), we first define the index set of its states as \( \Theta_i \). Then, we define \( \gamma_n \) as the \( n \)-th element of \( \gamma \). Clearly, \( \gamma_i \) consists of \( \gamma_n \) (\( n \in \Theta_i \)). Thus, based on (14), \( \gamma_n \) can be calculated as

\[ \gamma_n = \gamma'_n + H \psi_n, \quad n \in \Theta_i, \]

where

\[ \psi_n = \frac{1}{H} \sum_{i=1}^{H} \epsilon_n i \tilde{P}_i, \quad n \in \Theta_i. \]

In (22), \( \gamma_n \) is the \( n \)-th element of \( \gamma' \) and \( \epsilon_n \) is the element in row \( n \) and column \( i \) in \( \epsilon \). Note that both \( \gamma' \) and \( \epsilon \) are known by all ISOs after they solved (21) and further obtain \( \Lambda \) in (17). Thus, if ISO \( i \) has \( \psi_n \) (\( n \in \Theta_i \)) in (23), it can then compute \( \gamma \) by (22). However, ISO \( i \) only knows \( \tilde{P}_i \). In this case, ISO \( i \) needs to solve the problem of how to acquire \( \psi_n \) (\( n \in \Theta_i \)) when it only knows \( \tilde{P}_i \).

Remark 2: Calculating the constant vector \( \gamma_i \) for ISO \( i \) is essentially computing \( \psi_n (n \in \Theta_i) \) in (23) when the ISO only knows \( \tilde{P}_i \). Thus, a privacy-preserving distributed PLF method should enable each ISO to calculate (23) in a privacy-preserving and fully distributed manner.

III. PRIVACY-PRESERVING DISTRIBUTED APC

To enable every ISO to solve (21) in a privacy-preserving and fully distributed manner, we propose a privacy-preserving and fully distributed APC algorithm derived from the conventional APC algorithm [34].

A. APC Algorithm

The APC algorithm [34] aims to solve a system of linear equations that are partitioned such that each party only accesses a disjoint subset of the full set of equations and variables. In our case, the party is the ISO, and the system of linear equations is (21). To obtain \( \mathcal{X} \), ISO \( i \) first finds an initial solution \( \mathcal{X}_i(0) \in \mathbb{R}^{N \times M} \) that is available to all ISOs.
$\mathbb{R}^{N \times M}$ of $A_i X_i(0) = B_i$, among infinitely many solutions. This paper adopts the least squares solution as the initial solution. Then, ISO $i$ updates its initial solution via the APC algorithm as follows:

$$X_i(t + 1) = X_i(t) + \varphi \Gamma_i \left[ X(t) - X_i(t) \right],$$  \hspace{1cm} (24)

where $\Gamma_i \in \mathbb{R}^{N \times N}$ is the projection matrix onto the nullspace of $A_i$, as given in:

$$\Gamma_i = I - A_i^T (A_i A_i^T)^{-1} A_i = I - \Phi_i,$$  \hspace{1cm} (25)

$\bar{X}(t) \in \mathbb{R}^{N \times M}$ is the estimation of the global solution $X$ at iteration $t$, as given in:

$$\bar{X}(t) = (1 - \eta) \bar{X}(t - 1) + \frac{\eta}{H} \sum_{i=1}^{H} X_i(t)$$  \hspace{1cm} (26)

and $I$ is the $N$-dimensional identity matrix. Besides, the optimal parameters of $\varphi$ and $\eta$ in (24) and (26) are the solutions of the following equations:

$$\begin{align*}
u_{\text{max}} \varphi \eta &= (1 + \sqrt{(\varphi - 1)(\eta - 1)})^2, \\
u_{\text{min}} \varphi \eta &= (1 - \sqrt{(\varphi - 1)(\eta - 1)})^2, \\
\end{align*}$$  \hspace{1cm} (27)

where $\nu_{\text{max}}$ and $\nu_{\text{min}}$ are the maximal and minimal eigenvalues of $\Psi \in \mathbb{R}^{N \times N}$:

$$\Psi = \frac{1}{H} \sum_{i=1}^{H} \Phi_i.$$  \hspace{1cm} (28)

Using the iterative process defined by (24)-(28), $X_i(\forall i)$ converges to the global solution $X$ with the convergence rate [34]:

$$r = 1 - \frac{2 \nu_{\text{max}}}{\nu_{\text{min}}}.$$  \hspace{1cm} (29)

Note that computing (26) and (28) requires average calculations among all ISOs. For clarity, we summarize these average calculations as

$$G = \frac{1}{H} \sum_{i=1}^{H} L_i,$$  \hspace{1cm} (30)

where $L_i$ represents $X_i(t)$ or $\Phi_i$, while $G$ represents $\bar{X}(t)$ or $\Psi$ correspondingly. To compute (30), the authors in [34] use a center for data collection, calculation, and broadcasting. Once (30) is obtained by each ISO, other calculations of the APC algorithm can be performed independently.

B. Privacy-Preserving Distributed APC Algorithm

To develop a privacy-preserving and fully distributed APC algorithm without centralized data collection, each ISO needs to be able to compute (30) by local calculations and privacy-preserving communication with its neighbors. To this end, we use the privacy-preserving AAC algorithm proposed in [33].

The privacy-preserving AAC algorithm is based on graph theory. Specifically, all ISOs should form a connected and publicly available graph consisting of $H$ nodes and some edges, where each edge between a pair of nodes represents bidirectional noiseless communication between the two corresponding ISOs. The neighborhood of ISO $i$, denoted by $\Omega_i$, is defined as an index set of ISOs directly connected to ISO $i$. Meanwhile, the degree of ISO $i$ is represented by $d_i$, which is the number of ISO $i$’s neighbors. The graph should guarantee that if $j \in \Omega_i$, then $\Omega_j \cap \Omega_i$, as described in [33]. Under this graph, each ISO computes the elements in the so-called Metropolis weight matrix as follows:

$$W_{ij} = \begin{cases} \\
1 & \text{if } j \in \Omega_i \\
1 - \sum_{k \in \Omega_i} W_{ik} & \text{if } i = j \\
0 & \text{Otherwise} \\
\end{cases}$$  \hspace{1cm} (31)

Using the Metropolis weight matrix $W \in \mathbb{R}^{H \times H}$, each ISO further computes the accelerated Metropolis weight matrix $W^* \in \mathbb{R}^{H \times H}$ as follows:

$$W^* = (1 + \varepsilon)W - \varepsilon I,$$  \hspace{1cm} (32)

where $I$ is the $H$-dimensional identity matrix, and

$$\varepsilon = \frac{\lambda_{\text{min}} + \lambda_2}{2} - \lambda_{\text{min}} - \lambda_2$$  \hspace{1cm} (33)

is the optimal parameter for acceleration. Moreover, $\lambda_{\text{min}}$ is the minimal eigenvalue of $W$, and $\lambda_2$ is its second largest eigenvalue.

After obtaining $W^*$, ISO $i$ sets $y_i(0) = L_i$ and updates $y_i(t)$ through the privacy-preserving AAC algorithm in

$$y_i(t + 1) = W_{ii} y_i^+(t) + \sum_{j \in \Omega_i} W_{ij} y_j^+(t),$$  \hspace{1cm} (34)

where $y_i^+(t)$ represents the true value of $y_i(t)$ plus some noise:

$$y_i^+(t) = y_i(t) + \delta_i(t) - \delta_i(t - 1),$$  \hspace{1cm} (35)

and noise $\delta_i(t)$ is randomly selected from $[-\frac{\sqrt{t}}{2}, \frac{\sqrt{t}}{2}]$ by ISO $i$ with $\varphi > 0$ and $\varepsilon \in [0, 1]$. By the iterative process, $y_i$ converges to the average value of $L_i$ for $i = 1, \ldots, H$ as follows:

$$\lim_{t \to \infty} y_i(t) = \frac{1}{H} \sum_{i=1}^{M} L_i.$$  \hspace{1cm} (36)

For a detailed proof, please refer to [33].

Based on the privacy-preserving AAC algorithm, we derive the privacy-preserving and fully distributed APC algorithm, which is detailed in Algorithm 1.

It should be emphasized that, since $\varphi$ and $\eta$ in step 4 do not contain confidential information, ISOs could also share these values with others to guarantee that the parameters they use are absolutely the same. Besides, in the privacy-preserving distributed APC algorithm, each ISO only needs local calculations (steps 2, 4, 7, and 11) and neighboring communications (steps 3 and 8). Thus, the proposed algorithm is fully distributed and no center for data collection is required. Meanwhile, the only information that ISO $i$ shares with its neighbors is $y_i^+(t)$, which is masked by random noise. Therefore, the neighbors cannot deduce any private information from $y_i^+(t)$, resulting in strict privacy protection despite communication.
Algorithm 1: Privacy-preserving Distributed APC Algorithm for ISO $i$ (\$i\$).

**Input:** $A_i$ and $B_i$.

**Output:** Solution $\mathcal{X}$ of (21)

1. $t = 0$;
2. Compute $X_i(t)$ from $A_i X_i(t) = B_i$;
3. Obtain $\overline{X}(t)$ in (26) and $Y(t)$ in (28) by (34)-(36);
4. Compute $\Gamma_i$ by (25) and $\varphi, \eta$ by (27);
5. $t = t + 1$;
6. while APC convergence criterion is not met do

   7. Update $X_i(t)$ by (24);
   8. Obtain $\overline{X}(t)$ in (26) by (34)-(36);
   9. $t = t + 1$;
10. end
11. Return $\mathcal{X} = X_i(t)$;

IV. PRIVACY-PRESERVING AAC WITH FAKE INPUT

To enable ISO $i$ (\$i\$) to obtain $\psi_n$ in (23) via a privacy-preserving and fully distributed fashion, we note that (23) is mathematically equivalent to (30). Thus, ISO $i$ (\$i\$) can still use the privacy-preserving AAC algorithm in (34) to calculate (23).

However, to guarantee that ISO $i$ only obtains $\gamma_n (n \in \Theta_i)$ after performing the privacy-preserving AAC algorithm, we introduce a fake value $\hat{P}_i$ here. The fake value $\hat{P}_i$ is randomly generated and only available to ISO $i$. Using this fake value, ISO $i$ sets $y_{ni}(0) (n = 1, \ldots, N)$ as follows:

$$y_{ni}(0) = \begin{cases} \epsilon_{ni} \hat{P}_i, & n \notin \Theta_i \\ \epsilon_{ni} P_i, & n \in \Theta_i \end{cases} \quad (37)$$

where $y_{ni}(0)$ is the $n$-th element of $y_i(0)$ in (34). After performing the privacy-preserving AAC algorithm, $y_{ni}(t)$ converges to the following values for $n = 1, \ldots, N$:

$$\lim_{t \to \infty} y_{ni}(t) = \begin{cases} \frac{1}{H} \epsilon_{ni} \hat{P}_i + \frac{1}{H} \sum_{k=1, k \neq j}^{H} \epsilon_{nk} \hat{P}_k, & n \in \Theta_j \\ \frac{1}{H} \epsilon_{ni} P_i + \frac{1}{H} \sum_{k=1, k \neq i}^{H} \epsilon_{nk} \overline{P}_k, & n \in \Theta_i \end{cases} \quad (38)$$

where $\hat{P}_j$ is the random fake value chosen by any ISO $j$ ($j \neq i$). Clearly, ISO $i$ can compute the real $\psi_n (n \in \Theta_i)$ from (38) as

$$\psi_n = \lim_{t \to \infty} y_{ni}(t) - \frac{1}{H} \epsilon_{ni} \hat{P}_i + \frac{1}{H} \epsilon_{ni} P_i \quad (39)$$

Then, ISO $i$ acquires $\gamma_n (n \in \Theta_i)$ using (22) (i.e., $\gamma_i$). However, as ISO $i$ does not know $\hat{P}_j$ and $P_j$ of ISO $j$, it cannot deduce $\psi_n (n \in \Theta_j)$ from (38). Therefore, every $\gamma_n (n \in \Theta_i)$ remains unknown to ISO $i$, disabling it to derive the PLF of another region.

V. PRIVACY-PRESERVING DISTRIBUTED PLF METHOD

Based on the privacy-preserving distributed APC and privacy-preserving AAC algorithms, we propose the privacy-preserving distributed PLF method in Algorithm 2.

There are two points that should be noted about the proposed algorithm. First, except for steps 2 and 7 in Algorithm 2, the other steps are local calculations that can be conducted by each ISO. In addition, steps 2 and 7 are privacy-preserving distributed calculations. Therefore, Algorithm 2 provides a privacy-preserving distributed method that allows each ISO to only obtain the PLF of its own region through local calculations and privacy-preserving communication with its neighbors. No ISO can deduce private information of the other ISOs. Second, $g(x_i)$ is a joint probability distribution that characterizes the stochastic features of all states in region $i$ considering their correlations. This distribution can provide a simultaneous and exact evaluation for the probability of multiple states being out of bound [26]. Moreover, deriving the marginal or conditional probability distribution of a single state from $g(x_i)$ is also straightforward [35].

VI. CASE STUDY

A. Settings

First, we chose wind farms to represent the random power injections. The data from the Eastern Wind Integration Dataset published by the US National Renewable Energy Laboratory was used to simulate historical data of wind farms.

Besides, we used a modified IEEE 118-bus system and a modified 1354-bus system representing the European transmission grid [36] for conducting case studies. We randomly divided the two systems into nine regions. The 118-bus system is detailed in Fig. 1. Limitations of space prevent us from presenting the detailed structure of the 1354-bus system. Thus, we summarized the information of the 1354-bus system we used in Table I. Note that Sections VI-B, VI-C, and VI-D are based on the 118-bus system, while Section VI-E focuses on the simulations for the 1354-bus system.

| Region | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|---|---|---|---|---|---|---|---|
| Number of Buses | 365 | 189 | 261 | 39 | 154 | 141 | 148 | 68 | 49 |
| Number of Branches | 446 | 250 | 370 | 48 | 223 | 223 | 208 | 88 | 65 |
| Number of Wind Farms | 7 | 3 | 6 | 1 | 2 | 1 | 1 | 1 | 2 |
Furthermore, under different levels of power injections, the power flow results obtained from the DLPF model show small and approximate constant deviations compared to the real AC results [37]. So for better performance, one can estimate the deviations using a given set of power injections and then add the deviations to the corresponding results of the DLPF model to obtain a complemented version. In the following case studies, all the evaluated methods that embed the DLPF model use the complemented version.

Moreover, all the experiments were coded in MATLAB and run on an i5-7267 U 3.1 GHz processor with 8 GB RAM.

B. Correctness Verification

In this paper, correctness means that the calculation results of the distributed algorithm and its corresponding centralized algorithm should be consistent.

To verify the correctness of the proposed privacy-preserving distributed APC algorithm, we first computed \( \mathbf{X}_{\text{real}} = \mathbf{A}^{-1} \mathbf{B} \) in (20) in a centralized way and used the result as benchmark. Then, we used the proposed algorithm to solve \( \mathbf{A} \mathbf{X} = \mathbf{B} \) in a distributed, privacy preserving fashion and to obtain \( \mathbf{X}_i(t) \) for each ISO \( i \) (\( \forall i \)). Thereafter, we computed the average values of the relative errors of all the elements in \( \mathbf{X}_i(t) \) (\( \forall i \)) compared to \( \mathbf{X}_{\text{real}} \). The corresponding results are listed in Table II. As can be observed, the relative errors are negligibly small for the chosen stopping criterion of the iterative algorithm, indicating the correctness of the proposed privacy-preserving distributed APC algorithm.

To verify the correctness of the proposed privacy-preserving distributed PLF method, we used the centralized GMM-based PLF as benchmark. Then, we utilized the proposed PLF method for each ISO to obtain the PLF of its region. Both methods use the DLPF as the power flow model. After that, we used the Jensen–Shannon divergence (JSD) to measure the differences between the probability distributions obtained from the benchmark and proposed method. Note that the non-negative JSD between two probability distributions is bounded by 1, and smaller divergence indicates smaller differences between two probability distributions. As each region has its own probability distributions for its nodal voltages, angles, and branch flows, we computed the average and the maximal JSDs between the distributions built by the benchmark and proposed method for each region. The corresponding results are listed in Table III, where the bottom three rows are the maximums. Clearly, the average JSDs of all regions are negligible. Meanwhile, the maximal JSDs are all below \( 9.26 \times 10^{-5} \ll 1 \). Hence, the probability distributions obtained from the proposed method are basically the same as those obtained from the centralized method, verifying the correctness of the proposed distributed method.

For a more detailed comparison, we illustrate the voltage probability distribution functions (PDFs) of the buses with wind farms connected in Fig. 2, because these nodal voltages have large uncertainties. In the figure, legend ‘Proposed’ represents the PDFs obtained from the proposed method, and ‘Centralized’ represents those obtained from the benchmark. The PDFs obtained from the benchmark and proposed method agree. We also illustrate the 2D joint PDF of two randomly chosen branch flows in Fig. 3. Again, the joint PDFs obtained from the benchmark and proposed method agree.

C. Effective Verification

Effective verifications aim to demonstrate whether the PLFs generated by the proposed algorithm are close to the results of the corresponding Monte Carlo algorithm.
TABLE IV
EXPECTED VALUE ERROR USING DLFP-BASED MONTE CARLO METHOD AS BENCHMARK

| Region | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|---|---|---|---|---|---|---|---|
| Voltage (×10^-3) | 1.47 | 0.19 | 0.30 | 2.26 | 0.22 | 2.27 | 3.04 | 2.07 | 1.00 |
| Angle (×10^-4) | 4.76 | 3.73 | 5.11 | 3.44 | 2.39 | 1.99 | 2.67 | 2.71 | 3.43 |
| Flow (×10^-3) | 0.82 | 0.15 | 1.60 | 1.50 | 0.52 | 1.50 | 0.24 | 1.10 | 0.93 |

TABLE V
EXPECTED VALUE ERROR USING AC-BASED MONTE CARLO METHOD AS BENCHMARK (DC-MC: DC-BASED MONTE CARLO)

| Region | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|---|---|---|---|---|---|---|---|
| DC-MC (Angle) | 0.012 | 0.003 | 0.011 | 0.062 | 0.039 | 0.031 | 0.031 | 0.038 | 0.044 |
| Proposed (Angle) | 0.001 | 0.002 | 0.001 | 0.002 | 0.002 | 0.003 | 0.006 | 0.005 | 0.006 |
| DC-MC (Flow) | 0.005 | 0.037 | 0.191 | 0.085 | 0.085 | 0.375 | 0.227 | 0.091 | 0.050 |
| Proposed (Flow) | 0.003 | 0.002 | 0.013 | 0.004 | 0.003 | 0.016 | 0.006 | 0.009 | 0.006 |

We first compared the proposed method with the DLFP-based Monte Carlo method. Using this Monte Carlo method as benchmark, we calculated the average relative errors of the expected PLF values of each region obtained from the proposed method. Table IV shows that the relative errors of each region are again negligible. Thus, the Monte Carlo and proposed methods have comparable performance.

We also used the results of the AC-based Monte Carlo method as benchmark and compared the performances of the proposed method and the DC-based Monte Carlo method. The DC-based Monte Carlo method is the benchmark of the GMM-based PLF method in [26]. The average relative errors of the expected values in each region using the above mentioned methods are listed in Table V, with the minimal values being highlighted in bold. The relative errors of the proposed method are one to two orders of magnitude smaller than those of the DC-based Monte Carlo method. Furthermore, we used the JSD to measure the differences between the probability distributions obtained from the benchmark and the two evaluated methods. Then, we summarized the average JSD of each state and corresponding benchmark, obtaining the results depicted in Fig. 4. The JSDs of the DC-based Monte Carlo method exceed 0.1, even reaching 0.15, while the JSD of the proposed method remains below 0.05. Overall, the proposed method is superior in terms of expected value error and JSD.

For more intuitive comparisons, Fig. 5 shows the marginal cumulative distribution functions (CDFs) of the active branch flows on a number of 345 kV transmission lines. There are clear differences between the benchmark CDF and those obtained from the DC-based Monte Carlo method. However, the CDFs obtained from the proposed method suitably agree with those obtained from the benchmark. Moreover, Fig. 6 shows the joint CDFs of the branch flows on the 345 kV transmission lines 65-68 and 64-65. Again, the joint CDF obtained from the proposed method show better agreement with the benchmark than that obtained from the DC-based Monte Carlo method.

We would like to emphasize that, the above verifications are not intended to prove that our method is superior to all the existing centralized methods in terms of the PLF accuracy. These verifications can only demonstrate that our approach has satisfactory accuracy compared with the frequently used centralized PLF methods, i.e., the DC- and AC-based Monte Carlos. Such verification is sufficient as our actual focus is on enhancing a distributed PLF method to include the feature of being privacy-preserving, i.e., the goal is not to provide the most accurate PLF method which is normally what centralized PLF methods desire, but to strictly protect ISOs’ confidential information with a satisfactory PLF accuracy.

D. Efficiency Comparison

To verify the efficiency of the proposed privacy-preserving distributed PLF method, we measured the computational times
of the evaluated methods and listed the results in Table VI. Note that all the Monte Carlo methods require two steps for calculation: 1) running $10^6$ power flow simulations and 2) obtaining the PLF of states in each region using the corresponding samples. The distributed method was coded in a serial structure and the computation time is given for a serial execution of the regions.

As Table VI indicates, the proposed method requires about 37 seconds to obtain the PLF of a region, which is significantly faster than the Monte Carlo methods. Compared with the centralized GMM-based PLF method, the proposed method approximately costs an extra 22 seconds. This extra time could be regarded as the price of protecting ISOs' confidentiality. ISOs have to sacrifice computational efficiency in order to realize such protection.

Nevertheless, we still want to emphasize that the above efficiency comparisons might not be entirely appropriate because the methods for comparisons (Monte Carlos and the GMM-based PLF) are all centralized PLF methods. The JSD is again used to measure the differences between the probability distributions obtained by the benchmark and proposed methods. As each region has its own probability distributions for nodal voltages, angles, and branch flows, we computed the maximal JSDs between the distributions built by the benchmark and proposed methods for each region, and listed them in Table VII. Note that there are voltages and flows that are nearly constant, thus we did not take them into account here and below. Specifically, we excluded voltages whose variances are less than $10^{-10}$ and flows whose variances are less than $10^{-7}$.

In Table VII, the maximal JSDs of all regions are negligible. A lower JSD indicates a smaller difference between distributions. Therefore, the JSDs in Table VII demonstrate the correctness of the proposed method for the 1354-bus system.

For effective verifications, we chose three frequently used centralized PLF methods to illustrate the satisfactory accuracy of our approach. These methods include the classic two points estimation method (PEM), the DC-based Monte Carlo method, and the AC-based Monte Carlo method.

We first used the DLPF-based Monte Carlo method as a benchmark for evaluations. Based on this benchmark, we calculated the average relative errors of the expected voltages and flows in each region obtained from the proposed method and the PEM. Both of the evaluated methods use the DLPF as the power flow model. The JSD is again used to measure the differences between the probability distributions obtained by the benchmark and proposed methods. As each region has its own probability distributions for nodal voltages, angles, and branch flows, we computed the maximal JSDs between the distributions built by the benchmark and proposed methods for each region, and listed them in Table VII. Note that there are voltages and flows that are nearly constant, thus we did not take them into account here and below. Specifically, we excluded voltages whose variances are less than $10^{-10}$ and flows whose variances are less than $10^{-7}$.

In Table VII, the maximal JSDs of all regions are negligible. A lower JSD indicates a smaller difference between distributions. Therefore, the JSDs in Table VII demonstrate the correctness of the proposed method for the 1354-bus system.

For effective verifications, we chose three frequently used centralized PLF methods to illustrate the satisfactory accuracy of our approach. These methods include the classic two points estimation method (PEM), the DC-based Monte Carlo method, and the AC-based Monte Carlo method.

We first used the DLPF-based Monte Carlo method as a benchmark for evaluations. Based on this benchmark, we calculated the average relative errors of the expected voltages and flows in each region obtained from the proposed method and the PEM. Both of the evaluated methods use the DLPF as the power flow model. The JSD is again used to measure the differences between the probability distributions obtained by the benchmark and proposed methods. As each region has its own probability distributions for nodal voltages, angles, and branch flows, we computed the maximal JSDs between the distributions built by the benchmark and proposed methods for each region, and listed them in Table VII. Note that there are voltages and flows that are nearly constant, thus we did not take them into account here and below. Specifically, we excluded voltages whose variances are less than $10^{-10}$ and flows whose variances are less than $10^{-7}$.

In Table VII, the maximal JSDs of all regions are negligible. A lower JSD indicates a smaller difference between distributions. Therefore, the JSDs in Table VII demonstrate the correctness of the proposed method for the 1354-bus system.

For effective verifications, we chose three frequently used centralized PLF methods to illustrate the satisfactory accuracy of our approach. These methods include the classic two points estimation method (PEM), the DC-based Monte Carlo method, and the AC-based Monte Carlo method.

We first used the DLPF-based Monte Carlo method as a benchmark for evaluations. Based on this benchmark, we calculated the average relative errors of the expected voltages and flows in each region obtained from the proposed method and the PEM. Both of the evaluated methods use the DLPF as the power flow model. Note that the well known Cholesky decomposition was leveraged to deal with the correlation of wind power before performing the PEM. The relative errors are shown in Table VIII, where the minimal values are highlighted in bold. As can be observed, all the relative errors of the proposed method are smaller than the errors of the PEM, thereby demonstrating the satisfactory accuracy of the proposed method.

We then used the AC-based Monte Carlo method as a benchmark for evaluations. We calculated the average relative errors of the proposed method, PEM, and DC-based Monte Carlo method
TABLE VIII
EXPECTED VALUE ERROR USING DL-BASED MONTE CARLO METHOD AS BENCHMARK IN 1354-BUS SYSTEM (PEM: POINT ESTIMATION METHOD, DC-MC: DC-BASED MONTE CARLO)

| Region | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|---|---|---|---|---|---|---|---|
| PEM    | 6.460 | 1.310 | 2.695 | 13.90 | 4.102 | 2.189 | 2.325 | 6.445 | 19.44 |
| Proposed | 0.288 | 0.100 | 0.175 | 0.334 | 0.071 | 0.093 | 0.023 | 0.012 | 0.022 |

Flow × 10⁻¹

| Region | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|---|---|---|---|---|---|---|---|
| PEM    | 5.934 | 2.147 | 4.840 | 18.83 | 23.61 | 1.414 | 1.420 | 1.976 | 1.646 |
| Proposed | 0.035 | 0.014 | 0.029 | 0.041 | 0.009 | 0.009 | 0.015 | 0.009 | 0.003 |

TABLE IX
EXPECTED VALUE ERROR USING AC-BASED MONTE CARLO METHOD AS BENCHMARK IN 1354-BUS SYSTEM (PEM: POINT ESTIMATION METHOD, DC-MC: DC-BASED MONTE CARLO)

| Region | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|---|---|---|---|---|---|---|---|
| PEM    | 0.064 | 0.158 | 0.031 | 0.138 | 0.043 | 0.039 | 0.027 | 0.074 | 0.232 |
| Proposed | 0.010 | 0.001 | 0.006 | 0.013 | 0.003 | 0.004 | 0.001 | 0.001 | 0.003 |

Flow

| Region | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|---|---|---|---|---|---|---|---|---|
| PEM    | 0.440 | 0.207 | 1.348 | 1.796 | 0.226 | 0.091 | 0.239 | 0.172 | 0.236 |
| DC-MC  | 0.167 | 0.039 | 0.248 | 0.207 | 0.104 | 0.067 | 0.123 | 0.090 | 0.088 |
| Proposed | 0.009 | 0.002 | 0.019 | 0.008 | 0.003 | 0.003 | 0.006 | 0.002 | 0.001 |

in terms of the expected voltages and flows in each region. The errors are provided in Table IX. The minimal values are also highlighted in bold. Note that we did not show the errors of the DC-based Monte Carlo method regarding the voltages, as the voltages are all equal to 1. From this table, we can conclude that (1) the relative errors of the proposed method are the smallest compared with the other two methods; (2) the magnitude of the relative errors of the proposed method is also small. Therefore, the accuracy of the proposed method is indeed satisfactory for the 1354-bus system.

Note that for the 1354-bus system, the maximal computational time of the proposed method among all regions reaches 2527 seconds whereas the maximal computational time for the 118-bus system is around 37 seconds. That said, the computational time of the proposed method grows superlinear with system size. However, as mentioned earlier, ISOs have to sacrifice computational efficiency in order to realize the protection of their confidential information. Besides, although the computational time of the 1354-bus system exceeds 42 minutes, it still can meet requirements of many realistic applications, e.g., the reliability evaluation of power systems, the transmission expansion planning, and decision making in day-ahead markets.

VII. CONCLUSION

In this paper, we proposed a privacy-preserving distributed PLF method. This method allows every regional ISO to only obtain its regional joint PLF given the uncertain power injections across regions. Using the proposed method, each ISO only needs its own system parameters for computing the regional PLF. In addition, every ISO merely requires to communicate with its neighbors, and no center for data collection is needed.

Moreover, no ISO can deduce the PLF and parameters of other regions despite communication. To the best of our knowledge, this is the first privacy-preserving distributed approach for PLF calculation.

The findings of this paper are summarized as follows:

1) We revealed the key mathematical problem of developing a privacy-preserving distributed PLF method. It namely corresponds to solving a system of linear equations in a distributed manner. These linear equations are partitioned among ISOs, where each ISO can only access a disjoint subset of the full set of equations and variables.

2) Experiments on the 118- and 1354-bus systems show that the PLFs obtained by the proposed and corresponding centralized methods perfectly agree, which demonstrates the correctness of our method.

3) Tests for the 118- and 1354-bus systems indicate that the proposed method has satisfactory accuracy compared with other frequently used centralized PLF methods.

4) Regional ISOs have to sacrifice computational efficiency to realize the protection of their confidential information. Nevertheless, the simulation of the 1354-bus system shows that the computational time of the proposed method can still meet requirements of many realistic applications, e.g., the reliability evaluation of power systems, transmission expansion planning, and decision making in day-ahead markets.

There are still some open issues worth investigating. For a given system, there are clusters of buses located downstream of uncertain power injections. Using network reduction techniques, it might be possible to convert these clusters of buses into fewer nodes, thereby reducing the system size and the computational burden of the proposed method. Although in this paper we did not pursue improving computational efficiency, we will consider it as a potential area of our future works.

REFERENCES

[1] B. R. Prusty and D. Jena, “A critical review on probabilistic load flow studies in uncertainty constrained power systems with photovoltaic generation and a new approach,” Renewable Sustain. Energy Rev., vol. 69, pp. 1286–1302, 2017.

[2] J. Contreras, A. Losi, M. Russo, and F. F. Wu, “Simulation and evaluation of optimization problem solutions in distributed energy management systems,” IEEE Trans. Power Syst., vol. 17, no. 1, pp. 57–62, Feb. 2002.

[3] H. Zhang, B. Zhang, A. Bose, and H. Sun, “A distributed multi-control-center dynamic power flow algorithm based on asynchronous iteration scheme,” IEEE Trans. Power Syst., vol. 33, no. 2, pp. 1716–1724, Mar. 2018.

[4] ISONE and NYISO, “Inter-regional interchange scheduling (iris) analysis and options,” 2011. [Online]. Available: https://www.iso-ne.com/static-assets/documents/pubs/whtrpr/iris_white_paper.pdf

[5] L. Wu, “A transformation-based multi-area dynamic economic dispatch approach for preserving information privacy of individual areas,” IEEE Trans. Smart Grid, vol. 10, no. 1, pp. 722–731, Jan. 2019.

[6] Z. Li, W. Wu, B. Zhang, and B. Wang, “Decentralized multi-area dynamic economic dispatch using modified generalized benders decomposition,” IEEE Trans. Power Syst., vol. 31, no. 1, pp. 526–538, Jan. 2016.

[7] J. Zhu, T. Zhu, M. Liu, and W. Lu, “Decentralized optimal dispatch of multi-area power systems based on non-linear value-function approximation,” IET Gener., Transmiss. Distrib., vol. 13, no. 22, pp. 5083–5090, 2019.

[8] A. Ahmadi-Khatir, A. J. Conejo, and R. Cherkauer, “Multi-area unit scheduling and reserve allocation under wind power uncertainty,” IEEE Trans. Power Syst., vol. 29, no. 4, pp. 1701–1710, Jul. 2014.
[9] P. J. M., “Amended, restated operating agreement of pjm interconnection,” L.L.C, PJM, New York, NY, USA, 2017. [Online]. Available: http://www.pjm.com/media/documents/agreements/oa.aspx, Accessed: May 2017.

[10] G. Carpinelli, P. Caramia, and P. Varilone, “Multi-linear Monte Carlo simulation method for probabilistic load based distribution systems with wind and photovoltaic generation systems,” Renewable Energy, vol. 76, pp. 283–295, 2015.

[11] C. Carmona-Delgado, E. Romero-Ramos, and J. Riquelme-Santos, “Probabilistic load flow with versatile non-Gaussian power injections,” Elect. Power Syst. Res., vol. 119, pp. 266–277, 2015.

[12] Q. Xiao and S. Zhou, “Probabilistic power flow computation considering correlated wind speeds,” Appl. Energy, vol. 231, pp. 677–685, 2018.

[13] A. M. Leite da Silva and A. M. de Castro, “Risk assessment in probabilistic load flow via Monte Carlo simulation and cross-entropy method,” IEEE Trans. Power Syst., vol. 34, no. 2, pp. 1193–1202, Mar. 2019.

[14] Y. Liu, S. Gao, H. Cui, and L. Yu, “Probabilistic load flow considering correlations of input variables following arbitrary distributions,” Elect. Power Syst. Res., vol. 140, pp. 354–362, 2016.

[15] G. Zhou, R. Bo, L. Chien, X. Zhang, S. Yang, and D. Su, “GPU-accelerated algorithm for online probabilistic power flow,” IEEE Trans. Power Syst., vol. 33, no. 1, pp. 1132–1135, Jan. 2018.

[16] M. Aien, M. Fotuhi-Firuzabad, and F. Aminifar, “Probabilistic load flow in correlated uncertain environment using unscented transformation,” IEEE Trans. Power Syst., vol. 27, no. 4, pp. 2272–2281, Nov. 2012.

[17] Chun-Lien Su, “Probabilistic load-flow computation using point estimate method,” IEEE Trans. Power Syst., vol. 20, no. 4, pp. 1843–1851, Nov. 2005.

[18] J. M. Morales and J. Perez-Ruiz, “Point estimate schemes to solve the probabilistic power flow,” IEEE Trans. Power Syst., vol. 22, no. 4, pp. 1594–1601, Nov. 2007.

[19] M. Mohammadi, “Probabilistic harmonic load flow using fast point estimate method,” IET Gener., Transmiss., Distrib., vol. 9, no. 13, pp. 1790–1799, 2015.

[20] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” Proc. IEEE, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[21] R. H. Baghaee, M. Mirsalim, G. B. Gharehpetian, and A. Talebi, “Fuzzy unscented transform for uncertainty quantification of correlated wind/PV microgrids: probabilistic power flow based on rbfnn,” IET Renewable Power Gener., vol. 11, no. 6, pp. 867–877, 2017.

[22] Y. Wang, N. Zhang, Q. Chen, J. Yang, C. Kang, and J. Huang, “Dependent discrete convolution based probabilistic load flow for the active distribution system,” IEEE Trans. Sustain. Energy, vol. 8, no. 3, pp. 1000–1009, Jul. 2017.

[23] G. Wang, H. Xin, D. Wu, P. Ju, and X. Jiang, “Data-driven arbitrary polynomial chaos-based probabilistic load flow considering correlated uncertainties,” IEEE Trans. Power Syst., vol. 34, no. 4, pp. 3274–3276, Jul. 2019.

[24] H. Sheng and X. Wang, “Probabilistic power flow calculation using non-intrusive low-rank approximation method,” IEEE Trans. Power Syst., vol. 34, no. 4, pp. 3014–3025, Jul. 2019.

[25] D. Villanueva, A. E. Feij, and J. L. Pazos, “An analytical method to solve the probabilistic load flow considering load demand correlation using the dc load flow,” Electric Power Syst. Res., vol. 110, pp. 1–8, 2014.

[26] Z. Wang, C. Shen, F. Liu, and F. Gao, “Analytical expressions for joint distributions in probabilistic load flow,” IEEE Trans. Power Syst., vol. 32, no. 3, pp. 2473–2474, May 2017.

[27] C. P. Nguyen and A. J. Flueck, “A novel agent-based distributed power flow solver for smart grids,” IEEE Trans. Smart Grid, vol. 6, no. 3, pp. 1261–1270, May 2015.

[28] Hongbin Sun and Boming Zhang, “Distributed power flow calculation for whole networks including transmission and distribution,” in Proc. IEEE/IEEJ Trans. Transmiss. Distrib. Conf. Expo., Apr. 2008, pp. 1–6.

[29] J. Yang, N. Zhang, C. Kang, and Q. Xia, “A state-independent linear power flow model with accurate estimation of voltage magnitude,” IEEE Trans. Power Syst., vol. 32, no. 5, pp. 3607–3617, Sep. 2017.

[30] H. Shuai, J. Fang, X. Ai, Y. Tang, J. Wen, and H. He, “Stochastic optimization of economic dispatch for microgrid based on approximate dynamic programming,” IEEE Trans. Smart Grid, vol. 10, no. 3, pp. 2440–2452, May 2019.

[31] J. Zhan, W. Liu, and C. Y. Chung, “Stochastic transmission expansion planning considering uncertain dynamic thermal rating of overhead lines,” IEEE Trans. Power Syst., vol. 34, no. 1, pp. 432–443, Jan. 2019.

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

[33] M. Jia, Y. Wang, C. Shen, and G. Hug, “Privacy-preserving distributed clustering for electrical load profiling,” Feb. 2020, arXiv:2002.12709.

[34] N. Aizan-Ruhi, F. Lahouti, A. S. Avestimehr, and B. Hassibi, “Distributed solution of large-scale linear systems via accelerated projection-based consensus,” IEEE Trans. Signal Process., vol. 67, no. 14, pp. 6358–6362, 2018.

[35] Z. Wang, C. Shen, and F. Liu, “A conditional model of wind power forecast errors and its application in scenario generation,” Appl. Energy, vol. 212, pp. 771–785, 2018.

[36] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, “Continuity ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions,” IEEE Trans. Power Syst., vol. 28, no. 4, pp. 4909–4917, Nov. 2013.

[37] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang, “Data-driven power flow linearization: A regression approach,” IEEE Trans. Smart Grid, vol. 10, no. 3, pp. 2569–2580, May 2019.

[38] P. Chen, Z. Chen, and B. Bak-Jensen, “Probabilistic load flow: A review,” in Proc. 3rd Int. Conf. Elect. Utility Deregulation Restructuring Power Technol., Apr. 2008, pp. 1586–1591.

[39] M. O. Buygi, H. M. Shanesci, G. Balzer, and M. Shahidehpour, “Transmission planning approaches in restructured power systems,” in Proc. IEEE Bologna Power Tech Conf., 2003, p. 7.

[40] M. Rejc and M. Pantos, “Probabilistic short-term transmission-loss forecasting based on a two-point estimate method,” in Proc. 7th Int. Conf. Eur. Energy Marker, 2010, pp. 1–6.