

Topology-Adaptive Piecewise Linearization for Three-Phase Power Flow Calculations in Distribution Grids

Jiaqi Chen, and Line A. Roald

Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, USA.

Email: {jiaqi.chen, roald}@wisc.edu

Abstract—The rapidly growing number of distributed energy resources is increasing uncertainty and variability in distribution system operations. At the same time, better access to measurement data is enabling new, data-driven methods for state estimation and analysis. Unfortunately, topology changes pose a significant challenge to existing methods, which are geared towards approximating the power flow (PF) equations in a continuous manner. To address this gap, this paper proposes a data-driven topology-adaptive piecewise PF linearization approach which inherently adapts to topology changes in the distribution system. The approach leverages measurements of a topology-identifying variable that helps cluster the data according to the system topology, without requiring explicit information about the topology status. Specifically, when we fit a three-phase piecewise linear PF model using a K -plane regression algorithm while integrating the topology-identifying variables, we observe that each linear piece only incorporates samples from a single topology. Numerical tests demonstrate that the resulting piecewise linear PF model enables high accuracy PF calculations, even under system topology changes.

Index Terms—Topology-adaptive, data-driven, piecewise power flow linearization

I. INTRODUCTION

The rapid increase in the number of distributed energy resources (DERs) and the projected growth in electric vehicle (EV) charging can stress distribution grids, potentially resulting in power quality issues such as voltage magnitude violations and excessive voltage unbalance [1]. Access to accurate and efficient analysis tools is essential to adapt to these challenges, with the three-phase power flow (PF) equations forming the basis of the analysis. Unfortunately, solving traditional PF requires detailed feeder models, which many utilities lack. Even utilities that have accurate feeder models often struggle to keep them up to date as phase connections and switch statuses frequently change, either intentionally (such as in response to seasonal variabilities) or unintentionally (e.g. during restoration efforts after power outages). Furthermore, the nonlinear characteristic of the PF equations may result in non-convergence problems and high computation burdens [2]. The above issues can be resolved by linearizing the three-phase PF equations. However, existing linearization methods typically struggle to remain accurate under various operational

statuses, and particularly with changing topologies. In this paper, we seek to address this gap.

There are two main types of PF linearization approaches: the model-based approach and the data-driven approach. In the model-based approach, the PF equations are linearized around certain operating points [3] or assuming that, e.g., the voltage magnitude is close to one [4]. This approach relies on the knowledge of the feeder parameter and topology, which may not be available in practice. Furthermore, as the range of operational conditions increases, the validity of the underlying assumptions may be violated, causing the accuracy of the model-based linearization approach to deteriorate.

Data-driven methods seek to address the issue of missing model parameters and varying operating conditions by directly leveraging feeder data. This approach has become more viable in recent years, as the broad adoption of smart meters (SMs), microphaser measurement units (μ PMUs), and supervisory control and data acquisition (SCADA) systems provides an opportunity to obtain sufficient measurement data [5]. There exist a range of different data-driven approaches for state estimation and PF analysis in power systems, including several approaches based on machine learning such as [6], [7]. Here, we focus on linearized PF models which create a linear mapping between measurement data such as injections of active and reactive power and quantities of interest such as PFs and voltage magnitudes. Since the measurement data reflects the real and time-varying operational status of the system, we can obtain an accurate linear model even without details regarding the system parameters [8].

In recent years, data-driven PF calculations have received significant attention in the literature. Many references employ data-driven regression to obtain the linearized PF model, including a single-phase equivalent PF linearization approach based on partial least squares and Bayesian linear regression [9] and a support vector-based approach to obtain a robust three-phase PF linearization that addresses the issues of measurement outliers and data collinearity [10]. Other references use least squares regression [11], [12] or Gaussian process regression [13] to build the linearized PF model. In [14], an improved K -plane regression algorithm is proposed to train a three-phase piecewise linear PF model that has high accuracy across a larger range of operating conditions. A challenge of the above data-driven linearization approaches is that they

only work properly if all measurement data is obtained from a single topology. This is because a single set of power injections might lead to vastly different voltage distributions depending on the feeder configuration. It is however well known that the topology of distribution feeders changes due to switching actions (e.g., to manage transformer loading, reduce network losses or restore load after a fault) and changes in the phase connections (e.g., during restoration after a storm), and some of these changes may never be properly recorded. Thus, PF models that are able to automatically detect and adapt to new feeder topologies can offer a significant advantage to distribution operators. Related work includes [7], which proposes a neural-network-based PF calculation approach using voltage differences as the feature vector to characterize the system topology. Unfortunately, the neural network lacks the clear relationships and interpretability of a linearized PF model. Thus, we seek to address this gap.

Specifically, we propose a data-driven three-phase piecewise PF linearization approach that automatically adapts to topology changes in the distribution system. To achieve this, we first analyze the limitation of the existing piecewise PF linearization approach under feeder topology changes. Based on this analysis, we introduce a set of topology-identifying variables x_{add} that help differentiate between different feeder topologies. These variables are real-time voltage measurements at selected buses and are added to the other input data (i.e., active and reactive power injections) to help cluster the training data according to the feeder topology. We then improve the K -plane regression method from [14] to obtain a piecewise linear, three-phase PF representation. Once the model is trained, we can leverage measurements of power injections and the topology-identifying voltages to solve PF in real time. We note that neither the training of the method (i.e., the clustering and regression) nor the use of the method in real time require explicit knowledge of the system model.

The contributions of this paper can be summarized as follows. First, review the existing model and analyze its limitations. Second, we develop a topology-adaptive, piecewise linear three-phase PF model. Third, we devise a PF calculation process based on the trained model, which automatically identifies and uses the correct topology. Finally, we present a case study that illustrates that the proposed approach can identify the right topology for a given state and calculate the PF accurately under system topology changes.

The remainder of this paper is organized as follows. Section II reviews the data-driven three-phase piecewise PF linearization approach in our previous work [14] and how it cannot adequately adapt to changes in the feeder topology. Section III proposes extensions to the existing approach to enable better considerations of feeder topology changes. Section IV provides numerical results and Section V concludes.

II. PRELIMINARIES

We start by summarizing the data-driven three-phase piecewise PF linearization approach in our previous work [14], and analyze the limitation of this approach when the feeder topology changes.

A. Data-driven Three-phase Piecewise PF Linearization Approach with One Topology

Reference [14] proposes a data-driven three-phase piecewise PF linearization approach which builds piecewise linear relationships between the power injections and the voltage magnitude and branch power flow. The method splits the training data into K partitions, and derives a linearized PF model for each partition. Within the k^{th} partition, the corresponding linear model is given by

$$\mathbf{y} = \mathbf{w}_k^T \cdot \mathbf{x} + \mathbf{b}_k, \quad (1)$$

$$\mathbf{u}_{\text{root}} = \mathbf{u}_{\text{ref}}, \quad (2)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{PF}^T & \mathbf{QF}^T & \mathbf{u}^T & \mathbf{P}_{\text{root}}^T & \mathbf{Q}_{\text{root}}^T \end{bmatrix}^T, \quad (3)$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{P}^T & \mathbf{Q}^T & \mathbf{u}_{\text{root}}^T \end{bmatrix}^T,$$

where $\mathbf{x}$ is the vector of independent variables and $\mathbf{y}$ is the vector of dependent variables. The vectors $\mathbf{PF}$, $\mathbf{QF}$ represent the three-phase branch active and reactive power flow, respectively, and $\mathbf{P}$, $\mathbf{Q}$ and $\mathbf{u}$ are the vectors of three-phase active power injection, reactive power injection and square of voltage magnitude of all nodes except for the root node. The root node is the slack bus, with $\mathbf{u}_{\text{root}}$ and $\mathbf{u}_{\text{ref}}$ representing the vector of three-phase squared root node voltage magnitudes and reference voltage magnitudes, while $\mathbf{P}_{\text{root}}$, $\mathbf{Q}_{\text{root}}$ are the vectors of three-phase active and reactive power injections at the root node. The superscript T denotes transpose. The matrix $\mathbf{w}_k$ is the coefficient matrix in partition k and $\mathbf{b}_k$ is a vector of constants in partition k .

The coefficients of the piecewise linearized PF model can be fitted using the K -plane regression algorithm in [14]. The fitting can be done either separately (clustering and fitting data separately for each output variable) or together (clustering and fitting data for multiple output variables at the same time). In the following, we describe the fitting procedure for a single variable. Consider the K -piecewise linear function $f(x) = y = \mathbf{w}_k^T x + b_k$, with $k = 1, \dots, K$, where $\mathbf{w}_k$ and b_k are the parameters we are aiming to fit and K is the number of clusters. Assume that we have access to N observed samples $(\mathbf{x}_n, y_n)$ where $\mathbf{x}_n$ is the input variable and y_n is the output variable for sample $n = 1, \dots, N$, respectively. The vector $\boldsymbol{\mu}_k$ is the cluster center for cluster $k = 1, \dots, K$. The corresponding error function is given by

$$E = \sum_{n=1}^N \min_{\tilde{\mathbf{w}}_k, \boldsymbol{\mu}_k} \left[\left(\tilde{\mathbf{w}}_k^T \tilde{\mathbf{x}}_n - y_n \right)^2 + \gamma \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 + \alpha \|\tilde{\mathbf{w}}_k\|^2 \right] \quad (4)$$

Here, the first term represents the prediction error with the current coefficients. Note that the vectors $\tilde{\mathbf{w}}_k = \begin{bmatrix} \mathbf{w}_k^T & b_k \end{bmatrix}^T$ and $\tilde{\mathbf{x}}_n = \begin{bmatrix} \mathbf{x}_n^T & 1 \end{bmatrix}^T$, $n = 1, \dots, N$ have been extended to account for the constant term in (1). The second term of the error function minimizes the distance between the samples $\mathbf{x}_n$ and the cluster center $\boldsymbol{\mu}_k$, while the third term is a regularizing term that penalizes large coefficients and helps address the issues associated with data co-linearity. The parameters α and γ can be tuned to manage the relative weight of the terms.

1) *Training the Piecewise Linear PF Model:* To obtain the best possible parameters $(\tilde{\mathbf{w}}_k, \boldsymbol{\mu}_k)$, the error function is minimized by iterating between (i) choosing hyperplane coefficients $\tilde{\mathbf{w}}_k$ and cluster centers $\boldsymbol{\mu}_k$ to minimize the error for a given partition of the data points into sets, and (ii) updating which cluster each data point belongs to, given the hyperplane coefficient $\tilde{\mathbf{w}}_k$ and cluster center points $\boldsymbol{\mu}_k$. The detail of the iteration can be found in [14]. After learning all parameters $(\tilde{\mathbf{w}}_k, \boldsymbol{\mu}_k)$, the piecewise linearized PF model is obtained as $y = \tilde{\mathbf{w}}_k^T \tilde{\mathbf{x}}$ with the partition center $\boldsymbol{\mu}_k$, $k = 1, \dots, K$.

2) *Using the Piecewise Linear PF Model:* To solve the PF calculations using the piecewise linear model, we first identify the nearest partition j for the given vector of power injection $\mathbf{x}$. This partition is determined by

$$j = \arg \min_k \|\mathbf{x} - \boldsymbol{\mu}_k\|^2, k = 1, \dots, K. \quad (5)$$

Using partition j , we then calculate the branch power flow and bus voltage magnitude by solving

$$y = \tilde{\mathbf{w}}_j^T \tilde{\mathbf{x}}. \quad (6)$$

This can be repeated if separate models are trained for each output variable y .

B. Limitation of the Data-driven Three-phase Piecewise PF Linearization Approach with One Topology

For the data-driven three-phase piecewise PF linearization approach in [14], the training data consists of the power injections $\mathbf{x}$ (the feature) and the voltage magnitudes and branch power flows $\mathbf{y}$ (the label). During training, the power injection $\mathbf{x}$ is used to cluster the training data into different clusters. When the model is used, the power injection is compared with the obtained cluster centers $\boldsymbol{\mu}$ to determine which cluster it belongs to in the PF calculation process.

However, this proposed approach is only valid when the training data is obtained from one system topology. If there is more than one topology, one set of power injections $\mathbf{x}$ may correspond to different voltage magnitudes and branch power flows $\mathbf{y}$. In other words, for different topologies, the linear relationship between $\mathbf{x}$ and $\mathbf{y}$ is different, and mixing training data from different topologies may thus cause large inaccuracies. To calculate the PF accurately, we thus need to be able to differentiate between and derive different linearized PF relationships for each topology.

The method in [14] is unable to achieve this because it only relies on the power injection $\mathbf{x}$ to assign samples to the right clusters both during the training process and during the PF calculation. Since the power injections do not contain any information about the topology, we may include training data from different topologies into one cluster or identify the wrong cluster during computation. Therefore, the algorithm in [14] cannot obtain the different linear relationships for different topologies.

III. EXTENSION TO A TOPOLOGY-ADAPTIVE PIECEWISE LINEAR POWER FLOW MODEL

In the following, we seek to address the above concerns and develop a topology-adaptive version of the data-driven algorithm from [14].

A. Determining Topology-Identifying Variables

Our goal is to be able to (i) cluster the training data from different topologies into different clusters and (ii) to be able to identify the correct cluster (corresponding to the correct topology) when we use the model to solve PF. As observed above, simply using power injections to cluster the data is not sufficient as it does not contain any topology-identifying information. To address this, we propose to add additional topology-identifying variables x_{add} to the input data to help cluster the training data according to the topology. The topology-identifying variable x_{add} should be a quantity that will be (ideally significantly) different for different topologies, even if the power injections are the same. Here, we define x_{add} as the voltage magnitude at a small subset of nodes, though it could also be possible to use the PF of a subset of branches. When choosing which variables to include in x_{add} , it is important to consider the location of devices with real-time measurement and communication ability. It is also important to ensure that the chosen variables are able to provide the necessary information to distinguish between multiple different topologies. The choice of the x_{add} can be determined by cross-validation.

B. Topology-Aware Power Flow Model

In the topology-aware PF linearization approach, the resulting piecewise linearized PF model we want to obtain is the same as the one in [14], which is given by (1)-(3). However, the cluster center that we use to cluster the data during training and identify the correct cluster while we solve the PF is different, as we explain below.

To adapt to different topologies, we use both the power injection $\mathbf{x}$ and the identification variable x_{add} together as

$$\hat{\mathbf{x}} = [\mathbf{x}, \beta \cdot x_{add}]$$

to cluster the training data. Here, β is the a pre-defined parameter that decides the weight of the identification variable. We include this weight as we found that choosing $\beta > 1$ helps to increase the topology identification accuracy in some scenarios. The value of β can be determined by cross-validation. With this change to the clustering variable, the updated error function becomes

$$\hat{E} = \sum_{n=1}^N \min_{\tilde{\mathbf{w}}_k, \hat{\boldsymbol{\mu}}_k} \left[\left(\tilde{\mathbf{w}}_k^T \tilde{\mathbf{x}}_n - y_n \right)^2 + \gamma \|\hat{\mathbf{x}}_n - \hat{\boldsymbol{\mu}}_k\|^2 + \alpha \|\tilde{\mathbf{w}}_k\|^2 \right] \quad (7)$$

Here, $\hat{\boldsymbol{\mu}}_k = [\boldsymbol{\mu}_k^T, \boldsymbol{\mu}_{add,k}^T]^T$ is the new cluster center definition of the topology-aware approach, which contains entries corresponding both to the power injection $\mathbf{x}$ and the identification variable x_{add} . Note that this change only affects the second term of the error function, while the other terms and variables have a similar meaning as in (4). It should be noted that although equation (7) looks similar to equation (4), since we introduce the variable $\hat{\mathbf{x}}$, the two error functions are different and equation (7) can adapt to the system topology changes.

1) *Training the topology-adaptive model:* The iteration to decrease the error function (7) is improved in a similar manner as before:

Step 0) Initialization: We obtain the initial cluster set S^0 by clustering the training data $(\hat{\mathbf{x}}, \mathbf{y})$ to K clusters with K -means clustering algorithm. We initialize the linear coefficients to 0. *Step 1) Calculating the coefficients $\tilde{\mathbf{w}}_k$ and the cluster center $\hat{\boldsymbol{\mu}}_k$:* Assume that the linear coefficients and cluster centers after the c^{th} iteration is given by $\Theta^c = \{(\tilde{\mathbf{w}}_1^c, \hat{\boldsymbol{\mu}}_1^c), \dots, (\tilde{\mathbf{w}}_K^c, \hat{\boldsymbol{\mu}}_K^c)\}$ and that the training data is clustered into sets S_k^c , $k = 1, \dots, K$. The new cluster centers $\hat{\boldsymbol{\mu}}_k^{c+1}$ are computed as

$$\begin{aligned} \hat{\boldsymbol{\mu}}_k^{c+1} &= \arg \min_{\hat{\boldsymbol{\mu}}_k} \sum_{n=1}^K \hat{E}_k^c(\tilde{\mathbf{w}}_k^c, \hat{\boldsymbol{\mu}}_k) \\ &= \arg \min_{\hat{\boldsymbol{\mu}}_k} \sum_{\hat{\mathbf{x}}_n \in S_k^c} \|\hat{\mathbf{x}}_n - \hat{\boldsymbol{\mu}}_k\|^2 = \frac{1}{|S_k^c|} \sum_{\hat{\mathbf{x}}_n \in S_k^c} \hat{\mathbf{x}}_n. \end{aligned} \quad (8)$$

The new linear coefficients $\tilde{\mathbf{w}}_k^{c+1}$ are computed as

$$\begin{aligned} \tilde{\mathbf{w}}_k^{c+1} &= \arg \min_{\tilde{\mathbf{w}}_k} \sum_{n=1}^K \sum_{k=1}^K \hat{E}_k^c(\tilde{\mathbf{w}}_k^c, \hat{\boldsymbol{\mu}}_k) \\ &= \arg \min_{\tilde{\mathbf{w}}_k} \left[\sum_{\hat{\mathbf{x}}_n \in S_k^c} (\tilde{\mathbf{w}}_k^T \hat{\mathbf{x}}_n - y_n)^2 + \alpha \|\tilde{\mathbf{w}}_k\|^2 \right] \\ &= (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \alpha \mathbf{I})^{-1} \tilde{\mathbf{X}}^T \mathbf{Y}, \end{aligned} \quad (9)$$

where $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1 \dots \tilde{\mathbf{x}}_M]$ and $\mathbf{Y} = [y_1 \dots y_M]$ with $m = 1, \dots, M$ representing the samples in the set S_k^c . The matrix $\mathbf{I}$ is the identity matrix, and the value of parameter α is chosen by cross-validation.

Step 2) Clustering the training data: After obtaining the parameter Θ^{c+1} , keep Θ^{c+1} fixed and cluster the training data into new sets S_k^{c+1} as

$$S_k^{c+1} = \left\{ \hat{\mathbf{x}}_n | k = \arg \min_j \left[(\tilde{\mathbf{x}}_n^T \tilde{\mathbf{w}}_j^{c+1} - y_n)^2 + \gamma \|\hat{\mathbf{x}}_n - \hat{\boldsymbol{\mu}}_j^{c+1}\|^2 \right] \right\}$$

Here, the first term seeks to add training data to clusters where their corresponding PF calculation error would be small, while the second term minimizes the distance to the cluster center.

Step 3) Check convergence: If the clusters $S_k^{c+1} = S_k^c$ for all $k = 1, \dots, K$ (i.e., there is no change in clustering during the iteration), stop. Otherwise, move back to *step 1*.

After the iteration stops, we obtain the final piecewise linear model $y = \tilde{\mathbf{w}}_k^T \tilde{\mathbf{x}}$ with the linear coefficient $\tilde{\mathbf{w}}_k$ and the cluster center $\hat{\boldsymbol{\mu}}_k$ for each cluster k . Importantly, the cluster center $\hat{\boldsymbol{\mu}}_k$ is related to the power injection $\mathbf{x}$ and the identification variable $\mathbf{x}_{add}$ to help cluster the data considering different topologies. A summary of the proposed training approach is found in Table I.

2) *Using the topology-adaptive power flow model:* In the PF calculation process, we use the power injection $\mathbf{x}$ and the identification variable $\mathbf{x}_{add}$ together and compare them with the cluster center $\hat{\boldsymbol{\mu}}_k$ to determine which cluster the given data belongs to. The cluster j for a given data $\hat{\mathbf{x}}$ is the nearest cluster, which can be found by

$$j = \arg \min_k \|\hat{\mathbf{x}} - \hat{\boldsymbol{\mu}}_k\|^2, k = 1, \dots, K. \quad (10)$$

Then the branch power flow and bus voltage magnitude $\mathbf{y}$ for the given power injection is calculated with the linear function in cluster j with equation (6).

TABLE I
SUMMARY OF THE TRAINING ALGORITHM

Input: $\{(\hat{\mathbf{x}}_1, y_1), \dots, (\hat{\mathbf{x}}_N, y_N)\}$
Output: $\{(\tilde{\mathbf{w}}_1, \hat{\boldsymbol{\mu}}_1), \dots, (\tilde{\mathbf{w}}_K, \hat{\boldsymbol{\mu}}_K)\}$
Begin
<i>Step 0:</i> Initialize $c=0, (\tilde{\mathbf{w}}_1^0, \hat{\boldsymbol{\mu}}_1^0), \dots, (\tilde{\mathbf{w}}_K^0, \hat{\boldsymbol{\mu}}_K^0)$ and $S_1^0, \dots, S_K^0$.
<i>Step 1:</i> Find $\mu_k^{c+1}, \tilde{\mathbf{w}}_k^{c+1}, k = 1, \dots, K$ as follows,
$\hat{\boldsymbol{\mu}}_k^{c+1} = \frac{1}{ S_k^c } \sum_{\hat{\mathbf{x}}_n \in S_k^c} \hat{\mathbf{x}}_n,$
$\tilde{\mathbf{w}}_k^{c+1} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}} + \alpha \mathbf{I})^{-1} \tilde{\mathbf{X}}^T \mathbf{Y}.$
<i>Step 2:</i> Find $S_k^{c+1}, k = 1 \dots K$ as follows,
$S_k^{c+1} = \left\{ \hat{\mathbf{x}}_n k = \arg \min_j \left[(\tilde{\mathbf{x}}_n^T \tilde{\mathbf{w}}_j^{c+1} - y_n)^2 + \gamma \ \hat{\mathbf{x}}_n - \hat{\boldsymbol{\mu}}_j^{c+1}\ ^2 \right] \right\}$
<i>Step 3:</i> Termination Criteria:
if $S_k^{c+1} = S_k^c, \forall k$, then
stop.
Else
$c = c + 1$,
go to <i>Step 1</i> .
end
end

IV. NUMERICAL TEST

We present numerical tests to demonstrate that the proposed method to obtain a topology-aware PF model can accurately identify the correct topology and provides accurate PF calculation results without explicit knowledge of the topology.

A. Simulation Setup

The proposed approach is tested on a modified unbalanced IEEE 123-bus distribution system. The information of branch parameters and load profiles is available online [15]. We simulate 5 different topologies by opening and closing some branches, of which the information is provided in [16].

To create PF training data, we first generate a diverse set of power injections by multiplying the given load consumption with a random factor obtained from a mixed Gaussian distribution. The range of the mean values for this distribution is $[-15, 10]$ and the standard deviation is 0.5. Next, we solve the PF using the back-forward sweep (BFS) algorithm, and combine the power injection and the power solutions to create the input data $\{\hat{\mathbf{x}}, \mathbf{y}\}$ as the training dataset. The testing dataset is generated in the same way. For each topology, we generate 25000 training data and 5000 testing data points. We shuffle the training data together to simulate the scenario in which we do not know the topology information of each data in the training process. The total number of the training data and the testing data is 125000 and 25000, respectively.

For the topology-aware linearization approach, choice of the $\mathbf{x}_{add}$ and the value of parameters K, γ, α and β are determined by cross-validation. Here we use the square of the voltage magnitude of node 60, 61, 67, 76, 84, 86, 93, 95, 97, 101, 115 and 116 as the identification variable $\mathbf{x}_{add}$. The cluster number $K=50$, $\gamma = 1e^7$, $\alpha = 1e^{-7}$ and $\beta = 15$. All simulations were implemented using MATLAB on a personal laptop with an Apple M1 Pro processor and 32 GB of RAM.

B. Topology Identification in the Training Process

We first test if the proposed approach can cluster the training data into groups that correspond to different topologies in the training process. It is important to note that in the training process, we assume that we do not know the topology information of the training data. However, since we have generated the training data with 5 system topologies ourselves, we know which topology to each training data belongs. We can use this information to test the effectiveness of the topology identification of the proposed approach. The topology information is also a necessary input to the BFS algorithm to calculate the PF. In the training process, we use $K = 50$. We do not assign a given number of clusters to each topology, but let the training process determine how the samples is split. This would allow the training process to assign more clusters to topologies that have a larger variation in the voltage magnitudes, thus allowing for a better fit.

We first investigate the topology distributions for each cluster, i.e. how many samples from each topology is included in each cluster, for the case where we train the piecewise linear PF model for the voltage magnitudes in phase a . Table II shows the resulting topology distributions of the first 5 clusters of the proposed topology-aware approach. Table III shows the same results for the original approach in [14], which only considers the PF injections when clustering the data.

TABLE II
TOPOLOGY DISTRIBUTION OF THE FIRST 10 CLUSTERS OF THE PROPOSED APPROACH

Topology	1	2	3	4	5
Cluster 1	0	1774	0	0	0
Cluster 2	0	0	0	0	3043
Cluster 3	0	6250	0	0	0
Cluster 4	0	0	0	0	2889
Cluster 5	0	0	0	2223	0
⋮	⋮	⋮	⋮	⋮	⋮

TABLE III
TOPOLOGY DISTRIBUTION OF THE FIRST 10 CLUSTERS OF THE APPROACH IN [14]

Topology	1	2	3	4	5
Cluster 1	756	725	733	811	797
Cluster 2	428	447	436	447	426
Cluster 3	512	500	512	491	495
Cluster 4	449	491	439	445	441
Cluster 5	473	529	448	487	483
⋮	⋮	⋮	⋮	⋮	⋮

The data in Table II demonstrates that the proposed topology-aware approach separates the training data into clusters that only contain samples from one topology. This illustrates that the proposed approach can cluster the training data into groups corresponding to different topologies, which will allow us to use the clustered data to fit an accurate linear approximation for the topology and power injection range corresponding to that cluster. In contrast, Table III shows that when we apply the approach in [14], each cluster contains the training data from different topologies which may cause large errors when approximating the PF function. Although

we only show results for the 5 first clusters here due to space constraints, the results remain similar for the remaining clusters. This illustrates that the proposed approach can cluster the training data considering different topologies while the approach in [14] cannot. The topology distribution of all 50 clusters is provided in [16].

C. Accuracy of the PF Calculation

We next test the PF calculation accuracy of the proposed approach. The obtained piecewise linearized PF model is used to calculate the PF on the testing dataset. Since we use the square of the voltage magnitude of node 60, 61, 67, 76, 84, 86, 93, 95, 97, 101, 115 and 116 as the identification variable x_{add} in the training process, we assume we know the voltage magnitude of these nodes in real time and use them together with the power injection to determine the cluster to which a testing data belongs, as shown in equation (10).

1) *Ability to identify the correct topology:* We test if the proposed approach can pick up the right topology as follows: First, during the training process, we identify which topology each cluster corresponds to (note that we are able to establish a clear cluster-topology correspondence because the proposed approach only includes samples for one topology in each cluster). Then, in the testing process, we check which cluster each testing data sample is assigned to. Finally, we compare the identified topology and the true topology of each testing data sample and assess whether the two topologies match. We find that the proposed approach identifies the right topology in all cases, i.e., it has an accuracy of 100 percent.

2) *Accuracy of the PF calculation:* Next, the PF calculation result is compared with the result of the BFS approach to assess the accuracy of the calculations. We obtain the mean, median and maximum PF calculation errors for both voltage magnitudes and branch power flows. The errors of voltage magnitude V are measured as the mean absolute error. The errors of branch power flow PF , QF are shown in both as an absolute percentage error with the unit of 100% and as the mean absolute error. Since we find that any variable with a zero value is correctly predicted to be zero, we omit these zero values from the accuracy comparison. The mean, median and maximum errors of the proposed approach are compared with the approach in [14], which are shown in Table IV.

From this data, we observe that the mean and maximum errors of the proposed approach are very small. For voltage magnitudes, the maximum errors are less than < 0.0044 p.u., which is highly accurate. For the active and reactive power flow, the average percentage errors are also very small, while the maximum percentage errors reach values above 10% for reactive power flow. The higher percentage error observed in reactive power compared to active power might because the reactive power flow is less linear than the active power flow. When looking at the absolute error values, we can however see that these high percentage values likely arise on lines with small power flows, as the largest absolute error is 1.02 MVar and the median is less than 10^{-3} for both active and reactive power. For comparison, the maximum active and reactive

TABLE IV
POWER FLOW CALCULATION ERROR

		Proposed approach			Approach in [14]		
	Phase	Mean	Median	Max	Mean	Median	Max
$V(p.u.)$	a	1.06e-4	2.76e-5	5.84e-3	1.62e-2	6.27e-3	2.05e-1
	b	1.02e-4	2.63e-5	4.31e-3	1.26e-2	5.11e-3	1.61e-1
	c	1.04e-4	2.71e-5	4.72e-3	1.41e-2	5.73e-3	1.78e-1
$PF(\%)$	a	6.69e-3	3.56e-3	1.28e+0	3.63e+1	4.75e-1	1.56e+4
	b	6.15e-3	3.68e-3	6.94e+0	5.75e+1	4.14e+0	1.37e+4
	c	8.26e-3	4.23e-3	1.69e+0	3.24e+1	2.88e-1	3.47e+4
$QF(\%)$	a	1.23e-2	6.67e-3	4.08e+0	4.00e+1	1.37e+0	1.69e+4
	b	1.23e-2	7.00e-3	1.40e+1	6.51e+1	7.36e+0	1.55e+4
	c	1.41e-2	7.75e-3	8.29e+0	3.67e+1	8.63e-1	4.09e+4
$PF(MW)$	a	2.70e-3	2.18e-4	3.92e-1	2.36e+0	2.84e-1	6.30e+1
	b	2.59e-3	2.44e-4	1.67e-1	2.06e+0	4.24e-1	4.66e+1
	c	2.55e-3	2.65e-4	2.12e-1	2.16e+0	2.49e-1	6.14e+1
$QF(MVAr)$	a	1.54e-3	1.90e-4	1.02e+0	1.29e+0	1.76e-1	2.94e+1
	b	1.45e-3	2.35e-4	3.50e-1	1.12e+0	2.87e-1	2.09e+1
	c	1.43e-3	2.09e-4	4.35e-1	1.18e+0	1.61e-1	2.80e+1

power flows are 103.50 MW and 53.27 MVAr, respectively, with median values of 3.55 MW and 1.87 MVAr.

Comparing the proposed approach to the approach in [14], we observe that the error of our proposed method is at least two orders of magnitude lower than the method in [14]. This highlights the need for a topology-adaptive approach that is able to inherently defines a specific piecewise linearized PF model for a specific system topology during training and identifies the right topology in the PF calculation process, as the proposed method does. The approach in [14] cannot achieve this, and thus fails to calculate the PF accurately.

V. CONCLUSION

This paper proposes a topology-adaptive piecewise PF linearization approach for three-phase PF calculations in a distribution feeder. The key advantage of the proposed approach is its ability to seamlessly adapt to topology changes by integrating additional topology-identifying variables into the model. During the training process, the topology-identifying variable is leveraged during the data clustering process, enabling the algorithm to inherently identify clusters of data points that belong to the same topology without receiving information about the true topology status. For each cluster, we use a regression-based method to derive a linear PF formulation, giving rise to a piecewise linear PF representation. During the online PF calculation process, real-time access to measurements of the topology-identifying variable allows us to identify the correct cluster and obtain accurate PF results.

When tested on an IEEE 123-bus distribution feeder with the training data obtained from 5 topologies, we observe that the proposed approach can cluster the training data into groups corresponding to different topologies in the training process. In the PF calculation process, the proposed approach can pick up the right topology for a given state. The PF calculation result illustrates that the proposed approach can calculate the power flow more accurately, which indicates its promising implementation value under system topology changes.

Important benefits of the proposed approach include that it is model-free when sufficient measurement data is available. Furthermore, it does not need the information that to which topology the training data belongs, thus enabling the model to also adapt to unexpected or unintended changes in topology. In future work, we want to investigate more carefully how to choose the locations and number of topology-identifying variables, assess the impact of measurement noise, data outliers and missing data, investigate other types of topology changes such as changes in the phase connections and understand how much data would be needed in real-time to adapt to a previously unseen topology.

REFERENCES

- [1] J. Chen and L. A. Roald, "A data-driven linearization approach to analyze the three-phase unbalance in active distribution systems," *Electric Power Systems Research*, vol. 211, p. 108573, 2022.
- [2] H. Li, L. Pan, and Q. Liu, "A linear power flow solution for distribution power system including pv bus and zip load," *Journal of Electrical Engineering & Technology*, vol. 14, no. 5, pp. 1859–1870, 2019.
- [3] X. Chen, W. Wu, and B. Zhang, "Robust capacity assessment of distributed generation in unbalanced distribution networks incorporating anm techniques," *IEEE Transactions on Sustainable Energy*, vol. 9, no. 2, pp. 651–663, 2017.
- [4] K. Liu, C. Wang, W. Wang, Y. Chen, and H. Wu, "Linear power flow calculation of distribution networks with distributed generation," *IEEE Access*, vol. 7, pp. 44 686–44 695, 2019.
- [5] P. Huynh, H. Zhu, Q. Chen, and A. E. Elbanna, "Data-driven estimation of frequency response from ambient synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6590–6599, 2018.
- [6] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2082–2092, 2020.
- [7] M. Xiang, J. Yu, Z. Yang, Y. Yang, H. Yu, and H. He, "Probabilistic power flow with topology changes based on deep neural network," *International Journal of Electrical Power & Energy Systems*, vol. 117, p. 105650, 2020.
- [8] Y. Liu, Z. Li, and J. Zhao, "Robust data-driven linear power flow model with probability constrained worst-case errors," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 4113–4116, 2022.
- [9] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang, "Data-driven power flow linearization: A regression approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2569–2580, 2018.
- [10] J. Chen, W. Li, W. Wu, T. Zhu, Z. Wang, and C. Zhao, "Robust data-driven linearization for distribution three-phase power flow," in *2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, 2020, pp. 1527–1532.
- [11] Z. Shao, Q. Zhai, J. Wu, and X. Guan, "Data based linear power flow model: Investigation of a least-squares based approximation," *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 4246–4258, 2021.
- [12] Y. Liu, Z. Li, and Y. Zhou, "Data-driven-aided linear three-phase power flow model for distribution power systems," *IEEE Transactions on Power Systems*, 2021.
- [13] P. Pareek, W. Yu, and H. D. Nguyen, "Optimal steady-state voltage control using gaussian process learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7017–7027, 2020.
- [14] J. Chen, W. Wu, and L. A. Roald, "Data-driven piecewise linearization for distribution three-phase stochastic power flow," *IEEE Transactions on Smart Grid*, vol. 13, no. 2, pp. 1035–1048, 2021.
- [15] W. Zheng, W. Wu, B. Zhang, and C. Lin, "Distributed optimal residential demand response considering operational constraints of unbalanced distribution networks," *IET generation, transmission & distribution*, vol. 12, no. 9, pp. 1970–1979, 2018.
- [16] "Supplemental file for data-driven piecewise linearization for distribution three-phase power flow adapting to system topology changes," [ONLINE], available (2023/5/25) at <https://www.dropbox.com/s/c01jj1ubwdql7im/Supplemental%20File%20for%20System%20Topology%20Changes.docx?dl=0>, May 2023.