Check for updates

Research on data-driven power flow calculation method based on undirected-graph delooping-backtracking

Hong Zhu, Zijian Hu*, Zichen Liu and Yandi Wang

Nanjing Power Supply Company, State Grid Jiangsu Electric Power Co., Ltd., Nanjing, China

As the scale of the power grid expands and distributed energy sources are integrated, along with the emergence of random loads, topological control of distribution networks has become a novel means of control. Therefore, data-driven power flow calculations must be capable of rapidly and accurately computing power flow results even when there are changes in the network's topology. In this paper, a data-driven power flow calculation method is proposed to take topological changes into account. Based on initial loop data, we employ an undirected-graph delooping-backtracking method to generate a set of feasible topological samples. Using the Monte Carlo method on this basis, we generate feasible samples for the network's topology and power injection, thereby establishing a training dataset. By training a deep neural network on these samples and adjusting network parameters, we effectively address power flow calculations in the presence of topological changes. Case study results demonstrate that the data-driven power flow calculation method, considering topological changes, can rapidly and accurately compute power flow results when topology alterations occur.

KEYWORDS

data-driven, distributed energy sources, power flow calculation, delooping-backtracking, deep neural network

1 Introduction

Power flow calculation in electric power systems is the process of determining voltage, current, and power distribution throughout the entire system based on given operational conditions and network structure. It is a fundamental technique that underpins various aspects of electric power systems, including system planning, dispatch, stability analysis, and power market operations, playing a crucial role in ensuring the stability, reliability, and efficient operation of the power grid.

Traditional power flow calculation methods, including the Gauss-Seidel method, Newton's method, and the Fast Decoupled method, are widely used. However, these methods are all model-based approaches. On the one hand, these methods rely on complex mathematical models and detailed network parameter information, which can be difficult to obtain and prone to errors. On the other hand, these methods require iterative calculation for solving nonlinear power equations, posing challenges to computational speed in large-scale systems.

With the advancement of machine learning technology, data-driven power flow calculation has garnered significant attention from scholars. In comparison to

traditional power flow calculation methods, data-driven approaches offer notable advantages. On one hand, data-driven power flow calculation does not require an in-depth understanding of network parameters, thus improving its applicability and accuracy. On the other hand, it does not necessitate complex iterative computations and can rapidly provide results using pre-trained models, making it more suitable for real-time scheduling and emergency response in power systems.

In recent years, scholars have conducted extensive research around data-driven power flow calculation.

In reference (Tan et al., 2020), a hybrid physical model-driven and data-driven approach for linearizing power flow model is proposed. It can retain the useful inherent information from the physical model and utilize the ability of data analysis to extract the inexplicit linear relationship. A data-driven linearization approach of PF equations is proposed in (Liu et al., 2019). Both partial least squares and Bayesian linear regression based algorithms are introduced to address the collinearity. In (Cui et al., 2020), a data-driven slow dynamic characteristic extraction and state estimation method are proposed to overcome the shortcomings of the computational burden caused by the Jacobian matrix inversion of the traditional method. A data-driven chance-constrained optimal gas-power flow (OGPF) calculation method without any prior assumption on the distribution of uncertainties of wind power generation is proposed in reference (Wang et al., 2021). Chen J et al. (2022) proposes a data-driven power flow (PF) linearization approach for three-phase SPF calculation. An approach with high adaptability to the nonlinearity of power flow is proposed in 6, which can significantly improve the calculation accuracy. Reference (Xing et al., 2022; Xing et al., 2022) respectively introduce a single bus data-driven power estimation based on modified linear power flow model and a modified data-driven power flow model for power estimation with incomplete bus data. The models proposed in these references exhibit higher accuracy compared to the linear power flow model. In (Crozier and Baker, 2022), a data-driven method for determining constraints that may be excluded from the formulation is proposed. A novel machine learning (ML) based data-driven risk assessment model for early-warning of power system transmission congestion is proposed in (Zhang et al., 2022). Chen Y et al. (2022) presents a novel data-driven power flow (DDPF)calculation method based on exact linear regression equations (ELREs), which offers higher computation efficiency. Liu et al. (2022a) and Chen et al. (2020) introduce a data-driven-aided linear three-phase power flow model for distribution power systems (DPSs), which offers higher accuracy and robustness. In (Liu et al., 2022b), a robust data-driven linear power flow (RD-LPF) model is constructed, which can significantly reduce average errors and unacceptable worst-case linearization errors. Reference (Li et al., 2023) introduces a data-driven linear power flow calculation model that incorporates the Kirchhoff's Current Law(KCL). This model can be embedded in optimal power flow for distribution networks. A regression approach combining the principal component analysis (PCA), support vector regression (SVR) and ridge regression (RR) is

developed, which improves the accuracy of PF calculation especially in the presence of bad data. A novel multi energy flow analysis method for integrated energy systems is proposed in (Zhu and Zhou, 2023) to learn the mapping relationship between the given variable and the demanded variable from the historical operation data. In (Shao et al., 2023), a physical-model-aided data-driven linear power flow (PD-LPF) model is proposed as a solution for addressing the issue of insufficient training data. It introduces physical model parameters to assist the data-driven training process, demonstrating excellent accuracy and robustness under severe missing-data conditions.

However, the data-driven power flow calculation problems addressed in the aforementioned studies assume that the topology of the electrical grid remains fixed. This allows for accurate power flow calculations in networks with a static structure. However, for distribution networks where topology changes due to network reconfiguration operations are possible, this can result in inaccurate power flow calculation results. This paper focuses on data-driven power flow calculations that take into account topological changes. To obtain a set of topological samples for training, we employ an undirected-graph delooping-backtracking method. To establish a more accurate relationship between topology and voltage/power, we utilize deep neural networks to learn from these samples, effectively addressing power flow calculation challenges posed by topological changes. The overall framework of the proposed data-driven power flow calculation method is presented in Figure 1.

The contributions of this paper are as follows:

1) A feasible topological sample generation method based on undirected-graph delooping-backtracking is proposed, and feasible topological samples are established through loop data initialization, loop breaking and loop backtracking operations.
2) A data-driven power flow calculation method considering topological changes is proposed, and a power flow calculation model considering topological changes is obtained by encoding the topological data and integrating it into the input of the data-driven model, and using DBN for training.

The structure of this paper is as follows: Section 2 introduces the method for generating topological samples based on undirected-graph delooping-backtracking. Section 3 presents the data-driven power flow calculation model. Section 4 provides the case studies conducted in this paper. Finally, Section 5 offers the conclusions drawn from the research.

2 Topological samples generating based on undirected-graph delooping-backtracking

Radiation constraints are fundamental constraints in the operation of distribution networks. Neglecting radiation
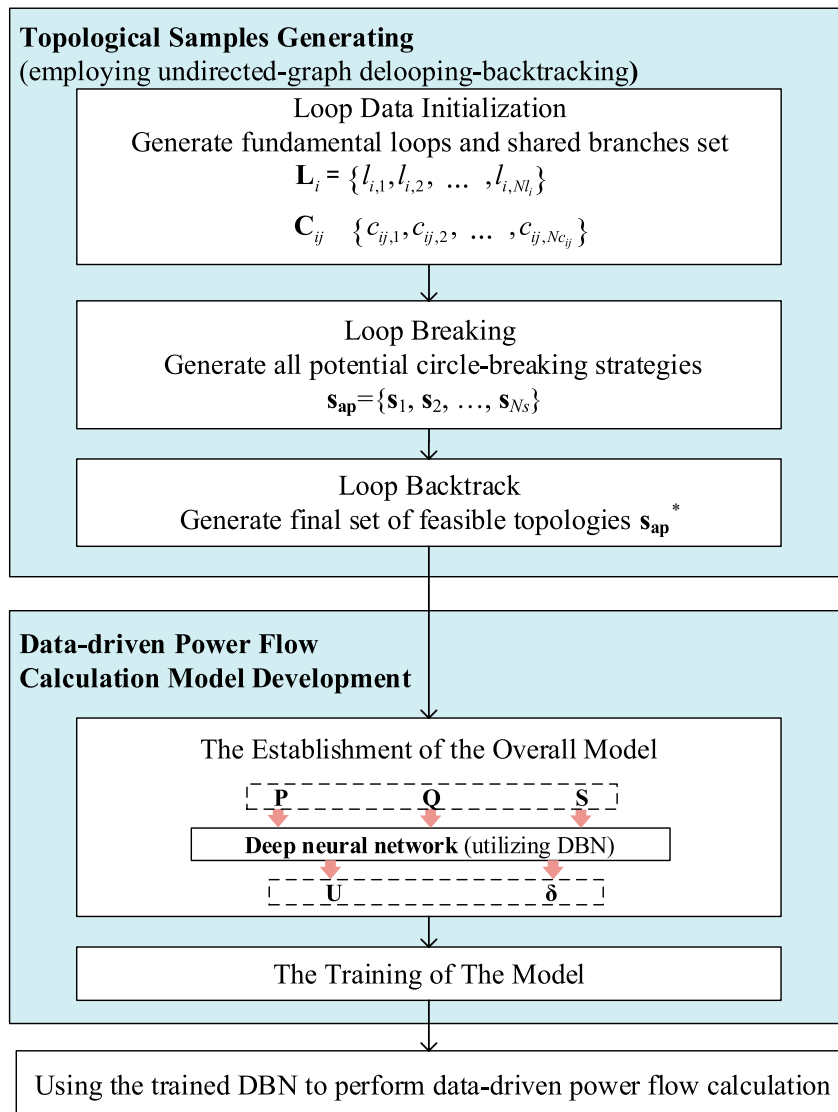
FIGURE 1
The overall framework of the proposed data-driven power flow calculation method.

constraints during the process of generating topological samples can result in a large number of infeasible topological samples being incorporated into the data-driven model. This section employs the undirected-graph delooping-backtracking method to generate a feasible set of network topologies. Based on this, topological samples for training neural networks are generated through Monte Carlo simulations, ensuring that each topological sample satisfies radiation constraints.

The undirected-graph delooping-backtracking method involves obtaining a feasible set of topologies by breaking loops and backtracking on the complete network graph. The specific process includes three main steps: loop data initialization, loop breaking, and loop backtracking.

2.1 Loop data initialization

The first step is loop data initialization, and the algorithm is given in Algorithm 1. Initially, identify all fundamental loops in the network, denoted as $\mathbf{L}_i$, and find a total of N_L fundamental loops. Assuming that the network branches are represented as $\mathbf{B} = \{b_1, b_2, \ldots, b_{N_B}\}$, the algorithm involves determining which branches are included in each fundamental loop $\mathbf{L}_i$. If a branch b_m belongs to $\mathbf{L}_i$, it is added to the fundamental loop $\mathbf{L}_i$. Additionally, shared branches among the fundamental loops should be identified, and be denoted as $\mathbf{C}_{ij}$. If b_m belongs to both $\mathbf{L}_i$ and $\mathbf{L}_j$, then it is added to the shared branches set $\mathbf{C}_{ij}$. In the end, the loop data for the network can be obtained as $\mathbf{L}_i = \{l_{i,1}, l_{i,2}, \ldots, l_{i,Nl_i}\}$, and

$\mathbf{C}_{ij} = \{c_{ij,1}, c_{ij,2}, \ldots, c_{ij,Nc_{ij}}\}$. Here, Nl_i represents the number of branches in $\mathbf{L}_i$, and Nc_{ij} represents the number of branches in $\mathbf{C}_{ij}$.

```
Input: Branch data of the network B = {b₁,b₂,…,b_{N_B}}
Output:      Loop      data      of      the      network
L_i = {l_{i,1},l_{i,2},…,l_{i,Nl_i}}, C_{ij} = {c_{ij,1},c_{ij,2},…,c_{ij,Nc_{ij}}}
Find N_L fundamental loops of the network
For m from 1 to N_B
  For i from 1 to N_L
    If b_m belongs to L_i (b_m∈L_i)
      Add b_m into L_i:
    End if
    For j from i+1 to Nl_i
      If b_m belongs to C_{ij} (b_m∈L_i & b_m∈L_j)
        Add b_m into C_{ij}:
      End if
    End for
  End for
End for
Output            loop            data
L_i = {l_{i,1},l_{i,2},…,l_{i,Nl_i}}, C_{ij} = {c_{ij,1},c_{ij,2},…,c_{ij,Nc_{ij}}}
```

Algorithm 1. Loop Data Initialization.

2.2 Loop breaking

After obtaining the set of branches for each fundamental loop, the next step is to break the loops, and the algorithm is presented in Algorithm 2. For each fundamental loop, we iteratively disconnect one branch at a time. Assuming that the branches disconnected from each fundamental loop are respectively $l_{1,k_1}, l_{1,k_2}, \ldots, l_{N_L,k_{N_L}}$, the set $\mathbf{s}_n = \{l_{1,k_1}, l_{1,k_2}, \ldots, l_{N_L,k_{N_L}}\}$ represents a potential loop-breaking strategy. By identifying all potential loop-breaking strategies, denoted as $\mathbf{s}_{ap} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{Ns}\}$, $\mathbf{s}_{ap}$ can be used to represent potential topological sets.

```
Input: Loop data of the network L_i = {l_{i,1},l_{i,2},…,l_{i,Nl_i}}
Output: Potential open branches s_{ap}
n=1
For k₁ from 1 to Nl₁
  For k₂ from 1 to Nl₂
    …
      For k_{NL} from 1 to Nl_{NL}
        s_n = {l_{1,k_1},l_{1,k_2},…,l_{N_L,k_{N_L}}}
        n=n+1
      End for
    …
  End for
End for
Output s_{ap}={s₁, s₂, …, s_{Ns}}
```

Algorithm 2. Loop Breaking.

2.3 Loop backtracking

The potential topological sets obtained through loop breaking may still contain islands that need to be further eliminated using loop

backtracking, as given in Algorithm 3. Assuming $\mathbf{s}_{ap}^*$ represents the final feasible topology, start by initializing $\mathbf{s}^*$ to be equal to $\mathbf{s}_{ap}$. Then, iterate through each element $\mathbf{s}_n$ in $\mathbf{s}_{ap}^*$. If $\mathbf{s}_n$ contains more than two elements from $\mathbf{C}_{ij}$, it inevitably indicates the presence of islands, so $\mathbf{s}_n$ needs to be removed from $\mathbf{s}_{ap}^*$ (Condition 1). Additionally, if $\mathbf{s}_n$ contains elements from $\mathbf{C}_{ij}$, $\mathbf{C}_{ik}$, and $\mathbf{C}_{jk}$ simultaneously, and the fundamental loops $\mathbf{L}_i$, $\mathbf{L}_j$ and $\mathbf{L}_k$ share nodes, their shared nodes will become islands as well, and $\mathbf{s}_n$ should also be removed from $\mathbf{s}_{ap}^*$ (Condition 2). After backtracking and eliminating infeasible topologies, $\mathbf{s}_{ap}^*$ represents the final set of feasible topologies obtained.

To illustrate the undirected-graph delooping-backtracking method proposed in this paper, a simple 5-node example system is considered. First, the fundamental loops and the shared branch sets between loops are determined, as shown in Figure 2. The network contains a total of 3 fundamental loops and 3 shared branch sets between loops, of which details are given in Table 1.

```
Input: Potential open branches s_{ap}, Loop data of the
network L_i = {l_{i,1},l_{i,2},…,l_{i,Nl_i}}, C_{ij} = {c_{ij,1},c_{ij,2},…,c_{ij,Nc_{ij}}}
Output: Reduced potential open branches s_{ap}*
Initialize s_{ap}*=s_{ap}
For n from 1 to N_s
  For i from 1 to N_L
    For j from i+1 to N_L
      If s_n satisfies Condition 1
        Delete s_n from s_{ap}*
      End if
      For k from j+1 to N_L
        If s_n satisfies Condition 2
          Delete s_n from s_{ap}*
        End if
      End for
    End for
  End for
End for
Output s_{ap}*
Condition 1: s_n contain more than one elements of C_{ij}.
Condition 2: s_n contain element in C_{ij}, C_{ik} and C_{jk} and L_i L_j
             and L_k share common nodes.
```

Algorithm 3. Loop Backtrack.

By breaking loops in the network as shown in Figure 2, all potential topological sets, denoted as $\mathbf{s}_{ap}$, can be obtain. Through loop backtracking, the final feasible topological set can be established, and denoted as $\mathbf{s}_{ap}^*$. During the loop backtracking process, an example of feasible topology (that satisfy neither Condition 1 nor Condition 2) is illustrated in Figure 3A. Infeasible topologies satisfying Condition 1 are depicted in Figure 3B, and those satisfying Condition 2 are shown in Figure 3C. From Figure 3B, it is apparent that due to the simultaneous disconnection of two branches in the shared branch set $\mathbf{C}_{12}$, Node 2 becomes an island, rendering the topology infeasible. From Figure 3C, it can be seen that the simultaneous disconnection of branches in the shared branch sets $\mathbf{C}_{12}$, $\mathbf{C}_{13}$, and $\mathbf{C}_{23}$ results in Node 3, a common node among $\mathbf{L}_1$, $\mathbf{L}_2$, and $\mathbf{L}_3$, becoming an island, making the topology infeasible. However, topologies that exclude both Condition 1 and Condition 2 can satisfy radiation constraints and prevent islands from forming.
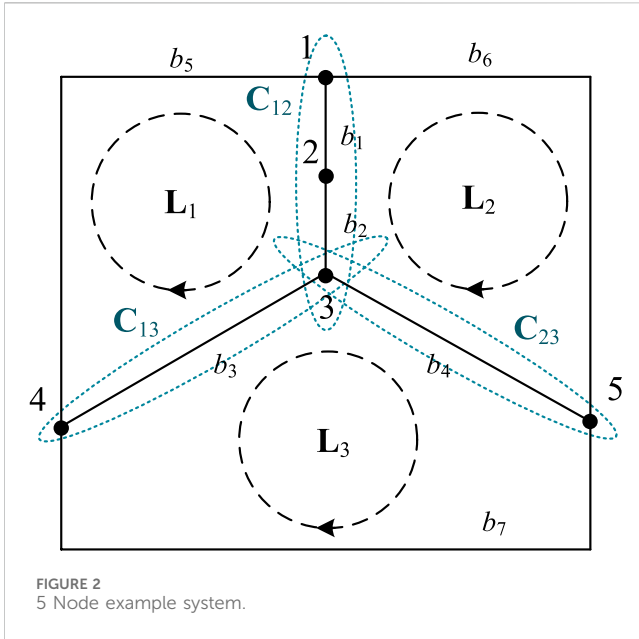
FIGURE 2
5 Node example system.

TABLE 1 The fundamental loops and shared branch sets for the 5-node system.

Loops		Branches
Fundamental Loops	L_1	b_1, b_2, b_3, b_5
	L_2	b_1, b_2, b_4, b_6
	L_3	b_3, b_4, b_7
Inter-loop shared branches	C_{12}	b_1, b_2
	C_{13}	b_3
	C_{23}	b_4

3 Data-driven power flow calculation model incorporating topological changes

This section develops the entire data-driven power flow calculation model based on the feasible topological set $\mathbf{s_{ap}}^\star$ established in the previous section.

3.1 The establishment of the overall model

A typical power flow calculation problem involves computing the voltage magnitudes and phase angles at various nodes under the condition of known injected power at each node. Assuming the set of active power injections at each node is denoted as $\mathbf{P}=\{P_1, P_2, \ldots, P_N\}$, the set of reactive power injections as $\mathbf{Q} = \{Q_1, Q_2, \ldots, Q_N\}$, the topological vector as $\mathbf{S}$, the set of voltage magnitudes as $\mathbf{U} = \{U_1, U_2, \ldots, U_N\}$, and the set of voltage phase angles as $\boldsymbol{\delta} = \{\delta_1, \delta_2, \ldots, \delta_N\}$ (where N is the number of nodes in the network), the inputs and outputs of the data-driven power flow calculation model can be defined as Eqs (1) and (2):

$$\mathbf{I} = \{\mathbf{P}, \mathbf{Q}, \mathbf{S}\} \tag{1}$$

$$\mathbf{O} = \{\mathbf{U}, \boldsymbol{\delta}\} \tag{2}$$

where $\mathbf{I}$ represents the set of input vectors for the data-driven model, $\mathbf{O}$ represents the set of output vectors for the data-driven model.

Unlike the potential loop-breaking strategy $\mathbf{s}_n$, the topological vector $\mathbf{S}$ records the switch states of each branch using a 0–1 representation, providing a more accurate reflection of the impact of branch switch states on power flow. Since the dimensions of the sets $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{U}$, and $\boldsymbol{\delta}$ are all N, while the dimension of $\mathbf{S}$ is N_B, the dimensions of the input vectors $\mathbf{I}$ and output vectors $\mathbf{O}$ are $2N + N_B$ and $2N$, respectively.

Based upon the definition of input and output variables, the entire data-driven power flow calculation model is further constructed using Deep Belief Networks (DBN) (Zhang et al., 2018). DBN, as a form of deep learning, consists of multiple layers of Restricted Boltzmann Machines (RBM) (Zhang et al., 2018; Tao et al., 2020; Wang et al., 2022). In this network architecture, there are connections between layers, but units within each layer are not interconnected. After training the neural network parameters layer by layer, DBNs are effective in fitting a large number of data samples, enabling estimation and prediction tasks (Wang et al., 2022).

The data-driven power flow calculation model based on DBN is shown in Figure 4. It can be seen that the input variables ($\mathbf{I}$) pass through the Input Layer, hidden layers, and output layer to ultimately yield the required power flow calculation results ($\mathbf{O}$).
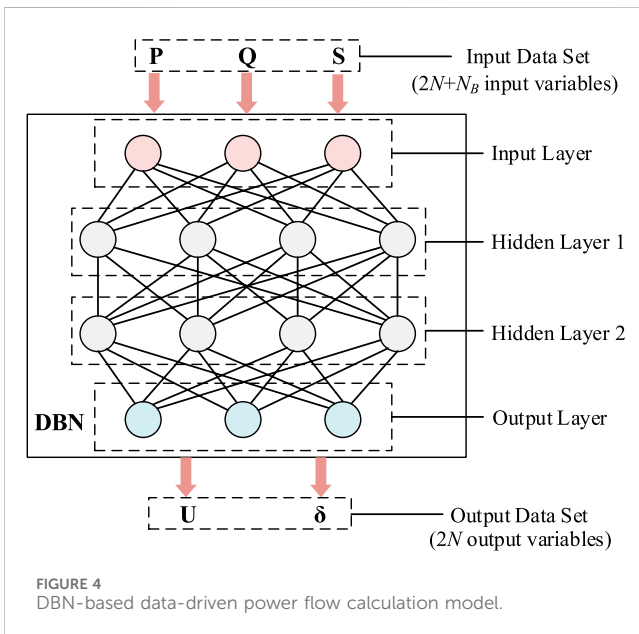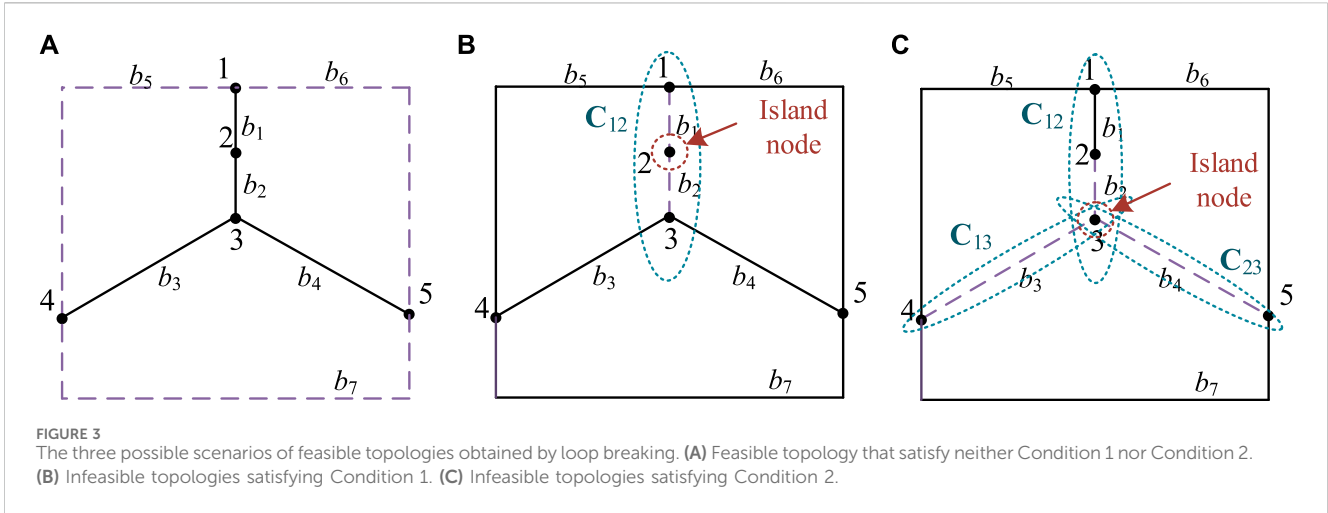
3.2 The training of the model

The data-driven power flow calculation model depicted in Figure 4 requires training before it can conduct rapid power flow calculations. Training samples play a crucial role in the accuracy of the power flow calculation model. In practical applications, training samples for the data-driven power flow calculation model can be sourced from actual measurements of power and voltage data in the electrical grid. However, on one hand, measured data samples are limited in quantity and may lack diversity, making it challenging to cover fewer common scenarios. On the other hand, from a model validation perspective, it is not easy to design a comprehensive validation method to verify the effectiveness of the model for various scenarios. In this paper, Monte Carlo simulations are employed to generate training samples.

Assuming that the rated active and reactive powers of the nodes in the network are represented as $\mathbf{P}_r = \{P_{r1}, P_{r2}, \ldots, P_{rN}\}$ and $\mathbf{Q}_r = \{Q_{r1}, Q_{r2}, \ldots, Q_{rN}\}$, respectively, the active and reactive power values for the kth node in the ith training sample can be obtained by adding noise to P_{rk} and Q_{rk} as follows:

$$P_{\text{train},i,k} = P_{r,k} + \varepsilon_{P,k} \tag{3}$$

$$Q_{\text{train},i,k} = Q_{r,k} + \varepsilon_{Q,k} \tag{4}$$

where $\varepsilon_{P,k}$ and $\varepsilon_{Q,k}$ represent the noise added to the active and reactive powers, respectively, and they follow a normal distribution:

FIGURE 3
The three possible scenarios of feasible topologies obtained by loop breaking. **(A)** Feasible topology that satisfy neither Condition 1 nor Condition 2. **(B)** Infeasible topologies satisfying Condition 1. **(C)** Infeasible topologies satisfying Condition 2.

FIGURE 4
DBN-based data-driven power flow calculation model.

$$\varepsilon_{\text{P},k} \sim N\left(0, \sigma_{\text{P},k}^2\right) \tag{5}$$

$$\varepsilon_{\text{Q},k} \sim N\left(0, \sigma_{\text{Q},k}^2\right) \tag{6}$$

where $\sigma_{\text{P},k}$ and $\sigma_{\text{Q},k}$ are the standard deviations of the noise. Through Eqs 3, 4, the sets of active and reactive powers for the ith training sample can be obtained as $\mathbf{P}_{\text{train},i} = \{P_{\text{train},i,1}, P_{\text{train},i,2}, \ldots, P_{\text{train},i,N}\}$ and $\mathbf{Q}_{\text{train},i} = \{Q_{\text{train},i,1}, Q_{\text{train},i,2}, \ldots, Q_{\text{train},i,N}\}$.

Topology set $\mathbf{S}_{\text{train},i}$ is obtained by sampling from the feasible topological set $\mathbf{s_{ap}}^\star$ and can be represented by the formula:

$$\mathbf{S}_{\text{train},i} \sim \mathbf{sap}^\star \tag{7}$$

Thus, the input variables for the ith training sample can be obtained as follows:

$$\mathbf{I}_{\text{train},i} = \{\mathbf{P}_{\text{train},i}, \mathbf{Q}_{\text{train},i}, \mathbf{S}_{\text{train},i}\} \tag{8}$$

Using $\mathbf{I}_{\text{train},i}$ and the network's own parameters, traditional power flow calculations can be performed. In this

case, the Newton-Raphson method is adopted to calculate the power flow for $\mathbf{I}_{\text{train},i}$, resulting in: $\mathbf{U}_{\text{train},i} = \{U_{\text{train},i,1}, U_{\text{train},i,2}, \ldots, U_{\text{train},i,N}\}$, $\boldsymbol{\delta}_{\text{train},i} = \{\delta_{\text{train},i,1}, \delta_{\text{train},i,2}, \ldots, \delta_{\text{train},i,N}\}$, which can be served as the output variables for the ith training sample in the data-driven power flow calculation model:

$$\mathbf{O}_{\text{train},i} = \{\mathbf{U}_{\text{train},i}, \boldsymbol{\delta}_{\text{train},i}\} \tag{9}$$

By repeatedly perform Formulas 3ormulas –Formulas 9, a large number of training samples can be generated. Eventually, a training sample set can be obtained as: $\mathbf{I}_{\text{train}} = \{\mathbf{I}_{\text{train},1}, \mathbf{I}_{\text{train},2}, \ldots, \mathbf{I}_{\text{train},Ntr}\}$, $\mathbf{O}_{\text{train}} = \{\mathbf{O}_{\text{train},1}, \mathbf{O}_{\text{train},2}, \ldots, \mathbf{O}_{\text{train},Ntr}\}$. Where Ntr is the number of training samples.

Using these training samples, the DBN can be trained, ultimately creating a data-driven power flow calculation model that takes into account topological changes and can rapidly compute power flow results.

4 Case study

The case study involves an IEEE 33-node distribution network system, as shown in Figure 5. This system comprises 33 nodes, 37 normally closed branches, and 10 branches equipped with controllable switches, denoted as b_4, b_7, b_{18}, b_{23}, b_{27}, b_{33}, b_{34}, b_{35}, b_{36}, b_{37}. The controllable branches allow for changes in the network topology while ensuring that the entire distribution network adheres to radiation constraints.

4.1 Topology sample set generation

From Figure 5, it can be observed that the IEEE 33-node system consists of a total of 5 fundamental loops. Since normally closed branches do not affect the network topology, only the branches equipped with controllable switches are listed in the loop data initialization. The results of loop data initialization are as shown in Table 2. Performing loop-breaking operations on the loop data yields a set of potential topologies, denoted as $\mathbf{s_{ap}}$, which contains a total of
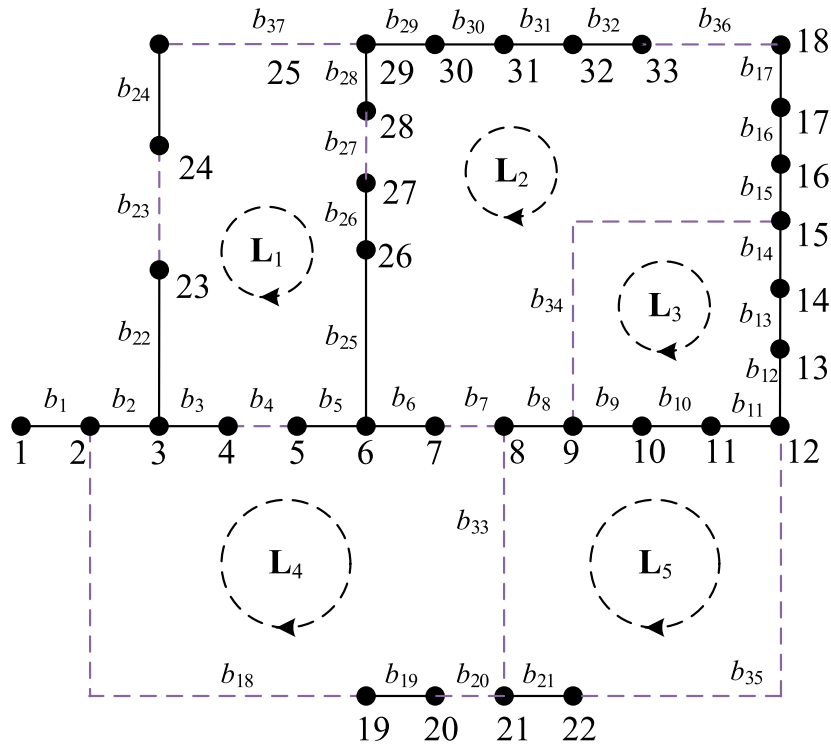
FIGURE 5
IEEE 33-node system.

TABLE 2 The basic loops and shared branches in the IEEE 33-node system (only listing branches equipped with controllable switches).

Loops		Branches
Fundamental Loops	L_1	$b_4, b_{23}, b_{27}, b_{37}$
	L_2	$b_7, b_{27}, b_{34}, b_{36}$
	L_3	b_{34}
	L_4	b_{18}, b_{20}, b_{33}
	L_5	b_{33}, b_{35}
Inter-loop shared branches	C_{12}	b_{27}
	C_{14}	b_4
	C_{23}	b_{34}
	C_{24}	b_3
	C_{45}	b_{33}

1024 potential topologies. Based on this set, loop backtracking operations are conducted to obtain the feasible topology set, denoted as $\mathbf{s_{ap}}^*$, which includes a total of 63 feasible topologies.

4.2 The power-flow calculation in the IEEE 33-node system with topology changes

The case study involves the validation of the data-driven power flow calculation method proposed in this paper for four different topological scenarios, as shown in the left column of Figure 6. To facilitate comparative analysis, the case study considers the following four methods:

1) Regression Method

Without considering the network's topology, a linear regression model is employed to establish the relationship between active power, reactive power, voltage magnitude, and phase angles.
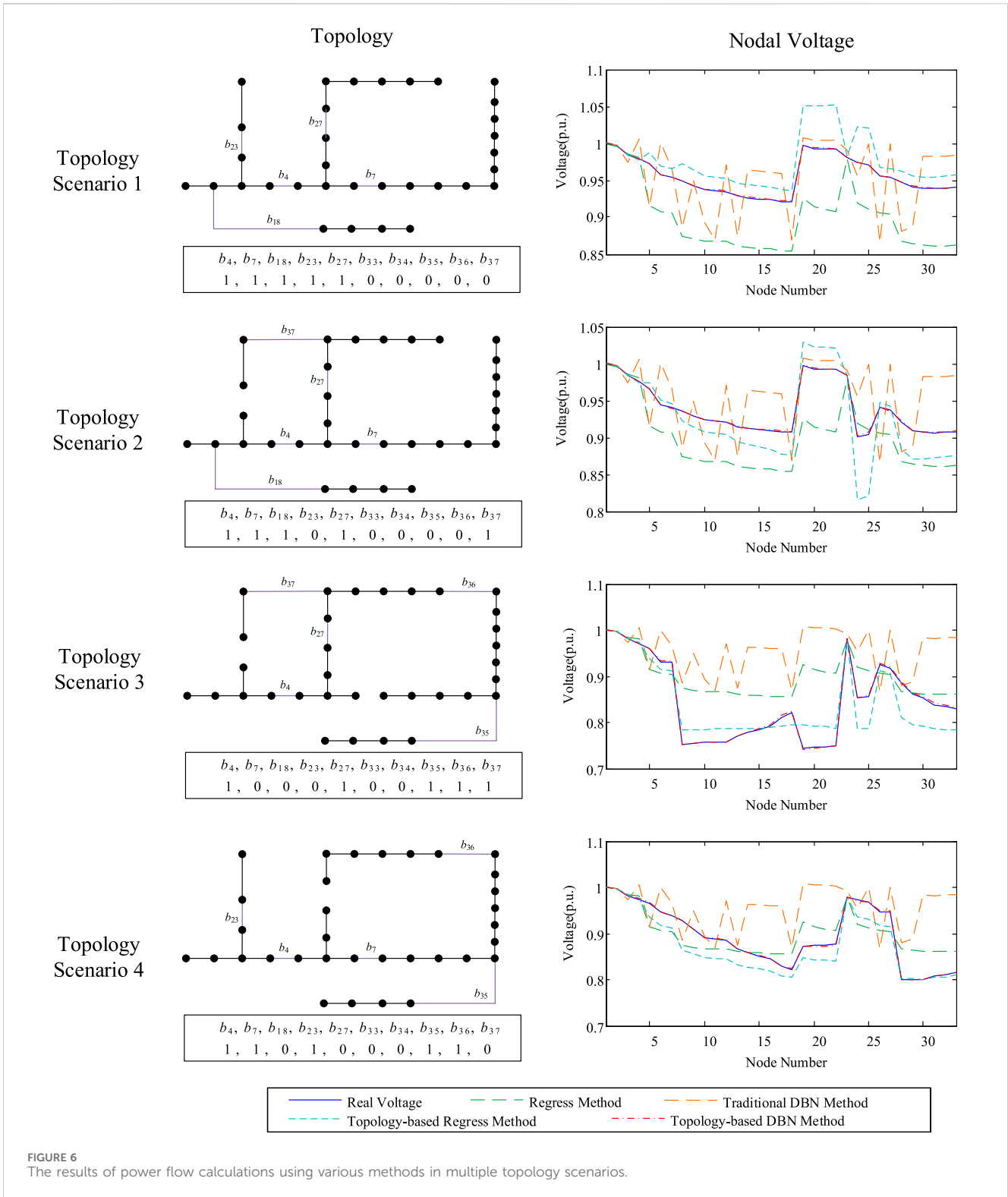
2) Traditional DRB Method

Without considering the network's topology, a Deep Belief Network (DBN) is used to establish the relationship between active power, reactive power, voltage magnitude, and phase angles.

3) Topology-based Regression Method

Considering the network's topology, a linear regression model is employed to establish the relationship between active power, reactive power, voltage magnitude, and phase angles while accounting for the network's topology.

4) Topology-based DRB Method (The proposed method)

The feasible topological sample set is generated using the undirected-graph delooping-backtracking method, and the DBN is used to establish the relationships between active power, reactive power, voltage magnitude, and phase angles while considering the

FIGURE 6
The results of power flow calculations using various methods in multiple topology scenarios.

network's topology. Due to the IEEE 33-node system comprising a total of 33 nodes and 10 controllable switches, according to the model described in Section 3.1, the dimensions of the input and output layers for the DBN are 76 and 66, respectively. In addition to the input and output layers, the DBN includes two hidden layers, each containing 500 neurons and the activation function chosen is the sigmoid function.

The comparison between the node voltages obtained from the four data-driven power flow calculation methods and the actual node voltages is shown in the right column of Figure 6. It can be seen that the Regression Method and the Traditional DRB Method do not consider the impact of topology changes on power flow, resulting in significant errors in the power flow calculation results. Topology-based Regression Method takes topology into account in its input
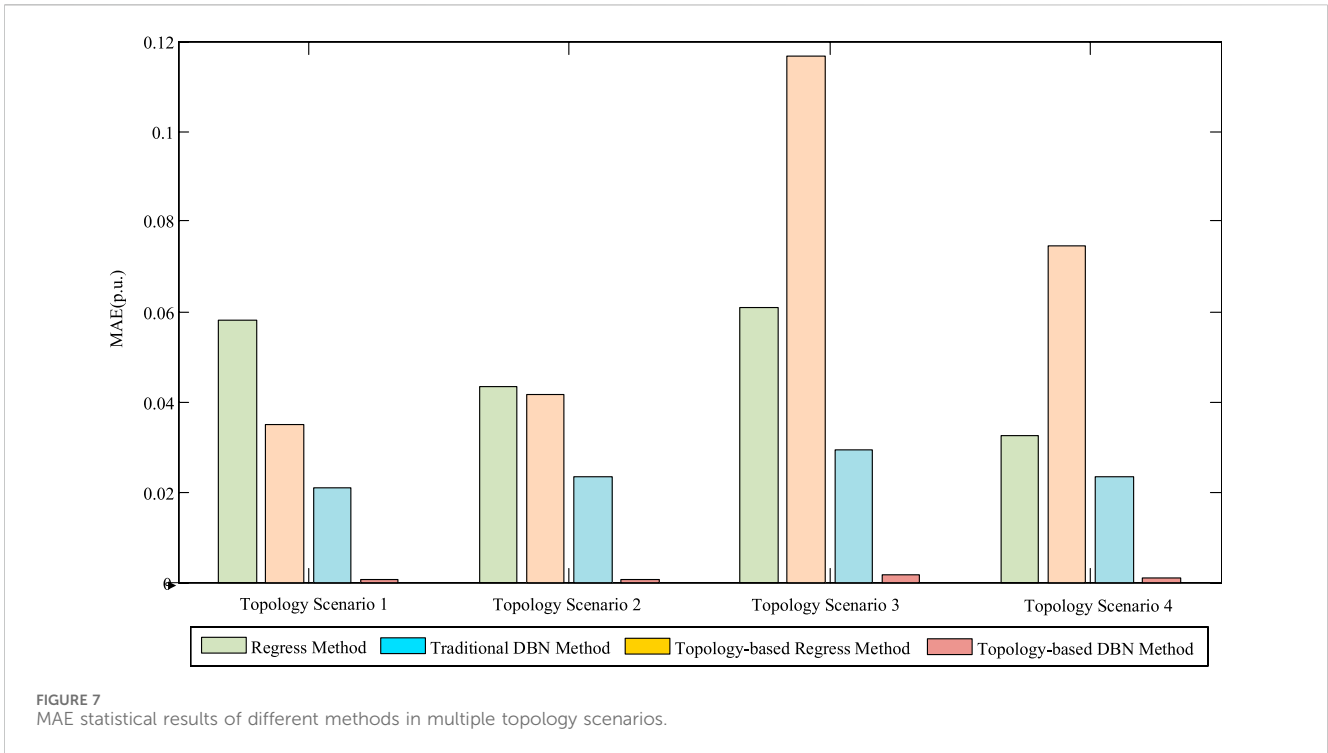
FIGURE 7
MAE statistical results of different methods in multiple topology scenarios.

TABLE 3 MAE statistics table with different methods.

Methods	MAE (p.u.)			
	Topology scenario 1	Topology scenario 2	Topology scenario 3	Topology scenario 4
Regression method	0.0581	0.0433	0.0607	0.0324
Traditional DRB method	0.0350	0.0414	0.1168	0.0746
Topology-based regression method	0.0207	0.0231	0.0293	0.0234
Topology-based DRB method (The proposed method)	0.0005	0.0006	0.0014	0.0008

variables, leading to a noticeable reduction in node voltage errors compared to the regression method. However, due to limitations in fitting nonlinear power equations, errors are still non-negligible. The Topology-based DRB Method (The proposed method), which combines variable topology and deep learning, output node voltage calculation results that are very close to the actual node voltages.

To further numerically compare the effectiveness of different data-driven power flow calculation methods, the Mean Absolute Error (MAE) is used to calculate the errors. The formula for calculating MAE is as Eq. (10):

$$MAE = \frac{\sum_{k=1}^{N}\left|U_{\text{test},k} - U_k\right|}{N} \qquad (10)$$

The bar chart in Figure 7 and the statistical results in Table 3 depict the MAE for power flow calculation results using different

methods across various topological scenarios. It is evident from Figure 7 and Table 3 that the MAE for node voltages obtained using the Topology-Based DRB Method (the proposed method) is significantly lower than that of the other three methods. This demonstrates that the proposed data-driven power flow calculation method proposed offers higher precision and is more suitable for scenarios involving network topology changes.

5 Conclusion

This paper introduces a data-driven power flow calculation method based on undirected-graph delooping-backtracking. It utilizes the undirected-graph delooping-backtracking technique to generate a feasible set of topological samples based on the initialization of loop data. Furthermore, it establishes a data-driven power flow calculation

model based on Deep Belief Networks (DBN). This approach enables accurate power flow calculations even in scenarios with changing network topologies. From the analysis results of the test case, we draw the following conclusions:

1) Through operations such as loop data initialization, loop-breaking, and loop backtracking, it is possible to effectively eliminate infeasible topologies and obtain feasible topologies on the basis of generating all potential topologies for the distribution network. For the 33-node testing system, by screening out infeasible topologies, the number of potential topologies is reduced from 1024 to 63, significantly reducing the computational complexity for training the data-driven power calculation model.

2) By incorporating topology information into the DBN-based numerical-driven power flow calculation model, it becomes possible to consider changes in the topology during the power flow calculation process. This allows the data-driven power flow calculation model to have an advantage over models that do not take topology information into account. Moreover, due to its excellent ability to learn complex features and handle large amounts of training data, the DBN achieves more accurate power flow calculation results compared to traditional neural networks.

The future work will focus on applying the proposed data-driven power flow calculation method to various aspects of power grid operations, including economic dispatch, topology optimization, and operational control, etc.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

References

Chen, J., Li, W., Wu, W., Zhu, T., Wang, Z., and Zhao, C. (2020). "Robust data-driven linearization for distribution three-phase power flow," in 2020 IEEE 4th Conference on Energy Internet and Energy System Integration (EI2), Wuhan, China, 30 October 2020 - 01 November 2020, 1527–1532. doi:10.1109/EI250167.2020.9347375

Chen, J., Wu, W., and Roald, L. (2022). Data-Driven piecewise linearization for distribution Three-Phase stochastic power flow. *IEEE Trans. Smart Grid* 13 (2), 1035–1048. doi:10.1109/tsg.2021.3137863

Chen, Y., Wu, C., and Qi, J. (2022). Data-driven power flow method based on exact linear regression equations. *J. Mod. Power Syst. Clean Energy* 10 (3), 800–804. doi:10.35833/mpce.2020.000738

Crozier, C., and Baker, K. (2022). "Data-driven probabilistic constraint elimination for accelerated optimal power flow," in 2022 IEEE Power &Amp; Energy Society General Meeting (PESGM), Denver, CO, USA, 17-21 July 2022, 1–5. doi:10.1109/pesgm48719.2022.9916838

Cui, X., Ma, K., Meng, X., Zhang, X., Duan, F., Li, T., et al. (2020). "A Data-Driven Slow Dynamic characteristic extraction and state estimation method for large power grid," in 2020 IEEE 3rd Student Conference on Electrical Machines and Systems (SCEMS), Jinan, China, 04-06 December 2020, 39–42. doi:10.1109/scems48876.2020.9352371

Guo, L., Zhang, Y., Li, X., Wang, Z., Liu, Y., Bai, L., et al. (2022). Data-driven power flow calculation method: a lifting dimension linear regression approach. *IEEE Trans. Power Syst.* 37 (3), 1798–1808. doi:10.1109/tpwrs.2021.3112461

Li, P., Wu, W., Wang, X., and Xu, B. (2023). A Data-Driven linear optimal power flow model for distribution networks. *IEEE Trans. Power Syst.* 38 (1), 956–959. doi:10.1109/tpwrs.2022.3216161

Liu, Y., Li, Z., and Zhao, J. (2022b). Robust data-driven linear power flow model with probability constrained worst-case errors. *IEEE Trans. Power Syst.* 37 (5), 4113–4116. doi:10.1109/tpwrs.2022.3189543

Liu, Y., Li, Z., and Zhou, Y. (2022a). Data-Driven-Aided linear Three-Phase power flow model for distribution power systems. *IEEE Trans. Power Syst.* 37 (4), 2783–2795. doi:10.1109/tpwrs.2021.3130301

Liu, Y., Zhang, N., Wang, Y., Yang, J., and Kang, C. (2019). Data-driven power flow linearization: a regression approach. *IEEE Trans. Smart Grid* 10 (3), 2569–2580. doi:10.1109/tsg.2018.2805169

Shao, Z., Zhai, Q., and Guan, X. (2023). Physical-model-aided data-driven linear power flow model: an approach to address missing training data. *IEEE Trans. Power Syst.* 38 (3), 2970–2973. doi:10.1109/tpwrs.2023.3256120

Tan, Y., Chen, Y., Li, Y., and Cao, Y. (2020). Linearizing power flow model: a hybrid physical model-driven and data-driven approach. *IEEE Trans. Power Syst.* 35 (3), 2475–2478. doi:10.1109/tpwrs.2020.2975455

Tao, C., Wang, X., Gao, F., and Wang, M. (2020). Fault diagnosis of photovoltaic array based on deep belief network optimized by genetic algorithm. *Chin. J. Electr. Eng.* 6, 106–114. doi:10.23919/cjee.2020.000024

Author contributions

HZ: Conceptualization, Investigation, Project administration, Supervision, Writing–review and editing. ZH: Investigation, Methodology, and Writing–Original Draft. ZL: Formal Analysis, Resources, Software, Writing–review and editing. YW: Resources, Validation, Visualization, Writing–review and editing.

Funding

Conflict of interest

Authors HZ, ZH, ZL, and YW were employed by Nanjing Power Supply Company, State Grid Jiangsu Electric Power Co., Ltd.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Wang, G., Qiao, J., Liu, C., and Shen, Z. (2022). How deep is deep enough for deep belief network for approximating model predictive control law. *IEEE Trans. Automation Sci. Eng.* 19, 2067–2078. doi:10.1109/tase.2021.3074984

Wang, J., Wang, C., Liang, Y., Bi, T., Shafie-khah, M., and Catalao, J. P. S. (2021). Data-driven chance-constrained optimal gas-power flow calculation: a bayesian nonparametric approach. *IEEE Trans. Power Syst.* 36 (5), 4683–4698. doi:10.1109/tpwrs.2021.3065465

Xing, Z., Gong, J., Lao, K. W., and Dai, N. (2021). "Single bus data-driven power estimation based on modified linear power flow model," in 2021 6th International Conference on Power and Renewable Energy (ICPRE), Shanghai, China, 17-20 September 2021, 755–758. doi:10.1109/ICPRE52634.2021.9635174

Xing, Z., Lao, K. W., Gao, H., and Dai, N. (2022). "A modified data-driven power flow model for power estimation with incomplete bus data," in 2022 12th International Conference on Power, Energy and Electrical Engineering (CPEEE), Shiga, Japan, 25-27 February 2022, 316–320. doi:10.1109/CPEEE54404.2022.9738684

Zhang, C., He, Y., Yuan, L., and Xiang, S. (2018). Analog Circuit Incipient Fault Diagnosis method using DBN based features extraction. *IEEE Access* 6, 23053–23064. doi:10.1109/access.2018.2823765

Zhang, Q., Li, X., Liu, X., Zhao, C., Shi, R., Jiao, Z., et al. (2022). "Data-driven risk assessment early-warning model for power system transmission congestions," in 2022 12th International Conference on Power, Energy and Electrical Engineering (CPEEE), Shiga, Japan, 25-27 February 2022, 201–206. doi:10.1109/CPEEE54404.2022.9738719

Zhu, Y., and Zhou, Y. (2023). "Data-driven multi-energy flow calculation method for integrated energy system," in 2023 IEEE 6th International Electrical and Energy Conference (CIEEC), Hefei, China, 12-14 May 2023, 501–505. doi:10.1109/CIEEC58067.2023.10166428