Data-driven Reactive Power Optimization for Distribution Networks Using Capsule Networks

Wenlong Liao, Jiejing Chen, Qi Liu, Ruijin Zhu, Like Song, and Zhe Yang

***Abstract*—The construction of advanced metering infrastructure and the rapid evolution of artificial intelligence bring opportunities to quickly searching for the optimal dispatching strategy for reactive power optimization. This can be realized by mining existing prior knowledge and massive data without explicitly constructing physical models. Therefore, a novel data-driven approach is proposed for reactive power optimization of distribution networks using capsule networks (CapsNet). The convolutional layers with strong feature extraction ability are used to project the power loads to the feature space to realize the automatic extraction of key features. Furthermore, the complex relationship between input features and dispatching strategies is captured accurately by capsule layers. The back propagation algorithm is utilized to complete the training process of the CapsNet. Case studies show that the accuracy and robustness of the CapsNet are better than those of popular baselines (e.g., convolutional neural network, multi-layer perceptron, and case-based reasoning). Besides, the computing time is much lower than that of traditional heuristic methods such as genetic algorithm, which can meet the real-time demand of reactive power optimization in distribution networks.**

***Index Terms*—Data-driven, reactive power optimization, distribution networks, deep learning, capsule networks.**

Nomenclature

δ_{ij}	Phase difference of voltage between the i^{th} node and the j^{th} node
$\varepsilon(\cdot)$	Step function
σ_i^{den}	Activation function of the i^{th} dense layer
σ_i^{con}	Activation function of the i^{th} convolutional layer
λ_1, λ_2	Penalty coefficients of voltage and current constraints
$*$	Convolutional operation
$\boldsymbol{B}_i^{\text{con}}$	Bias vector of the i^{th} convolutional layer
$\boldsymbol{B}_i^{\text{den}}$	Bias vector of the i^{th} dense layer
B_{ij}	Susceptance between the i^{th} node and the j^{th} node
$b_{i,j}$	Offset coefficient between the i^{th} and j^{th} primary capsule layers
$C_{i,j}$	Coupling coefficient between the i^{th} and j^{th} primary capsule layers
$d\boldsymbol{U}, d\boldsymbol{U}'$	Voltage offsets before and after reactive power optimization
f_1	Comprehensive objective function without penalty terms
f_2	Comprehensive objective function with penalty terms
G_{ij}	Conductance between the i^{th} node and the j^{th} node
$\boldsymbol{I}_i^{\max}$	The maximum current allowed to flow through the i^{th} branch
$\boldsymbol{I}_i$	Current of the i^{th} branch
$N_{i,\max}^{C}$	The maximum number of on/off switches for the i^{th} shunt capacitor bank in one day
$N_{i,t}^{C}$	Number of on/off switches for the i^{th} shunt capacitor bank at time t
$N_{i,t}^{T}$	Number of operations for the i^{th} on-load tap changer (OLTC) at time t
$N_{i,\max}^{T}$	The maximum number of operations for the i^{th} OLTC in one day
n_T	Number of positions with OLTC
n_{SVC}	Number of positions with static var compensator (SVC)
n_C	Number of positions with shunt capacitor bank
N	Total number of branches
n	Total number of nodes in distribution network
$P_{\text{loss}}, P'_{\text{loss}}$	Power losses before and after reactive power optimization
$Q_{\text{SVC}}^{\max}$	The maximum reactive power that SVC can generate
$Q_C^{\max}$	The maximum reactive power that can be generated by the i^{th} shunt capacitor bank
$Q_{\text{SVC}\,i,t}$	Reactive power generated by the i^{th} SVC at time t
$Q_{Ci,t}$	Reactive power generated by the i^{th} shunt ca-

Manuscript received: January 15, 2021; revised: April 23, 2021; accepted: August 30, 2021. Date of CrossCheck: August 30, 2021. Date of online publication: October 7, 2021.

W. Liao and Z. Yang are with the AAU Energy, Aalborg University, Aalborg, Denmark (e-mail: weli@energy.aau.dk; zya@energy.aau.dk).

J. Chen is with the School of Software and Microelectronics, Peking University, Beijing, China (e-mail: 1601210372@pku.edu.cn).

Q. Liu (corresponding author) is with the Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin, China (e-mail: liuqi1619@163.com).

R. Zhu is with the School of Electrical Engineering, Tibet Agriculture and Animal Husbandry University, Linzhi, China (e-mail: zhuruijin@xza.edu.cn).

L. Song is with the Maintenance Branch of State Grid Jibei Electric Power Co., Ltd., Beijing, China (e-mail: 2429747164@qq.com).

DOI: 10.35833/MPCE.2021.000033

	pacitor bank at time t
$\boldsymbol{S}_j$	Input vector of the j^{th} digital capsule layer
$\boldsymbol{T}_i^{\min}$, $\boldsymbol{T}_i^{\max}$	The minimum and the maximum tap positions of the i^{th} OLTC
$T_{i,t}$	Tap position of the i^{th} OLTC at time t
$\boldsymbol{U}_i^{\max}$, $\boldsymbol{U}_i^{\min}$	Upper and lower voltage limits of the i^{th} node
$\boldsymbol{U}_i^{\text{con}}$	Output data of the i^{th} convolutional layer
$\boldsymbol{U}_{j,i}^{\text{cap}}$	The j^{th} vector generated by the i^{th} input feature
$\boldsymbol{U}_N$	Rated voltage of distribution network
$\boldsymbol{U}_i$	Voltage of the i^{th} node
$\boldsymbol{W}_i^{\text{con}}$	Weight of the i^{th} convolutional layer
$\boldsymbol{W}_{j,i}^{\text{cap}}$	Weights of the primary capsule layers
W_{p}, W_{u}	Weights of power loss and voltage offset
$\boldsymbol{W}_i^{\text{den}}$	Weight of the i^{th} dense layer
$\boldsymbol{X}_i^{\text{con}}$	Input data of the i^{th} convolutional layer
$\boldsymbol{Y}_i^{\text{den}}$	Output data of the i^{th} dense layer
$y_{\text{e, OLTC}, i}$	Coding value of the i^{th} position
$y_{\text{d, OLTC}, i}$	Decoding value of the i^{th} position
$y_{\text{e, SVC}}$	Coding value of reactive power $\boldsymbol{Q}_{\text{SVC}}$
$y_{\text{d, SVC}}$	Reactive power
$\boldsymbol{Z}_j$	Output vector of the j^{th} digital capsule layers
$\boldsymbol{Z}_i^{\text{den}}$	Input data of the i^{th} dense layer

I. Introduction

REACTIVE power optimization is an effective means to change the power flows of distribution networks by adjusting the operation status of various power equipment at a given load level, so as to reduce power loss and improve the power quality. As an important part of distribution network scheduling and planning, reactive power optimization of distribution networks is of great significance for both theoretical research and practical applications [1].

Nowadays, the world faces more and more serious environmental problems and energy crisis. The traditional fossil energy gradually dries up, and the penetration of renewable energy sources (e. g., wind turbines and photovoltaic systems) in distribution networks continuously increases [2]. Although these renewable energy sources are abundant, sustainable, and environmentally friendly, their randomness and volatility also bring challenges to the safe operation of distribution networks [3]. In order to alleviate the negative impact of renewable energy sources on distribution networks, the types and quantity of power equipment integrated into distribution networks are increasing. In this case, a large number of renewable energy sources and power equipment make the physical models of reactive power optimization tend to be complex. Traditional methods for reactive power optimization of distribution networks need to assume and simplify these physical models, and then find the approximate optimal solution through iteration [4]. However, the integration of a large number of renewable energy sources and power equipment greatly increases the complexity of physical models. Traditional methods not only consume a lot of computing time, but also easily fall into the local optimal solution [5]. To ensure the economy and stability of distribution networks, there is an urgent need to develop a new approach independent of physical models, with high accuracy, strong adaptability, and fast computing speed.

With the rapid development of sensors and communication technologies, the amount of data in all walks of life shows an explosive increase, which has attracted the attention of experts and scholars in various fields [6]. Distribution networks are located at the end of the power system, which directly distribute electric energy to users. Its supervisory control and data acquisition (SCADA) system stores a large amount of historical load data. Since these power loads are collected from the adjacent regions, electrical users may have similar electricity consumption habits [7]. Moreover, previous publications have shown that there is a strong temporal dependency in the load curve, i. e., the current loads are relative to the historical loads [8], [9]. In this case, the dispatching strategy of historical cases may be used for the current case after fine-tuning parameters. Generally, massive historical data bring opportunities to the application of data-driven technologies for reactive power optimization of distribution networks. For example, some deep neural networks may be used to project the non-linear relationship between power loads and dispatching strategies.

The core idea of data-driven technologies is to mine the existing prior knowledge and massive data by supervised learning. Then, some valuable information is extracted to direct the reactive power optimization of distribution networks. The existing data-driven technologies for reactive power optimization of distribution networks can be summarized into two categories: similarity-based methods and model-based methods. For the first category, it mainly includes large random matrix theory, Apriori algorithm, expert system, and case-based reasoning (CBR) [10], which aims to compute the similarity metrics between historical samples and new samples. Specifically, the large random matrix theory calculates the similarity metrics between the forecasting random matrix and the real random matrix, so as to find the dispatching strategies of current power loads that are similar to historical power loads [11]. Similarly, the Apriori algorithm filters out the optimal dispatching strategy of the historical samples for the current sample based on frequent item set mining and association rule [12]. For the expert system, the user inputs the power loads of the distribution networks based on the human-machine interface, and then the inference engine matches the power loads with dispatching strategy in the knowledge base. The conclusions of the matched dispatching strategy are sent out to the system operators [13]. CBR searches historical cases similar to current cases, and then uses them to determine the dispatching strategy of unknown samples [14]. Generally, although these similarity-based methods make good use of prior knowledge by calculating distances between historical samples and current samples, they have difficulty in mining the complex non-linear relationship between power loads and dispatching strategies, resulting in their limited accuracy for reactive power optimization. Moreover, the spread diffusion of distributed generations and market-based behaviors of end-users may change the spatio-temporal correlation among the load curves, and the current load distribution and the historical load distribution may be dramatically different. Therefore, it is not appropriate to apply the dispatching strategies of historical cases to the current cases directly. For the second category, traditional model-based methods mainly include support vector machine, multi-layer perceptron (MLP), and deep neural networks such as the deep belief network (DBN) and convolutional neural network (CNN) [15], which train a model to predict suitable dispatching strategies by a lot of historical samples. Unlike similarity-based methods, the model-based methods are not to find a suitable historical dispatching strategy, but to create a new one for the current case. Specifically, supportive vector machine (SVM) is a supervised learning method for reactive power optimization by constructing a decision surface, where the difference between two different classes can be maximized [16]. MLP, DBN, and CNN construct neural networks using dense layers, restricted Boltzmann machines, and convolutional layers, respectively, to represent the relationship between power loads and dispatching strategies. Compared with MLP and DBN, CNN has higher accuracy in reactive power optimization, since the convolutional layers have more powerful feature extraction ability than dense layers and restricted Boltzmann machines [17]. However, the pooling operation of CNN will lose rich information of power loads, which restricts its accuracy to be further improved [18].

The capsule network (CapsNet) is a new deep neural network that originates from CNN. Compared with traditional CNN, CapsNet shows a much deeper semantic understanding of the scenario, and a stronger ability to represent high dimensional features, leading to its wide applications in various fields [19]. For example, a spectral-spatial CapsNet is proposed to significantly reduce the network complexity, and achieves highly accurate classification of hyper spectral imaging data in [20]. While in [21], CapsNet is designed to mine spatial relationships specialized in synthetic aperture radar automatic target recognition tasks. The simulation results show that CapsNet not only improves the accuracy of recognition, but also alleviates the computational burden. To improve the accuracy of gait recognition, CapsNet with convolutional and capsule layers is proposed to capture more discriminative features in [22]. The successful applications of CapsNet in images, videos, and speech signals prove that CapsNet is able to explore complex objective laws from high-dimensional data through supervised learning. Theoretically, the latent representations of power loads can be effectively extracted by deep convolutional layers with strong learning ability, and the complex non-linear relationship between dispatching strategies and power loads can be explored to improve the accuracy for reactive power optimization based on primary capsule layers and digital capsule layers.

However, most of the existing architectures of CapsNet are designed for computer vision (e. g., image and video) [23], which cannot handle the 1-dimensional power loads directly. Therefore, how to design a structure of CapsNet with high accuracy and strong feature extraction ability given data from distribution networks needs further research.

This paper aims to design a CapsNet to improve the accuracy of reactive power optimization for distribution networks. The performance of the proposed method is tested by the IEEE 33-bus radial distribution network and the IEEE 69-bus radial distribution network. The key contributions of this paper are as follows.

1) A new data-driven, model-free, and scalable method is proposed for reactive power optimization of distribution networks. The proposed CapsNet does not need to construct complex physical models explicitly. It makes full use of historical data and prior knowledge to quickly find dispatching strategies with similar accuracy to genetic algorithm (GA). In addition, the time cost is much lower than GA.

2) The pooling layers that lose rich feature information are replaced by capsule layers, which can more accurately explore the non-linear relationship between input features and dispatching strategies. Simulation results show that CapsNet has higher optimization accuracy in different volumes of the training set and stronger robustness than those of CBR, CNN, and MLP at different load levels.

3) This paper innovatively design the CapsNet with strong feature extraction capability and high optimization accuracy for reactive power optimization, according to the characteristics of the power loads from distribution networks. The influence of key parameters of CapsNet (e.g., the number of convolutional and capsule layers, batch size, the number of iterations, and the choice of optimizer) on the performance for reactive power optimization is analyzed by simulations on a smart meter dataset, and some default values of parameters are given.

The rest of this paper is organized as follows. Section II introduces the structure and parameters of the CapsNet. Section III formulates the reactive power optimization model. Section IV performs the simulations and analyzes the results. Section V summarizes the limitation and future works of the proposed method. Section VI presents the conclusions.

II. CapsNet

As shown in Fig. 1, CapsNet is a high-performance deep neural network, which is mainly composed of convolutional layers, capsule layers, and dense layers. Firstly, the convolutional layers with strong learning ability are used to extract latent representations of power loads (active power and reactive power), which serve as the input data of primary capsule layers. In the primary capsule layer, each capsule can be regarded as a neuron group. The weights between primary capsule layers and digital capsule layers are trained by the dynamic routing algorithm to get the output vector. Finally, the vectors from digital capsule layers are fed to dense layers to obtain the dispatching strategies.

A. Convolutional Layers

The main function of convolutional layers is to extract the latent representations of power loads by the convolutional operation. The feature matrix of power loads is obtained by multiplying the matrix elements with the convolutional kernel, and then adding an offset vector. Compared with the dense layers, convolutional layers have stronger feature mining ability and avoid the complex selection process caused by manual construction of input features. The relationship between input features and latent representations is given as:

$$\boldsymbol{U}_i^{\text{con}}=\sigma_i^{\text{con}}(\boldsymbol{X}_i^{\text{con}}*\boldsymbol{W}_i^{\text{con}}+\boldsymbol{B}_i^{\text{con}}) \tag{1}$$

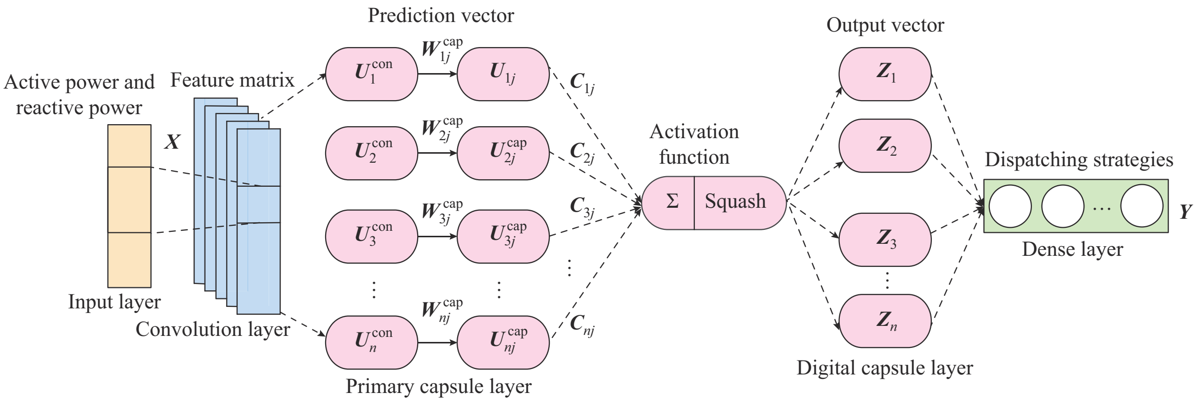

Prediction vector
Active power and reactive power
Feature matrix
$\boldsymbol{X}$
Input layer
Convolution layer
$\boldsymbol{U}_1^{\text{con}}$
$\boldsymbol{U}_2^{\text{con}}$
$\boldsymbol{U}_3^{\text{con}}$
$\boldsymbol{U}_n^{\text{con}}$
$\boldsymbol{W}_{1j}^{\text{cap}}$
$\boldsymbol{W}_{2j}^{\text{cap}}$
$\boldsymbol{W}_{3j}^{\text{cap}}$
$\boldsymbol{W}_{nj}^{\text{cap}}$
$\boldsymbol{U}_{1j}$
$\boldsymbol{U}_{2j}^{\text{cap}}$
$\boldsymbol{U}_{3j}^{\text{cap}}$
$\boldsymbol{U}_{nj}^{\text{cap}}$
$\boldsymbol{C}_{1j}$
$\boldsymbol{C}_{2j}$
$\boldsymbol{C}_{3j}$
$\boldsymbol{C}_{nj}$
Primary capsule layer
Activation function
Σ
Squash
Output vector
$\boldsymbol{Z}_1$
$\boldsymbol{Z}_2$
$\boldsymbol{Z}_3$
$\boldsymbol{Z}_n$
Digital capsule layer
Dispatching strategies
Dense layer
$\boldsymbol{Y}$

Fig. 1. Framework of CapsNet.

B. Capsule Layers

The capsule layers include primary capsule layers and digital capsule layers, in which the input scalar and output scalar of traditional deep neural networks are substituted with the input vector and output vector of capsule layers. The calculation of capsule layers includes two stages.

In the first stage, the prediction vector is obtained by multiplying the output features of convolutional layers with the weight matrix. The mathematical formula is:

$$\boldsymbol{U}_{j,i}^{\text{cap}}=\boldsymbol{U}_i^{\text{con}}\boldsymbol{W}_{j,i}^{\text{cap}} \tag{2}$$

In (2), $\boldsymbol{W}_{j,i}^{\text{cap}}$ are usually initialized by Gaussian noises.

In the second stage, the prediction vectors of primary capsule layers are sent out to the digital capsule layers by adjusting the coupling coefficient, whose mathematical formula are given as:

$$C_{i,j}=Softmax(b_{i,j})=\frac{\exp(b_{i,j})}{\sum_k \exp(b_{i,k})}\geq 0 \quad \sum_j C_{i,j}=1 \tag{3}$$

$$\boldsymbol{S}_j=\sum_i \boldsymbol{U}_{j,i}^{\text{cap}}C_{i,j} \tag{4}$$

$\boldsymbol{S}_j$ can be generated by multiplying coupling coefficients by prediction vectors of primary capsule layers. When $C_{i,j}=1$, all the generated features of the primary capsule layers are sent out to next digital capsule layers. When $C_{i,j}=0$, all the generated features of the primary capsule layers are not sent out to next digital capsule layers.

Unlike the CNN or MLP, which often uses other functions (e.g., sigmoid function, rectified linear unit (ReLU) function, hyperbolic tangent function) as activation function, CapsNet uses the squash function to obtain the output vector $\boldsymbol{Z}_j$ of digital capsule layers. The squash function $squash(\cdot)$ can be represented as:

$$\boldsymbol{Z}_j=squash(\boldsymbol{S}_j)=\frac{\left\|\boldsymbol{S}_j\right\|^2}{1+\left\|\boldsymbol{S}_j\right\|^2}\frac{\boldsymbol{S}_j}{\left\|\boldsymbol{S}_j\right\|} \tag{5}$$

The squash function can not only keep the direction of the input vector, but also project the modulus of the input vector into new values varying from 0 to 1. Further, Fig. 2 visualizes the relationship between the modulus of the input vector and reduction coefficient of squash function. As the modulus of vectors increases, the reduction coefficients get closer to 1, as shown in Fig. 2.

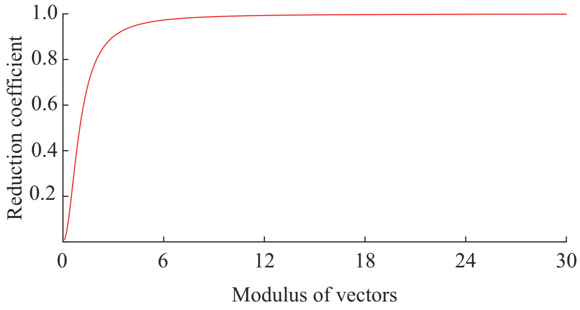


Fig. 2. Relationship between modulus of input vectors and reduction coefficient of squash function.

C. Dense Layers

Finally, the vectors from digital capsule layers are fed to dense layers to obtain dispatching strategies, whose output data are expressed as:

$$\boldsymbol{Y}_i^{\text{den}}=\sigma_i^{\text{den}}(\boldsymbol{Z}_i^{\text{den}}\boldsymbol{W}_i^{\text{den}}+\boldsymbol{B}_i^{\text{den}}) \tag{6}$$

D. Dynamic Routing Algorithm

After initializing the structure of the CapsNet, the dynamic routing algorithm is employed to update the parameters in capsule layers through the supervised training process, so that the rich information from the primary capsule layer is sent out to the digital capsule layers. The specific pseudo code of the dynamic routing algorithm is shown in Algorithm 1.

Firstly, the number of maximum iteration R is set and the bias coefficients $\boldsymbol{b}_{i,j}$ are initialized with 0 elements. Secondly, the coupling coefficients are calculated by (3), and $\boldsymbol{S}_j$ is obtained by linear weighting with $\boldsymbol{U}_{j,i}^{\text{cap}}$. Thirdly, the output vector $\boldsymbol{Z}_j$ can be obtained by inputting $\boldsymbol{S}_j$ to the squash activation function. Lastly, the output vector $\boldsymbol{Z}_j$ is utilized to update the bias coefficients $\boldsymbol{b}_{i,j}$, which can be obtained as:

$$\boldsymbol{b}_{i,j}=\boldsymbol{b}_{i,j}+\boldsymbol{Z}_j\boldsymbol{U}_{j,i}^{\text{cap}} \tag{7}$$

Algorithm 1: pseudo code of dynamic routing algorithm

```
initialize the number of maximum iterations $R$
    $b_{ij}=0$; $Step=0$
while ($Step<R$)
    $C_{i,j}=Softmax(b_{i,j})$
    $\boldsymbol{S}_j=\sum_i \boldsymbol{U}_{j,i}^{cap} C_{i,j}$
    $\boldsymbol{Z}_j=squash(\boldsymbol{S}_j)$
    $b_{i,j}=b_{i,j}+\boldsymbol{Z}_j \boldsymbol{U}_{j,i}^{\text{cap}}$
    $Step=Step+1$
end
Return $\boldsymbol{Z}_j$
```

E. Reactive Power Optimization Model via CapsNet Reactive Power Optimization Model

This paper takes the minimum power loss and voltage offset as the objective function to construct the reactive power optimization model of the distribution networks. The comprehensive objective function can be defined as the largest change in the power loss and voltage offset before and after optimization [24]:

$$\max f_1 = W_{\text{p}} \frac{P_{\text{loss}} - P'_{\text{loss}}}{P_{\text{loss}}} + W_{\text{u}} \frac{d\boldsymbol{U} - d\boldsymbol{U}'}{d\boldsymbol{U}} \tag{8}$$

$$d\boldsymbol{U} = \sum_{i=1}^{n} \left| \frac{\boldsymbol{U}_N - \boldsymbol{U}_i}{\boldsymbol{U}_N} \right| \tag{9}$$

The weights W_{p} and W_{u} can be adjusted flexibly. In this paper, the power loss and voltage offset are given the same weight (W_{p} and W_{u} are equal to 1) as an example to test the performance of different methods.

In addition, the reactive power optimization model of distribution networks should meet the following constraints.

1) Power Flow Constraints

$$\begin{cases} P_i - \boldsymbol{U}_i \sum_{j=1}^{n} \boldsymbol{U}_j (G_{ij}\cos\delta_{ij} + B_{ij}\sin\delta_{ij}) = 0 & i=1,2,\ldots,n \\ Q_i - \boldsymbol{U}_i \sum_{j=1}^{n} \boldsymbol{U}_j (G_{ij}\sin\delta_{ij} - B_{ij}\cos\delta_{ij}) = 0 & i=1,2,\ldots,n \end{cases} \tag{10}$$

2) Voltage and Current Constraints

$$\begin{cases} \boldsymbol{U}_i^{\min} \le \boldsymbol{U}_i \le \boldsymbol{U}_i^{\max} & i=1,2,\ldots,n \\ \boldsymbol{I}_i \le \boldsymbol{I}_i^{\max} & i=1,2,\ldots,N \end{cases} \tag{11}$$

For the voltage and current constraints, the constrained model is transformed into an unconstrained model by employing the penalty function method:

$$\max f_2 = f_1 - \lambda_1 \sum_{i=1}^{n} \left[\varepsilon(\boldsymbol{U}_i - \boldsymbol{U}_i^{\max}) + \varepsilon(\boldsymbol{U}_i^{\min} - \boldsymbol{U}_i) \right] - \lambda_2 \sum_{i=1}^{N} \varepsilon(\boldsymbol{I}_i - \boldsymbol{I}_i^{\max}) \tag{12}$$

The penalty coefficients should be much larger than f_1, and the penalty coefficients are equal to 10 in this paper. If the voltages and currents are within the boundary, f_1 and f_2 are equal.

3) Power Equipment Constraints

$$\begin{cases} 0 \le \boldsymbol{Q}_{Ci,t} \le \boldsymbol{Q}_C^{\max}, \sum_{t=1}^{24} N_{i,t}^C < N_{i,\max}^C & i=1,2,\ldots,n_C \\ \boldsymbol{T}_i^{\min} \le \boldsymbol{T}_{i,t} \le \boldsymbol{T}_i^{\max}, \sum_{t=1}^{24} N_{i,t}^T < N_{i,\max}^T & i=1,2,\ldots,n_T \\ 0 \le \boldsymbol{Q}_{\text{SVC}\,i,t} \le \boldsymbol{Q}_{\text{SVC}}^{\max} & i=1,2,\ldots,n_{\text{SVC}} \end{cases} \tag{13}$$

F. Coding Method of Power Equipment

The reactive power optimization of distribution networks is to search the suitable reactive power dispatching strategies within a certain range of load fluctuations to minimize the power loss and the voltage offset of each node. Its key is to construct the relationship between the power loads (active powers and reactive powers) and dispatching strategies by using the CapsNet.

The mainstream power equipment for reactive power optimization includes on shunt capacitor bank, on-load tap changer (OLTC), and SVC. The number of shunt capacitor banks, tap position in OLTC, and output reactive power of the SVC are variables to be predicted by the CapsNet. Obviously, shunt capacitor banks and OLTC are discrete variables, while the SVC is a continuous variable. If these variables are encoded using the binary encoding method, the number of encodings will be extremely long, which is not conducive to training the CapsNet. Therefore, this paper uniformly uses value encoding methods.

For discrete variables, the OLTC with m positions is taken as an example to illustrate the value coding method. The coding value of the i^{th} position is:

$$y_{\text{e,OLTC},i} = \frac{2i-1}{2m} \quad i=1,2,\ldots,m \tag{14}$$

where $y_{\text{e,OLTC},i}$ is less than 1 and greater than 0. Similarly, the positions $y_{\text{e,OLTC},i}$ predicted by the CapsNet can be decoded as:

$$y_{\text{d,OLTC},i} = \begin{cases} 1 & 0 \le y_{\text{e,OLTC},i} \le \dfrac{i}{m} \\ i & \dfrac{i-1}{m} < y_{\text{e,OLTC},i} \le \dfrac{i}{m} \end{cases} \tag{15}$$

where $y_{\text{d,OLTC},i}$ is less than m and greater than 1.

For continuous variables, the coding value of the reactive power is:

$$y_{\text{e,svc}} = \frac{Q_{\text{SVC}}}{Q_{\text{SVC}}^{\max}} \tag{16}$$

Similarly, the reactive power $y_{\text{e,svc}}$ predicted by the CapsNet can be decoded:

$$y_{\text{d,svc}} = y_{\text{e}} Q_{\text{SVC}}^{\max} \tag{17}$$

where $y_{\text{d,svc}}$ is less than $Q_{\text{SVC}}^{\max}$ and greater than 0.

So far, a bijective relationship has been established among the number of shunt capacitor banks, tap position in OLTC, and output reactive power of the SVC with the real numbers in the range of 0 to 1. For k variables to be optimized, the output layer at the top of the CapsNet outputs k numbers, which represent the optimized results. The mean absolute error (MAE) is used as the loss function and the weight of the network is updated by the gradient descent method.

G. Process of Reactive Power Optimization

The process of reactive power optimization based on the CapsNet is shown in Fig. 3, and the specific steps are as follows.

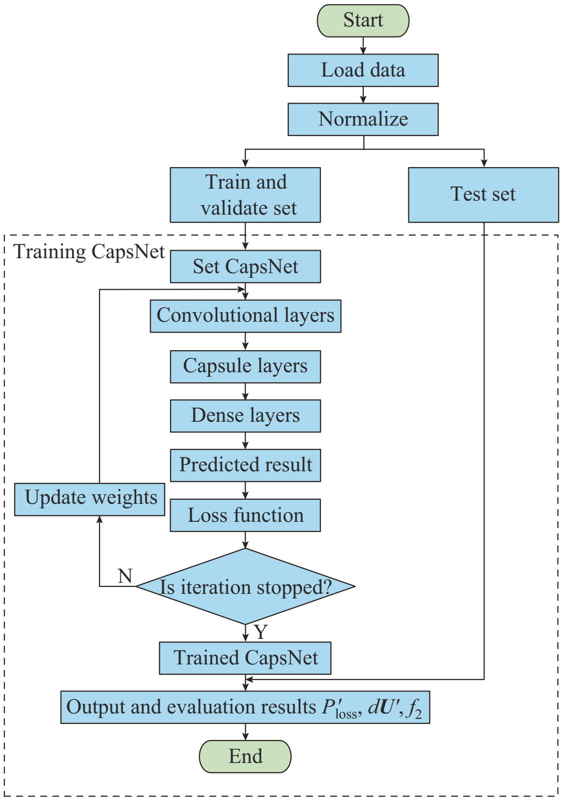


Fig. 3. Process of reactive power optimization based on CapsNet.

1) Load and normalize data. The topology of distribution networks, the historical power loads of each node, and the corresponding dispatching strategies of the power equipment are imported. The samples are randomly divided into training set, validation set, and test set. The training set and validation set are used to obtain the parameters of the CapsNet, and the test set is used to evaluate the performance of the CapsNet. To ensure the convergence of CapsNet, power loads are projected into the range of 0 to 1 by using the minimum-maximum normalization method.

2) Initialize the structure and parameters of the CapsNet. To improve the accuracy of reactive power optimization, it is necessary to explore suitable network structures and parameters before training the CapsNet. The structures and parameters of the CapsNet mainly include: the number of convolutional layers and capsule layers, the number of iterations, the batch size, and the selection of optimizer. Taking two convolutional layers and one capsule layer as examples, Table II shows a simple structure of the CapsNet. The size of convolutional filters is 8 and 16, respectively, the size of the convolutional kernels is 2 × 2, and the activation function is ReLU function. The capsule layer outputs two vectors of 1 × 16 scales, and the activation function is a squash function. In the dense layer, a vector of 1 × 4 scales is obtained to represent dispatching strategies of the power equipment.

3) Train the CapsNet. The CapsNet is trained by the back propagation algorithm, which consists of forward incentive propagation and backward weight update. In the forward incentive propagation, the power loads are processed by convolutional layers and capsule layers, and then transferred to the output layer, which outputs dispatching strategies of samples. The forecasting results and optimal results are used to calculate the loss function (e.g., MAE). In backward weight update, the loss function is transferred from the output layer to middle layers using the chain rule. Next, the weights of convolutional layers and capsule layers are updated by the gradient descent method. If the iteration is over, the test set is used to evaluate the performance of the trained CapsNet.

4) Evaluate the CapsNet. After training the CapsNet, it will be used for reactive power optimization. The power loads of nodes from the test set are input into the CapsNet to obtain the corresponding dispatching strategies. Then, the power flow of distribution networks is calculated to obtain the indices such as power loss P'_{loss}, voltage offset $d\boldsymbol{U}'$, and comprehensive objective function f_2.

TABLE II
A Simple Structure of CapsNet

Layer	Structure and parameter	Shape
1	X = Input(shape = (Powerload.shape))	1 × 64
2	Y = X.reshape(−1,8,8,1)	8 × 8 × 1
3	Y = Conv2D(filters = 8, kernel_size = 2, ‘ReLU’)(Y)	7 × 7 × 8
4	Y = Conv2D(filters = 16, kernel_size = 2, ‘ReLU’)(Y)	6 × 6 × 16
5	Y = Y.reshape(−1,36,16)	36 × 16
6	Y = Capsule(num_capsule = 2, dim = 16, ‘squash’)(Y)	2 × 16
7	Y = dense(unit = 4, ‘Sigmoid’)(Y)	1 × 4

III. Case Study

A. Dataset and Simulation Tools

In order to fully test the performance of the proposed method, the modified IEEE 33-bus radial distribution network is used for simulation and analysis. The resistance and reactance of the branch can be found in [25], and the topology is shown in Fig. 4.

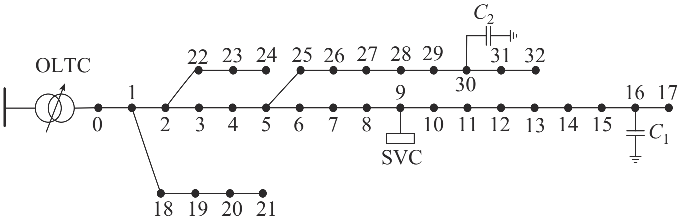


Fig. 4. Topology of modified IEEE 33-bus radial distribution network.

Specifically, the voltage magnitude base is 10 kV. The tap of the OLTC includes 17 positions, which range from −8 × 1.25% to 8 × 1.25%. Normally, SVCs and capacitor banks are located at the end of the feeder to improve the voltages and reduce the power loss [26], and the decentralized locations of these devices help reduce the voltage offset [27]. Therefore, the locations and capacities of SVCs and capacitor banks are assumed as follows. The SVC is added at the

9th node and the reactive power of the SVC ranges from 0 to 400 kvar. The 7 shunt capacitor banks are added at the 16th nodes and 6 shunt capacitor banks are added at the 30th nodes. The capacity of each bank is 100 kvar.

The power loads are collected from the London smart meter dataset [28], which includes the electricity consumption of each household in 112 blocks from November 2011 to February 2014. The time resolution of the load curve is 1 hour. The power loads of three adjacent blocks are combined to analog a node in the distribution network. Therefore, the power loads of 32 nodes can be obtained from the first 96 blocks. Since the collected time of each block is different, 5000 samples are reserved for simulation after data cleaning. 80% of the samples are randomly selected to train the CapsNet, and 10% of the samples are randomly selected as the validation set. The remaining samples are used as the test set to evaluate the performance of the trained CapsNet. Except for the slack node (node 0 in this paper), the remaining 32 nodes are considered as *PQ* nodes whose active power and reactive power are constant. Their active power and reactive power are used to form one sample. Therefore, each sample is a vector with one row and 64 columns for the modified IEEE 33-bus radial distribution network. Before training CapsNet, it needs to add a label to each sample, i.e., the corresponding dispatching strategies. For each sample, the GA is run 30 times independently, and the best dispatching strategy is selected as the label of the sample.

The programs of the CapsNet for reactive power optimization of distribution networks are implemented in Spyder 3.0 with Keras 2.0 and Tensorflow 1.0 deep learning library. The parameters of the computer are: Intel(R) Core(TM) i3-3110M, a dual-core CPU processor with a base clock speed of 2.40 GHz, and 6 GB of memory.

B. Discussion on Key Parameters

In order to intuitively understand the stability and convergence speed of the CapsNet, Fig. 5 shows the changing trend of loss function values in the training set and validation set.

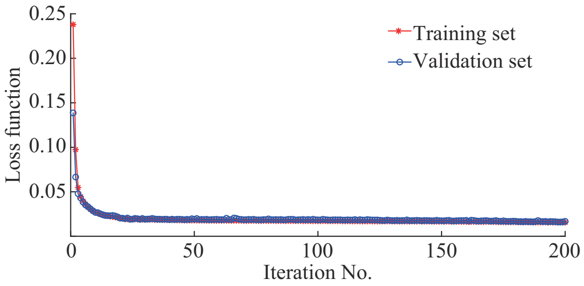
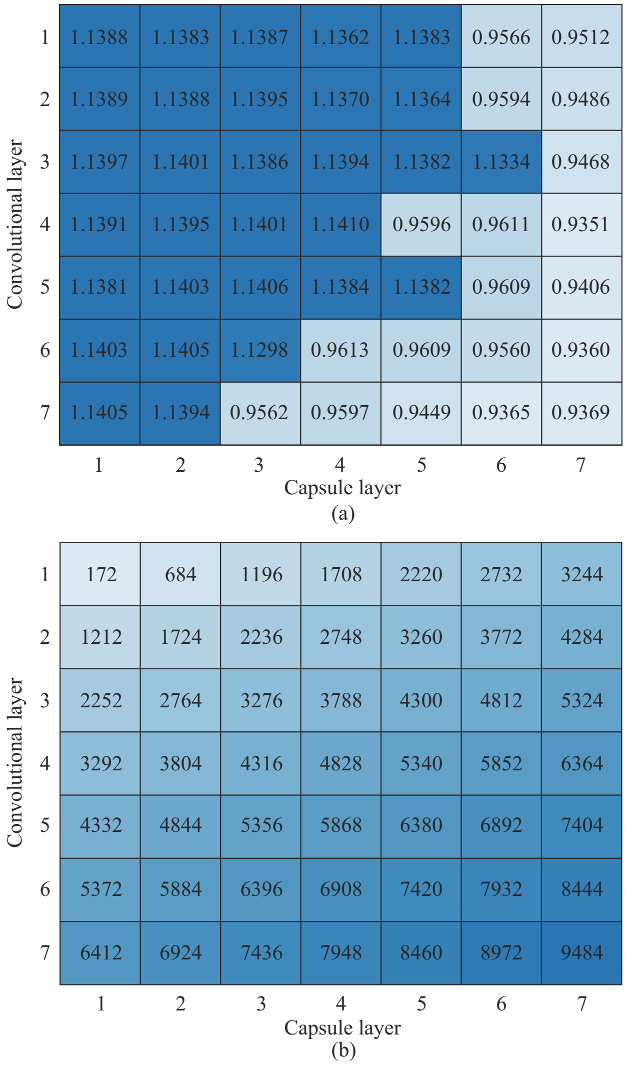


Fig. 5. Training process of CapsNet.

With the increase of iterations, the loss function values of training set and validation set decrease rapidly. After 50 rounds of training iterations, the loss function value of CapsNet becomes stable and no longer declines, which indicates that CapsNet has converged at this time. Comparing the loss function values of the training set and the validation set, it is found that they are very close and there is no fitting problem, which shows that CapsNet has strong generalization ability.

Normally, different combinations of parameters and structures need to be tried, so as to find a suitable structure for the neural network. Then, the loss function or accuracy of the validation set is calculated. The combination with the highest accuracy or the smallest loss function will be used for the neural network.

In order to analyze the influence of convolutional layers and capsule layers on the performance of the CapsNet, the numbers of convolutional layers and capsule layers are gradually increased, and the comprehensive objective functions of the validation set in different layers are counted, as shown in Fig. 6.



Convolutional layer	1	2	3	4	5	6	7
1	1.1388	1.1383	1.1387	1.1362	1.1383	0.9566	0.9512
2	1.1389	1.1388	1.1395	1.1370	1.1364	0.9594	0.9486
3	1.1397	1.1401	1.1386	1.1394	1.1382	1.1334	0.9468
4	1.1391	1.1395	1.1401	1.1410	0.9596	0.9611	0.9351
5	1.1381	1.1403	1.1406	1.1384	1.1382	0.9609	0.9406
6	1.1403	1.1405	1.1298	0.9613	0.9609	0.9560	0.9360
7	1.1405	1.1394	0.9562	0.9597	0.9449	0.9365	0.9369

(a)

Convolutional layer	1	2	3	4	5	6	7
1	172	684	1196	1708	2220	2732	3244
2	1212	1724	2236	2748	3260	3772	4284
3	2252	2764	3276	3788	4300	4812	5324
4	3292	3804	4316	4828	5340	5852	6364
5	4332	4844	5356	5868	6380	6892	7404
6	5372	5884	6396	6908	7420	7932	8444
7	6412	6924	7436	7948	8460	8972	9484

(b)

Fig. 6. Indicators of different layers. (a) Comprehensive objective functions. (b) Number of parameters to be trained.

The following conclusions can be drawn from Fig. 6.

1) In the early stage, the comprehensive objective functions of the test set increase as the number of convolutional layers and capsule layers increases, which indicates that the performance of the CapsNet is gradually becoming stronger. When the number of capsule layers and convolutional layers

are both 4, the comprehensive objective function of the CapsNet is the largest, and the performance of reactive power optimization is the best. This phenomenon shows that it is difficult to mine the complex nonlinear relationship between power loads and dispatching strategies by a small number of convolutional layers and capsule layers. Increasing the number of convolutional layers and capsule layers can improve the feature learning ability of the CapsNet, thus enhancing the accuracy of reactive power optimization.

2) Moreover, the number of parameters to be trained increases linearly with the increase of convolutional layers and capsule layers. When the numbers of convolutional layers and capsule layers are greater than 4, the performance of the CapsNet will be worse if more convolutional layers and capsule layers are added to the CapsNet. This is because the number of samples in the dataset is limited. Too many convolutional layers and capsule layers will not only increase the parameters of the CapsNet to be trained, but also easily lead to over-fitting problems.

3) In general, the number of convolutional layers and capsule layers should be determined according to the volume of the data set. If the number of training samples is 4000, the CapsNet can get good performance by setting the number of convolutional layers and capsule layers to be less than 4. For other datasets, 4 can be considered as a good default value for the numbers of convolutional layers and capsule layers, and larger values or smaller values may be fine.

The batch size determines the number of samples that will be propagated through the network, which will directly affect the training process. In order to explore the influence of the batch size on the performance of the CapsNet, the batch size is set from 8 to 512, and the comprehensive objective functions of the CapsNet on the validation set are counted, as shown in Table III.

TABLE III
COMPREHENSIVE OBJECTIVE FUNCTIONS IN DIFFERENT BATCH SIZES

Batch size	Power loss (MW)	Voltage offset (p.u.)	Objective function (p.u.)	Training time (s)
8	0.2319	0.7990	1.1404	325.79
16	0.2314	0.7970	1.1410	203.07
32	0.2319	0.8012	1.1397	154.76
64	0.2320	0.7990	1.1396	132.66
128	0.2318	0.8033	1.1385	118.47
256	0.2308	0.8179	1.1367	116.41
512	0.2340	0.8158	1.1315	98.42

The following conclusions can be drawn from Table III.

1) As the batch size increases, comprehensive objective functions of the CapsNet first increase and then decrease. When the batch size is 16, the comprehensive objective function is the largest, which indicates that the performance of reactive power optimization is the best. When the batch size is 512, the comprehensive objective function value is the smallest, which shows that the large batch size causes the performance of CapsNet to deteriorate.

2) In terms of training time, although too large batch size can reduce the training time, it also affects the generalization ability of the CapsNet, resulting in the degradation of performance for reactive power optimization. Generally, the batch size of the CapsNet should be determined according to the volume of the dataset and the computing resources of computers. When the number of training samples is about 4000, setting the batch size to 16 can make the CapsNet achieve good performance. The larger values or smaller values may be fine for other datasets.

After initializing the batch size, convolutional layers, capsule layers, and the number of iterations, the loss functions of neural networks are optimized by employing a gradient descent method. The mainstream methods for gradient descent include Adam, RMSprop, stochastic gradient descent (SGD), Adagrad, Adadelta, Nadam, and Adamax. In addition, the popular deep learning libraries (e. g., Keras, Pytorch, Tensorflow) include the implementations of these methods to update the weights. Normally, these methods are used as black boxes in practical engineering, since their principles are too complicated to be explained. To show how to choose a suitable optimizer for the CapsNet in reactive power optimization, the above optimizers are set up and simulated, and then the comprehensive objective functions of the validation set are counted, as shown in Table IV.

TABLE IV
COMPREHENSIVE OBJECTIVE FUNCTIONS IN DIFFERENT OPTIMIZERS

Optimizer	Power loss (MW)	Voltage offset (p.u.)	Objective function (p.u.)
Adadelta	0.2318	0.8040	1.1383
Adagrad	0.2313	0.8168	1.1338
Adam	0.2314	0.7970	1.1410
Adamax	0.2315	0.8012	1.1396
Nadam	0.2320	0.7988	1.1400
RMSprop	0.2321	0.7977	1.1404
SGD	0.2341	0.8173	1.1309

Table IV shows that the CapsNet has large objective function values when Nadam, RMSprop, Adadelta, Adamax, and Adam algorithms are utilized to train the model. Specifically, the Adam algorithm is the most optimal algorithms for the CapsNet in reactive power optimization, since the comprehensive objective function of the Adam algorithm is slightly larger than those of the first four optimizers. Furthermore, the comprehensive objective functions of SGD and Adagrad algorithms are all lower than 1.135, which indicates that they are not suitable for reactive power optimization based on the CapsNet.

C. Comparative Analysis with Existing Methods

In order to illustrate the effectiveness of the CapsNet, the traditional physical model-based method (e.g., GA) and popular data-driven technologies (e. g., CBR, CNN, and MLP) are used as the baselines. The controlled variable method is used to find the best parameters and structures of various algorithms [29].

1) For GA, the total number of chromosomes is 50. The

probability of chromosomal chiasma is 0.5, and the probability of chromosomal variation is 0.3. The number of iterations is 200.

2) The structures and parameters of the CBR are consistent with the algorithm proposed in [10].

3) CNN includes two convolutional layers, two max-pooling layers, two dropout layers, and a dense layer. The number of filters in the front and back convolutional layers is 8 and 16, respectively. The sizes of the kernels in the convolutional layers are 2. The pooling sizes in max-pooling layers are 2. The probabilities abandoning neurons in dropout layers are 0.25. The number of neurons in the dense layer is 4. The activation function of the density layer is the sigmoid function, and the rest are the ReLU functions. The number of iterations is 200, and the optimizer is the Adam algorithm.

4) In terms of MLP, 64 is the number of neurons in the input layer, and the numbers of neurons in the middle layer is 32, 16, and 8, respectively. The number of neurons in the output layer is 4. The number of iterations is 200, and the optimizer is the Adam algorithm. The activation function of the output layer is the sigmoid function, and the rest are the ReLU functions.

The above methods are independently repeated 30 times and the average results of the test set are shown in Table V.

TABLE V
AVERAGE RESULTS FOR DIFFERENT METHODS

Method	Power loss (MW)		Voltage offset (p.u.)		Comprehensive objective function (p.u.)		Computing time (s)
	Mean value	Variance	Mean value	Variance	Mean value	Variance	
CapsNet	0.2314	0.1281	0.7970	0.2855	1.1410	0.0289	0.068
CNN	0.2316	0.1286	0.8031	0.2859	1.1387	0.0292	0.069
MLP	0.2317	0.1297	0.8143	0.2879	1.1382	0.0294	0.042
CBR	0.2311	0.1278	0.8169	0.2933	1.1346	0.0326	4.002
GA	0.2315	0.1276	0.7949	0.2842	1.1430	0.0287	21.263

The following conclusions can be drawn from Table V.

1) The mean value of the comprehensive objective function of CBR is the smallest, which indicates that directly applying the dispatching strategies of historical cases to the current case will result in limited optimization accuracy. This is because the power loads of the historical cases and the current case are different, and the dispatching strategies found by the CBR are not well suited to current cases, especially when the historical load differs significantly from the current loads for some reasons (e. g., impacts of the spread distributed generations and market-based behavior of end-users). Comparing the performance of the CNN and MLP, it is found that the performance of CNN is slightly better than that of MLP, because convolutional layers have more powerful feature extraction ability than dense layers, which can mine the complex nonlinear relationship between power loads and dispatching strategies.

2) Notably, both CNN and CapsNet use convolutional layers to extract the features of the input data in the early stage. The difference is that they use pooling layers and capsule layers to explore the complex nonlinear relationship between the features and dispatching strategies in the later stage, respectively. The performance of CapsNet is better than that of CNN, which indicates that pooling operation loses parts of the feature information and limits the accuracy of reactive power optimization. In contrast, capsule layers in the CapsNet can accurately mine the relationship between the features and dispatching strategies, and provide better dispatching strategies to distribution networks.

3) Comparing the comprehensive objective functions of various algorithms, it is found that the variance of CapsNet is very close to that of GA, and less than those of the CNN, CBR, and MLP, which indicates that CapsNet has higher stability than CNN, MLP, and CBR.

4) Online computing speed is also one of the important indicators to measure the performance of the algorithm for reactive power optimization. For single reactive power optimization, the time consumptions of CapsNet, CNN, MLP, CBR, and GA are 0.068 s, 0.069 s, 0.042 s, 4.002 s, and 21.263 s, respectively. Traditional physical model-based methods rely on topology and parameters for online optimization, which takes a long time. Although data-driven methods require pre-trained models, they do not rely on grid model parameters, and the speed of decision-making is much lower than physical model based methods such as GA.

In order to test the performance of various algorithms for dynamic reactive power optimization, the information entropy method proposed is used to divide one day into several intervals. In this case, dynamic reactive power optimization can be simplified into several static reactive power optimization problems. Specifically, static reactive power optimization is performed in each interval to obtain several dispatching strategies, which form a dispatching strategy for dynamic reactive power optimization. The detailed inference process and implementation steps can be found in [30]. In general, the superiority of CapsNet for static reactive power optimization has been demonstrated by the previous Table V, and dynamic reactive power optimization can be simplified as multiple static reactive power optimizations. Therefore, it can be concluded that CapsNet is also applicable to dynamic reactive power optimization.

To verify this inference, five intervals are used as an example, i.e., dynamic reactive power optimization can be simplified as five static reactive power optimization problems. The average comprehensive objective functions of dynamic reactive power optimization of the test set are counted, as shown in Table VI.

Similar to the results of static reactive optimization, CapsNet outperforms other popular data-driven methods (e. g., CNN, CBR, and MLP) in terms of accuracy and stability.

Moreover, the CapsNet not only ensures that the optimization accuracy is very close to that of GA, but also ensures that the computing time is much lower than that of GA, which shows the effectiveness of the proposed CapsNet for dynamic reactive power optimization.

TABLE VI
AVERAGE COMPREHENSIVE OBJECTIVE FUNCTION OF EACH METHOD

Method	Mean value (p.u.)	Variance (p.u.)	Computing time (s)
CapsNet	1.1448	0.0295	0.340
CNN	1.1417	0.0298	0.345
MLP	1.1406	0.0230	0.210
CBR	1.1365	0.0337	20.010
GA	1.1472	0.0292	106.315

To visualize the results of dynamic reactive power optimization, one day is randomly selected, and various algorithms are used to solve the dynamic reactive power optimization model. The comprehensive objective function of each method is shown in Fig. 7.

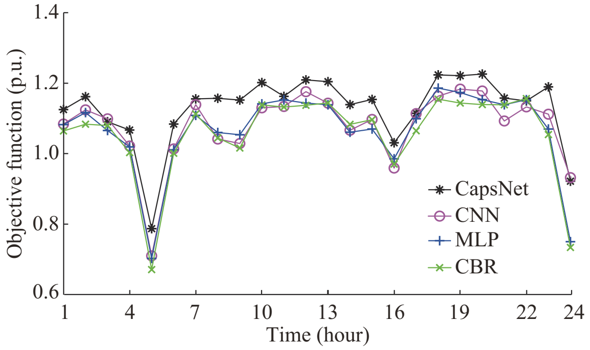


Fig. 7. Comprehensive objective function of each method on a randomly selected sample.

It can be observed from Fig. 7 that the comprehensive objective function value of CapsNet in one day is slightly higher than those of CNN, MLP, and CBR, which indicates that the performance of CapsNet in dynamic reactive power optimization is better than other data-driven methods.

D. Impact of Data Volume on Results

The smaller the number of samples in the training set, the less information the model can obtain from it. To analyze the impact of the volume of data in the training set on reactive power optimization, 12 simulation cases are set, and the number of samples in the training set of each case is shown in Table VII.

TABLE VII
NUMBER OF SAMPLES IN DIFFERENT CASES

Case	Number in training set	Case	Number in training set	Case	Number in training set
1	4000	5	2000	9	250
2	3500	6	1500	10	125
3	3000	7	1000	11	62
4	2500	8	500	12	31

The above methods are independently repeated 30 times and the average comprehensive objective functions of the test set are shown in Fig. 8.

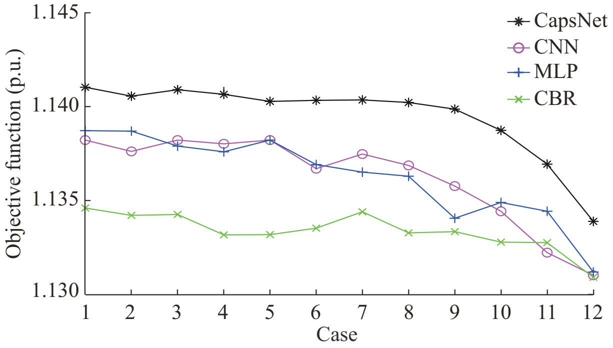


Fig. 8. Comprehensive objective function of each method.

The following conclusions can be drawn from Fig. 8.

1) When the size of samples in the training set is more than 500, the reactive power optimization results of CapsNet does not change much, and the comprehensive objective function is always greater than 1.14, keeping high accuracy. Furthermore, when the number of samples in the training set is less than 500, the comprehensive objective function of CapsNet decreases rapidly, since the limited numbers of samples reduce the generalization ability of the model.

2) Observing the comprehensive objective functions of MLP and CNN in different cases, it is found that they have higher requirements than CapsNet for the volumes of samples in the training set. When the volumes of samples in the training set is less than 1500, their comprehensive objective functions also begin to decrease rapidly, indicating that their generalization ability is weaker than CapsNet.

3) Although the comprehensive objective function of CBR is not sensitive to the volumes of samples in the training set, its optimization accuracy is lower than MLP, CNN, and CapsNet. In general, the performance of CapsNet is better than the existing data-driven technologies such as MLP, CNN, and CBR in different volumes of training set.

E. Robustness Analysis of CapsNet

In order to fully test the robustness of the proposed method, the sample with specified attributes (e.g., light loads and heavy loads) are removed from the training set. In other words, only the remaining samples (medium loads) are used to train models. Then, the trained models are utilized to obtain the dispatching strategies of the light loads and heavy loads in the test set. If the trained models also perform well for the samples with the specified attributes (e.g., light loads and heavy loads), it means that these models have good robustness, i.e., the CapsNet can be adapted to different scenarios that do not appear in the training set.

Firstly, 5000 samples are arranged in descending order according to the sum of the power loads of each node. Secondly, the first 30% of the samples are labeled as heavy loads and the last 30% as light loads. The remaining samples are considered as medium loads, which are used to train the models. Moreover, the optimization results of a randomly selected light loads and heavy loads are visualized, as shown in Fig. 9 and Table VIII.

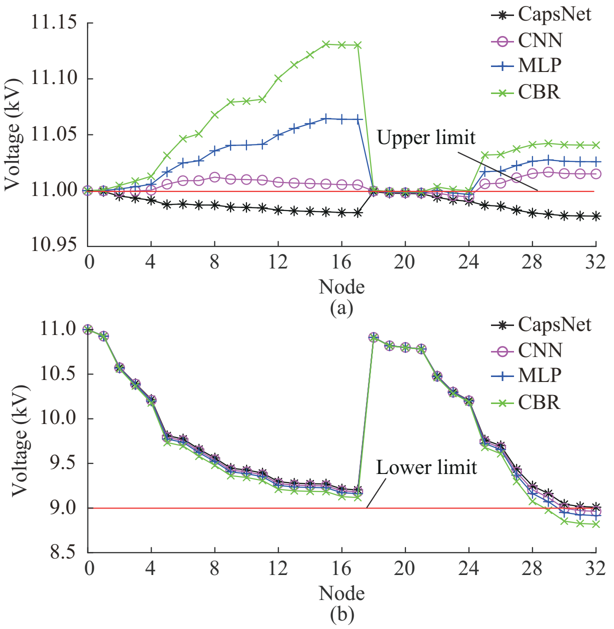


Fig. 9. Voltage corresponding to different loads. (a) Light loads. (b) Heavy loads.

TABLE VIII
SOLUTIONS OF DIFFERENT METHODS

Load type	Method	Position	Capacitor 1 (kvar)	Capacitor 2 (kvar)	SVC (kvar)
Light loads	CapsNet	8	0	0	30
	CNN	8	1	0	76
	MLP	8	1	1	65
	CBR	8	1	2	89
Heavy loads	CapsNet	8	7	6	400
	CNN	8	6	6	395
	MLP	8	5	6	392
	CBR	8	3	6	386

The following conclusions can be drawn from Fig. 9 and Table VIII.

1) Since CNN, MLP, and CBR are trained with the medium loads, the corresponding dispatching strategies of light loads and heavy loads obtained by these methods are too conservative, i.e., they tend to the corresponding solutions of medium loads. Specifically, when the distribution network is running at light load level, their solutions provide too much reactive power, which leads to voltages of some nodes exceeding the upper limit. In the same way, when the power loads are very heavy in distribution networks, the solutions of CNN, MLP, and CBR cannot provide enough reactive power, resulting in the voltage of some nodes exceeding the lower limit.

2) By contrast, although the training set does not include light loads and heavy loads, CapsNet makes good use of convolutional layers and capsule layers to explore the complex non-linear relationship between power loads and dispatching strategies, and the solutions always ensure that voltages are within the constraints, which indicates that CapsNet has stronger robustness than CNN, MLP, and CBR, and can adapt to reactive power optimization tasks at changeable load levels, which may be caused by spread distributed generation and market-based behavior of end-users.

F. Reactive Power Optimization of Distribution Networks with Renewable Energy

To test the performance of the proposed method for reactive power optimization of distribution networks with renewable energy, some photovoltaic (PV) systems and wind turbines (WTs) are added to the modified IEEE 33-bus radial distribution network, as shown in Fig. 10.

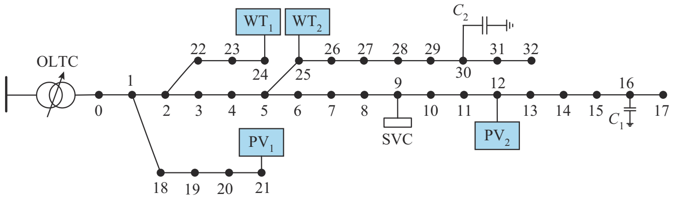


Fig. 10. Topology of modified IEEE 33-bus radial distribution network with renewable energy.

Specifically, the first PV system is added at the 21th node and the second PV system is added at the 12th node. Assume that the capacity of each PV system is 500 kVA. The first WT is added at the 24th node and the second WT is added at the 25th node. Assume that the capacity of each WT is 400 kVA. In addition, it is assumed that the nodes with renewable energy have capacitors that ensure a constant power factor (e.g., power factor is 0.9 in this paper). The data of PV systems and WT are collected from the National Renewable Energy Laboratory [31], [32] and the time resolution is 1 hour. Furthermore, the original PV and wind powers are scaled appropriately to ensure that the penetration of renewable energy in distribution networks ranges from 10% to 50%.

The different methods are independently repeated 30 times and the average results of the test set are shown in Tables IX-XI.

TABLE IX
AVERAGE POWER LOSS OF EACH METHOD

Penetration level (%)	Average power loss (MW)			
	CapsNet	CNN	MLP	CBR
10	0.1972	0.1985	0.1993	0.1998
20	0.1701	0.1712	0.1726	0.1729
30	0.1330	0.1338	0.1353	0.1356
40	0.1139	0.1144	0.1154	0.1177
50	0.0913	0.0919	0.0940	0.0947

The following conclusions can be drawn from the above Tables.

1) As the penetration of renewable energy increases, both power loss and voltage offset of the distribution network gradually decrease, because decentralized renewable energies can directly meet part of the load demand at the nodes, re-

ducing the power flowing in the lines. In addition, renewable energies are located at the end of feeder lines, which can improve the voltage amplitude of the nodes, and then reduce the voltage offset of distribution networks. Moreover, the power loss and voltage offset of whole distribution networks before optimization both decrease with the increase of penetration level, which causes the change not to be particularly large. Therefore, the comprehensive objective functions present a gradually downward trend.

TABLE X
AVERAGE VOLTAGE OFFSET OF EACH METHOD

Penetration level (%)	Average voltage offset (p.u.)			
	CapsNet	CNN	MLP	CBR
10	0.7568	0.7619	0.7723	0.7881
20	0.6845	0.6981	0.6967	0.7143
30	0.6578	0.6692	0.6701	0.6786
40	0.5962	0.6020	0.6154	0.6289
50	0.5774	0.5861	0.5992	0.6015

TABLE XI
AVERAGE COMPREHENSIVE OBJECTIVE FUNCTION OF EACH METHOD

Penetration level (%)	Average comprehensive objective function (p.u.)			
	CapsNet	CNN	MLP	CBR
10	1.1392	1.1268	1.1260	1.1053
20	1.1172	1.1112	1.1100	1.1045
30	1.1084	1.1065	1.0986	1.0952
40	1.0972	1.0948	1.0936	1.0869
50	1.0779	1.0678	1.0648	1.0641

2) Comparing the power losses, voltage offsets, and comprehensive objective functions of different methods under different penetration levels, it is found that the average power loss and voltage offset of CapsNet are lower than those of other methods, and the average comprehensive objective function is also the largest, which shows that CapsNet is superior to CNN, MLP, and CBR, and can adapt to the reactive power optimization of distribution networks with different penetration levels of renewable energy.

G. Impact of Different Scale Distribution Networks on Results

To analyze the impact of different scale distribution networks on the results, the modified IEEE 69-bus radial distribution network is used for simulation and analysis. The resistance and reactance of the branch can be found in [33], and the topology is shown in Fig. 11.

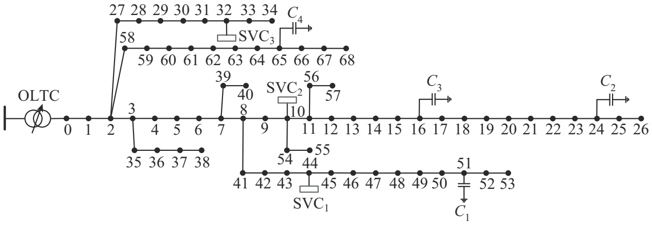


Fig. 11. Topology of modified IEEE 69-bus radial distribution network.

Specifically, the voltage magnitude base is 10 kV. The tap of the OLTC includes 17 positions, which range from $-8\times 1.25\%$ to $8\times 1.25\%$. The locations and capacities of SVCs and capacitor banks are assumed as follows. The SVCs are added at the 10th node, 32th node, and 44th node; the reactive power of all SVCs ranges from 0 to 400 kvar; seven shunt capacitor banks are added at the 16th node, 24th node, 51th node, and 65th node; and the capacity of each bank is 100 kvar.

In the same way, the London smart meter dataset are employed to form the power load of each node. Specifically, the power loads of three different blocks are combined to analog a node in the distribution network. Since the collected time of each block is different, 5000 samples are reserved for simulation after data cleaning. Except for the slack node (node 0 in this paper), the remaining 68 nodes are considered as *PQ* nodes whose active power and reactive power are constant. Their active power and reactive power are used to form one sample. Therefore, each sample is a vector with one row and 136 columns for the modified IEEE 69-bus radial distribution network. For each sample, the GA method is run 30 times independently, and the best dispatching strategy is selected as the label of the sample.

To illustrate the effectiveness of the CapsNet, the traditional physical model based method and popular data-driven methods are used as the baselines. Each method is independently repeated 30 times and the average comprehensive objective function of the test set are shown in Table XII.

TABLE XII
AVERAGE COMPREHENSIVE OBJECTIVE FUNCTION OF EACH METHOD

Method	Mean value (p.u.)	Variance (p.u.)	Computing time (s)
CapsNet	0.8343	0.0352	0.071
CNN	0.8326	0.0355	0.073
MLP	0.8319	0.0359	0.044
CBR	0.8217	0.0394	4.364
GA	0.8362	0.0347	64.775

The following conclusions can be drawn from Table XII.

1) Although the performance of CapsNet is slightly weaker than GA in terms of comprehensive objective function and its variance, CapsNet outperforms other data-driven methods such as CNN, MLP, and CBR.

2) Normally, the real time system requires that the suitable solutions should be obtained within 60 s, during which the power system gets the measurement data and then calculates the suitable dispatching strategies for all power equipment [34], [35]. Moreover, the computing time of GA increases significantly with the size of distribution networks (e.g., the number of power equipment and nodes), while the computing times of data-driven methods are not sensitive to the size of distribution networks, indicating that the proposed CapsNet is more suitable for real-time reactive power optimization than GA, especially for large-scale distribution networks.

IV. Discussion

This paper aims to apply the CapsNet to optimize the power loss and voltages of distribution networks. Moreover, the performance of the proposed CapsNet has been tested on the modified IEEE 33-bus radial distribution network and the modified IEEE 69-bus radial distribution network. The simulation results show that CapsNet achieves state-of-the-art performance with superior accuracy and less computing time for reactive power optimization of distribution networks. However, the proposed approach assumes that the topology of the distribution network is fixed, and it does not account for the influence of the dynamic topology (e.g., reconfiguration of distribution networks) on the results. Specifically, the tie switches and sectionalizing switches are also common devices used for regulation of voltages and power loss. Unlike the SVC, OLTC, and shunt capacitor banks, the open states of tie switches and sectionalizing switches may lead to outages at some nodes [36]. In addition, the impact of market-based behavior of end-users on reactive power optimization can be discussed in the future.

Moreover, the graph CapsNet is a possible extension of this paper to account for the influence of different topologies by inputting an adjacency matrix to neural networks [37]. In addition, the applications of the CapsNet are not limited to the reactive power optimization of distribution networks. It may be generalized to other tasks of power system such as energy management or demand-side response.

V. Conclusion

To improve the accuracy and computing speed of reactive power optimization, a novel machine learning model, the CapsNet, is presented for reactive power optimization of distribution networks. Through the simulation analysis on the IEEE 33-bus radial distribution network and the IEEE 69-bus radial distribution network, the following conclusions are obtained.

1) The numbers of convolutional layers and capsule layers, the batch size, the number of iterations, and the selection of optimizers have a great influence on the performance of reactive power optimization. Specifically, it is difficult to mine the complex nonlinear relationship between power loads and dispatching strategies by a small number of convolutional layers and capsule layers, while too many convolutional layers and capsule layers will not only increase the parameters of the CapsNet to be trained, but also easily lead to over-fitting problems. For the dataset with 5000 samples, the CapsNet can achieve good performance by setting the numbers of convolutional layers and capsule layers to 4. For other datasets, 4 can be considered as a good default value for the number of convolutional layers and capsule layers, and higher values or lower values may be fine. Similarly, the batch size should be determined according to the volume of the dataset and the computing resources of computers. Although small batch size can improve the performance of CapsNet, it consumes more training time. The convergence speed of CapsNet is very fast, and 50 iterations can ensure its convergence. Using Adam algorithm as the optimizer can enable the CapsNet with the best performance.

2) Compared with CNN, MLP, and CBR, CapsNet has not only higher optimization accuracy, but also better stability of optimization results. The pooling operation in the CNN loses parts of the feature information and limits the accuracy of reactive power optimization. In contrast, capsule layers in the CapsNet can accurately mine the relationship between features and dispatching strategies, which provides better dispatching strategies to distribution networks. The computing time of GA increases significantly with the size of distribution networks, while the computing time of data-driven-based methods is not sensitive to the size of distribution networks. Moreover, CapsNet does not rely on grid model parameters, and the speed of decision-making is much lower than physical model based methods such as GA.

3) Compared with other deep neural networks (e.g., MLP and CNN), CapsNet has lower requirements for the volume of the training set. The performance of CapsNet is better than the existing data-driven technologies such as MLP, CNN, and CBR.

4) The CNN, MLP, and CBR are too conservative. When the power loads are very heavy or light in distribution networks, some parts of the voltages may out of the limit, while CapsNet can ensure that the voltages are always within the constraints. In other words, the CapsNet has stronger robustness than CNN, MLP, and CBR, and can adapt to reactive power optimization tasks at different load levels.

5) For the reactive power optimization of distribution networks with different penetration levels of renewable energy, the average power loss and voltage offset of CapsNet are lower than those of other methods, and the average comprehensive objective function is also the largest, which shows that CapsNet is superior to CNN, MLP, and CBR.

References

[1] J. Liu, Y. Chen, C. Duan *et al.*, "Distributionally robust optimal reactive power dispatch with wasserstein distance in active distribution network," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 3, pp. 426-436, May 2020.

[2] R. Elavarasan, G. Shafiullah, S. Padmanaban *et al.*, "A comprehensive review on renewable energy development, challenges, and policies of leading Indian states with an international perspective," *IEEE Access*, vol. 8, pp. 74432-74457, Apr. 2020.

[3] Q. Zhao, W. Liao, S. Wang *et al.*, "Robust voltage control considering uncertainties of renewable energies and loads via improved generative adversarial network," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 6, pp. 1104-1114, Nov. 2020.

[4] T. Ding, S. Liu, W. Yuan *et al.*, "A two-stage robust reactive power optimization considering uncertain wind power integration in active distribution networks," *IEEE Transactions on Sustainable Energy*, vol. 7, vol. 1, pp. 301-311, Jan. 2016.

[5] L. Liu, H. Li, Y. Xue *et al.*, "Reactive power compensation and optimization strategy for grid-interactive cascaded photovoltaic systems," *IEEE Transactions on Power Electronics*, vol. 30, vol. 1, pp. 188-202, Jan. 2015.

[6] C. Si, S. Xu, C. Wan *et al.*, "Electric load clustering in smart grid: methodologies, applications, and future trends," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 2, pp. 237-252, Mar. 2021.

[7] Y. Wang, Q. Chen, T. Hong *et al.*, "Review of smart meter data analytics: applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125-3148, May 2019.

[8] D. Lee and R. Baldick, "Load and wind power scenario generation through the generalized dynamic factor model," *IEEE Transactions on Power Systems*, vol. 32, no. 1, pp. 400-410, Jan. 2017.

[9] L. Ge, W. Liao, S. Wang *et al.*, "Modeling daily load profiles of distribution network for scenario generation using flow-based generative network," *IEEE Access*, vol. 8, pp. 77587-77597, Apr. 2020.

[10] D. Yang, W. Liao, Y. Wang *et al.*, "Data-driven optimization control for dynamic reconfiguration of distribution network," *Energies*, vol. 11, no. 10, pp. 1-18, Oct. 2018.

[11] W. Sheng, K. Liu, H. Pei *et al.*, "A fast reactive power optimization in distribution network based on large random matrix theory and data analysis," *Applied Sciences*, vol. 6, no. 6, pp. 1-19, May 2016.

[12] G. Chen, Y. Zhang, S. Hao *et al.*, "Association mining based intelligent identification method of key parameters for reactive power optimization," *Automation of Electric Power Systems*, vol. 41, no. 23, pp. 109-116, Sept. 2017.

[13] G. Rao, S. Sundeep, and S. Reddy, "Optimal power flow controlling with TCSC using expert system," in *Proceedings of International Conference on Sustainable Energy and Intelligent Systems*, Chennai, India, Jul. 2011, pp. 379-383.

[14] L. Wenlong, S. Wang, L. Qi *et al.*, "Reactive power optimization of distribution network based on case-based reasoning," in *Proceedings of IEEE PES General Meeting*, Portland, USA, Aug. 2018, pp. 1-5.

[15] M. Shao, J. Wu, C. Shi *et al.*, "Reactive power optimization of distribution network based on data driven and deep belief network," *Power System Technology*, vol. 43, no. 6, pp. 1874-1885, Jun. 2019.

[16] X. Meng, G. Sun, J. Li *et al.*, "Automatic reactive power and voltage control for regional power grid based on SVM," in *Proceedings of IEEE International Conference of IEEE Region 10*, Xi'an, China, Oct. 2013, pp. 1-4.

[17] J. Choung, S. Lim, S. H. Lim *et al.*, "Automatic discontinuity classification of wind-turbine blades using A-scan-based convolutional neural network," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 1, pp. 210-218, Jan. 2021.

[18] S. Sabour, N. Frosst, and G. Hinton, "Dynamic routing between capsules," in *Proceedings of 31st Conference on Neural Information Processing Systems*, Los Angeles, USA, Dec. 2017, pp. 1-11.

[19] K. Shi, C. Gong, H. Lu *et al.*, "Wide-grained capsule network with sentence-level feature to detect meteorological event in social network," *Future Generation Computer Systems*, vol. 102, pp. 323-332, Jan. 2020.

[20] M. Paoletti, J. Haut, R. Beltran *et al.*, "Capsule networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 4, pp. 2145-2160, Apr. 2019.

[21] Y. Guo, Z. Pan, M. Wang *et al.*, "Learning capsules for SAR target recognition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4663-4673, Aug. 2020.

[22] Z. Xu, W. Lu, Q. Zhang *et al.*, "Gait recognition based on capsule network," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 159-167, Feb. 2019.

[23] X. Ding, N. Wang, X. Gao *et al.*, "Group feedback capsule network," *IEEE Transactions on Image Processing*, vol. 29, pp. 6789-6799, May 2020.

[24] W. Liao, Y. Yun, Y. Wang *et al.*, "Reactive power optimization of distribution network based on graph convolutional network," *Power System Technology*, vol. 45, no. 6, pp. 2150-2160, Sept. 2020.

[25] M. Baran and F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Transactions on Power Delivery*, vol. 4, no. 2, pp. 1401-1407, Apr. 1989.

[26] M. Masoum, M. Ladjevardi, A. Jafarian *et al.*, "Optimal placement, replacement and sizing of capacitor Banks in distorted distribution networks by genetic algorithms," *IEEE Transactions on Power Delivery*, vol. 19, no. 4, pp. 1794-1801, Oct. 2004.

[27] S. Wildenhues, J. Rueda, and I. Erlich, "Optimal allocation and sizing of dynamic Var sources using heuristic optimization," *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2538-2546, Sept. 2015.

[28] UK Power Networks. (2015, Jun.). Low carbon London project. [Online]. Available: https://data. london. gov. uk/dataset/smartmeter-energy-use-data-in-london-households

[29] W. Liao, D. Yang, Y. Wang *et al.*, "Fault diagnosis of power transformers using graph convolutional network," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 241-249, Mar. 2021.

[30] J. Zhao, H. Niu, and Y. Wang, "Dynamic reconfiguration of active distribution network based on information entropy of time intervals," *Power System Technology*, vol. 41, no. 2, pp. 402-408, Jan. 2017.

[31] C. Draxl, A. Clifton, B. Hodge *et al.*, "The wind integration national dataset (WIND) toolkit," *Applied Energy*, vol. 2015, pp. 355-366, Aug. 2015.

[32] National Renewable Energy Laboratory. (2012, Nov.). Solar integration national dataset toolkit. [Online]. Available: https://www.nrel.gov/grid/sind-toolkit.html

[33] H. Mojarrad, G. Gharehpetian, H. Rastegar *et al.*, "Optimal placement and sizing of DG (distributed generation) units in distribution networks by novel hybrid evolutionary algorithm," *Energy*, vol. 54, pp. 129-138, Jun. 2013.

[34] Q. Nan. (2016, Jun.). Voltage control in the future power transmission systems. [Online]. Available: https://vbn. aau. dk/ws/portalfiles/portal/254173904/

[35] R. Wu, W. Wang, Z. Chen *et al.*, "Coordinated voltage regulation methods in active distribution networks with soft open points," *Sustainability*, vol. 12, no. 22, pp. 1-18, Nov. 2020.

[36] Y. Gao, W. Wang, J. Shi *et al.*, "Batch-constrained reinforcement learning for dynamic distribution network reconfiguration," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5357-5369, Nov. 2020.

[37] W. Liao, B. Bak-Jensen, J. Pillai *et al.* (2021, Jan.). A review of graph neural networks and their applications in power systems. [Online]. Available: https://arxiv.org/abs/2101.10025

Wenlong Liao received the B.S. degree from China Agricultural University, Beijing, China, in 2017 and the M.S. degree from Tianjin University, Tianjin, China, in 2020. He is currently pursuing the Ph. D. degree in Aalborg University, Aalborg, Denmark. His research interests include smart grid, deep learning, and power system protection.

Jiejin Chen received the B. S. degree from China Agricultural University, Beijing, China, in 2016 and the M. S. degree from Peking University, Beijing, China, in 2019. Her research interests include deep learning, machine learning, and big data.

Qi Liu received the B. S. degree in 2015 and the M. S. degree in 2018 in electrical engineering from Shandong University of Science and Technology, Qingdao, China. He is currently working toward the Ph.D. degree in electrical engineering at Tianjin University, Tianjin, China. His research interests include distribution systems and renewable energy.

Ruijin Zhu received the M.S. degree in Gansu Agricultural University, Lanzhou, China, in 2014. He is currently an Associate Professor in School of Electrical Engineering, China Tibet Agriculture and Animal Husbandry University, Linzhi, China. His research interests include smart grid, machine learning, and optimization and operation of distribution network.

Like Song received the B. S. degree from Hebei University of Technology, Tianjin, China, in 2016 and the M. S. degree from Tianjin University, Tianjin, China, in 2019. He is currently working in Maintenance Branch of State Grid Jibei Electric Power Co., Ltd, Beijing, China. His research interests include big data and machine learning.

Zhe Yang received the B.S. degree from Northeast Electric Power University, Jilin, China, in 2017 and the M. S. degree from North China Electric Power University, Beijing, China, in 2020. He is currently pursuing the Ph. D. degree in Aalborg University, Aalborg, Denmark. His research interests include machine learning, renewable energy, and power system protection.