Reactive Power Optimization for Voltage Stability in Energy Internet Based on Graph Convolutional Networks and Deep Q-learning

1st Sheng Guo
Department of Automation
Tsinghua University
Beijing, China

2nd Junwei Cao
Beijing National Research Center for Information Science and Technology
Tsinghua University
Beijing, China
jcao@tsinghua.edu.cn

***Abstract*—The rapid response of reactive power compensation is crucial to guarantee the stable operation of energy Internet (EI) with variable loads and distributed power generations. Therefore, this paper proposes an intelligent approach for reactive power optimization in EI based on graph convolutional networks (GCN) and deep Q-learning (DQN). With the adjacency matrix that represents topology of EI, the GCN in the proposed approach fuses the monitoring data of EI nodes for a comprehensive feature extraction. Furthermore, reactive power optimization of EI during voltage sag is solved by DQN method in which GCN is used as the Q network. Thus, the optimized output of reactive power compensation device can be put into EI to ensure the voltage stability. The case study on the simulation data of an EI system that considers photovoltaic and battery storage system verifies the effectiveness of the proposed approach. The result shows that the proposed approach achieves fast response to the faults and sudden increase of load in EI, and gives more accurate reactive power compensation than the common control method of reactive power compensation device.**

***Index Terms*—Deep Q-learning, energy Internet, graph convolutional networks, microgrid, reactive power.**

I. Introduction

THE research on EI has been developed rapidly in recent years, as EI can absorb distributed energy resources and achieve effective energy management. The increase of distributed energy brings EI more prone to voltage stability problem with the variety of power generations and random loads, which results in voltage stability management to be an important aspect of EI protection [1]. The voltage instability in EI may lead to large-scale power system blackouts, which cause huge economic losses.

In the actual power system, reactive power compensation devices such as static var compensator and static var generator are usually used for the optimal control of reactive power. These devices often use the feedback control method to determine the reactive power compensation according to the deviation of bus voltage. It is effective to deal with not serious voltage instability, but it can not respond in time in the face of fault conditions that will cause a sharp voltage sag [2].

This work was sponsored by Tsinghua-Toyota Joint Research Institute Interdisciplinary Program. Corresponding author: Junwei Cao.

Due to the informatization and intellectualization of EI, the collected abundant data needs to be fully utilized in guaranteeing the safe operation of EI. Based on the collected real-time data, the voltage stability in EI can be transformed into an optimal control problem [3]. The optimal control value can be obtained by solving equations [2], [4] or using heuristic algorithms [5]. However, in the face of the changing parameters in EI, the calculation of these methods is not quick enough, and may fall into local extremum, which results in the optimal solution can not be obtained.

With the rapid development of artificial intelligence methods in recent years, deep learning has been applied in many power system control problems, such as fault detection [6], energy management [7] and renewable energy storage [8]. However, the aforementioned deep learning methods that only combine the information in EI as deep learning input are totally data-driven. The topology information that serves as domain knowledge of EI has not been considered properly, which results in the poor robustness of deep learning methods, especially under variable power generations in EI. The EI topology indicates the interconnection between the nodes and facilitates the precise determination of the required reactive power compensation.

With the topology, the information of EI can be regarded as data with graph structure. Motivated by applying convolutional neural networks (CNN) that has been successfully and widely used in computer vision to data with graph structure, graph convolutional networks (GCN) is proposed by defining convolution operations on a graph [9] with the adjacency matrix. Recently, GCN has been successfully applied and shows advantages than CNN in many fields such as traffic prediction [10] and virtual network embedding [11].

Therefore, to make full use of the informatization of EI and domain knowledge, GCN is introduced to the reactive power optimization during fault conditions in this paper. A novel reactive power optimization approach is proposed for voltage stability based on GCN and deep Q-learning (DQN). With the EI topology represented by the adjacency matrix and graph convolution operation in GCN, the features in the real time data of EI nodes and its adjacent nodes are extracted and fused. Then the action of reactive power input with the largest Q value is selected as the optimized reactive power compensation. The main contributions in this paper are summarized as follows:

1) For the first, GCN is introduced to reactive power optimization of EI in this paper. By considering EI topology, GCN achieves comprehensive feature fusion of EI information with the graph convolution operation.
2) To obtain the optimized output of reactive power compensation device, GCN is first combined with DQN. The voltage stability and reactive power output are considered in Q function simultaneously.
3) The case study on an EI system shows that the proposed method achieves better voltage stability and accurate reactive power compensation during voltage sag than the state-of-the-art methods based on deep learning.

The rest of the paper is organized as follows. Section II presents the basic principles of GCN and DQN. Section III introduces the proposed reactive power optimization approach for EI. Section IV verifies the proposed approach on the simulation data of an EI system. Section V concludes the paper.

II. Preliminaries

A. Graph Convolutional Networks

In recent years, with the strong feature extraction ability of convolution kernels, CNN has been successfully applied in many fields [12], [13]. However, the information with graph structure are common in our daily life, such as Internet, transportation network and power grid, the topology of which is significant for the state awareness of the system. Considering the multiple nodes interconnected in EI, the information of EI nodes is in a form of graph structure. Therefore, GCN is introduced to accomplish accurate feature extraction of EI information.

Aiming at applying convolution operation on graph structured data, GCN is proposed by deriving graph convolution operation with adjacency matrix [14]. Considering an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes $v_i \in \mathcal{V}$, K edges $(v_i, v_j) \in \mathcal{E}$. An adjacency matrix $A \in \mathbb{R}^{N\times N}$ is constructed to indicate the topology, and a degree matrix $D_{ii} = \sum_j A_{i,j}$ is constructed to indicate the degree of nodes. Then a normalized Laplacian matrix L of the graph is obtained by combining A and D [15]:

$$L = D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}} = U\Lambda U^T \tag{1}$$

where Λ is the matrix composed of eigenvalues of L, U is the matrix of eigenvectors, and ^{T} is matrix transposition.

Then graph convolution operation is defined by:

$$y = \sigma\Big(Ug_\theta(\Lambda)U^T x\Big) \tag{2}$$

where y is the output features of nodes, g_θ is the graph convolutional kernels. $g_\theta(\Lambda)$ represents the convolution operation of g_θ on Λ. x is the input node features, $\sigma(\cdot)$ is the activation function.

As L only represents the relationship between directly connected nodes, to extract the features of adjacent nodes with a larger distance, a Chebyshev polynomial is applied to Λ by [15]:

$$g_\theta(\Lambda) = \sum_{0}^{k-1} \beta_k T_k(\tilde{\Lambda}) \tag{3}$$

where $T_k(\cdot)$ is the k-th order Chebyshev polynomial, β_k is the convolutional kernels for k-th order features, $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - I$ is the rescaled Λ since $T_k(\cdot)$ needs the input between $[-1, 1]$. λ_{max} is the maximum of Λ, and I is the identity matrix. Therefore, we can derive the final formula of a graph convolutional layer by:

$$y = \sigma\Big(U\sum_{0}^{k-1} \beta_k T_k(\tilde{\Lambda})U^T x\Big) = \sigma\Big(\sum_{0}^{k-1} \beta_k T_k(\tilde{L})x\Big) \tag{4}$$

where $\tilde{L} = 2L/\lambda_{max} - I$ is the rescaled L.

As shown in Fig. 1, a complete GCN has multiple stacked graph convolutional layers to obtain the final features of nodes. x_i, y_i and o_i are the input, hidden and output features of the ith node, respectively. After each graph convolution layer, the features of each node are updated by combining the information of adjacent nodes. Finally, fully connected layers are constructed to accomplish target tasks such as classification and regression.

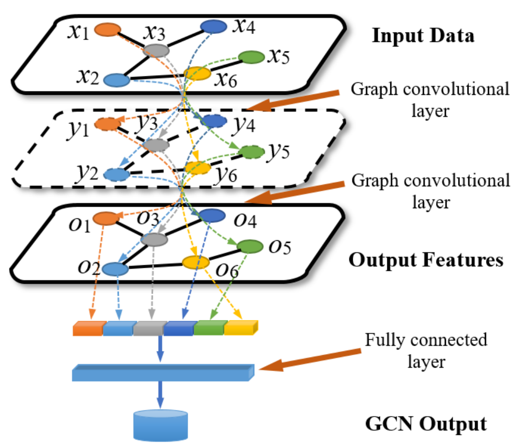

Fig. 1: Structure of GCN.

B. Deep Q-learning

Reinforcement learning is a method to solve decision-making or optimization problems, in which agents learn by trial and error and interact with the environment to get the maximum reward. The common model of reinforcement learning is the Markov decision process, which consists of a set of states s and actions a. State and reward r only depend on the state and action of the previous time step.

When deep learning is introduced into reinforcement learning, DQN is first proposed as a value based reinforcement learning method [16]. DQN learns how to evaluate the current state, and then selects the best action according to the Q value. The main process of DQN is as follows:

1) Get the state s from environment. Use s as the input of Q network to obtain the Q value for each action a.
2) Choose the action should be taken using $\epsilon - greedy$ algorithm: $a = argmax(Q(s, \theta))$.
3) Perform action a in the environment. The environment transit to the next state s' and return the reward r.
4) Save the sequence $< s, a, r, s' >$ into the replay memory.
5) Repeat 1) to 4) and obtain enough replay data.
6) Sample a batch data from replay memory and train the Q network. The loss during the training is calculated by: $loss = (r + \gamma max_{a'} Q(s', a'; \theta) - Q(s, a; \theta))^2$.

where θ is the parameters of Q network, and γ is the accumulation coefficient. After the training with replay data, the Q network is able to obtain the action with the largest reward based on the perception of the environment.

III. The Proposed Reactive Power Optimization Approach for Voltage Stability in EI

Based on the power grid architecture, energy Internet integrates information and power electronics technologies to connect a number of distributed generators with renewable energy and energy storage devices to power grid. Meanwhile, information acquisition devices are placed at each node to collect real-time data, which forms the information flow of EI. However, the variability of renewable energy brings great difficulties on the efficient utilization and balance of supply and demand to EI. At the same time, as EI advocates local consumption, the local microgrid should not only be integrated into the main power grid, but also operate in isolated island.

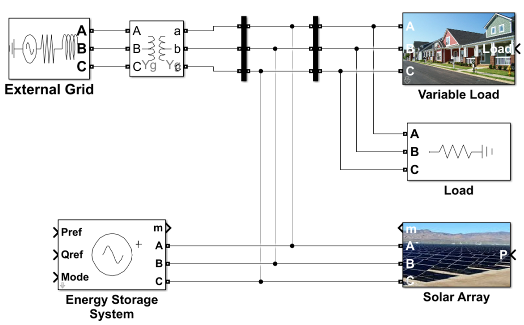

Fig. 2: Structure of a local microgrid in EI.

As shown in Fig.2, the photovoltaic power source, battery energy storage devices and two variable loads are connected to the main bus of the EI system. In the local consumption, there will be voltage stability problems, which affects the reliability of the system operation. When a large disturbance or a fault occurs in EI, transient voltage stability problems such as voltage sag may occur in the local network which is not robust. How to reasonably configure reactive power compensation after the voltage sag is the main problem to be solved in this paper.

To solve the problem, based on GCN and DQN, a novel reactive power compensation optimization approach is proposed for the voltage stability in EI during the voltage sag. Fig.3 shows the flow chart of the proposed approach, and the detail steps of the proposed approach are described as follows:

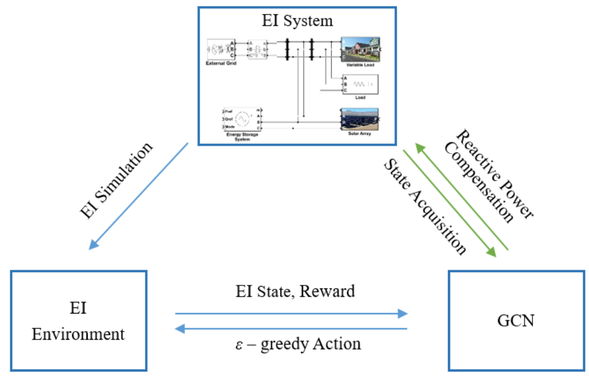

Fig. 3: Flow chart of the proposed approach.

1) Construction of EI Model: As the training of DQN uses a trail and error method, it is difficult to perform totally online training. Therefore, in addition to the data collected in the actual EI system, numerical simulation is needed to obtain more interactive data between reactive power compensation device and EI system. With the collected field data of EI nodes as initial power flow data, disturbances and faults are injected to the EI model to simulate the voltage sag in EI system.

2) Acquisition of EI State: For the optimized output of reactive power compensation device, it is necessary to obtain the complete information of the voltage sag. The monitoring data of all EI nodes, which consist of active power, reactive power, frequency and three-phase voltages of each node during a certain period with T samples, are combined into a matrix $S \in \mathbb{R}^{N \times T \times 6}$ as the state of EI system. In this paper, T is set as 60 with a sampling rate of 100Hz, which includes the monitoring data from 0.3 second before to 0.3 second after the occurrence of voltage sag.

3) Construction of Q network and Reward Function: With the advantage in considering the topology of EI, GCN is used as the Q network in the proposed approach. In GCN, the order of Chebyshev polynomial is set to 3, since the relationship between nodes with a large distance is weak. Number of graph convolutional layers is set to 2, which refers to [14]. Two fully connected layers is added after graph convolutional layers. The output size of GCN is equal to the number of actions.

Reward function in DQN is used to evaluate the state of environment after the action is executed. For evaluation of voltage stability, firstly, the voltages of nodes should be higher than 0.8 pu. Secondly, considering the loss of reactive power compensation device, the output of reactive power compensation device is added as a penalty in the reward function. The reward function in the proposed approach is defined as:

$$R = \frac{1}{N}\sum_{i}^{N} sgn(U - 0.8) \times (1 - \Delta U) - \alpha O_{rec} \quad (5)$$

where $sgn()$ is the signum function, U is line voltage of nodes, ΔU is the absolute value between U and 1 pu, O_{rec} is the output of reactive power compensation device, and α is a weight coefficient.

4) Offline Training: During the training of the DQN model, the EI model act as the environment that perform state transition to obtain the new state and the reward according to 5. GCN gives the optimized action with EI state as input. To obtain an accurate and robust model, the detail training process of the DQN model based on GCN is shown in Algorithm 1.

Algorithm 1 Training Process of DQN Model for Voltage Stability in EI.

Obtain the active and reactive power data of actual EI nodes with m samples.
for $i \in 1, 2, ..., m$ **do**
    With the i-th sample as the initial power flow of EI, simulate n voltage sag conditions by injecting disturbances and faults.
    Sample the EI state during voltage sag as s.
end for
repeat
    for $k \in 1, 2, ..., mn$ **do**
        Use the EI state s_k as the input of GCN to obtain $Q(s;\theta)$ for all action.
        Choose the action a by: $a = argmax(Q(s, \theta))$.
        Eexecute action a in EI environment and obtain the state s' and reward r.
        Save $< s, a, r, s' >$ into replay memory.
    end for
    Sample data batches that include $< s, a, r >$ from replay memory.
    Use the EI state s as the input of GCN to obtain $Q(s, a; \theta)$ for action a.
    Calculate the loss for action a: $loss = (r - Q(s, a; \theta))^2$.
    Carry out back propagation using Adam optimizer for all batches to update GCN parameters.
until $loss < loss_{end}$

5) Online Implementation: The trained DQN model is implemented to online reactive power compensation with the EI state from monitoring data as input. If a voltage sag is detected, the DQN model give the optimized output of reactive power compensation device to ensure voltage stability of EI.

IV. Numerical Simulation and Results

In this section, the effectiveness of the proposed approach is evaluated on an EI system. Moreover, the comparison with the state-of-the-art methods is carried out to show the superiority of the proposed approach.

A. EI Simulation

Based on the structure shown in Fig. 2, the simulation model of the EI system is constructed using PSD/BPA software. The EI system operates at 5 kV, 50 Hz. Photovoltaic power source (PV) is 1.0 MV and battery storage system (BES) is 0.5 MV. Load 1 and Load 2 are ZIP loads with different power curves and factors. The reactive power compensation device is installed on the main bus. Fig. 4 shows the power curves of DGs and loads used in the case study, which are with dynamic changes and contain the data of a whole day with an interval of 10 minutes. Negative values represent the load and positive values represent the power generation.

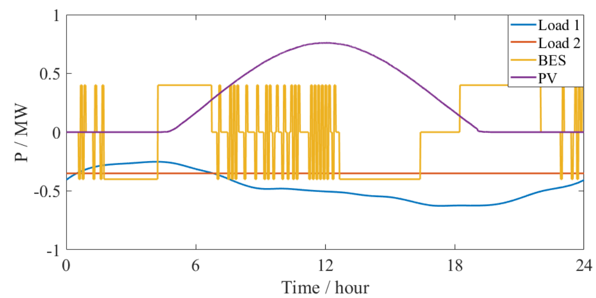

Fig. 4: Power curves of PV, BES and loads.

With the power curves at one moment as initial power flow, voltage sag conditions after disturbances and faults are injected. Disturbances include sudden increase of load on nodes. Faults include two-phase short circuit, two-phase ground and three-phase short circuit on lines. Fig. 5 shows the voltage curves of EI nodes when a two-phase ground on the line between main bus and load 1. The fault occurs at 0.9 second.

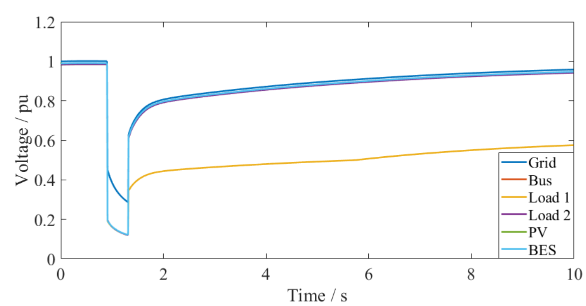

Fig. 5: Voltage curves of EI nodes during a two-phase ground fault.

B. Parameter Setup and Offline Training

In parameter setup, first, an adjacency matrix with a size of 6×6 is to describe the EI topology. Then a GCN with two graph convolutional layers and two fully connected layers is constructed, the detail parameters of which are list in Table I. The reactive power compensation values are discretized into 25 actions, which ranges from 0.2 MVar to 5 MVar.

Half of the voltage sag conditions that contains 3456 samples are used in the offline training, while half are used

TABLE I: Parameters of GCN

No.	Layer Type	Input Size	Parameters	Output Size
1	Graph Convolutional layer	6×360	$360 \times 100 \times 3$	6×100
2	Graph Convolutional layer	6×100	$100 \times 30 \times 3$	6×30
3	Fully connected layer	1×180	180 / 60	1×60
4	Fully connected layer	1×60	60 / 25	1×25

to evaluate the trained GCN model. The training of GCN is processed by Tensorflow in Python environment on a computer with a GTX 1070 GPU and 16 GB memory. The learning rate is set as 0.0001. After 1000 steps, the training of GCN achieves convergence during about 290 minutes. Most of the training time is spent on the interaction with the EI environment.

C. Results and Analysis

Table II lists the average losses and reward values of the training set and test set. It can be seen that all the training and test can achieve voltage stability during voltage sag conditions.

TABLE II: Average Losses and Reward Values of the Training and Test Set

Data Set	Loss	Reward
Training	0.0085	0.904
Test	0.0101	0.908

Fig. 6 shows the reactive power compensation result of the voltage sag condition in Fig. 5. 2.4 MVar reactive power is put into the EI system, which raises the voltage of load 1 to above 0.8 pu and ensures the voltage stability of the EI system. The reward is 0.842. The dosh lines shows the reactive power compensation result given by static var generator. As static var generator only receives the voltage of the main bus, the line voltages cannot recover above 0.8 pu.

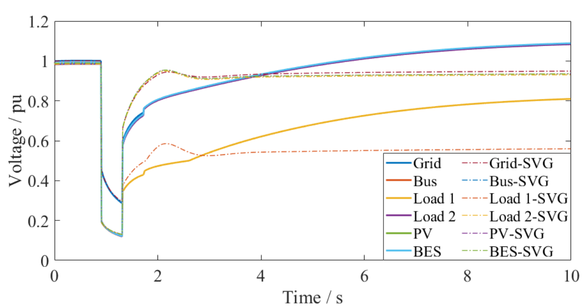

Fig. 6: Reactive power compensation result of GCN action.

To show the superiority of the proposed approach, GCN is further compared with the state-of-the-art methods, which include deep reinforcement learning (DRL) [17] and static var generator (SVG). Table III shows the average reward of all methods using the training and test set.

The comparison result shows that the proposed approach has 1.1% and 5.0% higher average reward than DRL and SVG for voltage stability, respectively. SVG has a slow response to the voltage sag, which results in a low average reward. DRL do not take EI topology into consideration, therefore, reactive power compensation is not accurate enough. Compared with the state-of-the-art methods, GCN has better feature fusion ability of adjacent EI nodes and rapid response capacity to be an effective approach for voltage stability in EI.

TABLE III: Comparison with the State-of-the-art Methods

Methods	Training Set Reward	Test Set Reward	Total
DRL	0.895	0.897	0.896
SVG	0.873	0.852	0.863
GCN	0.904	0.908	0.906

V. Conclusion

This paper proposed a novel reactive power optimization approach for voltage stability in EI based on GCN and DQN. GCN that considers EI topology is first introduced to reactive power optimization. The training of GCN uses the method of DQN to get the optimal reactive power compensation value. The case study on an EI model shows that the proposed approach can ensure the voltage stability of EI during voltage sag. The comparison study shows that the proposed approach provide more accurate reactive power compensation than the state-of-the-art methods.

In future work, the GCN can be further used in EI with more nodes and reactive power compensation devices. With the topology information and special structure of GCN, the optimization problems in EI with multi-agents can be solved.

References

[1] Y. Qin, H. Hua, and J. Cao, "Stochastic optimal control scheme for battery lifetime extension in islanded microgrid via a novel modeling approach," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 4467–4475, 2019.

[2] A. Camacho, M. Castilla, J. Miret, R. Guzman, and A. Borrell, "Reactive power control for distributed generation power plants to comply with voltage limits during grid faults," *IEEE Transactions on Power Electronics*, vol. 29, no. 11, pp. 6224–6234, 2014.

[3] T. Sousa, H. Morais, Z. Vale, and R. Castro, "A multi-objective optimization of the active and reactive resource scheduling at a distribution level in a smart grid context," *Energy*, vol. 85, pp. 236 – 250, 2015.

[4] R. Anilkumar, G. Devriese, and A. K. Srivastava, "Voltage and reactive power control to maximize the energy savings in power distribution system with wind energy," *IEEE Transactions on Industry Applications*, vol. 54, no. 1, pp. 656–664, 2018.

[5] H. Shi, H. Wen, Y. Hu, and L. Jiang, "Reactive power minimization in bidirectional dc–dc converters using a unified-phasor-based particle swarm optimization," *IEEE Transactions on Power Electronics*, vol. 33, no. 12, pp. 10 990–11 006, 2018.

[6] W. Li, D. Deka, M. Chertkov, and M. Wang, "Real-time faulted line localization and pmu placement in power systems through convolutional neural networks," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 4640–4651, 2019.

[7] H. Hua, Y. Qin, C. Hao, and J. Cao, "Optimal energy management strategies for energy internet via deep reinforcement learning approach," *Applied Energy*, vol. 239, pp. 598 – 609, 2019.

[8] T. Fu, C. Wang, and N. Cheng, "Deep-learning-based joint optimization of renewable energy storage and routing in vehicular energy network," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6229–6241, 2020.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–21, 2020.

[10] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, 2020.

[11] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1040–1057, 2020.

[12] Z. Zheng, Y. Yang, X. Niu, H. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Trans. Ind. Inform.*, vol. 14, no. 4, pp. 1606–1615, 2018.

[13] S. Guo, B. Zhang, T. Yang, D. Lyu, and W. Gao, "Multitask convolutional neural network with information fusion for bearing fault diagnosis and localization," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 8005–8015, 2020.

[14] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *arXiv e-prints*, p. arXiv:1609.02907, Sep. 2016.

[15] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29*, pp. 3844–3852. Curran Associates, Inc., 2016.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, and J. Veness, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[17] J. Cao, W. Zhang, Z. Xiao, and H. Hua, "Reactive power optimization for transient voltage stability in energy internet via deep reinforcement learning approach," *Energies*, vol. 12, p. 1556, 04 2019.