Service Restoration Using Deep Reinforcement Learning and Dynamic Microgrid Formation in Distribution Networks

Mosayeb Afshari Igder, *Graduate Student Member, IEEE*, and Xiaodong Liang, *Senior Member, IEEE*

***Abstract*—A resilient power distribution network can reduce length and impact of power outages, maintain continuous services, and improve reliability. One effective way to enhance the system's resilience is to form microgrids during outages. In this article, a novel dynamic microgrid formation-based service restoration method using deep reinforcement learning is proposed, and it is treated as a Markov decision process (MDP) while taking operational and structural limitations of microgrids into account. The deep Q-network is employed to obtain optimal control strategies for microgrid formation. We have introduced a new way for the agent to choose actions when building a microgrid using the deep Q-learning method, which ensures that the microgrid has a feasible radial structure. The proposed service restoration method enables real-time computing to facilitate online formation of dynamic microgrids and adapts to changing conditions. The influence of optimal switch placement on service restoration using proposed method is also investigated. The effectiveness of proposed service restoration method is validated by case studies using the modified IEEE 33-node test system and a real 404-node distribution system operated by Saskatoon Light and Power in Saskatoon, Canada.**

***Index Terms*—Deep Q-learning, deep reinforcement learning, distribution network, microgrid formation, service restoration.**

I. Introduction

IMPROVING resiliency is crucial to ensure that power grids can withstand and recover from disruptions, such as natural disasters and cyber-attacks [1], [2]. Service restoration capability to critical loads after disruptions on the main grid is a key indicator of a resilient distribution system. The traditional practice is to redirect affected loads from areas without power to areas with power through the network reconfiguration [3]. Today, forming microgrids (MGs) with dynamic boundaries is a promising service restoration solution to improve resiliency of distribution systems [4], [5]. By incorporating a variety of energy sources, such as solar panels, wind turbines and backup generators, along with remotely control switches, a distribution network can be partitioned into multiple self-adequate MGs, through which the restoration is improved and the power supply continuity to critical loads is maintained [6], [7], [8], [9], [10]. Optimal construction of multiple MGs to restore critical loads during a fault in the primary grid is proposed in [11], [12]. In [13], distributed energy resources (DER) and isolate switches are allocated in a distribution network to form resilience-oriented MGs. In [14], a two-stage service restoration method is proposed for the mobile emergency resource allocation, MG formation, and sequential service restoration for contingencies. In [15], a self-healing control strategy is proposed to minimize unused capacities of distributed generation (DG) for service restoration in islanded mode during a contingency.

The main challenge of a multi-MGs formation problem is to find the most suitable topology that satisfies various operational constraints. Mathematical programming [16], [17], [18] and heuristic search [19], [20] approaches are commonly used to achieve the topology determination. In [16], a novel mixed-integer linear programming (MILP) method is used to create an optimization model to form MGs in distribution networks after a disturbance. In [17], a MILP-based service restoration method is developed to determine the optimal hardening plan, allocation of DGs, and topology reconfiguration, which ensures that a predetermined level of load is supplied after natural disasters. The MG formation problem is formulated as a mixed-integer non-linear programming (MINLP) model in [18], and a commercial solver, DCOPT, is used to address the optimization problem. A heuristic-based method is proposed to identify optimal reconfiguration of a large-scale distribution network in [19]. In [20], a tabu search optimization algorithm and a graph theory-based method are used to form several virtual MGs in a distribution network. These MG formation strategies are primarily focused on present conditions/environment. However, conditions during a natural disaster may be uncertain and prone to change [21], causing constructed MGs lose their effectiveness or get damaged.

To use mathematical programming methods, creating a precise mathematical model is essential, but this process can be time-consuming, and thus, may not be practical for large systems. Model-based schemes may also have difficulty to

Manuscript received 29 April 2023; accepted 13 June 2023. Date of publication 20 June 2023; date of current version 20 September 2023. Paper 2023-PSEC-0648, presented at the 2023 IEEE/IAS 59th Industrial and Commercial Power Systems Technical Conference, Las Vegas, NV, USA, May 21–25, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Power Systems Engineering Committee of the IEEE Industry Applications Society [DOI: 10.1109/ICPS57144.2023.10142118]. This work was supported by Research Junction Development co-funded by the University of Saskatchewan and the City of Saskatoon, Canada. *(Corresponding author: Xiaodong Liang.)*

The authors are with the Department of Electrical and Computer Engineering, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada (e-mail: moa443@mail.usask.ca; xil659@mail.usask.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIA.2023.3287944.

Digital Object Identifier 10.1109/TIA.2023.3287944

model complex objects, integrate new features, and maintain efficiency for large systems. To overcome these challenges, novel adaptive dynamic MG formation strategies are urgently needed.

Deep reinforcement learning (DRL) can continuously interact with the environment and gather feedback, so it can form MGs adaptable to changing conditions [22]. DRL is effective in handling Markov decision process (MDP), and becomes popular to tackle many problems in power systems, such as voltage control [23], electrical vehicles charging navigation [24], demand response [25], MG power management [26], and resiliency improvement of distribution networks [27]. However, service restoration using DRL-based MG formation has not been reported in the literature yet.

In this article, a novel dynamic microgrid formation-based service restoration method using deep reinforcement learning for distribution networks is proposed for the very first time. We have adopted the node cell and route model concept in [2] to start the process. Each DG with the black-start capability is treated as an energization agent, and it travels through the system to energize lines and nodes it visits. Using the node cell concept, nodes connected by non-switchable lines are grouped into a single unit, known as a "node cell", and all nodes within a node cell are activated simultaneously when an energization agent visits, which greatly reduces the search space. Next, the deep Q-network algorithm guides the agent to select the node cell and pick up as many loads as possible while following operational constraints. To ensure radiality constraints for each constructed MG, we propose a novel algorithm to pick up node cells by an energization agent. A simulator-based environment is created using the software, OpenDSS, integrated for power flow studies. A simulator offers a variety of features and can perform a range of tasks, and thus, facilitates complex object-oriented learning, and new features can be integrated through agent-environment interactions in a simulator. This adaptable capability makes simulators suitable for real-world application.

The main contributions of this article include:

- A novel dynamic MG formation-based service restoration method using DRL for distribution networks is proposed.
- A new DRL framework through MDP is developed to form dynamic MGs by incorporating the node cell and route model concept (The deep Q-network is trained offline, and then used for online decision-making to provide fast, near-optimal solutions).
- The proposed method is validated using the IEEE 33-node test system and a real unbalanced three-phase 404-node distribution system operated by Saskatoon Light and Power, a Canadian electric utility in Saskatoon, Canada.

The article is organized as follows: the proposed method is introduced in Section II; its problem formulation is given in Section III; the DRL algorithm is provided in Section IV; the proposed approach is validated by case studies in Sections V and VI; optimal switch placement and its effect on service restoration is studied in Section VII; and conclusions are drawn in Section VIII.

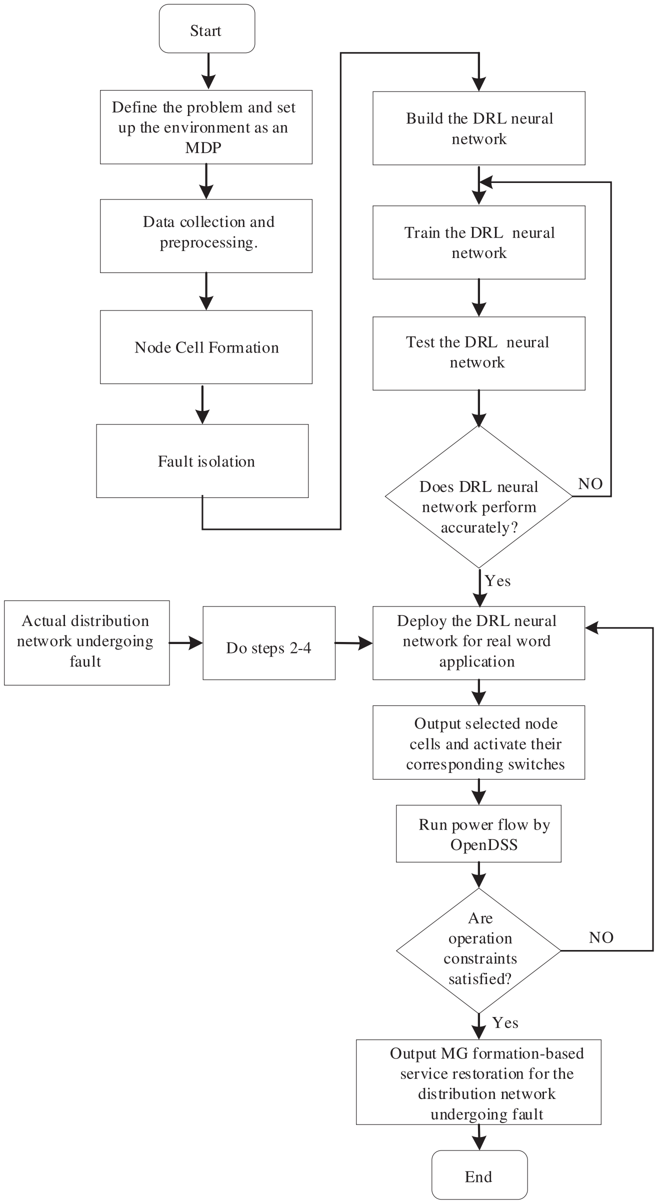
Start
Define the problem and set up the environment as an MDP
Data collection and preprocessing.
Node Cell Formation
Fault isolation
Build the DRL neural network
Train the DRL neural network
Test the DRL neural network
Does DRL neural network perform accurately?
NO
Yes
Actual distribution network undergoing fault
Do steps 2-4
Deploy the DRL neural network for real word application
Output selected node cells and activate their corresponding switches
Run power flow by OpenDSS
Are operation constraints satisfied?
NO
Yes
Output MG formation-based service restoration for the distribution network undergoing fault
End

Fig. 1. Flow chart of the proposed service restoration method through dynamic MG formation and DRL in distribution networks.

II. The Proposed Service Restoration Method Using Deep Reinforcement Learning

Modern distribution systems with renewable DGs and controllable devices have significant potential to enhance service restoration. In this article, the proposed service restoration method can maximize the restored load by forming multiple dynamic microgrids. DRL is trained offline and then applied online for fast and efficient decision-making. DRL is used to create control actions, while ensures that operational and topological constraints are met.

The proposed service restoration method can be implemented by the following eight steps (Fig. 1).

Step 1. Define the problem and set up the environment as an MDP: The objective of service restoration, state and action spaces, and rewards/penalties related to different actions are defined.

Step 2. Data collection and preprocessing: Gather data for an electrical distribution system, including the details of generation and load. To establish an appropriate input format for neural network, the data are preprocessed.

Step 3. Node Cell Formation: The node cell concept is used to convert a distribution network to a smaller simplified network with only switchable lines.

Step 4. Fault isolation: Open upstream switches of node cells containing faults.

Step 5. Deep reinforcement learning model construction: A deep neural network-based reinforcement learning model that can make decisions for service restoration is constructed. The model consists of an input layer that receives the state data of the distribution network, and an output layer that generates a probability distribution for possible actions.

Step 6. Model training using the deep Q-learning approach: The DRL model is trained using the preprocessed data and the Q-learning approach, which involves optimizing the model's decision-making process based on rewards and penalties associated with different actions.

Step 7. Model testing and evaluation: The trained DRL model is tested using new data for service restoration in distribution networks.

Step 8. Real-world deployment of the trained model: The trained DRL model can then be used in a real-world distribution network for service restoration. This involves integrating the model into the operational environment of a distribution network and utilizing it to guide actual service restoration decisions based on its learned decision-making capabilities.

III. Problem Formulation

In this section, the MG formation-based service restoration and its MDP are formulated.

A. Microgrid Formation-Based Service Restoration

A graph *G* = *(V, E)* is used to represent a distribution network, where *V* and *E* refer to the sets of nodes and edges, respectively, and $|V| = N$, $|E| = L$. At each node *i* in V, there is a load with active and reactive power demand, p_i and q_i, respectively. Before converting the MG formation-based service restoration problem into a MDP, the objective function, and operational and topological constraints are defined.

1) Objective Function: The restoration aims to maximize the restored priority-weighted loads based on the capacity of DGs. The objective function is expressed by

$$\mathrm{OF} = \max \sum_{i \in V} z_i w_i \mathcal{P}_i \tag{1}$$

where $\mathcal{P}_i$ is the restored load (active power). z_i is a binary variable determining if load *i* is picked up. w_i is the priority weight of load *i*.

2) Operational Constraints: The linearized DistFlow model [11] is used to represent operational constraints in the MG formation-based service restoration. Equations (2) and (3) ensure active and reactive power balance, respectively.

$$\mathcal{P}_i^{g} - z_i p_i + \sum_{(j,i) \in E} \mathcal{P}_{ji}^{\mathcal{BR}} - \sum_{(i,j) \in E} \mathcal{P}_{ij}^{\mathcal{BR}} = 0, \forall i \in V \tag{2}$$

$$\mathcal{Q}_i^{g} - z_i q_i + \sum_{(j,i) \in E} \mathcal{Q}_{ji}^{\mathcal{BR}} - \sum_{(i,j) \in E} \mathcal{Q}_{ij}^{\mathcal{BR}} = 0, \forall i \in V \tag{3}$$

where $\mathcal{P}_i^{g}$ and $\mathcal{Q}_i^{g}$ are active and reactive power output of DGs at node *i*, respectively. p_i and q_i are real and reactive demand of the load at node *i*, respectively. Real and reactive power flow on branch *(i, j)* are $\mathcal{P}_{ij}^{\mathcal{BR}}$ and $\mathcal{Q}_{ij}^{\mathcal{BR}}$, respectively. Limits for DG output power are expressed by (4).

$$\underline{\mathcal{P}}_m^{\min} \le \mathcal{P}_m^{g} \le \overline{\mathcal{P}}_m^{\max}, \underline{\mathcal{Q}}_m^{\min} \le \mathcal{Q}_m^{g} \le \overline{\mathcal{Q}}_m^{\max}, \forall m \in V_{DG} \tag{4}$$

Where $\overline{\mathcal{P}}_m^{\max}$ and $\underline{\mathcal{P}}_m^{\min}$ are upper and lower limits of active power output of DGs, respectively. $\overline{\mathcal{Q}}_m^{\max}$ and $\underline{\mathcal{Q}}_m^{\min}$ are upper and lower limits of reactive power output of DGs, respectively. V_{DG} is the set of nodes containing DGs. Equation (5) enforces the relation between the voltage difference of two end nodes and the power flow of each closed line.

$$\begin{aligned} & - M\left(1 - \alpha_{ij}\right) + \frac{\left(r_{ij}.\mathcal{P}_{ij}^{\mathcal{BR}} + x_{ij}.\mathcal{Q}_{ij}^{\mathcal{BR}}\right)}{\nu_0} \le \upsilon_j - \upsilon_i \\ & \le M\left(1 - \alpha_{ij}\right) + \frac{\left(r_{ij}.\mathcal{P}_{ij}^{\mathcal{BR}} + x_{ij}.\mathcal{Q}_{ij}^{\mathcal{BR}}\right)}{\nu_0}, \forall\left(i, j\right) \in E \end{aligned} \tag{5}$$

where α_{ij} is a binary variable with a value of 1 if line *(i, j)* is closed, and 0 if otherwise. r_{ij} and x_{ij} are the resistance and reactance of line *(i, j)*, respectively. ν_0 is the reference voltage. To ensure that voltages at two unconnected buses are separated, the Big M method is used. The voltage range (υ_i) is defined by

$$\upsilon^{\min} \le \upsilon_i \le \upsilon^{\max} \tag{6}$$

where $\nu^{\max}$ and $\nu^{\min}$ are the maximum and minimum voltage magnitude squared. Safe margins should be maintained for line loading conditions.

$$- \mathcal{P}_{ij}^{\max} \le \alpha_{ij}.\mathcal{P}_{ij}^{\mathcal{BR}} \le \mathcal{P}_{ij}^{\max} \tag{7}$$

$$- Q_{ij}^{\max} \le \alpha_{ij}.Q_{ij}^{\mathcal{BR}} \le Q_{ij}^{\max} \tag{8}$$

3) Topological Constraints: The multicommodity flow-based model [15] is used to ensure a radial topology of MGs. The topological constrains are provided as follows:

$$\sum_{(j,i_s) \in E} \mathcal{F}_{ji_s}^k - \sum_{(i_s,j) \in E} \mathcal{F}_{i_s j}^k + 1 = 0, \forall\, k \in V \backslash i_s \tag{9}$$

$$\sum_{(j,k) \in E} \mathcal{F}_{jk}^k - \sum_{(k,j) \in E} \mathcal{F}_{kj}^k - 1 = 0\,, \quad \forall\, k \in V \backslash i_s \tag{10}$$

$$\sum_{(j,i) \in E} \mathcal{F}_{ji}^k - \sum_{(i,j) \in E} \mathcal{F}_{ij}^k = 0, \forall\, k \in V \backslash i_s, \forall\, i \in V\, \{k, i_s\} \tag{11}$$

$$0 \le \mathcal{F}_{ij}^k \le \lambda_{ij}, 0 \le \mathcal{F}_{ji}^k \le \lambda_{ji}, \forall\, k \in V \backslash i_s, (\mathrm{i,j}) \in E \tag{12}$$

$$\sum_{(i,j) \in E} \lambda_{ij} + \lambda_{ji} - |V| + 1 = 0 \tag{13}$$

$$\lambda_{ij} + \lambda_{ji} = \sigma_{ij}\ , \forall\, (\mathrm{i,j}) \in E \tag{14}$$

$$\alpha_{ij} - \sigma_{ij}\ \leq 0 \tag{15}$$

where $\mathcal{F}_{ij}^{k}$ is the flow of electric power (commodity) k from nodes *i* to *j*. λ_{ij} is a binary variable with a value of 1 if arc *(i, j)* is part of the directed spanning tree, and 0 if otherwise. σ_{ij} represents a fictitious status of branch *(i, j)*, it is 1 when closed, and 0 when otherwise. i_s is the index for the substation or DG buses. The constraints (9)–(14) are used to create a fictitious tree-like structure, known as “spanning tree”, by connecting all nodes in the network, but it may not be the same as the actual physical layout of the network. The constraint (15) restricts the network to only use certain connections in the spanning tree that are specified by the variable, σ [16].

B. Modeling Microgrid Formation-Based Service Restoration as a Markov Decision Process

In MDP, the decision-maker makes a sequence of decisions over time; outcomes of these decisions depend on current state of the system, and actions taken by the decision-maker. By formulating MG formation-based service restoration as a MDP, this decision-making problem can be broken down into a sequence of simpler decision problems, where the decision-maker (agent) must choose which load should be picked up based on the current state of the network, and expected future outcomes of each decision.

State $s_{m,t} \in S$: The state of the MG formation for service restoration consists of: 1) the current location of the agent, $LA_{m,t}$; 2) all visited node cells, VN_t; 3) the load condition, P_t; and 4) the remaining capacity of the DG, $C_{m,t}$.

$$s_{m,t} = [LA_{m,t}, VN_t, P_t, C_{m,t}] \tag{16}$$

Action $a_{m,t} \in A$: The action is to pick up a node cell and switch on a branch. Nodes connected by non-switchable lines are treated as a node cell, and all nodes within a node cell are activated and supplied power simultaneously if this node cell is visited by an energization agent. The simplified network topology only contains node cells and switchable lines, and the action space contains only candidate node cells. A agent travels through the network, and chooses node cells by following topological constraints: 1) each MG is isolated from other MGs, and 2) all constructed MGs operate in a tree topology to ensure radiality. This problem is built over a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where the node cell is represented by node $n \in \mathcal{V}$, and the energization path is indicated by edge $e \in E$, which connects two node cells. The adjacent $\mathcal{V} \times \mathcal{V}$ matrix A is used to represent this graph, its element a_{ij} is 1 when there is an edge from node cells i to j, and 0 when there is no such edge. For an agent to choose a node cell as an action, there should be a connection between its current position/previously visited node cells and the candidate node cell to maintain connectivity within a MG.

To ensure the radiality of a MG, the following steps should be implemented when an agent takes an action:

Step 1: The energization path must begin from either a DG or a substation.

Algorithm 1: Action Selection.

```
for agent m = 1 to the number of agents
  Observe state $s_{m,t}$ and adjacent matrix
    Condition1 = False, Condtion2 = False, Condtion3
    = False
  Select action Via exploration or exploitation
  if selected action not in the visited nodes
      Condition1 = True
    end if
  if selected action load ≤ agent DG capacity
      Condition2 = True
      end if
    Check the connection between the selected action
    and the agent current position or visited nodes
      if there is any connection
      Condition3 = True
      end if
    if Condition1 and Condtion2 and Condition3
          Take action
          break
        else:
          select another action and check three conditions
          again
      end if
    if agent couldn't take any action
      agent's done = True
    end if
end for
```

Step 2: When an agent travels, its path shouldn’t include any loop, i.e., a node cell cannot be visited more than once.

Step 3: A agent can pick up a node cell only if its upstream node cell has already been energized.

An agent selects the next node cell to visit at each step based on its current power generation capacity, and all loads within the selected node cell will be restored at the same time. When an agent restores a node cell, its capacity is updated as follows:

$$\mathcal{P}^{\max}_{m,t} = \overline{\mathcal{P}}^{\max}_{m} - \sum_{l \in N^L_{t-1}} \sum_{k=1}^{t-1} \mathcal{P}^L_{l,k} \tag{17}$$

$$\mathcal{Q}^{\max}_{m,t} = \overline{\mathcal{Q}}^{\max}_{m} - \sum_{l \in N^L_{t-1}} \sum_{k=1}^{t-1} \mathcal{Q}^L_{l,k} \tag{18}$$

Where $\mathcal{P}^{\max}_{m,t}$ and $\mathcal{Q}^{\max}_{m,t}$ are the maximum active and reactive power of DG m at time *t*, respectively. $\overline{\mathcal{P}}^{\max}_{m}$ and $\overline{\mathcal{Q}}^{\max}_{m}$ are the maximum active and reactive power capacities of DG m, respectively. $\mathcal{P}^L_l$ and $\mathcal{Q}^L_l$ are active and reactive power demand of node cell *l*. The second terms of (17) and (18) indicate the amount of active and reactive power being restored already.

The procedure of selecting an action by an agent is demonstrated in Algorithm 1 below.

Reward $r_{m,t}(s_t, a_t)$: The agents fulfill the power balance, branch limits, and voltage constraints while taking actions. There are two methods in the literature to address the MDP under constraints: 1) use a safety layer to tune control actions, and 2) add penalty functions to the reward signal. In this article, the second method is adopted using penalty terms to solve a constrained MDP with the following reward signal:

$$\mathscr{r}_{m,t} = w_l \mathcal{D}_l \ + C_p + C_q + C_b + C_v \tag{19}$$

$$\mathcal{D}_l = \sqrt{\mathcal{P}_l^{L2} + \mathcal{Q}_l^{L2}} \tag{20}$$

$$C_p = \begin{cases} -\left|\Delta\mathcal{P}\right| & \left|\Delta\mathcal{P}\right| \leq \partial_p \\ -\eta * \left|\Delta\mathcal{P}\right| & \left|\Delta\mathcal{P}\right| > \partial_p \end{cases} \tag{21}$$

$$C_q = \begin{cases} -\left|\Delta\mathcal{Q}\right| & \left|\Delta\mathcal{Q}\right| \leq \partial_q \\ -\eta * \left|\Delta\mathcal{Q}\right| & \left|\Delta\mathcal{Q}\right| > \partial_q \end{cases} \tag{22}$$

$$\Delta\mathcal{P} = \sum_{i=1}^{N} \left(\sum_{h:(h,i)\in B} \mathcal{P}_{hi}^{\mathcal{BR}} + \mathcal{P}_i^{\mathscr{g}} - \sum_{j:(i,j)\in B} \mathcal{P}_{ij}^{\mathcal{BR}} - \mathcal{P}_t^L \right) \tag{23}$$

$$\Delta\mathcal{Q} = \sum_{i=1}^{N} \left(\sum_{h:(h,i)\in B} \mathcal{Q}_{hi}^{\mathcal{BR}} + \mathcal{Q}_i^{\mathscr{g}} - \sum_{j:(i,j)\in B} \mathcal{Q}_{ij}^{\mathcal{BR}} - \mathcal{Q}_t^L \right) \tag{24}$$

$$\begin{aligned} C_b = & - \sum_{(i,j)\in B} (\max(0, \mathcal{P}_{ij}^{\mathcal{BR}} - \mathcal{P}_{ij}^{\max}) \\ & + \max(0, \mathcal{P}_{ij}^{\max} - \mathcal{P}_{ij}^{\mathcal{BR}})) \\ & - \sum_{(i,j)} \in B(\max(0, \mathcal{Q}_{ij}^{\mathcal{BR}} - \mathcal{Q}_{ij}^{\max}) \\ & + \max(0, \mathcal{Q}_{ij}^{\max} - \mathcal{Q}_{ij}^{\mathcal{BR}})) \end{aligned} \tag{25}$$

$$C_v = - \sum_{i=1}^{N} \left(\max\left(\mathcal{V}_i - \mathcal{V}^{\max}, 0\right) + \max\left(\mathcal{V}^{\min} - \mathcal{V}_i, 0\right) \right) \tag{26}$$

where $\mathcal{D}_l$ in (20) is the demand of node cell l. The first term of (19), $w_l \mathcal{D}_l$, represents the load pick-up with its priority weight. C_p and C_q in (21) and (22) are penalty terms for active and reactive power imbalance, respectively. $\Delta\mathcal{P}$ and $\Delta\mathcal{Q}$ in (23) and (24) are active and reactive power imbalance, respectively. C_b in (25) is the penalty term for the branch limit violation. C_v in (26) is a penalty term for the voltage limit violation. ∂_p and ∂_q are thresholds to maintain power imbalance within a permissible range. η is a penalty factor when the power imbalance is greater than a threshold.

IV. Deep Reinforcement Learning Algorithm for Service Restoration

Reinforcement learning (RL) concerns about decision making, and identifies the best strategy that yields the maximum cumulative reward through the best combination of actions by intelligent agents [28]. The agent interacts with the environment in discrete time steps, and receives the state of an environment, based on which the agent takes action to alter the state of the environment. Then the environment will send the agent a reward for the current time step, and the state for the next time step. A RL algorithm can tackle a problem expressed as the MDP, and the optimal solution of this problem can then be identified using the deep Q-learning algorithm.

A. Deep Q-Learning

The deep Q-learning algorithm is the core concept in RL, and uses deep learning to improve the learning capability [30]. A deep Q-network (DQN) is used to approximate Q-values (the estimated optimal future values) that the agent will learn. The input of a DQN is the state of the environment, its outputs are Q-values for available actions, and these Q-values are updated iteratively. For a policy, φ, which is a neural network mapping from its input to output, the Q-value function $\mathcal{Q}^{\varphi}(s_{m,t}, a_{m,t})$ is formulated as follows [30]:

$$\begin{aligned} \mathcal{Q}^{\varphi}\left(s_{m,t}, a_{m,t}\right) &= \mathbb{E}\left[\sum_{t'=t}^{T} \gamma^{t'-t} \left[\mathscr{r}_{m,t'}\left(s_{m,t'}, a_{m,t'}\right)\right]\right] \\ &= \mathbb{E}\left[\mathscr{r}_{m,t} + \gamma \mathcal{Q}^{\varphi}\left(s_{m,t+1}, \varphi\left(s_{m,t+1}\right)\right]\right. \end{aligned} \tag{27}$$

Where $\mathcal{Q}^{\varphi}(s_{m,t}, a_{m,t})$ is the expected Q-value for the state-action pair, $s_{m,t}$ and $a_{m,t}$, at time step t. γ is the discount factor. $\mathscr{r}_{m,t}$ is the immediate reward at time step t. The Q-value is a measure of the expected cumulative reward that the agent can achieve by taking a specific action $a_{m,t}$ in a particular state $s_{m,t}$, considering the policy φ as the decision-making mechanism. The DRL aims to find an optimal policy φ^* to achieve the maximum expected cumulative reward. In a DQN, the neural network with the parameters ϑ, $\mathcal{Q}(s_{m,t}, a_{m,t}|\vartheta)$, is trained to minimize the following loss function, $\mathcal{L}(\vartheta)$, known as a mean-squared Bellman error [30]:

$$\begin{aligned} \mathcal{L}\ (\vartheta) = & \Big[\mathscr{r}_{m,t} + \gamma \max_{a_{m,t+1}} \mathcal{Q}\left(s_{m,t+1}, a_{m,t+1}\left|\vartheta\right.\right) \\ & - \mathcal{Q}\left(s_{m,t}, a_{m,t}\left|\vartheta\right.\right)\Big]^2 \end{aligned} \tag{28}$$

Where the 1st term in (28) represents immediate reward obtained by the agent for taking action $a_{m,t}$ in state $s_{m,t}$; the 2nd term reflects the estimated future rewards that the agent can obtain by selecting the best action in the next state; and the 3rd term is the current estimated Q-value for the current state-action pair ($s_{m,t}$, $a_{m,t}$), obtained from the neural network with parameters, ϑ. By minimizing the loss function in (28), a DQN learns to produce Q-values, leading to a proper selection of actions. To minimize this loss, the Adam optimization algorithm is used:

$$\vartheta \leftarrow Adam\left(\vartheta, \nabla_{\vartheta} \mathcal{L}\left(\vartheta\right)\right) \tag{29}$$

where ∇_{ϑ} is the policy gradient.

B. Enhance Learning Process of DRL

One essential part of DRL is to ensure the Q-network learn appropriate reactions in the MDP. Three most successful approaches are the Epsilon-greedy-based exploration, the fixed Q-network, and the experience replay.

1) Epsilon-Greedy-Based Exploration: Due to the random initialization of weights and biases in the Q-network prior to training, it is challenging to recognize actions that result in the greatest long-term reward. At the start of training, instead of relying solely on the Q-network to select actions, the agent can randomly select actions to explore all potential outcomes and receive a satisfactory reward. After being trained, the agent can then exploit the environment by selecting actions based on the DQN approximation. The epsilon-greedy strategy is an effective way to balance exploration and exploitation in DRL, allowing the agent to efficiently learn and adapt its actions to achieve optimal performance in the environment. The agent makes random action choices with a probability of epsilon (ε), allowing it to explore different actions and gather information about their associated rewards and consequences. The exploration rate, ε, can be adjusted using an exponential decay, and its value gradually decreases over the course of the Q-network training. At each time step, the agent generates a random number between 0 and 1. If the generated number is below ε, the agent selects the action with the highest Q-value; otherwise, the agent randomly selects an action. ε is formulated by

$$\varepsilon \;= \varepsilon_F \;+ (\varepsilon_{int} - \varepsilon_F) * e^{-\varepsilon_{decay} * episode} \tag{30}$$

where ε_F, ε_{int}, and ε_{decay} are the final, initial, and decay rate of exploration, respectively. $episode$ is the episode number.

2) Fixed Q-Network: Determining the maximum Q-value of the next state and the Q-value of the current state using the same Q-network during an update process, the loss calculation may cause overestimation in deep Q-learning. To address this concern, a separate Q-network, known as the target Q-network, is established to obtain the Q-value of the next state. The target Q-network has the same structure and parameters as the Q-network, and is periodically updated with the Q-network parameters during training.

$$\mathcal{L}(\vartheta) = \begin{cases} \left[r_{m,t} - Q\left(s_{m,t}, a_{m,t} \left| \vartheta \right.\right)\right]^2 & t = T \\ \left[\begin{array}{c} r_{m,t} + \gamma \max\limits_{a_{m,t+1}} Q^{Target}\left(s_{m,t+1}, a_{m,t+1} \left| \vartheta^{Target} \right.\right) \\ - Q\left(s_{m,t}, a_{m,t} \left| \vartheta \right.\right) \end{array}\right]^2 & t \neq T \end{cases} \tag{31}$$

Where Q^{Target} is the expected Q-value for the target Q-network. ϑ^{Target} are the neural network parameters for the target Q-network.

3) Experience Replay and Multi-Buffers: The DQN can be trained based on prior experiences. For this purpose, an experience replay mechanism with memory and replay components is employed. This mechanism stores the agent's experience (the state, action, reward, and next state at time step t) in the memory. Once enough experience has been stored, the replay process is initiated, and a random selection of experience is drawn from the memory. This helps to remove any bias due to correlations between data samples. The learning of DQN also relies heavily on the stored experience, and aligning experience with high reward values can enhance the learning and lead to an improved selection of actions. As a result, a multi-buffer approach with two memories is employed. The first memory (*memo1*), denoted as the "original memory," is used to store all the experiences collected during the training process. The second memory (*memo2*), known as the "reward-based memory," is specifically designed to store the experience with high reward values extracted from the original memory. Subsequently, the Q-network is trained using the mini-batch that is generated from two separate memories, *memo1* and *memo2*. During the training process, the mini-batch in (34) is randomly sampled from both memories, which allows a diverse set of experience to be used for training.

TABLE I
DG Parameters for the Modified IEEE 33-Node Test System

Parameters	DG1	DG2	DG3
Node	1	33	18
$P_g^{max}(MW)$	1.5	1.2	1
$P_g^{\min}(MW)$	0.15	0.1	0.1
$Q_g^{max}(MVar)$	1.2	1	0.8
$Q_g^{min}(MVar)$	-0.9	-0.8	-0.8
Status	BSDG/DDG	BSDG/DDG	BSDG/DDG

$$memo1 \;=\; [\ldots, (s_{m,t}, a_{m,t}, r_{m,t}, s_{m,t+1}), \ldots] \tag{32}$$

$$memo2 \;=\; \left[\ldots, \left(s_{m,t}, a_{m,t}, r^{\dagger}_{m,t}, s_{m,t+1}\right), \ldots\right] \tag{33}$$

$$minibatch \;=\; \left[\begin{array}{c} random.sample(memo1, BatchSize1 \\ random.sample(memo2, BatchSize2 \end{array}\right] \tag{34}$$

Where the batch sizes, denoted as *BatchSize1* and *BatchSize2*, represent the number of samples used in each training step from the first and second memories, respectively.

C. Offline Training and Online Application of the Proposed Method

The framework of the proposed method, including offline training and online application, is shown in Fig. 2.

During the training phase, the agent interacts with a simulator-based environment that provides feedback (S_t, A_t, r_t, S_{t+1}) based on the agent's actions. The feedback, referred to as experience, is saved in the replay memory and later utilized to train the DQN and adjust parameters of the neural network. The training ends when a set number of steps have been completed and the loss function reaches the threshold. Afterwards, the trained Q-network is saved as the policy Q-network, and can be used for online applications for service restoration during faults. The input layer of the policy Q-network receives the state of a distribution system, and optimal MGs are formed by selecting actions with the highest Q-values in the output layer of the policy Q-network.

V. Validation Using IEEE 33-Node Test System

The proposed method is first validated through case studies using the modified IEEE 33-node test system, as shown in Fig. 3. The system has the load demand of 3.715 MW and 2.3 MVar, ten switchable lines and three DGs. Table I shows parameters of DGs (BSDG means "black-start DG", and DDG means "dispatchable DG"). The node cell concept is used to convert the original 33-node system into a simplified 14-node-cell system with only switchable lines, as shown in Fig. 4. The detailed

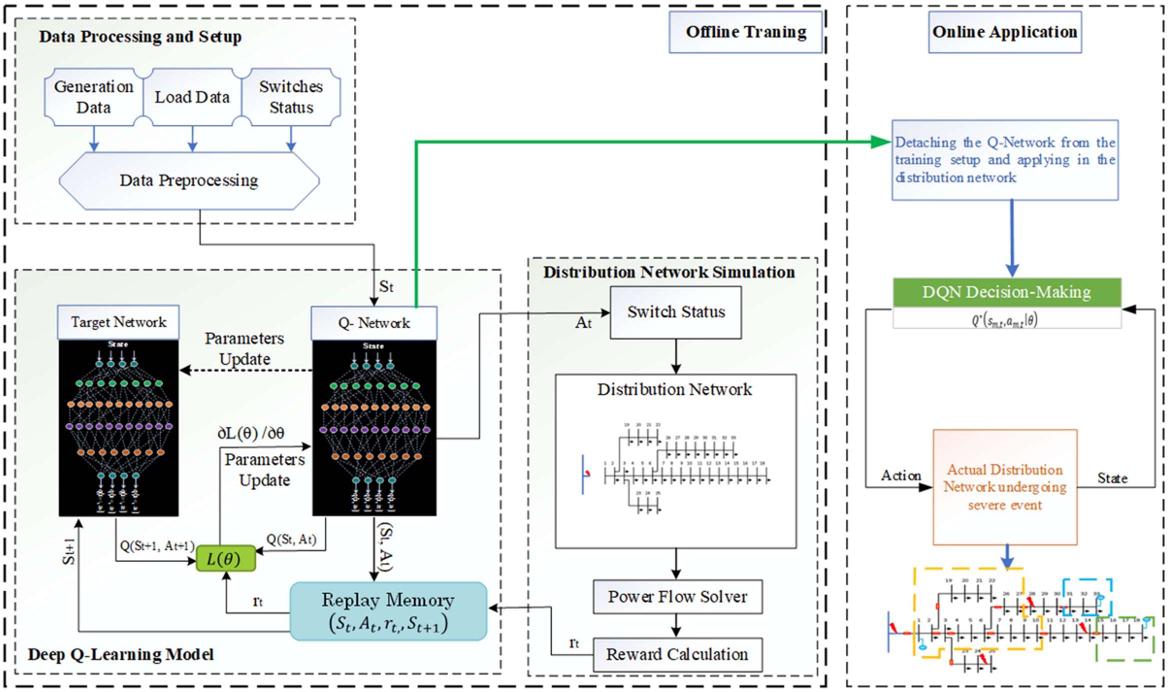

Fig. 2. Framework of DRL for MG-formation-based service restoration.

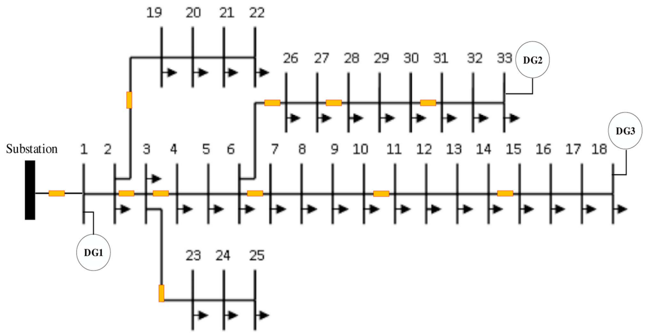

Fig. 3. Modified IEEE 33-node test system.

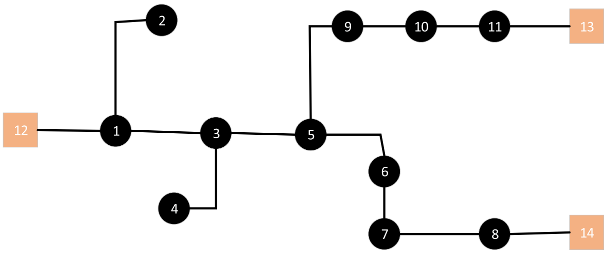

Fig. 4. Simplified IEEE 33-node test system with 14 node cells.

information about each node cell containing certain number of nodes in this system is shown in Appendix (Table XXIV). The hyper-parameter settings of the DQN for the test system are shown in Table II. Pytorch 1.10 and Python 3.9 are used to solve this service restoration problem.

TABLE II
HYPER-PARAMETER SETTINGS OF THE DQN FOR THE MODIFIED IEEE 33-NODE TEST SYSTEM

Hyper-parameters	Values
Number of Hidden Layer	3
No. of neurons in hidden layers	50, 50, 50
Learning rate	0.001
Reward discount factor	0.999
Activation function of output layer	Linear
Activation function of hidden layers	ReLU
Optimizer	Adam
Replay memory size	100000
ε_{min}	0.01
ε_{max}	1
ε_{decay}	0.001
Target update	10
Hyper-parameters	Values

A. Training

The structure of the Q-network is established during training as a neural network with three fully-connected linear layers, all of which have rectified linear units (ReLU). The input layer has the same number of neurons as the state parameters; the hidden layers contain 50 ReLU neurons each; and the output layer has the same number of neurons as the size of the action space.

The training of the proposed method using the test system is performed for 10000 episodes. The policy Q-network parameters ϑ are initialized by random values. During each episode, an agent selects an action based on the current state of the environment, then this action is applied to the environment, which transitions the agent to a new state, and generates a reward signal. Afterward, the current state, action, reward, and next state

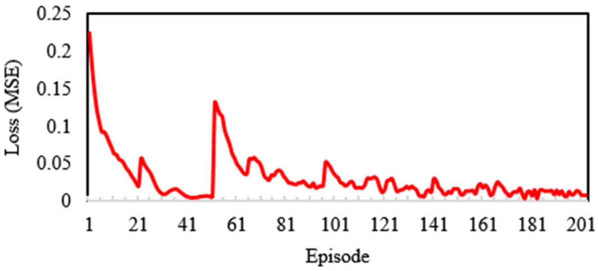

Fig. 5. Average loss in the training process.

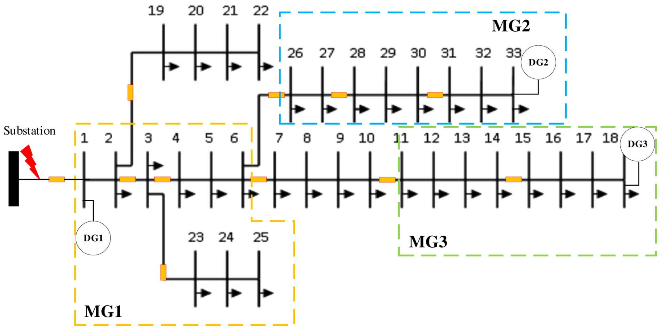

Fig. 6. Three microgrids formed in the modified IEEE 33-node test system using DQN (case 1).

are saved in a replay memory. After enough experiences are stored, they can be used to train the Q-network by minimizing the loss function calculated in (31). The convergence process of the training loss is shown in Fig. 5. Fig. 5 shows, as the episode number increases, the average loss of training decreases. After the loss becomes relatively stable with small oscillations, the original DQN can reasonably judge the performance of actions and form dynamic MGs for service restoration during outages/faults.

B. Testing and Implementation

After the DQN is trained properly and learned to make decisions, the following four case studies with different fault scenarios are conducted to test the DQN model and implement it for online applications.

1) Case1: In Case 1, a fault occurs at the substation (Fig. 6), and the substation must be isolated from the rest of the system. Accordingly, a switch is opened between Node cells 1 and 12 (Fig. 4) to clear the fault. Note when a faulty area is isolated by opening switches, its connection status in the adjacent matrix will be changed from 1 to 0. In Case 1, before the fault, the connection status between Node cells 1 and 12 was 1 in the adjacent matrix; after the fault, it became 0, i.e., no energization path between the two node cells, so energization agents will not visit node cell 12 during their travels through the system.

As shown in Fig. 6, three MGs powered by three black-start DGs are formed in Case 1. They can restore 76.31% of active power and 82.61% of reactive power of the entire load, and the computational time for restoration is 0.0004951 seconds (Table III). The system is only partially restored as the total capacity of DGs are less than the total load demand. Node numbers and the amount of power restored by each MG are also shown in Table III. Table IV shows the output power of DGs when restoring load in Case 1.

TABLE III
Restored Nodes and Power of the Modified IEEE 33-Node Test System Using DQN (Case 1)

Microgrid	Restored nodes	Restored loads (kW)	Restored loads (kVar)
MG1	1, 2, 3, 4, 5, 6, 23, 24, 25	1,360	680
MG2	26, 27, 28, 29, 30, 31, 32, 33	920	950
MG3	11, 12, 13, 14, 15, 16, 17, 18	555	270
$\frac{Total\ resotoredload}{Total\ load}*100$		76.31%	82.61%
Computational time (seconds)		0.0004951	

TABLE IV
DG Output Power of the Modified IEEE 33-Node Test System (Case 1)

Generation	Active power (kW)	Reactive power (kVar)
DG1	1,383.08	694.84
DG2	935.09	967.74
DG3	560.91	276

Fig. 7 provides a sequential visualization of the restoration path using a learned DQN. Energization agents are coordinated to restore load by the following rules: 1) no loops are formed, 2) the maximum loads are picked up, and 3) DGs energize node cells based on their power capacity. For example, the energization agent connected to Node 18 in Fig. 6 doesn't have enough capacity to energize Node cell 6 in Step 4. In Step 5, energization agents in yellow circles can pick up Node cell 2, 4 or 6, and node cell 4 is eventually selected due to the highest load pick-up and power balance constraints.

The voltage profile of the system after forming three MGs is shown in Fig. 8. Some nodal voltages are omitted in Fig. 8 because they are not energized. Phase voltages of all energized buses are within an acceptable range between 0.95 p.u. and 1.05 p.u. Active and reactive power losses for each formed MG are shown in Fig. 9.

2) Case2: In Case 2, the substation and the line between nodes 24 and 25 are both faulted, as shown in Fig. 10. Accordingly, Node cells 12 and 4 (Fig. 4) are isolated from the rest of the system by opening switches, their connection status with the upstream node cells are altered in the adjacent matrix, and they will not be picked up by an energization agent and remain isolated without power during the fault.

The system is partitioned into three MGs in Fig. 10, each is powered by one black-start DG. Table V shows the restored node numbers, and restored active and reactive power. 74.97% of active power and 80.43% of reactive power for the entire load are restored with an execution time of 0.0004345 seconds. Table VI shows output power of DGs. The system's voltage profile in Case 2 is shown in Fig. 11. Active and reactive power losses for each formed MG in Case 2 are shown in Fig. 12.

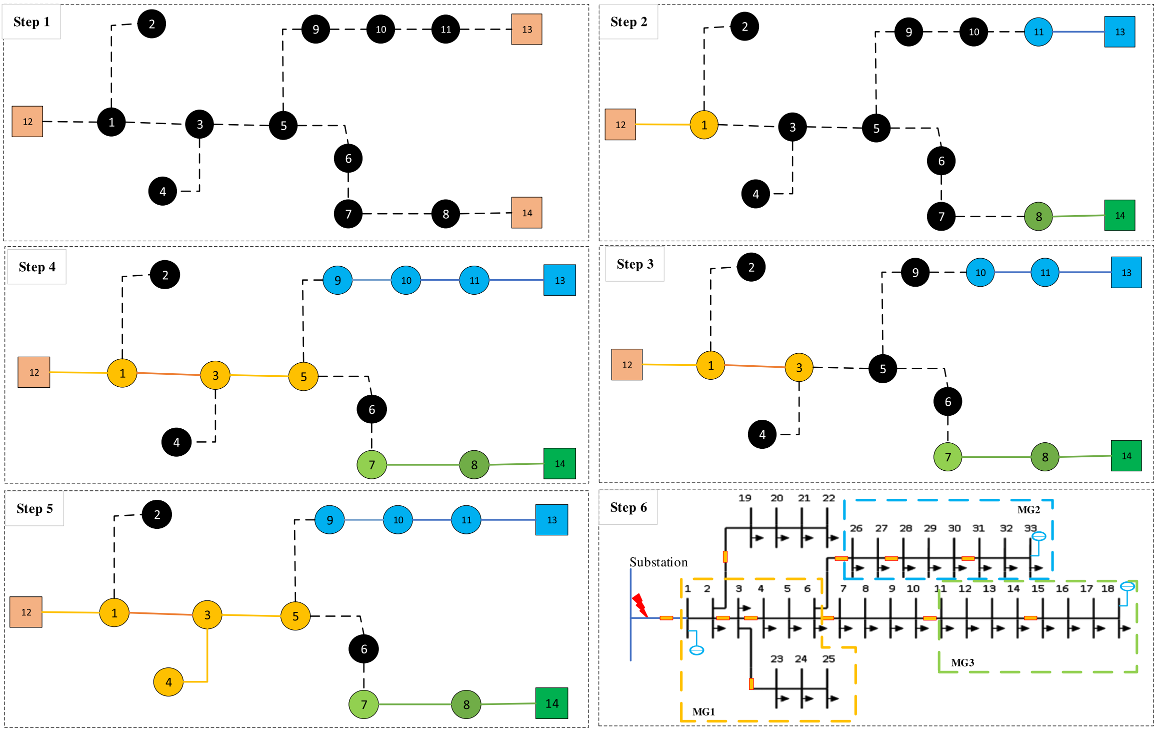

Fig. 7. Restoration sequences of modified IEEE 33-node system using the DQN in case1.

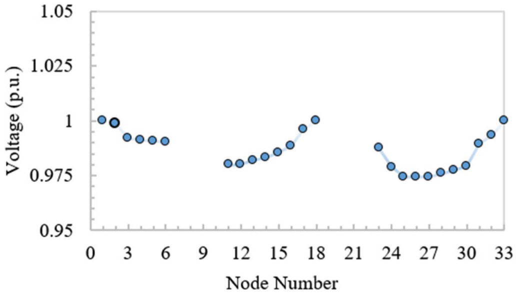

Fig. 8. Voltage profile of the modified IEEE 33-node test system after forming three MGs (case 1).

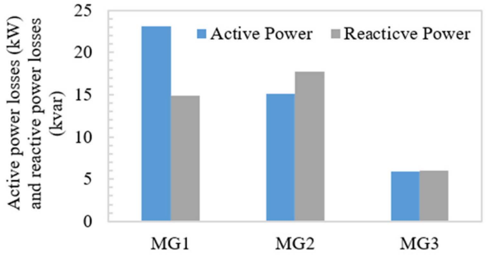

Fig. 9. Power losses of the modified IEEE 33-node test system in each MG (case 1).

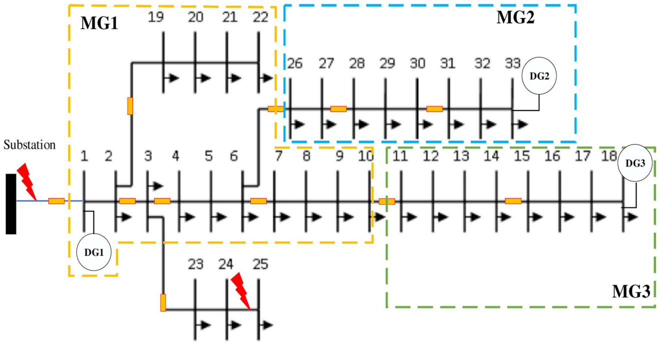

Fig. 10. Microgrid formation of the modified IEEE 33-node test system using DQN (case 2).

TABLE V
RESTORED NODES AND POWER OF THE MODIFIED IEEE 33-NODE TEST SYSTEM USING DQN (CASE 2)

Microgrid	Restored nodes	Restored loads (kW)	Restored loads (kVar)
MG1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 19, 20, 21, 22	1310	630
MG2	26, 27, 28, 29, 30, 31, 32, 33	920	950
MG3	11, 12, 13, 14, 15, 16, 17, 18	555	270
$\frac{Total\ resotoredload}{Total\ load} * 100$		74.97%	80.43%
Computational time (seconds)		0.0004345	

3) Case3: In Case 3, the substation and DG3 connected to node 18 have a fault, as shown in Fig. 13, and they are disconnected from the rest of the system by opening switches. With two available black-start DGs, DG1 and DG2, two MGs are formed, 61.37% of active power and 70.87% of reactive power for the entire load are restored with the computational

TABLE VI
DG OUTPUT POWER OF THE MODIFIED IEEE 33-NODE TEST SYSTEM (CASE 2)

Generation	Active power (kW)	Reactive power (kVar)
DG1	1325.07	640.73
DG2	935.09	967.74
DG3	560.91	276

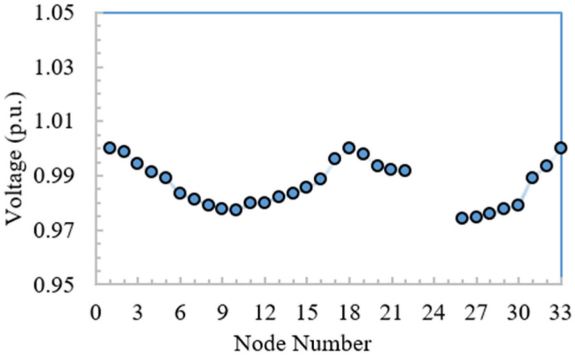

Fig. 11. Voltage profile of the modified IEEE 33-node test system after forming three MGs (case 2).

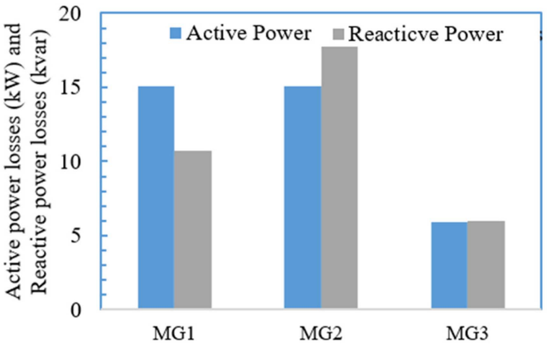

Fig. 12. Power losses of the modified IEEE 33-node test system in each MG (case 2).

TABLE VII
RESTORED NODES AND POWER OF THE MODIFIED IEEE 33-NODE TEST SYSTEM USING DQN (CASE 3)

Microgrid	Restored Nodes	Restored loads (kW)	Restored loads (kVar)
MG1	1, 2, 3, 4, 5, 6, 23, 24, 25	1360	680
MG2	26, 27, 28, 29, 30, 31, 32, 33	920	950
$\frac{Total\ resotoredload}{Total\ load}*100$		61.37 %	70.87%
computational time (seconds)		0.0004035	

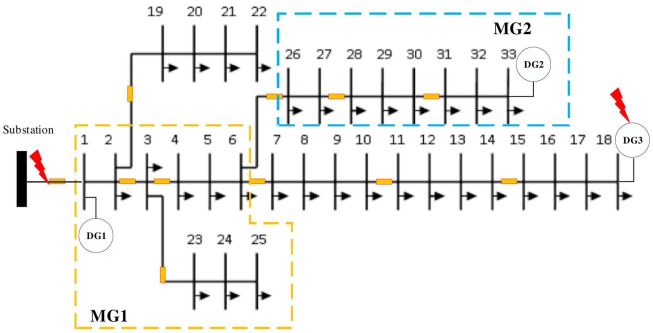

Fig. 13. Microgrid formation of the modified IEEE 33-node system using DQN (case 3).

TABLE VIII
DG OUTPUT POWER OF THE MODIFIED IEEE 33-NODE TEST SYSTEM (CASE 3)

Generation	Active power (kW)	Reactive power (kVar)
DG1	1383.08	694.83
DG2	935.09	967.74
DG3	-	-

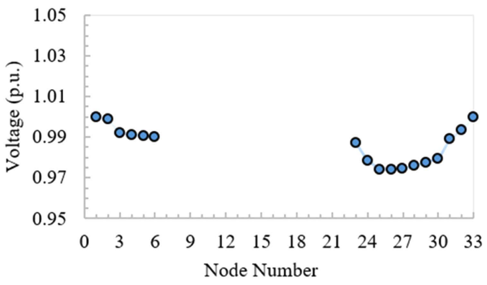

Fig. 14. Voltage profile of the modified IEEE 33-node test system after forming two MGs (case 3).

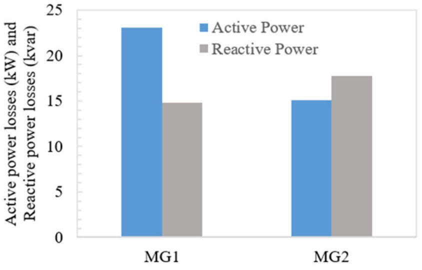

Fig. 15. Power losses of the modified IEEE 33-node test system in each MG (case 3).

time of 0.0004035 seconds (Table VII). Table VII also shows the restored node numbers, and restored active and reactive power. Table VIII shows output power of DGs in Case 3. Figs. 14 and 15 show the voltage profile of the system after service restoration and power losses in each MG, respectively.

4) Case4: In Case 4, four faults occur at the substation and the three lines between nodes 24 and 25, nodes 13 and 14, and nodes 28 and 29 (Fig. 16). Accordingly, node cells 12, 4, 7, and 10 (Fig. 4) are isolated from the rest of the system by opening switches, and they will not be selected by energization agents. With three black-start DGs, three MGs are formed to restore loads in Fig. 16. The restored node numbers and restored active and reactive power are shown in Table IX. 57.07% of active power and 42.61% of reactive power of the entire load are restored with the computational time of 0.0004714 seconds. Table X shows the output power of DGs. The system's voltage profile after restoration and power losses in each MG are shown in Figs. 17 and 18, respectively.

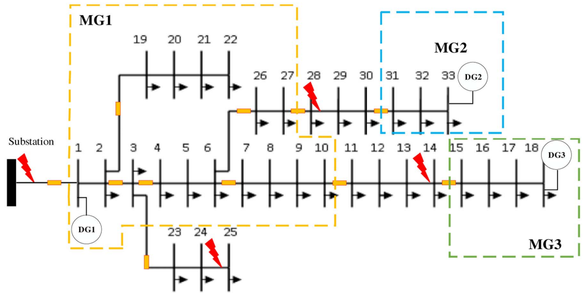

Fig. 16. Microgrid formation of the modified IEEE 33-node test system using DQN (case 4).

TABLE IX
RESTORED NODES AND POWER OF THE MODIFIED IEEE 33-NODE TEST SYSTEM USING DQN (CASE 4)

Microgrid	Restored Nodes	Restored loads (kW)	Restored loads (kVar)
MG1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 19, 20, 26, 27	1430	680
MG2	31, 32, 33	420	210
MG3	15, 16, 17, 18	270	90
$\frac{Total\ resotoredload}{Total\ load} * 100$		57.07%	42.61%
computational time (seconds)		0.0004714	

TABLE X
DG OUTPUT POWER OF THE MODIFIED IEEE 33-NODE TEST SYSTEM (CASE 4)

Generation	Active power (kW)	Reactive power (kVar)
DG1	1449.28	693.34
DG2	420.52	210.78
DG3	270.40	90.40

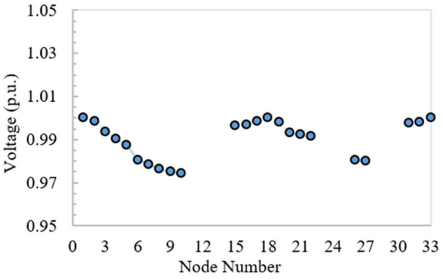

Fig. 17. Voltage profile of the modified IEEE 33-node test system after forming three MGs (case 4).

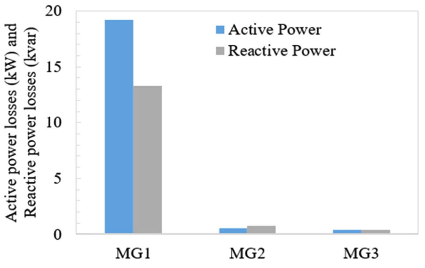

Fig. 18. Power losses of the modified IEEE 33-node test system in each MG (case 4).

TABLE XI
COMPARISON OF THE PROPOSED METHOD AND THREE EXISTING METHODS IN [11], [12], AND [16] FOR SERVICE RESTORATION USING THE MODIFIED IEEE 33-NODE TEST SYSTEM

Model	Restored load (kW)	Restoration ratio (%)	Computational time (s)
[12]	1500	40.37%	0.56
[11]	1820	49%	0.75
[16]	2575	69.31%	0.49
The proposed method	2575	69.31%	0.003912

TABLE XII
NUMBER OF VARIABLES AND CONSTRAINTS FOR THREE EXISTING METHODS [16]

	Model [12]	Model [11]	Model [16]
Number of binary Variables	$2N(N_r + N_g) + N + L$	$N + L$	$2N + 3L$
Number of continuous variable	$4N(N_r + N_g)$	$N + 3(N_r + N_g) + 3L$	$2N.L + N + 2(N_r + N_g) + L$
Number of constraints	$(11N - 2N_r - 2N_g + 1)(N_r + N_g) + 2N + N_o + N_c + L + L_o + L_c$	$3N + 3(N_r + N_g) + N_o + N_c + 5L + L_o + L_c + 1$	$N^2 + 2N.L + 3N + 2(N_r + N_g) + N_o + N_c + 3L + L_o + L_c + 1$

C. Performance Comparison With Existing Methods

The proposed restoration method is compared with three existing methods in [11], [12] and [16] for critical loads restoration through MG formation. Ref. [11] uses normal single-commodity flow-based radiality constraints, requires that each subgraph must be a connected graph, and the quantity of closed branches must match the number of subgraphs subtracted from the total number of nodes. Ref. [12] employs the node clustering method to form MGs, allocates each node to a DG, and ensure constraints for connectivity, branch-node, and clustering nodes are met. Ref. [16] uses the directed multicommodity flow-based model of the spanning tree constraints to form MGs. All three models in [11], [12] and [16] are formulated as a MILP problem and solved using commercial optimization solvers.

To do the comparison, a case study in [16] with the IEEE 33-node test system is used. The detailed system topology, including fault locations, and DGs and switches locations, can be found in [16]. Table XI shows a comparison of the restored loads and computational time of the proposed method and three existing methods. The proposed method can restore more loads than the methods in [11] and [12], and restore a similar amount of load to the method in [16]. However, from the computational time perspective, the proposed method outperforms all three existing methods. Table XII shows the number of variables and constraints in MG formation models in [11], [12] and [16], in which N, N_r, N_g, N_o, N_c, L, L_o, and L_c represent the total nodes, substation nodes, DG nodes, faulted open nodes, faulted closed nodes, branches, faulted open branches, and faulted close branches, respectively. For a large system, a large number of

TABLE XIII
DG Parameters in the Modified 404-Node System Model

Name	Node	P_g^{max}/ $P_g^{min}(MW)$	Q_g^{max}/ $Q_g^{min}(MVar)$	Status
DG1	1	1.05/0.1	0.8/-0.5	BSDG/DDG
DG2	313	0.9/0.09	0.7/-0.5	BSDG/DDG
DG3	82	0.9/0.09	0.7/-0.5	BSDG/DDG
DG4	147	1.05/0.1	0.8/-0.5	BSDG/DDG
DG5	230	1.2/0.12	0.9/-0.6	BSDG/DDG
DG6	101	1.2/0.12	0.9/-0.6	BSDG/DDG
DG7	183	1.5/0.15	1.2/-0.9	BSDG/DDG
DG8	391	0.5/0.05	0.4/-0.2	BSDG/DDG
DG9	337	0.5/0.05	0.4/-0.2	BSDG/DDG
DG10	279	1.5/0.15	1.2/-0.9	BSDG/DDG

variables/constraints are involved, which can greatly increase the computational time and make the practical use of methods in [11], [12] and [16] infeasible. The proposed method is model-free, can be trained offline and used online to make quick and efficient decisions.

VI. Validation Using a Real 404-Node Distribution System

The proposed method is further validated using a real 404-node distribution system. This system is one substation within a large distribution network operated by Saskatoon Light and Power in Saskatoon, Canada. The system model is developed using the practical data of lines and load of two feeders within the substation. The feeders are supplied by the 25/14.4 kV, 33.3 MVA substation. The load demand is 11.605 MW/7.192 MVar. The system model in this study is further modified with ten black-start DGs, and DG parameters are shown in Table XIII. The system is converted to a simplified system with only switchable lines through the node cell concept, as shown in Fig. 19. The detailed information about each node cell containing certain number of nodes in this system can be found in Appendix (Table XXV). The switch locations are assumed in the system model. To assess online performance of the proposed algorithm after training, three scenarios are conducted for this system using Python and OpenDSS.

A. Scenario 1

In Scenario 1, a fault occurs at the substation, as shown in Fig. 20. To isolate the affected region, the substation switches are activated and isolate the fault. Ten MGs have been formed (Fig. 20). Restoration agents energize node cells while ensuring operational and topological constraints are met. The voltage profile of the restored system in Scenario 1 is shown in Fig. 21, and all bus voltages are within acceptable limits. Power losses within each formed MG in Scenario 1 are shown in Fig. 22.

Table XIV shows the restored node cell numbers and restored loads. 80.66% of the total load is restored with a computational time of 0.013406 seconds. Due to security and power flow constraints, 19.34% of the load has been shed. The DG output power are shown in Table XV.

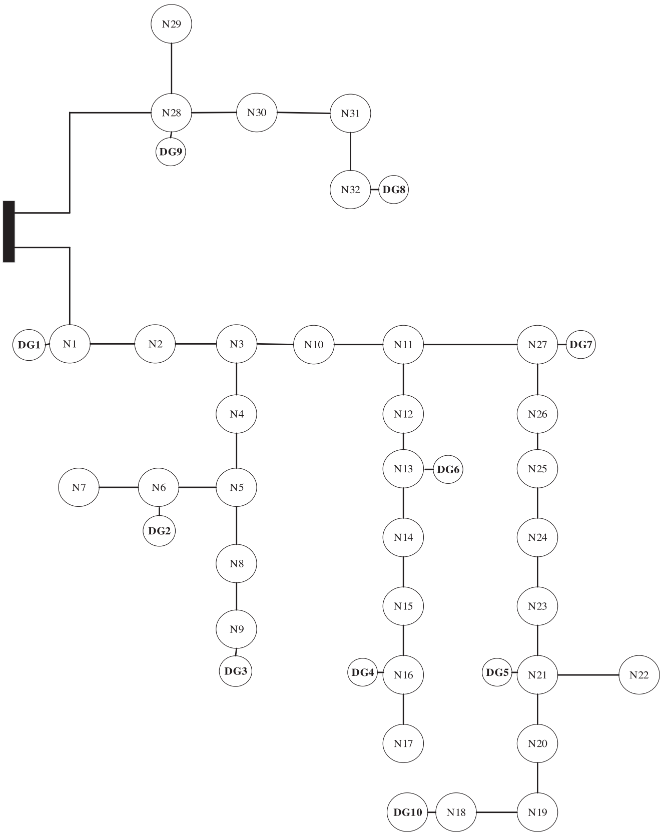

Fig. 19. Simplified system with 32 node cells through the node cell concept for the real 404-node distribution system.

TABLE XIV
Restored Node Cells and Power of the Modified 404-Node System Using DQN (Scenario 1)

Microgrid	Restored node cells	Restored loads (kW)	Restored loads (kVar)
MG1	N1, N2, N3, N4, N5, N10,	1041.2500	645.3088
MG2	N6, N7	795.8125	493.2003
MG3	N8, N9	825.5625	511.6377
MG4	N16, N17	937.1250	580.7780
MG5	N20, N21, N23	1041.2500	645.3089
MG6	N12, N13	1194.4750	740.2614
MG7	N24, N25, N26, N27	1404.9438	870.7086
MG8	N32	446.2500	276.5635
MG9	N28, N30, N31	476.0000	294.9983
MG10	N18	1197.4375	742.1051
$\frac{Total\ resotoredload}{Total\ load}*100$		80.66%	80.66%
Computational time (seconds)		0.013406	

B. Scenario 2

In Scenario 2, 10 faults are applied to the substation and several places in the system, as shown in Fig. 23, and ten MGs are formed. Each MG is energized by a black-start DG. The voltage profile following the service restoration in Scenario 2 is shown in Fig. 24. Power losses of each formed MG in Scenario 2 are shown in Fig. 25. In an effort to restore load, restoration agents work within bounds of topological and operational constraints

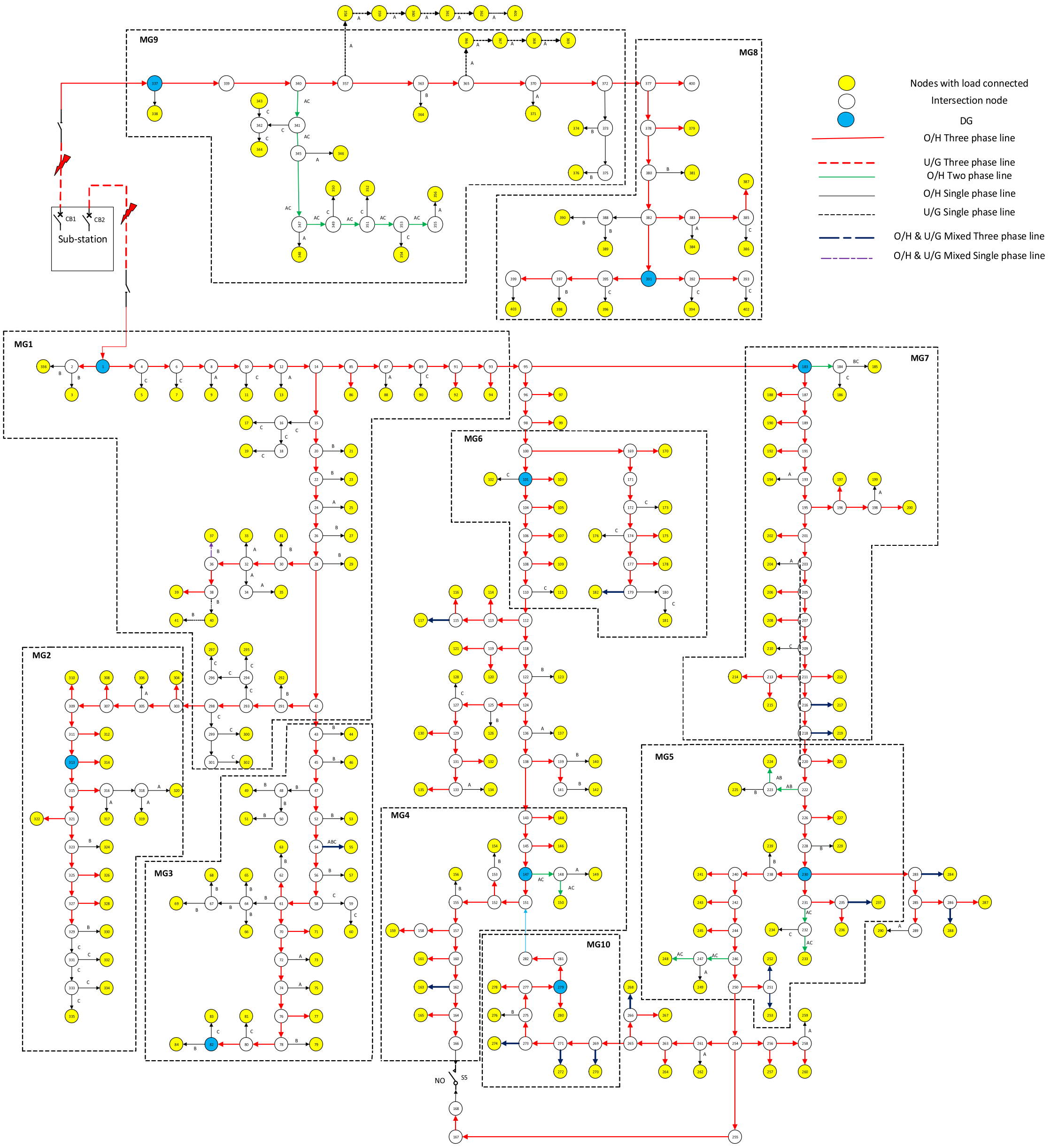

Fig. 20. Microgrid formation of the 404-node system using DQN (scenario 1).

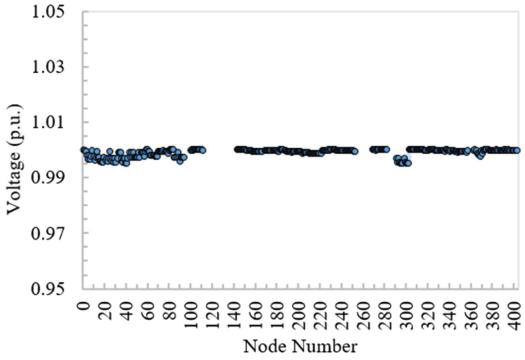

Fig. 21. Voltage profile of the 404-node system after restoration (scenario 1).

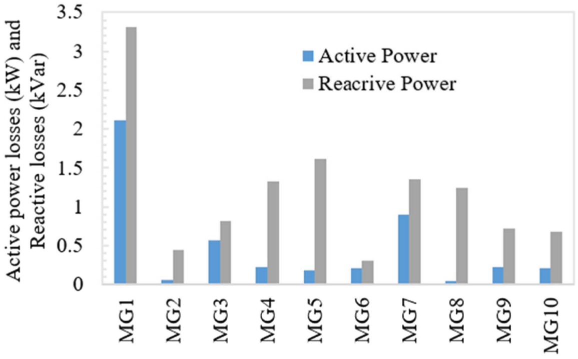

Fig. 22. Power losses in each MG of the 404-node system (scenario 1).

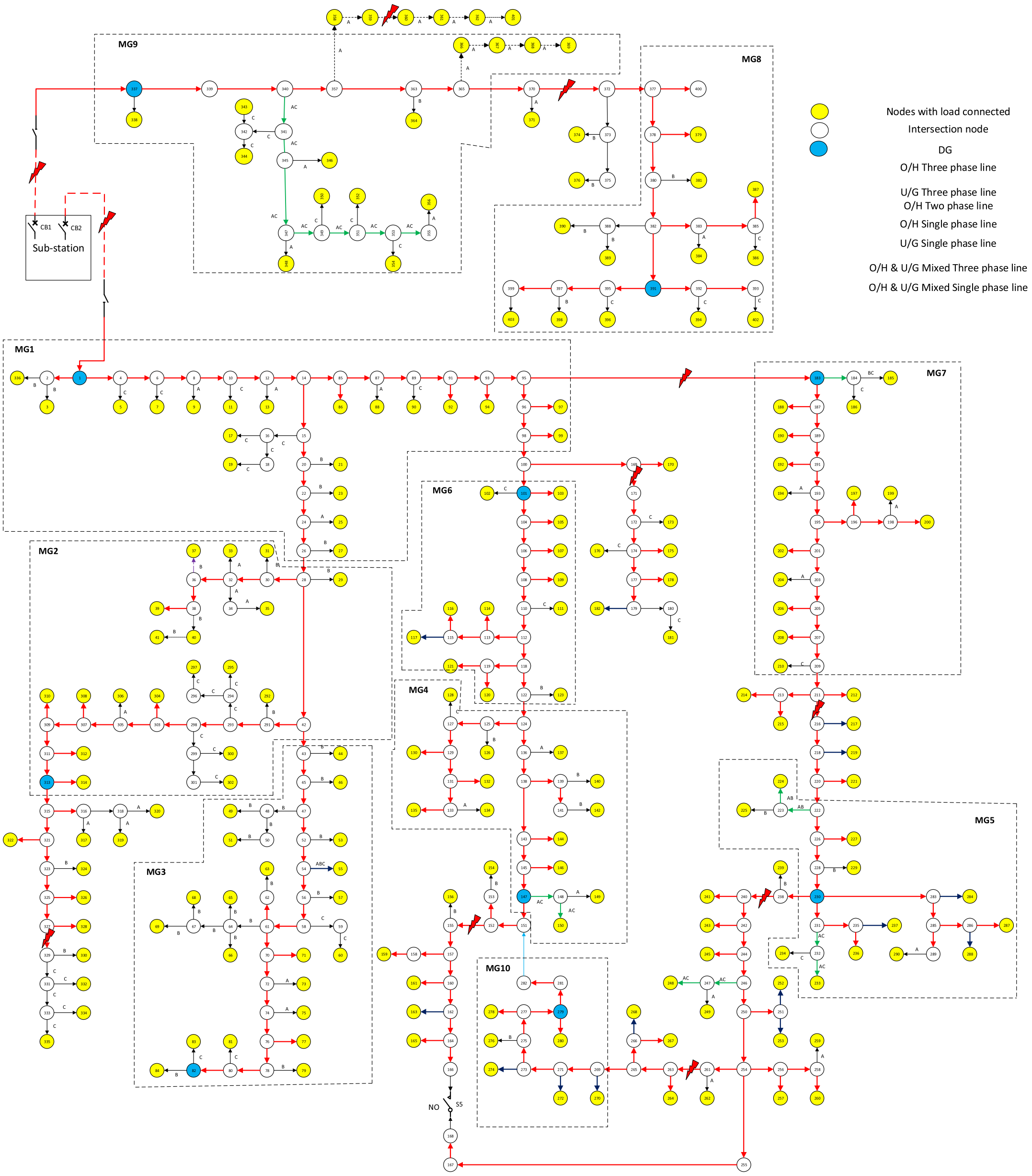

Fig. 23. Microgrid formation of the 404-node system using DQN (scenario 2).

TABLE XV
DG Output Power for the Modified 404-Node System (Scenario 1)

Generation	Active power (kW)	Reactive power (kVar)
DG1	1043.3627	648.6241
DG2	795.8640	493.6421
DG3	826.1352	512.4591
DG4	937.3487	582.1107
DG5	1041.4337	646.9206
DG6	1194.6908	740.5647
DG7	1405.8361	872.0588
DG8	446.2943	277.8089
DG9	476.2291	250.7190
DG10	1197.6440	742.7900

to select node cells. Table XVI shows the restored node cell numbers and restored loads. 63.72% of the total load restored with a computational time of 0.013716 seconds. The DG output power are shown in Table XVII.

C. Scenario 3

In Scenario 3, a new fault develops at DG3 at Scenario 2, i.e., DG3 has a permanent fault in addition to all faults of Scenario 2. As shown in Fig. 26, nine MGs are formed using nine black-start DGs in the system. Following DG3's fault, operational boundaries of MG1 and MG2 must be adjusted from Scenario 2 to ensure the maximum load pickup. The voltage profile after restoration in Scenario 3 is shown in Fig. 27. Fig. 28 shows power losses in each MG in Scenario 3.

Table XVIII shows the restored node cells and restored loads. 58.72% of the total load restored with an execution

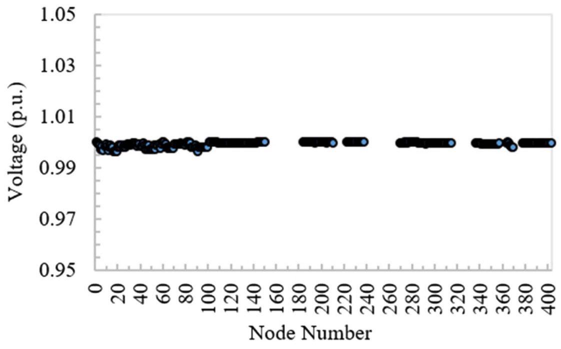

Fig. 24. Voltage profile of the 404-node system after restoration (scenario 2).

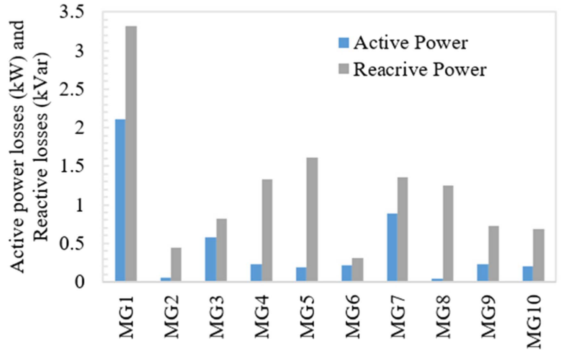

Fig. 25. Power losses in each MG for the 404-node system (scenario 2).

TABLE XVI
RESTORED NODE CELLS AND POWER OF THE MODIFIED 404-NODE SYSTEM USING DQN (SCENARIO 2)

Microgrid	Restored node cells	Restored loads (kW)	Restored loads (kVar)
MG1	N1, N2, N3, N10, N11	699.1250	433.2788
MG2	N4, N5, N6	810.6875	502.4190
MG3	N8, N9	825.5625	511.6377
MG4	N15, N16	699.1250	433.2788
MG5	N21, N22, N23	899.9375	557.7318
MG6	N13, N14	820.8150	508.6877
MG7	N25, N26, N27	594.2563	368.2896
MG8	N32	446.2500	276.5635
MG9	N28, N30	401.6250	248.9048
MG10	N18	1197.4375	742.1051
$\frac{Total\ resotoredload}{Total\ load}*100$		63.72%	6373%
Computational time (seconds)		0.013716	

TABLE XVII
DG OUTPUT POWER FOR THE MODIFIED 404-NODE SYSTEM (SCENARIO 2)

Generation	Active power (kW)	Reactive power (kVar)
DG1	700.1227	438.2869
DG2	810.8950	503.0603
DG3	826.1352	512.4591
DG4	699.1989	433.8761
DG5	900.0551	560.7991
DG6	820.9023	509.2373
DG7	594.3471	368.8613
DG8	446.2943	277.8089
DG9	401.8250	249.5012
DG10	1197.6440	742.78998

TABLE XVIII
RESTORED NODE CELLS AND POWER OF THE 404-NODE SYSTEM USING DQN (SCENARIO 3)

Microgrid	Restored node cells	Restored loads (kW)	Restored loads (kVar)
MG1	N1, N2, N3, N4, N10, N11	1026.3750	636.0901
MG2	N5, N6, N8	728.8750	451.7161
MG3	N15, N16	699.1250	433.2788
MG4	N21, N22, N23	899.9375	557.7318
MG5	N13, N14	820.8150	508.6877
MG6	N25, N26, N27	594.2563	368.2896
MG7	N32	446.2500	276.5635
MG8	N28, N30	401.6250	248.9048
MG9	N18	1197.4375	742.1051
$\frac{Total\ resotoredload}{Total\ load}*100$		58.72%	58.72%
Computational time (seconds)		0.013825	

TABLE XIX
DG OUTPUT POWER FOR THE 404-NODE SYSTEM (SCENARIO 3)

Generation	Active power (kW)	Reactive power (kVar)
DG1	1028.2354	442.0463
DG2	729.0638	452.6724
DG3	-	-
DG4	699.1989	433.8761
DG5	900.0551	560.7991
DG6	820.9023	509.2373
DG7	594.3471	368.8613
DG8	446.2943	277.8089
DG9	401.8250	249.5012
DG10	1197.6440	742.7900

time of 0.013825 seconds. The DG output power after service restoration is shown in Table XIX. The proposed method has shown promising results in this real 404-node distribution system through the three scenarios. When a new condition arises, boundaries of MGs can be adjusted accordingly.

VII. OPTIMAL SWITCH PLACEMENT AND ITS EFFECT ON SERVICE RESTORATION

In this section, an effective optimal switch placement method is proposed using the binary particle swarm optimization (BPSO) to minimize unsupplied loads and the total number of switches as the objective function (OF) through a multi-objective optimization technique. We also demonstrate that optimally placed switches offer a significant improved service restoration when compared to randomly placed switches in the IEEE 33-node test system and the real 404-node distribution system in Sections V and VI.

Furthermore, since we have actual switches information for the 404-node distribution system from Saskatoon Light and Power, actually installed switches, randomly placed switches, and optimally placed switches for this system are compared from the service restoration perspective using the proposed restoration method in this article. It is found that optimally placed switches can significantly improve service restoration compared to actual installed switches and randomly placed switches in this system.

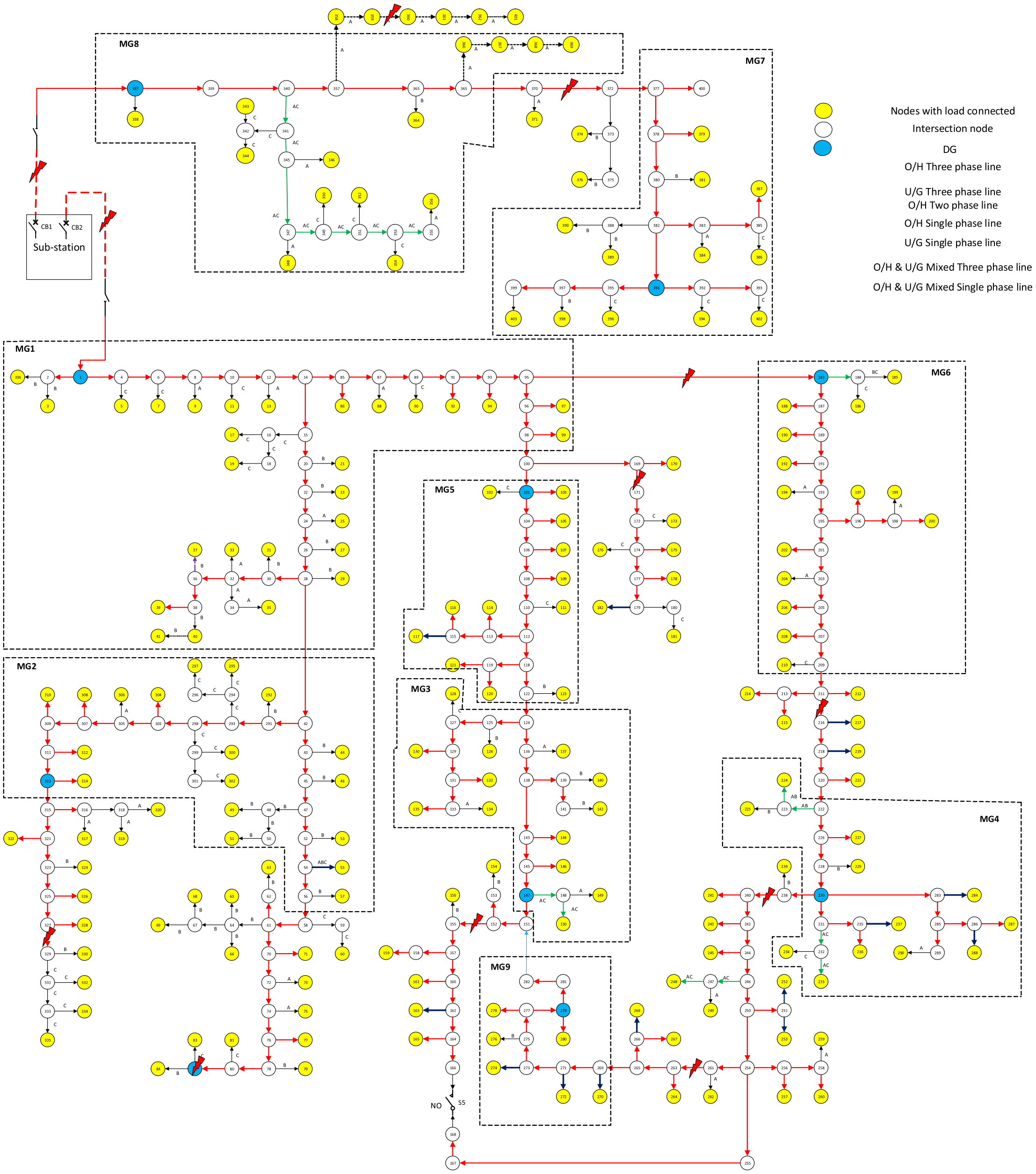

Fig. 26. Microgrid formation of the 404-node system using DQN (scenario 3).

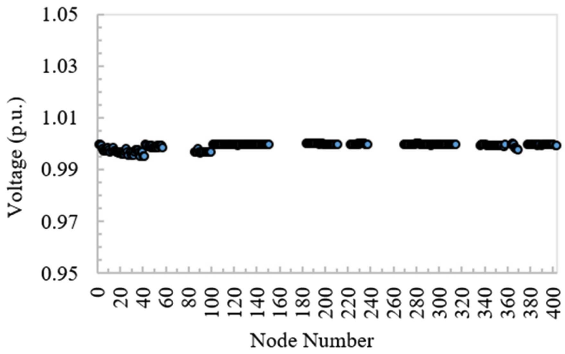

Fig. 27. Voltage profile of the 404-node system after restoration (scenario 3).

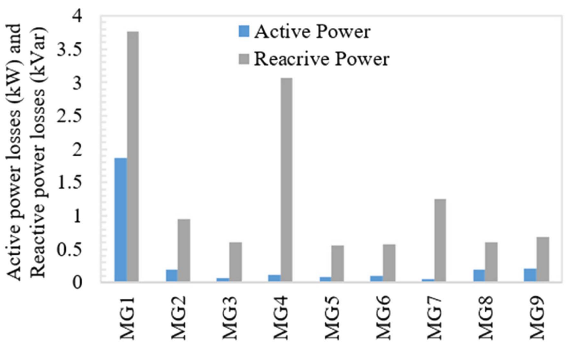

Fig. 28. Power losses in each MG for the 404-node system (scenario 3).

TABLE XX
OPTIMAL SWITCH PLACEMENT RESULTS WITH DIFFERENT OBJECTIVE WEIGHTS FOR IEEE 33-NODE TEST SYSTEM

Solution (#)	Priority Weights of Objectives	Number of Switches Placed	Best OF value	Worst OF Value	Mean	STD	Switch Locations
1	W_u = 0.5 W_r = 0.5	15	0.7182	0.7869	0.7467	0.0169	(Substation, 1), (3, 4), (6, 7), (8, 9), (10, 11), (14, 15), (2, 19), (20, 21), (3, 23), (23, 24), (24, 25), (6, 26), (27, 28), (29, 30), (31, 32)
2	W_u = 0.6 W_r = 0.4	19	0.6243	0.6754	0.6507	0.0193	(Substation, 1), (2, 3), (3, 4), (5, 6), (6, 7), (7, 8), (8, 9), (11, 12), (14, 15), (16, 17), (19, 20), (3, 23), (23, 24), (24, 25), (6, 26), (27, 28), (29, 30), (30, 31), (31, 32)
3	W_u = 0.4 W_r = 0.6	7	0.7901	0.8484	0.8199	0.0209	(Substation, 1) (12, 13), (13, 14), (14, 15), (3, 23), (23, 24), (24, 25)

A. The Proposed Switch Placement Algorithm

In the switch placement algorithm proposed in this article, the objective function (OF) includes two member functions, F_{UL} and F_{NSW}. The function F_{UL} calculates the number of unsupplied loads due to power outages, and the function F_{NSW} determines the number of switches included in the solution *x*. When a fault occurs in the network, related switches are opened to isolate the fault while ensuring that the maximum possible load is supplied. The objective function is expressed as follows:

$$OF\ (x) = w_u F_{UL}\,(x) + w_r F_{NSW}\,(x) \tag{35}$$

where w_u and w_r are weight coefficients used to minimize unsupplied loads and the number of switches, respectively. Their values determine the relative importance of different objective terms, and can be selected based on priorities of stakeholders. These weight coefficients must follow constrains in (36) and (37).

$$w_u + w_r = 1 \tag{36}$$

$$w_u, w_r \geq 0 \tag{37}$$

The F_{UL} function can be expressed as follows:

$$F_{UL} = \frac{\sum_{j=1}^{B} d_j}{TD} \tag{38}$$

where B is the total number of branches in the distribution network. d_j is the amount of load demand that is switched off due to a fault on the *jth* branch. *TD* is the total amount of load demand that are normally supplied by a distribution network.

The F_{NSW} function can be expressed as follows:

$$F_{NSW} = \frac{N_x}{TS} \tag{39}$$

where F_{NSW} is calculated considering the number of switches in the current iteration solution denoted by N_x, and the total number of switches denoted by *TS*.

B. Optimal Switch Placement in IEEE 33-Node Test System

The BPSO [32] is applied to find an optimal configuration of switches for the IEEE 33-node test system, and the results are shown in Table XX. In this table, three solutions are compared based on the priority weights of objectives, number of switches, best and worst OF values, mean and standard deviation (STD) of OF values, and switch locations. For Solution 1, the weight coefficients w_u and w_r are both set to 0.5, leading to a total of 15 switches optimally placed.

TABLE XXI
SERVICE RESTORATION USING THE PROPOSED RESTORATION METHOD WITH OPTIMALLY AND RANDOMLY PLACED SWITCHES FOR IEEE 33-NODE TEST SYSTEM

Methods	Total restored active load (%)	Total restored reactive load (%)	Restoration improvement over randomly placed switches (active load/ reactive load), %
Randomly placed switches	57.07	42.61	-
Optimally placed Switches	63.26	70.43	**10.84% / 65.29%**

To demonstrate the effect of weight coefficients, in addition to equal weights, two other sets of w_u and w_r values are considered for Solutions 2 and 3 in Table XX. Solution 2 puts more weight on reducing unsupplied loads, resulting in a lower OF value of 0.62431 and a total of 19 switches placed. Solution 3 puts more weight on minimizing the number of switches placed, resulting in a higher OF value of 0.7901 and a total of 7 switches placed. Therefore, different weight coefficients can be selected to achieve the desired balance between unsupplied loads and the number of switches placed.

The OF values are also analyzed according to their best and worst values, and their mean and standard deviation (STD) values for all solutions. For example, in Solution 1, the best OF value is the lowest OF value of 0.7182, and the worst OF value is the highest OF value of 0.7869. The mean of the OF values is 0.7467, which is the average OF value of all solutions found by the algorithm; and its STD OF is 0.0169, which represents the spread of OF values around the mean. A low STD indicates that solutions found by the algorithm are consistently good.

Using the proposed restoration method in the IEEE 33-node test system, optimally placed switches lead to significantly improved service restoration compared to randomly placed switches with arbitrary numbers and locations, as shown in Table XXI. In this table, compared to Case 4 in Section V using randomly placed switches, the improvement of 10.84% of active power and 65.29% of reactive power restoration can be achieved with optimally placed switches in the same system.

C. Optimal Switch Placement in the 404-Node Distribution System

The proposed optimal switch placement method is applied to the real 404-node distribution system operated by Saskatoon

TABLE XXII
COMPARATIVE ANALYSIS OF SWITCHING SOLUTIONS WITH DIFFERENT OBJECTIVE WEIGHTS IN THE REAL 404-NODE DISTRIBUTION SYSTEM

Solution (#)	Priority Weights of Objectives	Number of Switches placed	Best OF value	Worst OF value	Mean	STD	Switch Location
1	$W_u = 0.5$ $W_r = 0.5$	40	0.60771	0.61545	0.6128	0.0027	(Substation, 1), (4, 6), (14, 15), (28, 30), (293, 298), (298, 299), (307, 309), (311, 313), (316, 318), (325, 327), (327, 329), (42, 43), (56, 58), (61, 64), (74, 76), (95, 96), (100, 169), (172, 174), (101, 104), (108, 110), (112, 113), (125, 127), (122, 124), (147, 151), (152, 155), (95, 183), (207, 209), (218, 220), (226, 228), (230, 238), (240, 242), (246, 250), (263, 265), (265, 269), (279, 151), (Substation, 337), (357, 358), (365, 366), (365, 370), (377, 378)
2	$W_u = 0.6$ $W_r = 0.4$	56	0.5957	0.61005	0.60468	0.005	(Substation, 1), (6, 8), (12, 14), (15, 20), (22, 24), (28, 30), (36, 38), (293, 294), (299, 301), (305, 307), (311, 313), (316, 318), (323, 325), (325, 327), (42, 43), (47, 48), (52, 54), (58, 61), (61, 64), (74, 76), (14, 85), (85, 87), (87, 89), (95, 96), (100, 169), (101, 104), (112, 113), (118, 122), (122, 124), (124, 136), (139, 141), (145, 147), (155, 157), (160, 162), (162, 164), ('95', '183'), ('189', '191'), ('195', '196'), (205, 207), (220, 222), (222, 226), (228, 230), (230, 231), (244, 246), (261, 263), (265, 269), (269, 271), (271, 273), (279, 151), (Substation, 337), (345, 347), (351, 353), (353, 355), (365, 370), (377, 378), (382, 391)
3	$W_u = 0.4$ $W_r = 0.6$	27	0.84098	0.85605	0.84830	0.0057	(Substation, 1), (28, 30), (298, 303), (313, 315), (321, 323), (329, 331), (47, 48), (72, 74), (78, 80), (89, 91), (98, 100), (110, 112), (118, 122), (129, 131), (124, 136), (138, 139), (191, 193), (193, 195), ('196', 198), (218, 220), (230, 283), (250, 254), (279, 151), (0, 337), (347, 349), (372, 373), (395, 397)

TABLE XXIII
COMPARISON OF MG-BASED SERVICE RESTORATION USING DRL IN THE REAL 404-NODE DISTRIBUTION SYSTEM WITH OPTIMAL, RANDOM, AND ACTUAL SWITCH PLACEMENT

Method	Total restored active load (%)	Total restored reactive load (%)	Restoration improvement over actual switches (active load/ reactive load), %	Restoration improvement over randomly placed switches (active load/reactive load),%
Actual switches	76.03	76.23		
Randomly placed switches	80.66	80.66		
Optimally placed Switches	83.72	83.46	**10.11%/9.48%**	**3.79% / 3.47%**

Light and Power. The results are shown in Table XXII. Similarly, three sets of weight coefficients w_u and w_r values are considered. For Solution 1, a total of 40 switches are placed with the equal weight of 0.5 for w_u and w_r. Solution 2 prioritizes minimizing the unsupplied loads, leading to a lower OF value of 0.5957 and 56 switches. Solution 3 prioritizes reducing the number of switches, leading to a higher OF value of 0.84098 and 27 switches.

Because this is a real system and we have the number, types and locations of actually installed switches, so we are able to conduct service restoration using the proposed restoration method for faults in Scenario 1 (Section VI-A) by using: 1) actually installed switches in the real system, 2) randomly placed switches as assumed in Scenario 1 in Section VI, and 3) optimally placed switches in this section. The comparison results are shown in Table XXIII. Using optimally placed switches, the improvement of 3.79% in active power and 3.47% in reactive power restoration can be achieved over randomly placed switches; while the improvement of 10.11% in active power and 9.48% in reactive power restoration can be achieved over actual installed switches in this system. Therefore, optimal switch placement in distribution networks can result in significantly improved service restoration. It should be noted that during microgrid formation, load shedding may be needed to establish a self-adequate microgrid to balance the power generation and load demand.

VIII. CONCLUSION

In this article, a novel dynamic microgrid formation-based service restoration method using deep reinforcement learning for distribution networks is proposed. Through the deep Q-learning technique and a simulator-based environment, the deep reinforcement learning algorithm can effectively learn the optimal control policy and form dynamic microgrids during contingencies for rapid service restoration. It can be trained offline first, and then used in online applications. The proposed method is validated using the modified IEEE 33-node test system and a real large 404-node distribution system from Saskatoon Light and Power in Saskatoon, Canada. Case studies using a real large distribution system with a very short computational time prove great potential of the proposed method in online applications for large scale distribution networks. Further development of this method can lead to a powerful tool for electric utilities to realize self-healing and service restoration. Finally, effective optimal switch placement is proposed using BPSO through a multi-objective optimization technique, the effect of optimal switch placement on service restoration is evaluated. It is found that optimal switch placement can lead to significant improvement of service restoration in distribution networks.

APPENDIX

As the first step for a given distribution system, the node cell concept can be used to convert the original system into a simplified system with only node cells and switchable lines. The following two tables show each node cell corresponds to the node numbers of the original system within it for the modified IEEE 33-node test system in Section V, and the large 404-node system in Section VI.

TABLE XXIV
NODE CELL MAPPING FOR THE MODIFIED IEEE 33-NODE TEST SYSTEM

Node Cell	Corresponding Nodes
Node cell 1	1, 2
Node cell 2	19, 20, 21, 22
Node cell 3	3
Node cell 4	23, 24, 25
Node cell 5	4, 5, 6
Node cell 6	7, 8, 9, 10
Node cell 7	11, 12, 13, 14
Node cell 8	15, 16, 17, 18
Node cell 9	26, 27
Node cell 10	28, 29, 30
Node cell 11	31, 32, 33

TABLE XXV
NODE CELL MAPPING FOR THE 404-NODE DISTRIBUTION SYSTEM

Node Cell	Corresponding Nodes
Node cell 1	1, 2, 3, 336
Node cell 2	4, 5, 6, 7, 8, 9, 10, 11, 12, 13
Node cell 3	14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
Node cell 4	28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41
Node cell 5	291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302
Node cell 6	303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314
Node cell 7	315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335
Node cell 8	42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57
Node cell 9	58, 59. 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84
Node cell 10	85, 86, 87, 88, 89, 90, 91, 92, 93, 94
Node cell 11	95, 96, 97, 98, 99
Node cell 12	169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182
Node cell 13	101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111
Node cell 14	112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123
Node cell 15	124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142
Node cell 16	143, 144, 145, 146, 147, 148, 149, 150
Node cell 17	151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166
Node cell 18	269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282
Node cell 19	254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268
Node cell 20	238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253
Node cell 21	230, 231, 232, 233, 234, 235, 236, 237
Node cell 22	283, 284, 285, 286, 287, 288, 289, 290
Node cell 23	222, 223, 224, 225, 226, 227, 228, 229
Node cell 24	211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221
Node cell 25	201, 202, 203, 204, 205, 206, 207, 208, 209, 210
Node cell 26	183, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200
Node cell 27	184, 185, 186
Node cell 28	337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356
Node cell 29	358, 359, 360, 361, 362, 401
Node cell 30	357, 363, 364, 365, 366, 367, 368, 369
Node cell 31	370, 371, 372, 373, 374, 375, 376
Node cell 32	377, 400, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 402, 403

REFERENCES

[1] M. A. Igder and X. Liang, "Dynamic microgrid formation-based service restoration using deep reinforcement learning in distribution networks," in *Proc. IEEE 59th Ind. Commercial Power Syst. Tech. Conf.*, 2023, pp. 1–8.

[2] B. Chen, Z. Ye, C. Chen, and J. Wang, "Toward a MILP modeling framework for distribution system restoration," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 1749–1760, May 2018.

[3] Y.-L. Ke, "Distribution feeder reconfiguration for load balancing and service restoration by using G-nets inference mechanism," *IEEE Trans. Power Del.*, vol. 19, no. 3, pp. 1426–1433, Jul. 2004.

[4] X. Liang, M. A. Saaklayen, M. A. Igder, S. M. R. H. Shawon, S. O. Faried, and M. Janbakhsh, "Planning and service restoration through microgrid formation and soft open points for distribution network modernization: A review," *IEEE Trans. Ind. Appl.*, vol. 58, no. 2, pp. 1843–1857, Mar./Apr. 2022.

[5] M. A. Igder, X. Liang, and M. Mitolo, "Service restoration through microgrid formation in distribution networks: A review," *IEEE Access*, vol. 10, pp. 46618–46632, 2022.

[6] J. Zhao, F. Li, S. Mukherjee, and C. Sticht, "Deep reinforcement learning based model-free on-line dynamic multi-microgrid formation to enhance resilience," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 2557–2567, Jul. 2022.

[7] Y. Huang, G. Li, C. Chen, Y. Bian, T. Qian, and Z. Bie, "Resilient distribution networks by microgrid formation using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 13, no. 6, pp. 4918–4930, Nov. 2022.

[8] F. S. Gazijahani, J. Salehi, M. Shafie-Khah, and J. P. S. Catalão, "Spatiotemporal splitting of distribution networks into self-healing resilient microgrids using an adjustable interval optimization," *IEEE Trans. Ind. Inform.*, vol. 17, no. 8, pp. 5218–5229, Aug. 2021.

[9] X. Liu, M. Shahidehpour, Z. Li, X. Liu, Y. Cao, and Z. Bie, "Microgrids for enhancing the power grid resilience in extreme conditions," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 589–597, Mar. 2017.

[10] C. Abbey et al., "Powering through the storm: Microgrids operation for more efficient disaster recovery," *IEEE Power Energy Mag.*, vol. 12, no. 3, pp. 67–76, May/Jun. 2014.

[11] T. Ding, Y. Lin, G. Li, and Z. Bie, "A new model for resilient distribution systems by microgrids formation," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 4145–4147, Sep. 2017.

[12] C. Chen, J. Wang, F. Qiu, and D. Zhao, "Resilient distribution system by microgrids formation after natural disasters," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 958–966, Mar. 2016.

[13] Y. Vilaisarn, Y. R. Rodrigues, M. M. A. Abdelaziz, and J. Cros, "A deep learning based multiobjective optimization for the planning of resilience oriented microgrids in active distribution system," *IEEE Access*, vol. 10, pp. 84330–84364, 2022.

[14] S. Cai, Y. Xie, Q. Wu, M. Zhang, X. Jin, and Z. Xiang, "Distributionally robust microgrid formation approach for service restoration under random contingency," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 4926–4937, Nov. 2021.

[15] M. N. Ambia, K. Meng, W. Xiao, and Z. Y. Dong, "Nested formation approach for networked microgrid self-healing in islanded mode," *IEEE Trans. Power Del.*, vol. 36, no. 1, pp. 452–464, Feb. 2021.

[16] S. Lei, C. Chen, Y. Song, and Y. Hou, "Radiality constraints for resilient reconfiguration of distribution systems: Formulation and application to microgrid formation," *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 3944–3956, Sep. 2020.

[17] M. Salimi, M.-A. Nasr, S. H. Hosseinian, G. B. Gharehpetian, and M. Shahidehpour, "Information gap decision theory-based active distribution system planning for resilience enhancement," *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 4390–4402, Sep. 2020.

[18] Z. Wang and J. Wang, "Self-healing resilient distribution systems based on sectionalization into microgrids," *IEEE Trans. Power Syst.*, vol. 30, no. 6, pp. 3139–3149, Nov. 2015.

[19] K. S. A. Sedzro, X. Shi, A. J. Lamadrid, and L. F. Zuluaga, "A heuristic approach to the post-disturbance and stochastic pre-disturbance microgrid formation problem," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5574–5586, Sep. 2019.

[20] S. A. Arefifar, Y. A.-R. I. Mohamed, and T. H. M. El-Fouly, "Supply-adequacy-based optimal construction of microgrids in smart distribution systems," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1491–1502, Sep. 2012.

[21] Y. Du, H. Tu, X. Lu, J. Wang, and S. Lukic, "Black-start and service restoration in resilient distribution systems with dynamic microgrids," *IEEE J. Emerg. Sel. Topics Power Electron.*, vol. 10, no. 4, pp. 3975–3986, Aug. 2022.

[22] T. Zhao and J. Wang, "Learning sequential distribution system restoration via graph-reinforcement learning," *IEEE Trans. Power Syst.*, vol. 37, no. 2, pp. 1601–1611, Mar. 2022.

[23] Q. Huang, R. Huang, W. Hao, J. Tan, R. Fan, and Z. Huang, "Adaptive power system emergency control using deep reinforcement learning," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1171–1182, Mar. 2020.

[24] T. Qian, C. Shao, X. Wang, and M. Shahidehpour, "Deep reinforcement learning for EV charging navigation by coordinating smart grid and intelligent transportation system," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1714–1723, Mar. 2020.

[25] B. Wang, Y. Li, W. Ming, and S. Wang, "Deep reinforcement learning method for demand response management of interruptible load," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3146–3155, Jul. 2020.

[26] Q. Zhang, K. Dehghanpour, Z. Wang, and Q. Huang, "A learning-based power management method for networked microgrids under incomplete information," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1193–1204, Mar. 2020.

[27] M. M. Hosseini and M. Parvania, "Resilient operation of distribution grids using deep reinforcement learning," *IEEE Trans. Ind. Inform.*, vol. 18, no. 3, pp. 2100–2109, Mar. 2022.

[28] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[29] Z. Zhang, D. Zhang, and R. C. Qiu, "Deep reinforcement learning for power system applications: An overview," *Chin. Soc. Elect. Eng. J. Power Energy Syst.*, vol. 6, no. 1, pp. 213–225, 2019.

[30] S. H. Oh, Y. T. Yoon, and S. W. Kim, "Online reconfiguration scheme of self-sufficient distribution network based on a reinforcement learning approach," *Appl. Energy*, vol. 280, 2020, Art. no. 115900.

[31] X. Wang et al., "Deep reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 28, 2022, doi: 10.1109/TNNLS.2022.3207346.

[32] J. R. Bezerra, G. C. Barroso, and R. P. Leao, "Switch placement algorithm for reducing customers outage impacts on radial distribution networks," in *Proc. IEEE TENCON Region 10 Conf.*, 2012, pp. 1–6.

Mosayeb Afshari Igder (Graduate Student Member, IEEE) received the M.S. degree in electrical engineering from the Shiraz University of Technology, Shiraz, Iran, in 2016. He is currently working toward the second Master of Science degree with the University of Saskatchewan, Saskatoon, SK, Canada. From 2016 to 2020, he had a research collaboration with the Department of Electrical and Electronics Engineering, Shiraz University of Technology. His research interests include power system operation, renewable energy, and marine power systems.

Xiaodong Liang (Senior Member, IEEE) was born in Lingyuan, Liaoning, China. She received the B.Eng. and M.Eng. degrees in electrical engineering from Shenyang Polytechnic University, Shenyang, China, in 1992 and 1995, respectively, the M.Sc. degree in electrical engineering from the University of Saskatchewan, Saskatoon, SK, Canada, in 2004, and the Ph.D. degree in electrical engineering from the University of Alberta, Edmonton, AB, Canada, in 2013. From 1995 to 1999, she was a Lecturer at Northeastern University, Shenyang. In 2001, she joined Schlumberger in Edmonton, Canada, and was promoted to a Principal Power Systems Engineer with this world's leading oil field service company in 2009. She was with Schlumberger for almost 12 years till 2013. From 2013 to 2019, she was with Washington State University, Vancouver, WA, USA, and Memorial University of Newfoundland, St. John's, NL, Canada, as an Assistant Professor and later an Associate Professor. In 2019, she joined the University of Saskatchewan, where she is currently an Associate Professor and Canada Research Chair with Technology Solutions for Energy Security in Remote, Northern, and Indigenous Communities. From 2019 to 2022, she was an Adjunct Professor at the Memorial University of Newfoundland. Her research interests include power systems, renewable energy, and electric machines. Dr. Liang is the registered Professional Engineer in the province of Saskatchewan, Canada, a Fellow of IET, and the Deputy Editor-in-Chief of IEEE Transactions on Industry Applications.