A Two-Stage Stacked Neural Network Framework for Computationally Efficient Reliability Optimization

Runze Han, Tianyang Zhao, Gengfeng Li, Zhaohong Bie
School of Electrical Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
Hanrunze@stu.xjtu.edu.cn

***Abstract*—Circuit breaker placement optimization is critical for distribution network reliability but faces computational challenges: traditional methods suffer from intractability or suboptimality, while two-stage stochastic optimization remains prohibitive due to expensive reliability assessments. Neural network-based optimization enables rapid evaluation through mixed-integer linear programming (MILP), but embedding binary neural networks (BNNs) introduces exponential growth of binary variables. This paper proposes a two-stage NN framework where Stage 1 performs sensitivity analysis to reduce the decision space, and Stage 2 constructs a compact BNN-MILP with augmented Lagrangian constraints ensuring conservative predictions for critical nodes. Case studies demonstrate 64.5% reduction in binary variables and 7.3× speedup.**

***Index Terms*—circuit breaker placement, binary neural network, MILP, sensitivity analysis, distribution reliability**

I. Introduction

Distribution network reliability is a critical performance metric affecting customer satisfaction and utility revenue. Circuit breakers enable rapid fault isolation and service restoration, significantly improving reliability indices. However, high installation costs necessitate systematic optimization to balance reliability and economic investment [1], [2].

The circuit breaker placement problem is inherently combinatorial. For a network with N candidate lines and allowing at most B circuit breakers, there exist $\sum_{i=1}^{B}\binom{N}{i}$ feasible configurations, each requiring comprehensive reliability assessment. Traditional methods face fundamental limitations: 1) Exhaustive enumeration becomes intractable when the search space exceeds thousands of configurations [3]; 2) Metaheuristic algorithms may converge to local optima [4].

Two-stage stochastic optimization frameworks have been proposed to handle uncertainty in fault scenarios [5], [6], decomposing the problem into investment decisions (outer stage) and reliability assessment (inner stage). However, exhaustive evaluation of all feasible configurations requires extensive Monte Carlo simulations (5,000-10,000 samples per configuration), taking hours to days for practical networks [7].

Neural network-based optimization offers a promising alternative by learning the implicit configuration-to-reliability mapping from pre-simulated data. Training on offline samples reduces evaluation time from hours to milliseconds, enabling rapid exploration of the configuration space [8], [9]. BNNs are particularly attractive as they can be exactly reformulated as mixed-integer linear programming (MILP) constraints [10], [11], enabling inverse optimization: given target reliability requirements, the optimal configuration is obtained by solving the embedded BNN-MILP model.

However, this introduces a new computational bottleneck. Each neuron requires binary variables for activation states. A typical BNN with 3 hidden layers and 128 neurons per layer generates 384 binary variables and over 50,000 constraints, requiring hours to solve [12]. This creates a fundamental dilemma: larger BNN models improve accuracy but exponentially increase optimization difficulty.

To address these challenges, this paper proposes a two-stage stacked neural network framework. Stage 1 employs gradient-based sensitivity analysis to reduce the decision space by 71–94%. Stage 2 constructs a compact BNN-MILP with augmented Lagrangian constraints ensuring conservative reliability predictions for critical nodes.

The main contributions are:

- A hierarchical framework achieving significant reduction in binary variables through sensitivity-driven search space pruning while preserving solution optimality.
- Augmented Lagrangian constraints guaranteeing conservative predictions that ensure feasible solutions meet reliability requirements as the sufficient condition.

II. Problem Framework

This section formulates the circuit breaker placement problem as a constrained optimization problem. Section II-A defines the decision variables, objectives, constraints, and the reliability assessment model. Section II-B discusses the computational challenges.

A. Problem Formulation and Reliability Assessment

Consider a radial distribution network with m critical nodes and N candidate lines $\mathcal{L} = \{1, 2, \ldots, N\}$ for circuit breaker installation. The binary decision variable $x_i \in \{0, 1\}$ indicates whether a circuit breaker is installed on line i.

Supported by Smart Grid-National Science and Technology Major Project (2030) Pilot Project for Low-Carbon and High-Reliability Urban Distribution Power Systems (2024ZD0800700). Corresponding author: Tianyang Zhao.

979-8-3315-8138-1

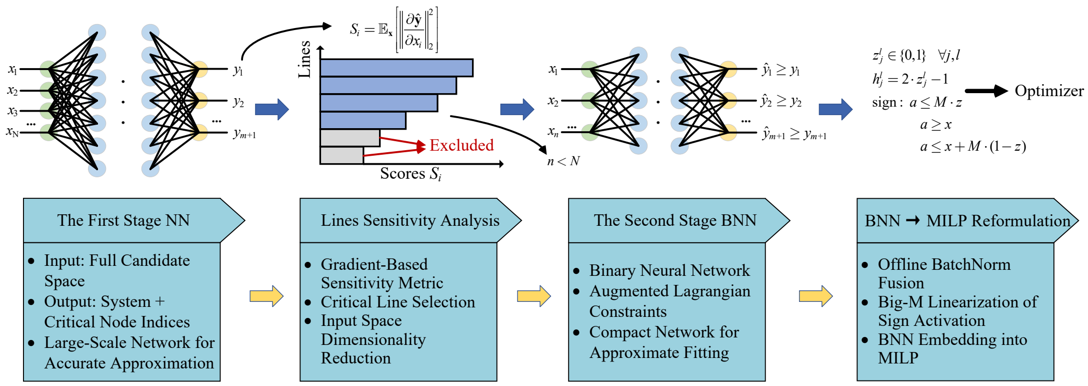

Fig. 1. Overview diagram

The typical problems is budget-constrained reliability maximization. Given a fixed budget, minimize system-level Expected Energy Not Supplied (EENS) while ensuring critical nodes meet Customer Interruption Frequency (CIF) requirements [13],

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \mathrm{EENS}(\mathbf{x}) \\
\text{s.t.} \quad & \sum_{i=1}^{n} x_i \le B \\
& \mathrm{CIF}_j(\mathbf{x}) \le \mathrm{CIF}_j^{\max}, \quad \forall j \in \mathcal{N}_{\text{crit}} \\
& \mathbf{y}(\mathbf{x}) = \mathrm{RA}(\mathbf{x}) \\
& x_i \in \{0,1\}, \quad \forall i \in \mathcal{L}
\end{aligned}
\tag{1}
$$

where B is total budget constraint; $\mathcal{N}_{\text{crit}} \subseteq \mathcal{N}$ is set of critical nodes (hospitals, data centers, industrial facilities). $\mathbf{y}(\mathbf{x}) = [\mathrm{EENS}(\mathbf{x}), \mathrm{CIF}_1(\mathbf{x}), ..., \mathrm{CIF}_m(\mathbf{x})]^T$ denotes the vector of reliability indices evaluated through the three-stage fault recovery process $\mathrm{RA}(\cdot)$ that models fault isolation, microgrid reconfiguration, and permanent repair operations [15]. The circuit breaker configuration $\mathbf{x}$ determines fault isolation boundaries, which subsequently affects load restoration capabilities across recovery stages.

B. Computational Challenge

The formulations (1) present significant computational challenges:

- Combinatorial explosion: For N candidate lines with budget constraint allowing at most B circuit breakers, there exist $\sum_{i=1}^{B}\binom{N}{i}$ possible configurations. For example, with $n = 30$ and $B = 8$, this yields approximately 5.7×10^8 configurations.
- Expensive evaluation: Each reliability assessment requires solving $|\mathcal{F}| \times 3$ mixed-integer linear programs (one for each stage of each $N-1$ fault scenario).

III. Two-Stage Stacked Neural Network Framework

This section presents the proposed two-stage stacked neural network framework. The framework addresses the computational challenges through hierarchical decomposition: Stage 1 trains a neural network and performs sensitivity analysis to reduce the decision space, and Stage 2 constructs a compact BNN-MILP model for exact optimization with conservative constraints, as Fig. 1.

A. Stage 1 Neural Network Training

The Stage 1 neural network serves as a surrogate model to rapidly evaluate reliability indices for arbitrary circuit breaker configurations. This enables data-driven sensitivity analysis (Section III-B) to identify critical lines where circuit breakers have the most significant reliability impact.

1) Network Architecture: The Stage 1 model can employ any feedforward neural network architecture capable of accurately approximating the reliability evaluation function. Without loss of generality, we adopt a multi-task architecture with shared encoder and task-specific heads to capture both system-level and node-level reliability metrics. With m critical nodes, the network maps circuit breaker configurations to reliability indices:

$$f_{\boldsymbol{\theta}} : \mathbf{x} \in \{0,1\}^N \to \{\mathbf{y}_{\text{sys}}, \mathbf{y}_{\text{node},1}, \ldots, \mathbf{y}_{\text{node},m}\} \tag{2}$$

where $\mathbf{x} \in \{0,1\}^N$ is the circuit breaker configuration vector for N candidate lines; $\mathbf{y}_{\text{sys}} = [\mathrm{SAIFI}, \mathrm{SAIDI}, \mathrm{EENS}]^T \in \mathbb{R}^3$ represents system-level reliability indices; $\mathbf{y}_{\text{node},k} = [\mathrm{CIF}_k, \mathrm{CID}_k, \mathrm{EENS}_k]^T \in \mathbb{R}^3$ represents node-level indices for critical node $k = 1, \ldots, m$; $\boldsymbol{\theta}$ denotes the trainable network parameters.

The network structure consists of:

$$
\begin{cases}
\mathbf{h} = g_{\text{enc}}(\mathbf{x}; \boldsymbol{\theta}_{\text{enc}}) \\
\mathbf{y}_{\text{sys}} = g_{\text{sys}}(\mathbf{h}; \boldsymbol{\theta}_{\text{sys}}) \\
\mathbf{y}_{\text{node},k} = g_{\text{node},k}(\mathbf{h}; \boldsymbol{\theta}_{\text{node},k})
\end{cases}
, k = 1, \ldots, m \tag{3}
$$

where $g_{\text{enc}}(\cdot)$ is a shared encoder extracting common features, and $g_{\text{sys}}(\cdot)$, $g_{\text{node},k}(\cdot)$ are task-specific heads predicting reliability indices for the system and each critical node, respectively.

2) Training Procedure: The Stage 1 model is trained using uncertainty-weighted multi-task learning [12] to balance system-level EENS and node-level CIF predictions:

$$\mathcal{L}_{\text{total}} = \sum_{k=1}^{m+1} \left[\frac{1}{2\sigma_k^2} \|\mathbf{y}_k - \hat{\mathbf{y}}_k\|_2^2 + \frac{1}{2} \log \sigma_k^2 \right] \tag{4}$$

where σ_k^2 are learnable task uncertainties. The weighting scheme automatically down-weights tasks with higher intrinsic

979-8-3315-8138-1

noise while preventing trivial solutions via the log-barrier regularization term.

The model is trained on a dataset $\mathcal{D} = \{(\mathbf{x}^{(j)}, \mathbf{y}^{(j)})\}_{j=1}^{N_s}$ generated offline via Monte Carlo simulation with the three-stage fault recovery model [15].

B. Sensitivity Analysis and Search Space Reduction

The trained Stage 1 neural network enables efficient sensitivity analysis to identify critical lines where circuit breakers have the most significant reliability impact. This reduces the decision space for Stage 2 BNN-MILP optimization.

1) Gradient-Based Sensitivity Metric: For each candidate line i, we compute a sensitivity score that quantifies its marginal contribution to reliability improvement. The sensitivity is defined as the expected magnitude of reliability change when installing a circuit breaker on line i:

$$S_i = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left\|\frac{\partial \hat{\mathbf{y}}}{\partial x_i}\right\|_2^2\right] \tag{5}$$

where $\mathbf{y} = [\mathbf{y}_{\text{sys}}^T, \mathbf{y}_{\text{node},1}^T, \dots, \mathbf{y}_{\text{node},m}^T]^T \in \mathbb{R}^{3(m+1)}$ is the concatenated output vector of all reliability indices, and $\mathcal{D}$ is the training dataset. High S_i indicates strong influence on reliability, while low S_i suggests marginal impact.

2) Finite Difference Approximation: For BNNs with non-differentiable activations, we approximate gradients using finite differences. The sensitivity score is computed by averaging over the training dataset:

$$S_i = \frac{1}{N_s}\sum_{j=1}^{N_s}\left\|f_{\boldsymbol{\theta}}(\mathbf{x}^{(j)}|_{x_i=1}) - f_{\boldsymbol{\theta}}(\mathbf{x}^{(j)}|_{x_i=0})\right\|_2^2 \tag{6}$$

where $\{\mathbf{x}^{(j)}\}_{j=1}^{N_s}$ are sampled configurations from the training set, and N_s is the sample size.

3) Critical Line Selection: Lines are ranked by sensitivity score S_i in descending order, and the top n lines are selected as $\mathcal{L}_{\text{crit}}$ for Stage 2 optimization. Reducing decision variables from N to n decreases MILP complexity by a factor of 2^{N-n}. Retaining lines with cumulative sensitivity $\sum_{i\in\mathcal{L}_{\text{crit}}} S_i \geq \tau \sum_{i=1}^{n} S_i$ preserves at least τ fraction (typically $\tau = 0.85$-0.90) of total sensitivity.

The reduced candidate set $\mathcal{L}_{\text{crit}}$ serves as the input to Stage 2, where a compact BNN with conservative constraints is trained and embedded into MILP for exact optimization.

C. Stage 2: BNN Construction with Conservative Constraints

The Stage 2 model operates on the reduced candidate set $\mathcal{L}_{\text{crit}}$ identified by Stage 1 sensitivity analysis. To enable exact MILP-based optimization, we employ a Binary Neural Network (BNN) architecture that can be reformulated as mixed-integer linear constraints.

1) Binary Neural Network Architecture: BNNs replace standard activation functions with sign activations $\text{sign}(\cdot) \in \{-1, +1\}$ and binarize weights during forward propagation, enabling exact MILP reformulation without approximation errors. The Stage 2 BNN consists of:

$$\begin{cases} \mathbf{z}^{(1)} = \text{BN}(\mathbf{W}_{\text{bin}}^{(1)}\mathbf{x}_{\text{reduced}} + \mathbf{b}^{(1)}) \\ \mathbf{h}^{(i)} = \text{sign}(\mathbf{z}^{(i)}), \quad i = 1, \dots, L \\ \mathbf{z}^{(i+1)} = \text{BN}(\mathbf{W}_{\text{bin}}^{(i+1)}\mathbf{h}^{(i)} + \mathbf{b}^{(i+1)}), \quad i = 1, \dots, L-1 \\ \hat{\mathbf{y}} = \mathbf{W}^{(L+1)}\mathbf{h}^{(L)} + \mathbf{b}^{(L+1)} \end{cases} \tag{7}$$

where $\mathbf{x}_{\text{reduced}} \in \{0,1\}^n$ is the reduced configuration vector; $\mathbf{W}_{\text{bin}}^{(i)} = \alpha^{(i)} \cdot \text{sign}(\mathbf{W}^{(i)})$ are binarized weights with scaling factors $\alpha^{(i)} = \text{mean}(|\mathbf{W}^{(i)}|)$; $\text{BN}(\cdot)$ denotes batch normalization; $\mathbf{h}^{(i)} \in \{-1,+1\}^{d_i}$ are binary activations; L is the number of hidden layers; d_i denotes the number of neurons in the i-th hidden layer; $\hat{\mathbf{y}} \in \mathbb{R}^{m+1}$ predicts $[\text{EENS}, \text{CIF}_1, ..., \text{CIF}_m]^T$ for critical nodes; $b^{(i)}$ is the vector of bias parameters.

This design focuses on system-level EENS (economic objective) and node-level CIF for the critical node (service quality metric), preserving all decision-relevant metrics while using compact network architecture.

2) Conservative Constraint via Augmented Lagrangian: To ensure reliability requirements are satisfied under model uncertainty, we enforce conservative prediction constraints during BNN training. The objective is to produce slightly pessimistic predictions (i.e., $\hat{y}_j \geq y_j$) that avoid under-estimating system risk, thereby preventing violations of critical node reliability thresholds in the subsequent MILP optimization.

For critical node outputs $j \in \mathcal{N}_{\text{crit}}$ (e.g., CIF for hospitals/data centers), we impose a soft lower bound on predictions to penalize under-estimation:

$$\hat{y}_j(\mathbf{x}) \geq y_j(\mathbf{x}) + \epsilon_{\text{tol}}, \quad \forall \mathbf{x} \in \mathcal{D},\ j \in \mathcal{N}_{\text{crit}} \tag{8}$$

where $y_j(\mathbf{x})$ is the true reliability index, $\hat{y}_j(\mathbf{x})$ is the BNN prediction, and ϵ_{tol} is a small tolerance margin (typically $\epsilon_{\text{tol}} = 0.02\cdot\text{std}(y_j)$) to accommodate minor prediction errors without over-conservatism.

The constraint violation $v_j(\mathbf{x})$ quantifies the extent to which predictions under-estimate true values beyond tolerance ϵ_{tol}. The total training loss combines prediction accuracy with augmented Lagrangian penalties:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}}(\hat{\mathbf{y}}, \mathbf{y}) + \sum_{j\in\mathcal{N}_{\text{crit}}}\sum_{\mathbf{x}\in\mathcal{D}}\left[\lambda_j \cdot v_j(\mathbf{x}) + \frac{\rho}{2}v_j(\mathbf{x})^2\right] \tag{9}$$

where $\lambda_j \geq 0$ is the Lagrange multiplier for output j, dynamically adjusted during training; $\rho > 0$ is the penalty parameter, controlling the quadratic penalty strength; The linear term $\lambda_j v_j$ enforces constraint satisfaction in expectation; The quadratic term $\frac{\rho}{2}v_j^2$ provides robustness against large violations.

The Lagrange multipliers are updated every T_{update} epochs, using the gradient ascent rule:

$$\lambda_j \leftarrow \text{clip}\left(\lambda_j + \rho \cdot \bar{v}_j, 0, \lambda_{\max}\right) \tag{10}$$

where $\bar{v}_j = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{x}\in\mathcal{D}} v_j(\mathbf{x})$ is the average violation over the training set, and $\lambda_{\max}$ prevents unbounded growth.

979-8-3315-8138-1

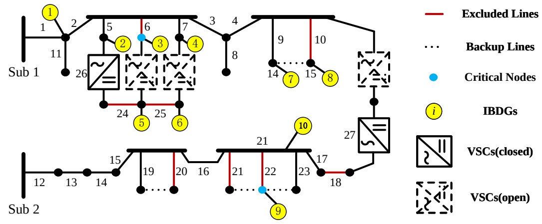

Fig. 2. A real hybrid AC/DC distribution network in a southern Chinese city

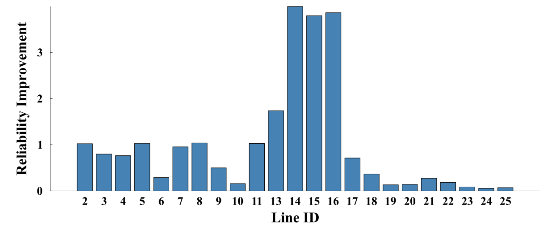

Fig. 3. All lines sensitivity analysis

The penalty parameter ρ is adaptively increased when violations persist:

$$\rho \leftarrow \min\left(\rho \cdot \gamma, \rho_{\max}\right) \quad \text{if} \quad \bar{v}_j > \delta_{\text{tol}} \tag{11}$$

where γ is growth factor; δ_{tol} is violation tolerance threshold. This adaptive scheme balances constraint satisfaction with training stability.

D. BNN-to-MILP Encoding for Exact Optimization

After training and pruning the BNN surrogate, we embed it into a MILP formulation to enable exact optimization of circuit breaker placement. This section describes the encoding of binary activations, Big-M linearization, and the final optimization problem.

1) Offline BatchNorm Fusion: To reduce MILP complexity, we fuse BatchNorm layers into the preceding linear transformations offline. For each BNN layer with weight $\mathbf{W}^{(i)}$, bias $\mathbf{b}^{(i)}$, and BatchNorm parameters $(\gamma^{(i)}, \beta^{(i)}, \mu^{(i)}, \sigma^{2(i)})$, the fused parameters $\tilde{\mathbf{W}}^{(i)}$ and $\tilde{\mathbf{b}}^{(i)}$ are computed by absorbing the affine transformation into the linear layer [14]. This eliminates BatchNorm from the MILP, reducing the number of auxiliary variables.

2) Big-M Linearization of Sign Activation: Each BNN layer computes $\mathbf{h}^{(i)} = \text{sign}(\mathbf{z}^{(i)})$ where $\mathbf{z}^{(i)} = \tilde{\mathbf{W}}^{(i)}\mathbf{h}^{(i-1)} + \tilde{\mathbf{b}}^{(i)}$. We encode this using binary indicator variables $\delta_j^{(i)} \in \{0,1\}$ and continuous activation variables $h_j^{(i)} \in \{-1,+1\}$ via Big-M constraints:

$$\begin{cases} z_j^{(i)} = \sum_{k=1}^{d_{i-1}} \tilde{W}_{jk}^{(i)} h_k^{(i-1)} + \tilde{b}_j^{(i)} \\ z_j^{(i)} \geq -M_j^{(i)}(1-\delta_j^{(i)}) \\ z_j^{(i)} \leq M_j^{(i)}\delta_j^{(i)} \\ h_j^{(i)} = 2\delta_j^{(i)} - 1 \end{cases} \tag{12}$$

where $M_j^{(i)}$ is the tightest bound on $|z_j^{(i)}|$, computed as $M_j^{(i)} = \sum_{k=1}^{d_{i-1}} |\tilde{W}_{jk}^{(i)}| + |\tilde{b}_j^{(i)}|$ based on the fact that $h_k^{(i-1)} \in \{-1,+1\}$. Using neuron-specific bounds instead of a global constant improves MILP solver efficiency.

3) Complete MILP Formulation: The circuit breaker placement problem is formulated as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \text{EENS}(\mathbf{x}) \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_i \leq B \\ & \text{CIF}_j(\mathbf{x}) \leq \text{CIF}_j^{\max}, \quad \forall j \in \mathcal{N}_{\text{crit}} \\ & \hat{\mathbf{y}}(\mathbf{x}) = \text{BNN}(\mathbf{x}) \\ & \mathbf{x} \in \{0,1\}^n \end{aligned} \tag{13}$$

where $\text{BNN}(\mathbf{x})$ represents the embedded neural network. Compared to formulation (1) where reliability indices $\mathbf{y}(\mathbf{x})$ are evaluated through $\text{RA}(\cdot)$, the BNN-embedded formulation (13) uses $\text{BNN}(\cdot)$, enabling rapid evaluation through mixed-integer linear constraints.

IV. Case Study

A. Test System

The proposed two-stage NN framework is validated using a real hybrid AC/DC distribution network from a southern Chinese city, as illustrated in Fig. 2.

23 candidate lines (excluding substation feeders, VSC connections and backup lines) are evaluated. Node 27 (hospital) should follow $\text{CIF}_{27}^{\max} = 2.5$ interruptions/year and node 9 (data center) should follow $\text{CIF}_{9}^{\max} = 2.0$ interruptions/year.

The Stage 1 neural network is trained on a dataset of more than 8,000 samples generated through Monte Carlo simulation with random circuit breaker configurations (1-5 breakers per configuration).

B. Sensitivity Analysis Validation

The Stage 1 of the proposed framework employs NN-based sensitivity analysis to identify critical lines. Note that Lines 1 and 12, serving as substation feeders with inherent circuit breaker protection, are excluded from the candidate set. This leaves 23 distribution lines (Lines 2-11, 13-25) for optimization.

TABLE I
Search Space Reduction via Tier 4 Elimination

Budget (B breakers)	Full Space ($n = 23$)	Reduced Space ($n = 15$)	Reduction (%)
3	$2,024$	576	71.5
5	$67,298$	$3,808$	94.3
7	$1,144,066$	$13,328$	98.8

Fig. 3 illustrates the reliability improvement potential of all 23 candidate lines. The sensitivity analysis reveals a clear hierarchical structure: Tier 1 (Lines 13-16) exhibits the highest scores, corresponding to main feeder segments; Tier 2 (Lines 2-5, 7, 8, 11) represents primary distribution paths; Tier 3 (Lines 3, 6, 9, 17, 18, 21) serves intermediate positions with limited upstream protection; Tier 4 (Lines 10, 19-22, 24-25) comprises terminal branches with scores below 0.20.

A critical observation is that high sensitivity scores do not guarantee priority in optimal configurations. Lines in Tiers 1-2, while exhibiting strong individual performance, are predominantly on main feeders where upstream protection already mitigates fault impacts. Conversely, low-scoring lines

979-8-3315-8138-1

TABLE II
PERFORMANCE COMPARISON UNDER BUDGET CONSTRAINTS

Budget (B)	Method	Optimal Config.	System EENS (MWh)	Node 9 CIF	Node 27 CIF	Solution Time (s)	Speedup Factor
3	Benchmark	{2, 4, 14}	40.752	0.600	0.900	2,135	–
	Proposed	{2, 4, 14}	41.308	0.971	1.014	68	31.4×
4	Benchmark	{2, 4, 11, 14}	39.247	0.500	0.900	6,846	–
	Proposed	{2, 4, 11, 14}	39.291	0.941	1.010	370	18.4×
5	Benchmark	{2, 4, 9, 11, 14}	31.580	0.500	0.900	24,444	–
	Proposed	{2, 4, 9, 11, 14}	31.767	0.862	0.958	3,349	7.3×

(e.g., Line 18) may appear in optimal solutions due to complementary fault isolation when combined with high-sensitivity lines. Tier 4 lines consistently rank at the bottom with minimal impact on optimal solutions. Removing these 8 lines reduces the search space to 15 lines while preserving solution quality. Table I quantifies the combinatorial complexity reduction. The pruning strategy achieves 71.5-98.8% search space reduction across different budgets while preserving solution optimality, enabling efficient BNN-MILP optimization.

C. Method Effectiveness Comparison

This section validates the proposed two-stage framework by comparing it against the benchmark method. The benchmark employs a Two-Stage Stochastic Optimization framework where the inner-stage reliability assessment is performed via rigorous Monte Carlo Simulation (MCS).

Table II presents the optimal circuit breaker configurations and resulting reliability metrics for budget limits $B \in \{3, 4, 5\}$. The benchmark method employs a two-stage optimization approach. This provides a reference for solution quality at the cost of prohibitive computational expense. The proposed framework identifies identical optimal configurations as the benchmark across all scenarios, validating the two-stage approach's effectiveness. The proposed method exhibits minor conservative bias (EENS error 0.59%, Node 27 CIF error 6.4% for $B = 5$). This conservative bias serves a critical engineering purpose: it ensures that planned configurations meet reliability requirements with safety margins. Specifically, the over-estimation mechanism guarantees that: 1) If the BNN-MILP finds a feasible solution under budget B, the actual system reliability will exceed predicted values, providing a safety buffer against model uncertainty. 2) If the optimization becomes infeasible (no solution satisfies CIF constraints), it certifies that budget B is insufficient, and additional breakers ($B + 1$, $B + 2$, ...) are required. The proposed framework achieves 7.3-31.4× speedup compared to the benchmark method across different budgets. This acceleration stems from two factors: 1) 71.5-94.3% search space reduction via sensitivity analysis, and 2) BNN-MILP's ability to evaluate reliability constraints via embedded neural network inference rather than iterative power flow simulations.

V. CONCLUSION

This paper presents a two-stage stacked neural network framework for circuit breaker placement optimization, achieving 7.3-31.4× speedup via sensitivity-based search space reduction (71.5-94.3%) and compact BNN-MILP with conservative constraints. Case studies show $< 0.6\%$ EENS error while guaranteeing critical node requirements, eliminating under-protection risks for critical infrastructure.

REFERENCES

[1] Z. Li, W. Wu, X. Tai, et al., "Optimization model-based reliability assessment for distribution networks considering detailed placement of circuit breakers and switches," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 3991–4004, Sep. 2020.

[2] A. Heidari et al., "Reliability optimization of automated distribution networks with probability customer interruption cost model in the presence of DG units," *IEEE Trans. Smart Grid*, vol. 8, no. 1, pp. 305–315, Jan. 2017.

[3] Z. Guo, W. Wei, Y. Liu, et al., "Real-time distribution network reconfiguration using multi-stage robust optimization," *IEEE Trans. Smart Grid*, vol. 16, no. 6, pp. 4475–4487, Nov. 2025.

[4] A. M. Helmi, R. Carli, M. Dotoli, et al., "Efficient and sustainable reconfiguration of distribution networks via metaheuristic optimization," *IEEE Trans. Autom. Sci. Eng.*, vol. 19, no. 1, pp. 82–98, Jan. 2022.

[5] C. Wang, K. Pang, M. Shahidehpour, et al., "MILP-based fault diagnosis model in active power distribution networks," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 3847–3857, Sep. 2021.

[6] A. Shahbazian, A. Fereidunian and S. D. Manshadi, "Optimal switch placement in distribution systems: A high-accuracy MILP formulation," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5009–5018, Nov. 2020.

[7] S. Cho, J. Tovar-Facio and I. E. Grossmann, "Impact of reliability formulations on the optimal planning and operation of power systems," *IEEE Trans. Power Syst.*, early access, 2025, doi: 10.1109/TPWRS.2025.3614510.

[8] D. Cao, Y. Liu, Y. Wang et al., "Deep reinforcement learning in power systems resilience: A review," *IEEE Trans. Rel.*, early access, 2025, doi:10.1109/TR.2025.3606215.

[9] Q. Yuan, X. Liu and M. Mao, "Efficient Deep Reinforcement Learning Based Coordination for Unlocking Electric Vehicle Decarbonization Potential in Coupled Transportation and Power Networks," *IEEE Trans. Power Syst.*, vol. 40, no. 4, pp. 2943-2954, July 2025.

[10] W. Zhao, T. Ma, X. Gong et al., "A review of recent advances of binary neural networks for edge computing," *IEEE J. Miniaturization Air Space Syst.*, vol. 2, no. 1, pp. 25–35, Mar. 2021.

[11] X. Luo, S. Dong, J. Hong, et al., "Deep learning-based mixed-integer co-optimization of velocity planning and powertrain control in urban environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 11, pp. 18581–18596, Nov. 2025.

[12] R. Sayed, H. Azmi, H. Shawkey, et al., "A systematic literature review on binary neural networks," *IEEE Access*, vol. 11, pp. 27546–27578, 2023.

[13] Z. Li, W. Wu, B. Zhang and X. Tai, "Analytical Reliability Assessment Method for Complex Distribution Networks Considering Post-Fault Network Reconfiguration," in IEEE Transactions on Power Systems, vol. 35, no. 2, pp. 1457-1467, March 2020.

[14] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 7482–7491.

[15] R. Han and T. Zhao, "A three-stage reliability optimization scheme for hybrid AC/DC distribution systems: Supplementary material," 2025. [Online]. Available: https://github.com/thunderwizard0/ReliabilityAssessment-Model.