ELECTRICAL ENGINEERING AND ENERGY

OPEN ACCESS

End to end prediction of optimal power flow in sustainable power system using physics-informed spatiotemporal graph neural network

Abdul-Majid Issah Malori[1], Emmanuel Asuming Frimpong[1], Francis Effah Boafo[1] Elvis Twumasi[1], Peter Asigri[1], and Bernard Marfo Adjei[1]

[1] Department of Electrical and Electronic Engineering, Kwame Nkrumah University of Science and Technology, PMB University Post Office. Kumasi, Ashanti Region, Ghana

Submitted: 6 May 2026
Accepted: 1 June 2026
Online First: 10 June 2026

Corresponding author
Abdul-Majid Issah Malori,
issahmalori@bolgatu.edu.gh

DOI: 10.64470/elene.2026.33

Abstract: This paper proposes a physics-informed spatiotemporal graph neural network (PIStGNN) for end-to-end ACOPF prediction in sustainable power systems. The proposed framework embeds physical power flow constraints, grid topology, and temporal dynamics of renewable generation. The model is validated on IEEE 33-, 57-, and 118-bus systems using time-series 60,048 datasets generated in Pandapower. All experiments are implemented in Python and performance are evaluated using prediction accuracy, physical feasibility, and computational efficiency. The results show that system voltage profiles remain within acceptable operational limits of 0.96–1.04 p.u. across all scenarios. The system efficiency analysis reveals that moderate renewable penetration reduces transmission losses, while excessive penetration increases losses due to reverse power flow and congestion in larger systems. The proposed PIStGNN achieves prediction errors on the order of 10^{-2} p.u. for voltage magnitude and angle while delivering speedups of up to 10,135× compared with Newton–Raphson-based solvers. Furthermore, physics-informed recurrent graph models achieve minimum uncertainty values of 1.06×10^{-3}, 2.88×10^{-3}, and 1.19×10^{-3} for the IEEE 33-, 57-, and 118-bus systems, respectively. The results demonstrate that integrating physics-informed learning with spatiotemporal graph modeling provides an efficient surrogate for ACOPF prediction in sustainable power systems.

Keywords Alternating current optimal power flow, Renewable energy, Reconfiguration, Physics-informed, Spatiotemporal graph neural network.

1. Introduction

The efforts to lower greenhouse gas emissions in modern power grids have intensified due to climate change (Roald, et al., 2023) (Zhou, et al, 2024). It is estimated that one-third of global carbon dioxide emissions is produced by electricity generation (Y. Zhou, et al, 2015) (Xie, et al, 2023) (Lin, et al, 2014). An important step toward power system sustainability is renewable energy sources (Xie, Wei , Li, Dong, & Mei, 2023). However, these sources introduce significant intermittency, nonlinearity, and uncertainty into power system operations (C,. Lowery and M., O'Malley, 2012) (Jordehi, 2018) (Abdi, 2021) (Kaffash, 2022) (Roald, et al.,

2023) (Singh & Powar, 2024). Consequently, the rapid growth of renewable integration, dynamic load variations, and topological changes makes conventional techniques struggle with computational scalability of Alternating Current Optimal Power Flow (ACOPF) (Hmida, et al, 2018) (Aravena, et al., 2023). Traditional methods such as Interior Point, Newton–Raphson, and Sequential Quadratic Programming, suffer from slow convergence, and local optimality, particularly in uncertain situations (Lei, et al., 2021).The ACOPF problem has long been a mainstay of power system operation and planning. It determines the optimal dispatch of generation units to achieve specific objective while satisfying power network constraints (Yang, et al, 2017) (Halilbasic, et al, 2018). Significant economic benefits can result from exact ACOPF estimation (Kilwein, et al., 2021).Therefore, there is urgent need to implement efficient ACOPF technique for sustainable power system.

To address computational challenges, the data-driven method provides a promising tool for ACOPF assessment with stochastic renewable energy. These methods for solving the ACOPF problem can be divided into two main groups; End to End (E2E) and learning to optimize (L2O) (Hentenryck, 2021).The application of L2O and E2E learning techniques for ACOPF problem require further in-depth improvement (Hasan, et al, 2020). In modern grid, intermittency, uncertainty and topology changes increase the learning difficulties and the demand for learning samples (Gao, et al, 2024).This limits the efficacy of Euclidean-based models like convolutional neural networks (CNNs) and multilayer perceptrons. These methods suffer from over-smoothing, and poor interpretability (Saffari & Khodayar, 2024). Graph Neural Networks (GNNs) have become a potent tool for modeling grid-structured data in order to address these drawbacks. GNNs naturally capture the spatial dependencies of power systems by representing transmission lines as edges and buses as nodes (Zideh, et al, 2024) (Jalving, et al., 2024). However, of GNNs are efficient for static topology and neglects the dynamic evolution of the power network. Moreover, the spatiotemporal GNNs (StGNNs) models extend traditional GNNs by learning both spatial and temporal dynamics of dynamic systems (Wang, 2023). StGNNs are divided into two subcategories based on their learning strategy: the successive spatiotemporal GNNs architectures treat spatial and temporal learning as separate but sequential processes. The spatial features are extracted using the graph convolutional layers and temporal modeling using recurrent mechanisms. In contrast, simultaneous spatiotemporal GNNs learn spatial and temporal features within a unified framework. Its major limitation is it may not fully capture detailed temporal transitions where chronological time modeling is needed. Power system has a dynamic structure as a result of maintenance operations, faults and planned or unplanned outages. This implies that power system is represented by a dynamic graph topology (Yaniv, et al, 2023). Graphs may be homogeneous (uniform node and edge types) or heterogeneous (distinct node and edge properties), with the latter being ideal for modelling power systems. Buses are categorized into slack (S), generator (PV), and load (PQ) nodes to capture their specific operational roles. The evolution of node and edge features over short time intervals is analyzed by short-term spatiotemporal graph neural networks (StGNNs). The Short term StGNNs model abrupt temporal changes in graph-structured data. There are basically two groups namely; 1D-CNN-based and RNN-based models. The 1D-CNN-based models process longer sequences more effectively but have trouble with uncertain data. The RNN-based StGNNs capture variable-length temporal dependencies.

Furthermore, purely data-driven StGNNs run the risk of breaking basic physical laws like Kirchhoff's current and voltage laws. The physics informed models embed physical constraints directly into the learning process to guarantee feasibility. That notwithstanding physics-informed deep learning for ACOPF formulations is underexplored in literature (Huang & Wang, 2023). Motivated by these gaps, the paper seeks to solve ACOPF problem using PIStGNNs across different renewable penetration levels and networks. The central research hypothesis is "*physical regularization with spatiotemporal graph learning improves solution quality and generalization performance*". The null hypothesis assumes that no performance enhancement is achieved with PIStGNN modeling. To ascertain the stated hypotheses, the following research questions are formulated:

1) How can PIStGNN be designed to learn physics constraints, topology, and renewable variability?

2) To what extent does PIStGNN improve ACOPF performance metrics relative to classical approaches?
3) How robust is the PIStGNN models under different renewable penetration levels, and networks?

The proposed technique combines dynamic graph convolution networks (GCNs) with long short-term memory (LSTM) or gated recurrent units (GRU). The dynamic GCN layers captures the electrical topology using adjacency matrix in the graph convolution operation. The LSTM or GRU layers learn variability from renewable generation, load variations, and operational fluctuations. The PIStGNN algorithm is validated using time-series datasets generated in Pandapower on benchmark IEEE 33-, 57-, and 118-bus systems. The remainder of this paper is organized as follows. Section 2 formulates the ACOPF problem and its physical constraints. It also presents the theoretical foundation of the proposed PIStGNN architecture. Lastly the methodological framework and experimental implementation is described. Section 3 analyzes and discusses the experimental results, while Section 4 concludes with key findings and recommendations for future research.

2. Materials and Methods

2.1 Formulating ACOPF problem

Given that the power system network is modeled as graph network with n buses and m edges. For each bus i, voltage magnitude (V_i), voltage angle (θ_i), active power generated (P_i^G), reactive power generated (Q_i^G), active load (P_i^D) and reactive load (Q_i^D). P_{ij}^f and Q_{ij}^f also denote the active and reactive power flowing from bus i to bus j. The admittance between buses i and j is $g_{ij} - jb_{ij}$. θ_{ij} represent the shorthand for $\theta_i - \theta_j$. In this context buses are categorized into slack (S), generator (PV), and load (PQ) nodes to capture their specific operational roles. Let the power system at time t be a graph $G_t = (V_t, E_t)$ with $N = |V_t|$ buses and $E_t = |\mathcal{E}_t|$ lines. Node features are stacked as $X_t \in \mathbb{R}^{N\times F}$ and the model predicts ACOPF-related states/controls $\hat{Y}_t \in \mathbb{R}^{N\times O}$ (e.g.,$|V_i|, \theta_i, P_i^G, Q_i^G, P_i^D, Q_i^D, P_{ij}^f, Q_{ij}^f$). For a temporal window length T, the input sequence is $\{X_{t-T+1}, \dots, X_t\}$.The ACOPF problem is formulated as shown below:

$$\min_{V,\theta} ci(Pi\ G) \tag{1}$$

$$\text{St } P_i^G = P_i^D + \sum_{j=1}^{N} P_{ij}^f \tag{2}$$

$$Q_i^G = Q_i^D + \sum_{j=1}^{N} Q_{ij}^f \tag{3}$$

$$P_{ij}^f = V_i^2 g_{ij} - V_i V_j \left(g_{ij}\cos(\theta_{ij}) - b_{ij}\sin(\theta_{ij})\right) \tag{4}$$

$$Q_{ij}^f = V_i^2 b_{ij} - V_i V_j \left(b_{ij}\cos(\theta_{ij}) + g_{ij}\sin(\theta_{ij})\right) \tag{5}$$

$$\underline{V_i} \le V_i \le \overline{V}_i \tag{6}$$

$$\underline{P_i^G} \le P_i^G \le \overline{P}_i^G \tag{7}$$

$$\underline{Q_i^G} \le Q_i^G \le \overline{Q}_i^G \tag{8}$$

$$(P_{ij}^f)^2 - (Q_{ij}^f)^2 \le (S_{ij}^{max})^2 \tag{9}$$

where $\hat{b}_{ij} = b_{ij} - 0.5b_{ij}^C$ and b_{ij}^C is the line charging susceptance. The constraints (2) and (3) enforce power balance, (4) and (5) are the AC power flow equations, (6) limits the bus voltage magnitudes, (7) and (8) represent the active and reactive limits and (9) are the line flow limits.

2.2 Successive Spatiotemporal Graph Neural Network

Figure 1 illustrates the successive spatiotemporal architecture used for ACOPF prediction. The model receives a time series of grid measurements $X = \{X_1 \ldots, X_T\}$. These include voltage magnitude, voltage angle, active/reactive load, and generation at each bus. In Stage 1, a GCN or its physics-informed variant processes the grid topology at each timestep. The GCN captures spatial dependencies between buses by performing message passing along transmission lines. These produces spatial embeddings $\boldsymbol{S_1}, \ldots, \boldsymbol{S_T}$ that encode the network structure. In Stage 2, the sequence of spatial embeddings is passed to a temporal recurrent model, either an LSTM or GRU. It learns the temporal dynamics of the system caused by load fluctuations, renewable variability, and switching events. This stage models how the spatial states evolve over time. Finally, the output layer maps the learned representation to predicted system states. The outputs are denormalized physical quantities providing an estimate of the system operating point at the final timestep.

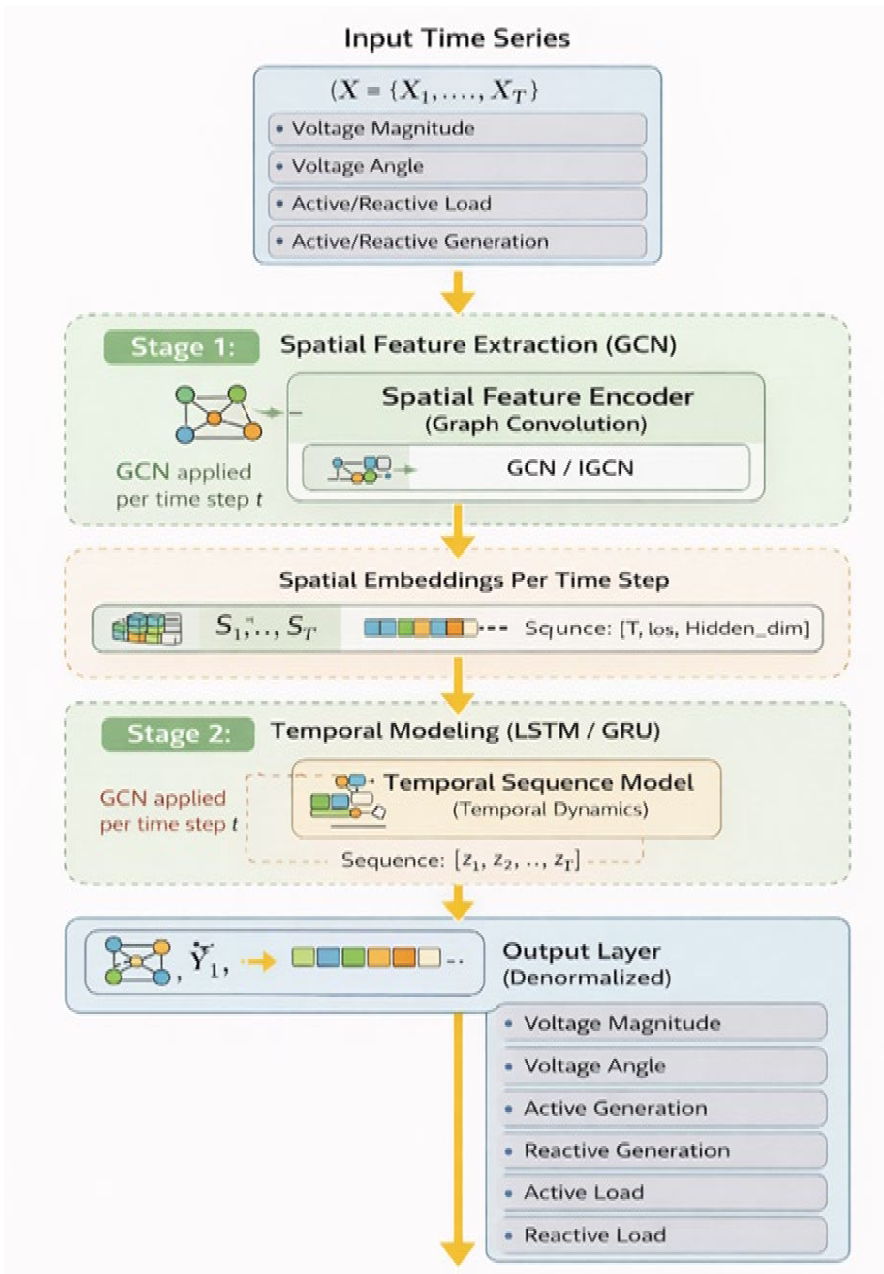

Figure 1 Successive spatiotemporal architecture used for power system state prediction

2.2.1 Graph Convolution Network Block Based on Physical Grid Topology

The structures of electric power systems constantly change as a result of maintenance operations, the integration of renewable energy sources, equipment failures, and outages. A dynamic graph, in which the nodes (buses) and edges (lines) vary over time is an efficient way to represent this power system. To reflect the changes in connections and node attributes, the adjacency matrix and feature matrices increase in dimension as the graph moves from $G_{t-1} = (V_{t-1}, E_{t-1})\ to\ G_t = (V_t, E_t)$. As shown in Figure 2 the evolution of the transmission network between two-time steps. While red edges in G_t represent reconfigured line introduced to optimize operational goals, blue edges in G_{t-1} indicate stable connections. These topological shifts are correspondingly mathematically encoded by the adjacency matrices A_{t-1} and A_t . Red cells indicate newly created connections during reconfiguration, while blue cells indicate active lines.

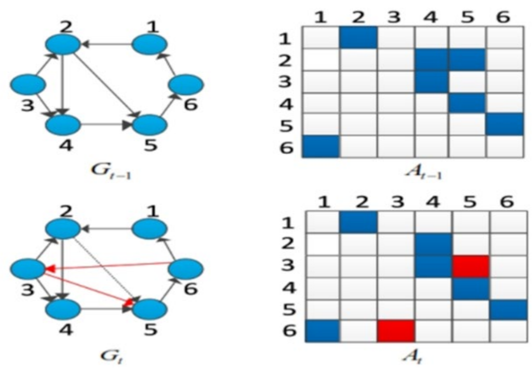

Figure 2 Illustration of network structure evolution overtime

The topology may change over time due to switching operations, contingencies, or network reconfiguration. These structural changes are captured through the time-varying adjacency matrix $A_t \in \mathbb{R}^{N\times N}$. where N is the number of buses and Aij=1if buses i and j are electrically connected. The proposed model strictly uses the physical power grid topology derived from the current network configuration. Therefore, message passing occurs only along actual transmission lines. The normalized adjacency matrix used for graph convolution is defined as

$$\tilde{A}_t = D_t^{-\frac{1}{2}}(A_t + I)D_t^{-\frac{1}{2}} \tag{10}$$

where D_t is the degree matrix of A_t and I is the identity matrix introducing self-loops. The graph convolution operation at layer l is then expressed as

$$H_t^{(l+1)} = \sigma\left(\tilde{A}_t H_t^{(l)} \mathrm{W}(l)\right) \tag{11}$$

where

- $H_t^{(l)}$ represents the node embeddings at layer *l*,
- $W(l)$ denotes the trainable weight matrix, and
- $\sigma(\cdot)$is a nonlinear activation function such as ReLU.

The initial layer uses the raw node features: $H_t^{(0)} = X_t$. After L graph convolution layers, the spatial representation becomes $S_t = H_t^{(L)}$. Which encodes topology-aware electrical interactions among buses. This formulation aligns with the implementation based on PyTorch Geometric's GCNConv operator, where graph message passing occurs strictly along the actual transmission network topology determined by the current breaker and switch states.

2.2.2 Long short-term memory block

Long Short-Term Memory (LSTM) networks are well suited for time-series prediction in dynamic systems. Therefore, highly appropriate for sustainable power system operation analysis. Power grids with high renewable penetration exhibit strong temporal characteristics. For accurate spatiotemporal modeling of ACOPF states, it essential to capture both short-term and long-term operational fluctuations. The LSTM architecture consist of memory cell, which functions as an information accumulator. The memory control is achieved through three gating mechanisms;

1) Input Gate: determines which components of the current input should be written into the memory cell.
2) Forget Gate: decides which information from the previous cell state should be removed.
3) Output Gate: regulates how the updated cell state contributes to the hidden state and network output.

Through these gating operations, LSTMs mitigate the vanishing and exploding gradient problems in recurrent neural networks. This ensures stable gradient propagation during training. Figure 3 shows the LSTM computational unit. X_t, denotes the input vector at time step t. While h_{t-1} and c_{t-1} represent the previous hidden and cell states, respectively. These states are updated to the current hidden state h_t and cell state c_t. It achieved through nonlinear transformations by sigmoid (σ) and hyperbolic tangent $(tanh)$ activation functions.

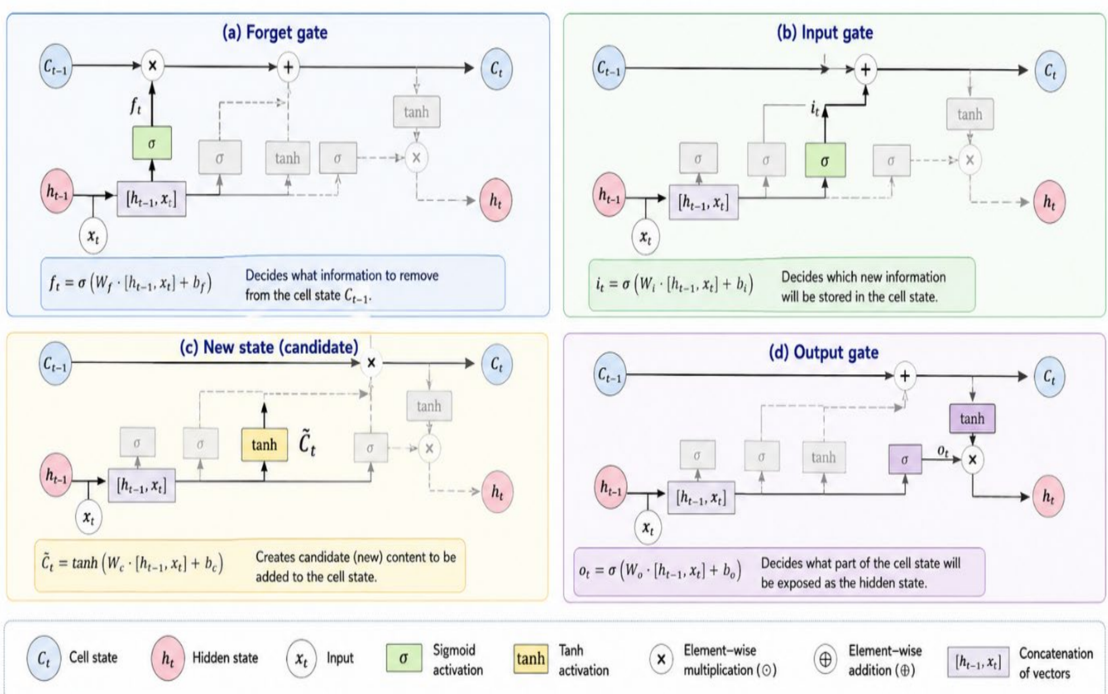

Figure 3 The LSTM internal gating structure

Figure 3 depicts the gating structure of the LSTM unit. It comprises, forget, input, and output gates that regulate temporal information flow.

1) Forget Gate

The forget gate (Figure 4a) determines which information is discarded from the previous cell state. It receives the current input X_t and previous hidden state h_{t-1}:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{12}$$

where (σ) outputs values in $(0,1)$, represents the degree of information retention. Values close to zero indicate forget, while values near one retain information.

2) Input Gate

The input gate (Figure 4b) controls the incorporation of new information into the cell state through a two-stage process:

First, a sigmoid layer determines which elements will be updated:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{13}$$

Second, a candidate memory vector is generated:

$$\tilde{c}_t = \tanh\,(W_c x_t + U_c h_{t-1} + b_c) \tag{14}$$

The cell state is then updated by combining retained historical information and new candidate content:

$$c_t = f_t^i \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{15}$$

3) Output Gate

The output gate regulates the information propagated to the hidden state and subsequent time steps:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{16}$$
$$h_t = o_t \odot \tanh\,(c_t) \tag{17}$$

This mechanism ensures relevant components of the updated memory cell influence the output.

2.2.3 Gated Recurrent Unit (GRU) block

The GRU block captures the temporal dependencies using a simplified gating mechanism as shown in Figure 4. The GRU merges memory and hidden representations into a single hidden state unlike LSTM. This reduction in structural complexity often yields faster training and lower parameter count. This can be advantageous for real-time ACOPF surrogate learning.

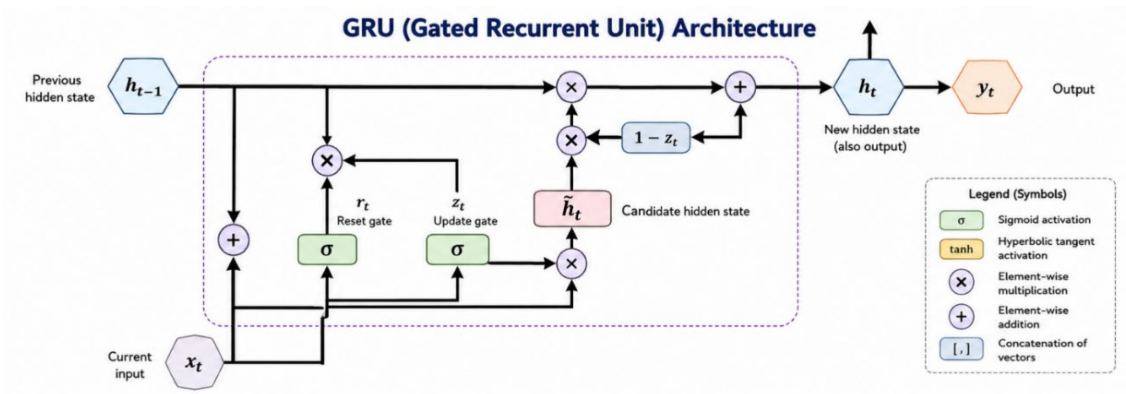

Figure 4 The gated recurrent unit network

Given input X_t and previous hidden state h_{t-1}, the standard GRU equations are defined as:

update gate $z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$ (18)

reset gate $r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$ (19)

candidate hidden state $\tilde{h}_t = tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$ (20)

hidden state update $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$ (21)

Where $\sigma(\cdot)$ is the sigmoid function, $tanh(\cdot)$ is the hyperbolic tangent, and $\odot$ denotes elementwise multiplication.

2.2.4 Successive Spatiotemporal GNN (GCN–LSTM)

Let the input sequence over a temporal window of length T be

$$X = \{X_1, X_2, \dots, X_T\}, X_t \in R^{N \times F} \quad (22)$$

where N is the number of buses and F is the node feature dimension. For each time step t, the graph convolution block extracts spatial features using the grid adjacency: $H_t^{(o)} = X_t$

$$H_t^{\ell+1} = \sigma\left(\tilde{A}_t H_t^{(\ell)} W^{\ell} + b(\ell)\right), \qquad \ell = 0,1, \dots, L-1 \quad (23)$$

where $\tilde{A}$ is the normalized adjacency matrix, $W(\ell)$ and $b(\ell)$ are trainable parameters, L is the number of graph convolution layers, and $\sigma(\cdot)$ is the nonlinear activation function. The final spatial embedding at time step t is $S_t = H_t^{(L)} \in \mathbb{R}^{N \times D}$ where D is the spatial embedding dimension. Since the recurrent block in the implementation uses a standard PyTorch LSTM, the spatial embedding is reshaped into a vector sequence before temporal modeling:

$$z_t = vec(S_t) \in \mathbb{R}^{N \times D} \quad (24)$$

The temporal dynamics are then modeled sequentially using a standard LSTM:

$$(h_t, c_t) = LSTM(z_t, h_{t-1}, c_{t-1}) \quad (25)$$

where h_t and c_t denote the hidden state and cell state at time step t, respectively. The final temporal representation is mapped to the output prediction through a fully connected layer:

$$\hat{Y} = W_o h_T + b_o \tag{26}$$

where $\hat{Y}$ contains the predicted ACOPF-related quantities.

2.2.5 Successive Spatiotemporal GNN (GCN–GRU)

For the GRU-based variant, the same spatial encoder is used, and the recurrent stage becomes

$$h_t = GRU(z_t, h_{t-1}) \tag{27}$$

with output prediction $\hat{Y}$, this formulation reflects the actual implementation in which graph convolution is applied. The temporal sequence learning is performed afterward using a standard recurrent unit.

2.2.6 Residual Successive Spatiotemporal GNN

For the residual variants, a skip connection is added between the projected spatial input and the recurrent output. Let the projected residual input be

$$r_t = W_r z_t + b_r \tag{28}$$

Then the residual recurrent representation is $\tilde{h}_t = h_t + r_t$ and the final output becomes $\hat{Y}$. The residual connection improves gradient flow during training.

2.3 Customized Loss Function

The physics-informed loss function is a linear combination of Mean Squared Error (MSE) and a regularization term based on the equality and inequality constraints. The PIStGNN is trained to predict the ACOPF optimality while maintaining physical feasibility.

$$L_{total} = w_1 L_{data} + w_2 L_{phys} + w_3 L_{safe} + w_4 L_{branch} \tag{29}$$

Where $L_i \in \{L_{data}, L_{phys}, L_{safe}, L_{const}\}$ denotes the loss terms.

1) Data fidelity loss

It ensures supervised consistency between the predicted state$\hat{y}$ and the ground-truth ACOPF solution y:

$$L_{data} = \|\hat{y} - y\|_2^2 \tag{30}$$

where all variables are normalized to ensure numerical stability. This term ensures that the PIStGNN accurately approximates reference ACOPF solutions generated using conventional solvers.

2) Physics consistency loss

This regularization term embeds AC power flow physics into the training process. The complex power injection at each bus is defined as:

$$S = P + jQ = V \cdot (Y_{bus}V)^* \tag{31}$$

Where V and Y_{bus} represent the complex bus voltage vector and admittance matrix respectively. P and Q stand for active and reactive power respectively.

$$L_{phys} = |P_{net} - Re(V \cdot (Y_{bus}V)^*)|^2 + |Q_{net} - Im(Y_{bus}V)^2|^2 \tag{32}$$

This constrains satisfy nonlinear AC power flow equations to reduce the risk of non-compliant predictions as observed in purely data-driven models.

3) Operational safety loss

A soft penalty function is used to maintain voltage security within permissible bounds;

$$L_{safe} = ReLU(|V| - V_{max})^2 + ReLU(V_{min} - |V|)^2 \tag{33}$$

where $ReLU(x) = \max(0, x)$.

4. Branch capacity loss (Inequality constraint)

$$L_{branch} = \frac{1}{LB}\sum[ReLU\left(|S_k| - S_{k,max}\right)^2] \tag{34}$$

where the branch apparent power flow is:$I_R = Y_{branch,k}\left(V_{from,k} - V_{to,k}\right)$, $|S_k| = \left|V_{from,k} \cdot I_k^*\right|$. $Y_{branch,k}$ is the series admittance of branch $\boldsymbol{k}$, extracted as $Y_{\{\mathbf{bus}\}[\mathbf{from},\mathbf{to}]}$. $S_{k,max}$ is the thermal rating of the branch in per-unit.

2.4 Performance Evaluation Metrics in Learning-based ACOPF

1. Optimality

The optimality measures how close the objective function value obtained by a learning-based scheme is to the value produce by a benchmark solver (ground truth). The optimality is evaluated by first computing the optimality loss (gap). This is obtained by calculating the Mean Absolute Error (MAE) of the objective function across a set of test samples, as expressed in equation (35). Subsequently, the optimality score is computed based on equation (36).

$$Optimality\ loss = \frac{1}{N}\sum_{i=1}^{N}\left|VM_i^{\circ} - VM_i^*\right| \tag{35}$$

$$Optimality = 1 - \frac{1}{N}\sum_{i=1}^{N}\left|VM_i^{\circ} - VM_i^*\right| \tag{36}$$

where i denotes the index of the test samples and N represents the total number of test samples. The variables VM_t° and VM_t^{*} denote the voltage magnitude obtained from the learning-based scheme and the benchmark solver, respectively. Similarly, VA_t° and VA_t^{*} represent the voltage angle associated with the machine learning-driven model and the benchmark solver. The optimality is quantified such that smaller MAE values indicate closer agreement with the benchmark solver. And higher optimality corresponds to better performance of the learning-based method.

2. Constraint satisfaction rate:
The quality of solutions obtained by machine learning techniques can be assessed by measuring the constraint satisfaction rates. These metrics are defined as:

$$Constraint\ satisfaction\ sate\ (\%) = \frac{n_s}{n_c} \times 100\% \tag{37}$$

Where n_s, n_c denote the total number of satisfied equality and inequality constraints, and total number of constraints, respectively

3. Computational efficiency
The computational efficiency metrics offer valuable insights into the efficiency and speed of model execution. The computational efficiency is typically analyzed via *speedup factor*. The computational efficiency metrics are described as follows:

$$Speed\ factor = \frac{\frac{1}{T}\sum_{t=1}^{T} CT_t^{*}}{\frac{1}{T}\sum_{t=1}^{T} CT_t^{\circ}} \tag{38}$$

CT_t° and CT_t^{*} denote the computation time for solving the OPF problem using a learning-based model and a benchmark solver, respectively.

2.5 Detailed Implementation of PIStGNN

Figure 5 shows an integrated framework that combines data-driven learning and physics-informed modeling, ACOPF under renewable energy integration. The overall methodology consists of two main stages: (i) data generation, and (ii) supervised physics-informed spatiotemporal learning,

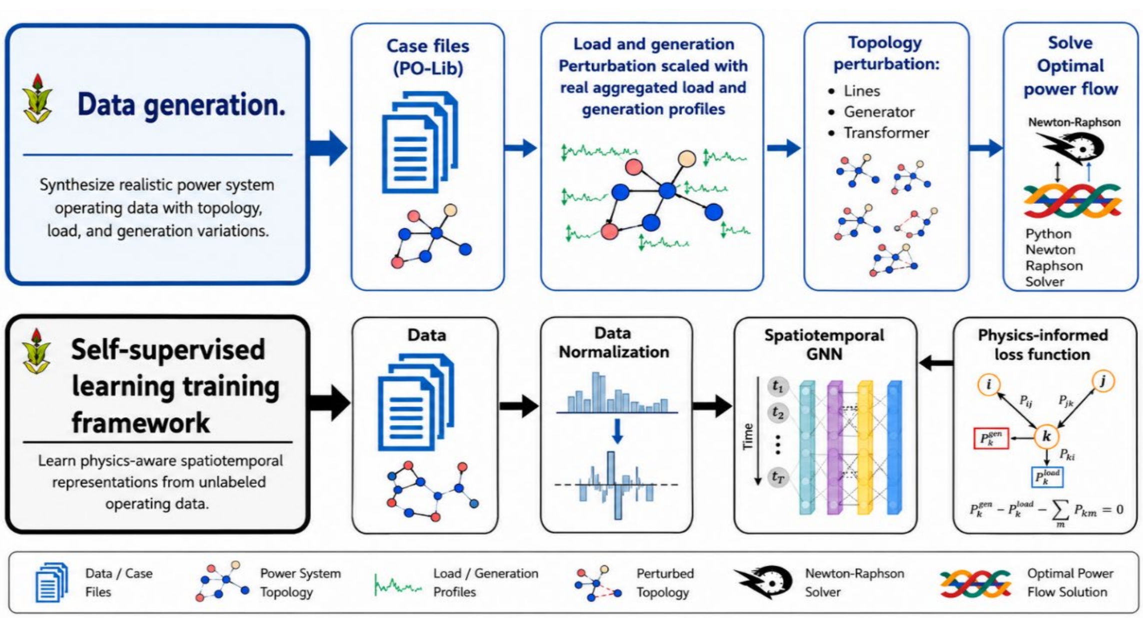

Figure 5 Detailed implementation of PIStGNN

2.5.1 Data generation pipeline

The data pipeline generates spatiotemporal power system datasets for training and evaluating the proposed graph neural network models. Three benchmark systems from the Pandapower library are used: the IEEE 33-bus, IEEE 57-bus, and IEEE 118-bus networks. These systems represent distribution and transmission grids with increasing structural complexity. The IEEE 33-bus system is a radial distribution network with normally open tie-lines enabling reconfiguration, whereas the IEEE 57-bus and IEEE 118-bus systems are meshed transmission networks without predefined tie-lines. The base power values are 10 MVA for the 33-bus system and 100 MVA for the 57-bus and 118-bus systems. Each bus at every timestep is described by an 11-dimensional feature vector capturing local electrical conditions. The features include active and reactive load demand, power injections from the external grid, conventional generation, renewable generation, voltage magnitude, voltage angle, and the normalized bus degree representing local connectivity. The power quantities are converted to per-unit values using the system base power. Voltage magnitude is mean-centered around 1.0 p.u., and the node degree is normalized by the maximum degree in the network. To emulate measurement uncertainty, Gaussian sensor noise is added to the primary electrical variables during data generation. Ground-truth targets are obtained from AC power flow solutions computed in Pandapower. Although the full target vector contains multiple electrical quantities, the graph neural networks are trained to predict only voltage magnitude deviation and voltage angle for each bus. Temporal operating conditions are generated using synthetic load and renewable generation profiles. Load demand follows a 24-hour "camel-shaped" curve with morning and evening peaks, with an additional $\pm 5\%$ random perturbation to represent short-term variability. Solar generation follows a bell-shaped daytime curve. It is also affected by seasonal variation and stochastic cloud conditions. While wind generation follows a night-peaking coastal profile governed by weather-dependent wind speed ranges. Weather evolution is also modeled using a first-order Markov chain. It enables temporally correlated renewable generation patterns. The data generator also simulates network reconfiguration events. In the radial IEEE 33-bus system, switching occurs by closing a tie-line and opening another line within the resulting loop to maintain radiality. For the meshed IEEE 57-bus and 118-bus systems, cycles are identified and a line is opened to create a temporary tie-line before reconfiguration. Each switching configuration is assigned a unique topology identifier, and the corresponding bus admittance matrix (Y_{bus}) is recomputed. Before training, the generated data undergo preprocessing. Raw simulation outputs are aggregated, converted to per-unit values, and split chronologically into training (70%), validation (15%), and testing (15%) subsets. The processed datasets, including node features, targets, topology identifiers, admittance matrices, and network connectivity information, are stored as PyTorch tensors to support efficient training and evaluation of the proposed models.

2.5.2 Model Architectures

The graph neural network architectures are implemented to evaluate spatiotemporal learning for AC optimal power flow prediction. The models are grouped into spatial-only and spatiotemporal categories and are implemented using PyTorch and PyTorch Geometric. Graph message passing is performed using the GCNConv operator. The node interactions occur strictly along the physical transmission network defined by the grid topology. The spatial-only models process a single system snapshot with input tensor shape (B, N, F). Where B is the batch size, N the number of buses, and F the number of node features. The standardGCN model uses a fixed adjacency matrix corresponding to the base network topology. The dynamicGCN extends this model by updating the adjacency matrix for each sample to reflect the current switching configuration or contingency state. The physics informed graph convolution network (PIGCN) architecture uses the same structure as dynamicGCN but applies physics-informed loss function. This combines equation (30) with penalties enforcing equations (31) – (34). These models follow a successive spatiotemporal pipeline. The spatiotemporal models operate on sequential inputs with tensor shape (B, T, N, F), where T denotes the temporal window length. For each timestep, graph convolution extracts spatial embeddings equation (24). These are stacked across time and reshaped into per-node sequences. The sequences are then processed using equation (25) or (27) to capture temporal dependencies caused by renewable variability and load dynamics. Two variants are considered: physics informed graph convolution long short-term memory (PIGCLSTM) and physics informed graph convolution gated recurrent unit (PIGCGRU). To mitigate oversmoothing in deeper graph networks equation (28) is applied. This introduces two residual variants: physics-informed residual graph convolutional long short-term memory network (PIResnetGCLSTM) and physics-informed residual graph convolutional gated recurrent unit network (PIResnetGCGRU). The residual graph convolution blocks consist of two GCNConv layers with skip connections. This allows stable training of deeper spatial representations. For all spatiotemporal models, the recurrent layer outputs the hidden representation at the final timestep. The output is mapped through a fully connected layer to predict the network state as expressed in equation (26). The shape of the final output is presented as $(B, N, 2)$. The voltage magnitude deviation and voltage angle recorded as output at each bus.

2.5.3 Software environment and dependencies

The study is implemented in Python (version 3.10 or later) and trained in either GPU enabled through CUDA 12.x or CPU-only environments. The project repository is obtained using a standard Git workflow. Subsequently, a Python virtual environment is created and activated. This setup ensures compatibility across different hardware configurations and maintain consistent software dependencies. The core machine learning components are implemented using PyTorch (version 2.10.0). This provides tensor computation and automatic differentiation capabilities. PyTorch Geometric (version 2.7.0) is used to implement the graph neural network operations. The GCNConv operator is used for graph convolution. The training workflow, include training loops, callbacks, and logging utilities. Lightning (version 2.6.1) is applied to manage the training. Additionally, NetworkX, a dependency of Pandapower, is used for graph operations such as cycle detection during topology switching events. This software environment ensures efficient training of graph neural network models while supporting reproducible experimentation across different computational platforms. The power system simulations and benchmark network models are generated using Pandapower (version 2.14.0 or later). The Pandapower provides standard IEEE test systems and AC power flow solvers. Finally, Weights & Biases (version 0.25.0) is used to track the performance.

2.5.4 Configuration and hyperparameter settings

The training processes are controlled through configuration files. It defines parameters for data generation and model training. These parameters enable systematic experimentation across different operating conditions. The data generation is configured using configs/data_generation.yaml. A 10,008 timesteps dataset is generated which represents different operation conditions under uncertainty. The renewable penetration levels are given $[0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$ to evaluate system behavior from conventional operation to fully renewable scenarios. The grid reconfiguration events occur probabilistically with a switching rate of 10% per timestep. This allows the network topology to change dynamically during simulation. The system uncertainty is measured using gaussian noise to electrical variables. With standard deviations given as 0.005 p.u. for voltage magnitude, 0.01 p.u. for power injections, and 0.02 radians for voltage angles. Overall, 30% of buses are assumed to have PMU-quality measurements. This emulates realistic monitoring infrastructure. The renewable generation is constrained with practical operating limits of 0.98 maximum utilization coefficient. The weather weight vectors define initial probability distributions for solar and wind states. It ensures weather-driven renewable variability is modeled. In addition, model training parameters are specified in configs/training.yaml. The graph neural networks are trained using a batch size of 32. For spatiotemporal models, a temporal sequence length of four timesteps. Each bus is represented by 11 input features. The models predict voltage magnitude deviation and voltage angle as outputs per bus. Both the graph convolution layers and recurrent layers (LSTM or GRU) use a hidden dimension of 64. The spatial encoder consists of four GCN layers or residual blocks. The Adam optimizer with an initial learning rate of 0.001 is used optimize the model. The learning rate scheduling follows a ReduceLROnPlateau strategy. This strategy reduces the learning rate by a factor of 0.5 if the validation loss does not improve for 10 epochs. The training is limited to a maximum epoch of 200. A fixed random seed of 42 is used to ensure reproducibility. The training objective incorporates additional penalty terms for physics-informed models. These include weights for power balance $(\lambda = 0.1)$, voltage magnitude limits $(\lambda = 0.01)$, and branch capacity constraints $(\lambda = 0.01)$. Finally, a comparisons analysis is conducted with classical solvers.

2.5.5 Evaluation and benchmarking

The proposed models' performance is evaluated using a dedicated pipeline implemented in scripts/evaluate.py. Each trained model is tested on a held-out dataset and assessed along three primary dimensions. These include prediction accuracy, physical feasibility, and computational efficiency. The prediction accuracy is measured using MAE (as shown in equation (35)) between the predicted and ground-truth. This metric quantifies the model's ability to approximate the steady-state operating point of the power system. The physical feasibility is assessed by verifying whether the predicted system states satisfy key electrical constraints. Equation (37) measures constraint satisfaction rates. The constraints computed are active power balance (P), reactive power balance (Q), voltage magnitude limits (V), and branch flow capacity limits (S). The power balance violations are evaluated using a tolerance threshold of 0.5%. At the same time, voltage and branch constraints are checked against their respective operational limits. These metrics show how the predicted states adhere to the physical laws governing power system operation. The performance is evaluated by comparing the inference speed of the proposed GNN models with four classical solvers available in Pandapower. These include: Newton–Raphson (NR), Newton–Raphson with the Iwamoto multiplier (NRI), Gauss–Seidel (GS), and Backward/Forward Sweep (BFS). Multiple runtime trials are conducted to obtain stable timing measurements. The Test-Time Augmentation (TTA) approach is also used to analyze predictive uncertainty. The model performs inference across multiple perturbed inputs and the variance of the resulting predictions is computed. This variance provides an estimate of the model's sensitivity to input perturbations and serves as a measure of predictive uncertainty.

3. Result and Discussion

3.1 Data Profile Analysis of Power Systems with Renewable Integration

Figure 6–8 shows the data profile analysis of system behavior under high renewable penetration for the IEEE 33-bus, 57-bus, and 118-bus systems. The analysis considers demand–supply balance, generation mix, renewable generation variability, and voltage health across buses. The indicators provide insight into the operational feasibility and stability of renewable power systems. The demand–supply balance curves demonstrate daily load patterns for all systems. In Figure 6 peak demand occurs around 18:00 and demand varies between 1.1 MW and 3.8 MW. The generation profile closely follows the demand curve with reserve margin of 0.05–0.15 MW. A similar trend is observed in the Figure 7, where demand ranges from 380 MW to 1300 MW. In Figure 10, demand varies between 1300 MW and 4300 MW. The generation mix shows that renewable energy sources have the largest share. In the IEEE 33 bus system, renewable generation ranges between 1.2 MW and 3.5 MW. As displayed in Figure 8 and Figure 8, renewable output increases to 1200 MW and 4500 MW, respectively. The conventional generators and external grid support serve as balancing resources during periods of renewable variability. The analysis demonstrates that there is direct correlation between renewable generation capacity increases and power system variability. In IEEE 33 bus system, renewable output increases from 0 MW at 0% penetration to 9 MW at 100% penetration. In the IEEE 57 and IEEE 118 bus systems, maximum renewable outputs reach approximately 1300 MW and 3600 MW respectively. The overall distribution indicates that renewable generation increases with system size. Finally, the voltage analysis confirms that the network maintains stable operating conditions across all buses. For the IEEE 33 system, bus voltages remain within 0.98–1.01 p.u. The IEEE 57 system maintains voltages between 0.99–1.02 p.u. In the larger IEEE 118 system, voltages vary between 0.96–1.04 p.u., but remain within the acceptable limits of 0.9–1.1 p.u. These results indicate that even under high renewable penetration scenarios, the systems maintain adequate voltage regulation and operational stability.

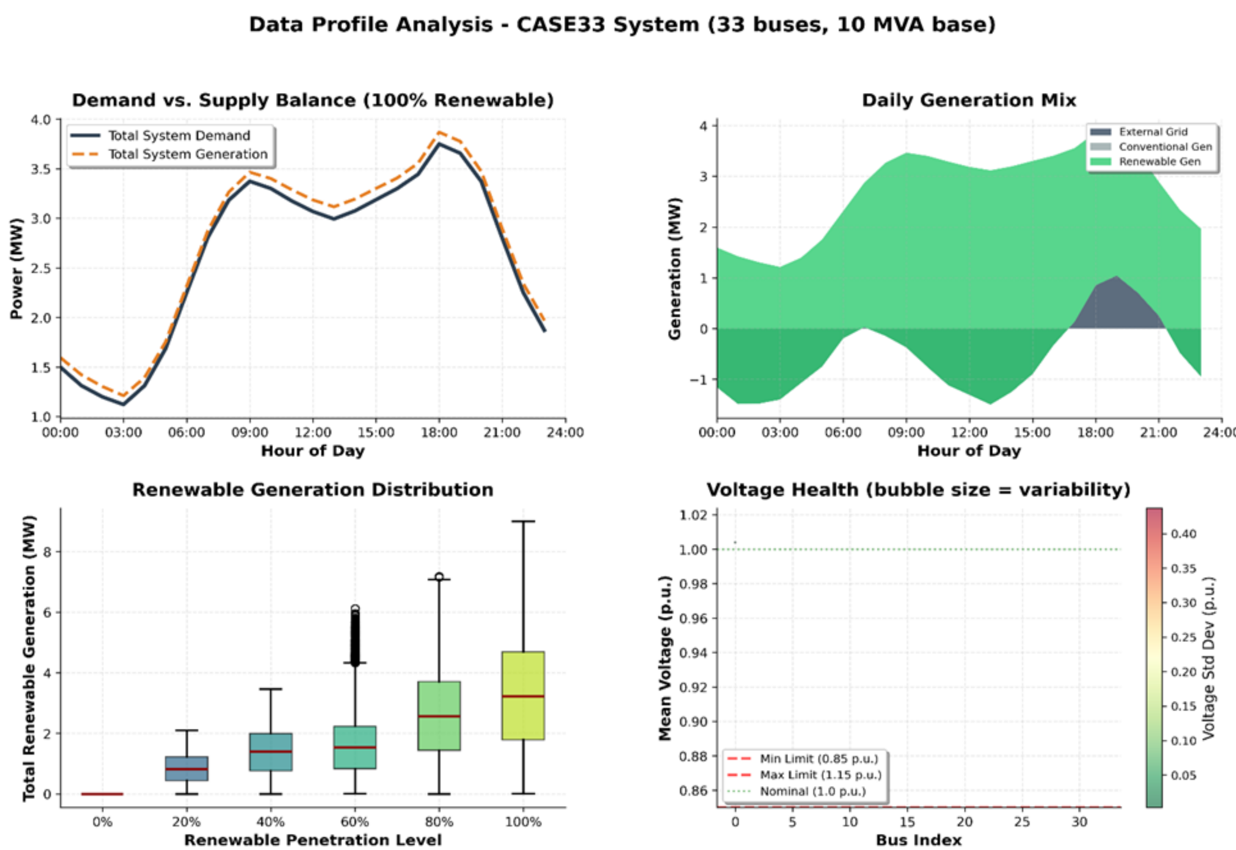

Figure 6 Data profile analysis for the IEEE 33-bus system

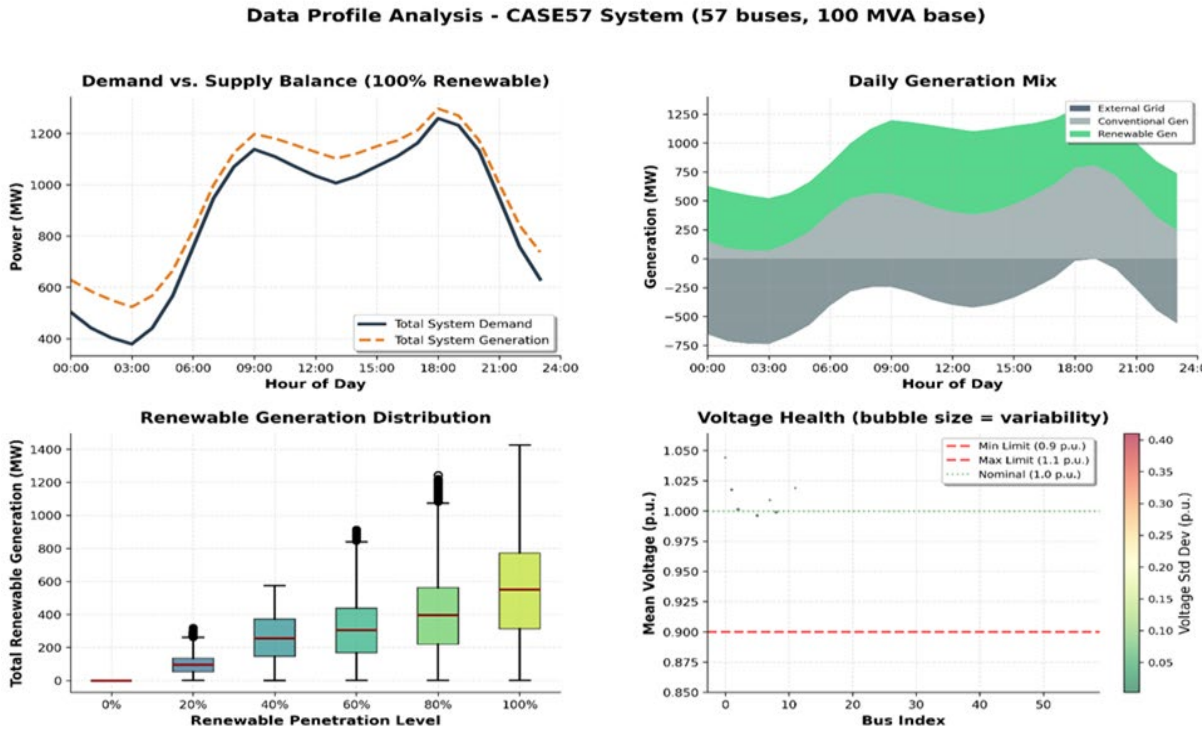

Figure 7 Data profile analysis for the IEEE 57-bus system

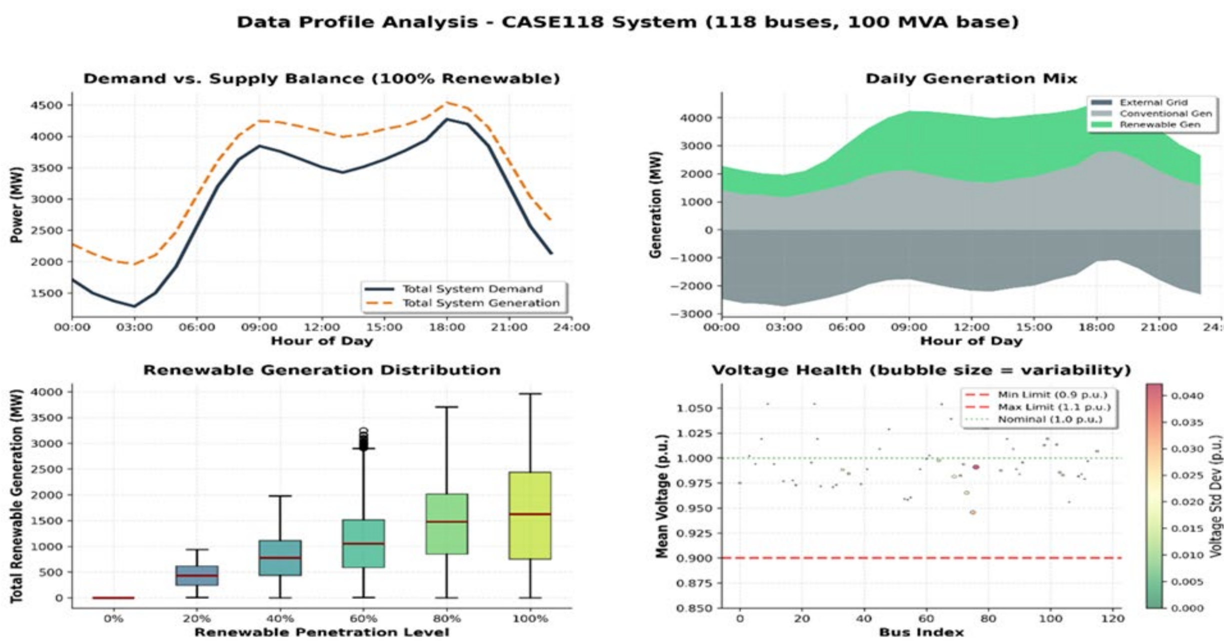

Figure 8 Data profile analysis for the IEEE 118-bus system

3.1.2 Impact of distributed renewable generation on power flow patterns and system efficiency

The results shown in Figure 9–11 highlight how distributed renewable generation affects power flow patterns and system efficiency. Figure 9 represents IEEE 33 bus system. The total system line losses decrease as renewable penetration increases from 0% to 60%. At 0% penetration, the mean active power loss is 0.11 MW, which decreases to about 0.06 MW at 60% penetration. This represents nearly a 45% reduction in losses. The 60% penetration level is the optimal operating point, where local renewable generation effectively supplies nearby loads. This reduces power transfer line losses. However, beyond 60% penetration, losses begin to increase, reaching 0.115 MW at 100% penetration. The increase is attributed to reverse power flow and network congestion. Which occur when excess distributed generation exceeds local demand. A different trend is observed in Figure 10. The minimum system losses are recorded at 0% renewable penetration. As renewable penetration increases, the losses rise, reaching around 94 MW at full penetration. This indicates that in larger transmission networks, excessive distributed generation significantly alter power flow patterns causing increased line loading and higher resistive losses. The wider shaded region represents ±1 standard deviation also indicates increased variability in losses as renewable penetration increases. In Figure 11 the impact of renewable penetration on losses becomes even more pronounced. The system exhibits the lowest losses of 183.41 MW at 0% penetration, which increase with higher renewable integration. At 100% penetration, losses reach 540 MW, representing nearly a threefold increase compared with the base case. This trend reflects the challenges of integrating high levels of distributed generation in large-scale transmission systems. The power flow reversals and longer transmission paths can lead to increased resistive losses. Across all three systems, the results demonstrate a consistent relationship between renewable penetration and network efficiency.

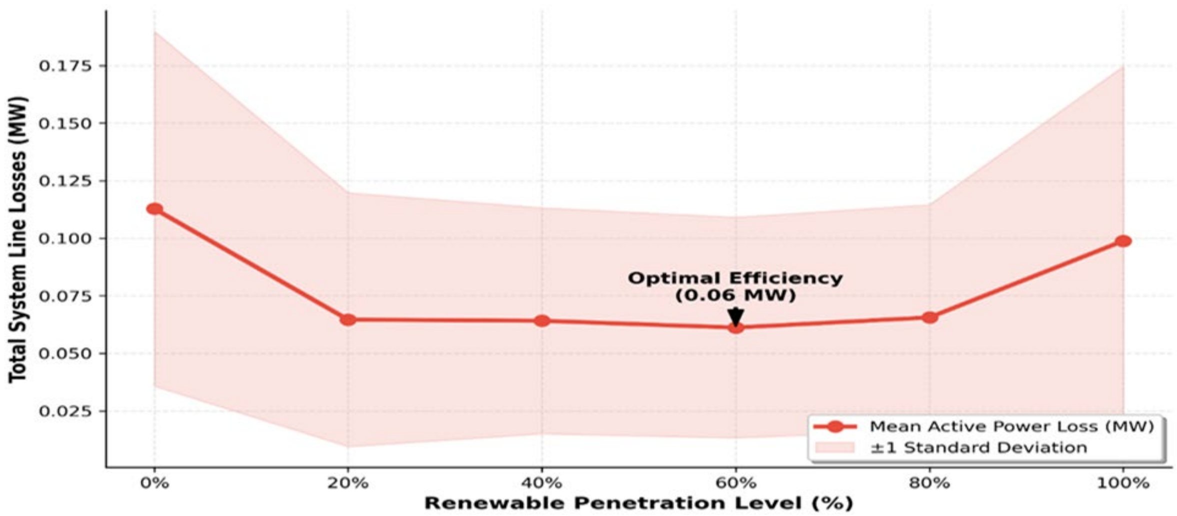

Figure 9 Impact of distributed renewable generation on power flow patterns and system efficiency for IEEE 33 bus system

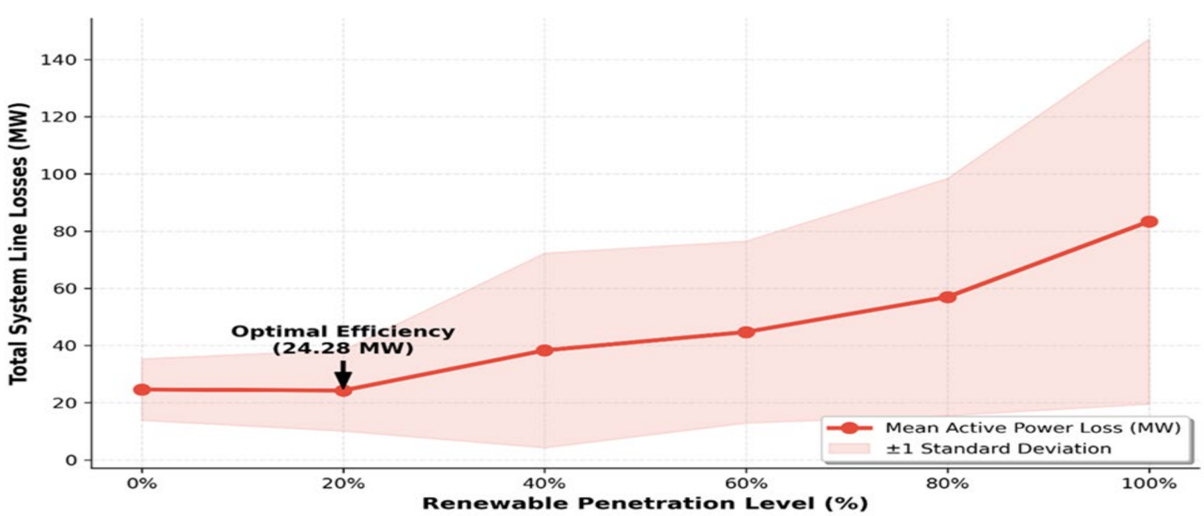

Figure 10 Impact of distributed renewable generation on power flow patterns and system efficiency for IEEE 57 bus system

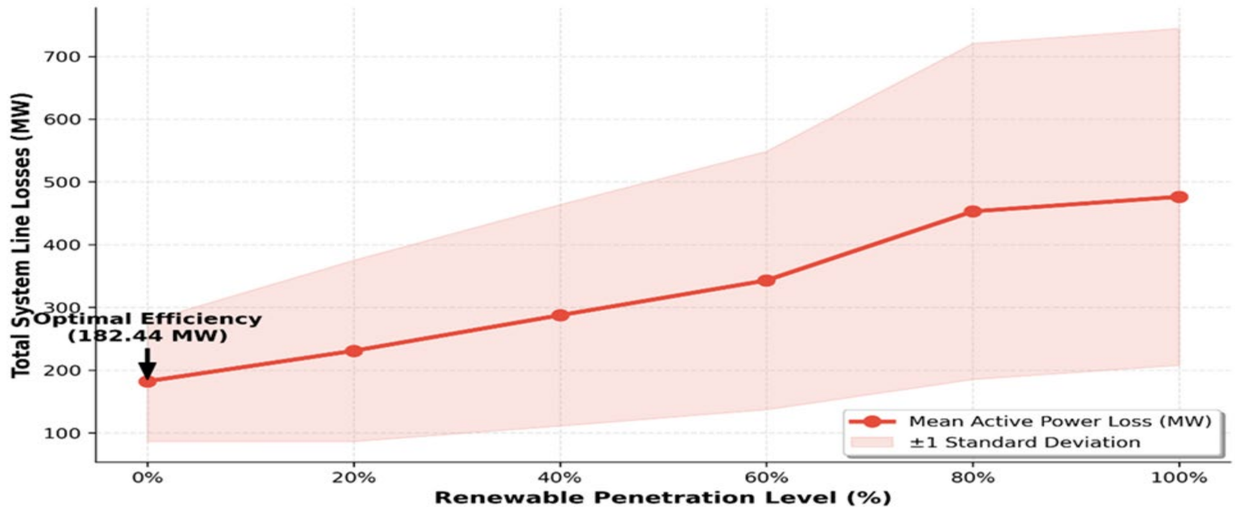

Figure 11 Impact of distributed renewable generation on power flow patterns and system efficiency for IEEE 118 bus system

3.1.3 Topology Stability and adaptability analysis for the IEEE bus systems

Reconfiguration is an operational strategy used to maintain system stability under varying renewable penetration levels. The results shown in Figure 12–14 give insights into topology adaptability and how transmission lines are reconfigured. Figure 12 indicate 55 rerouting events observed across the timeline in IEEE 33 bus system. Most distribution lines depict stable topology and maintain their base configuration. However, tie-line closures occur on higher-index distribution lines intermittently. The lines serve as backup paths for reliable power flow during disturbances or load fluctuations. Figure 13 exhibits 60 rerouting events for IEEE 57 bus system. It displays greater topology adaptation compared to the IEEE 33 network. The rerouting events are distributed across multiple transmission lines. As renewable penetration increases dynamic switching actions also increase. The distribution of these events suggests that alternative transmission paths is used decongest or balance power flow. In Figure 14, the IEEE 118 bus system demonstrates the highest level of topology dynamics. It records 66 rerouting events during the simulation period. The events are distributed across a larger number of transmission lines. It reflects the complexity of the larger transmission network. Which show that the system depends on adaptive network reconfiguration. It is therefore verified that stable power flow under varying operational conditions is maintained. Furthermore, stability resolution strategy analysis is performed to show how the system responds to potential instability conditions. The majority of timesteps fall under strict normal operating conditions for all three networks. However, smaller portion of timesteps requires strict switching actions, which indicate topology reconfiguration events. Also, the proportion switching actions increase with increase in renewable penetration (0% to 100%). It is also evident that generator trips and restoration events appear at higher penetration levels in Figure 14.

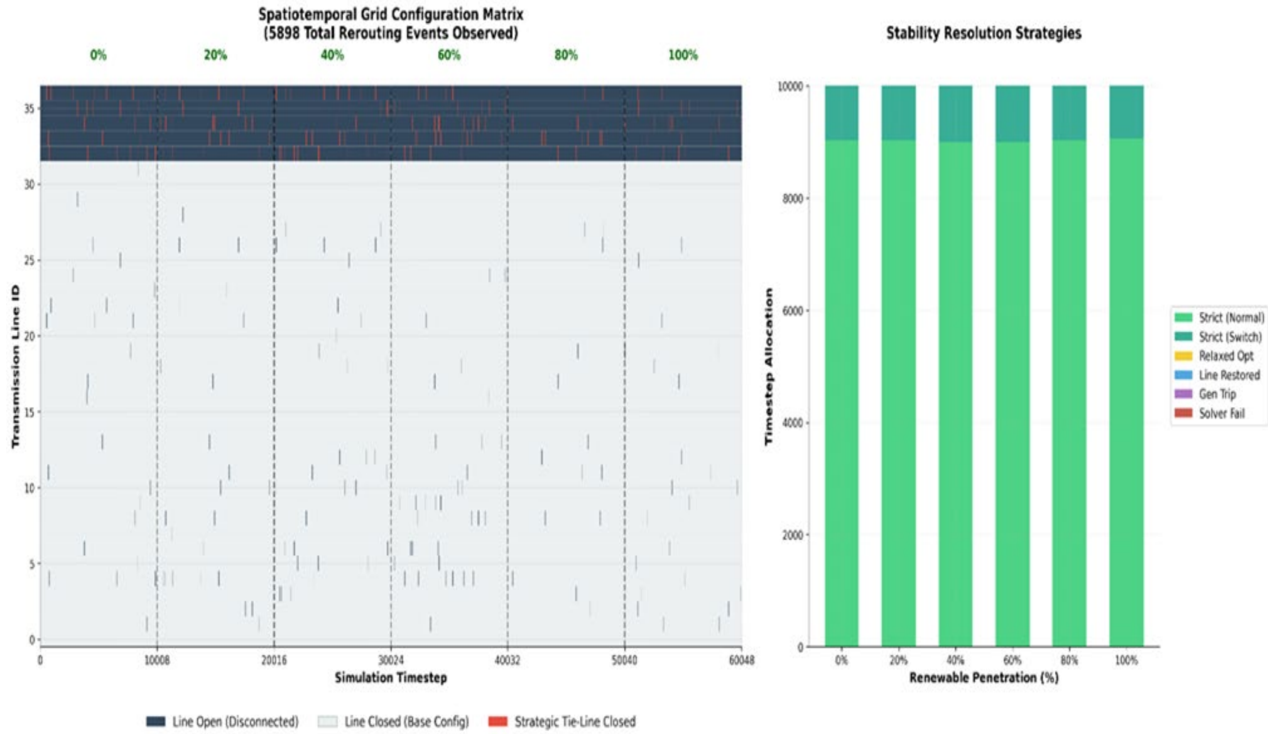

Figure 12 Impact of renewable penetration on grid topology and operational stability strategies for the IEEE 33-bus system

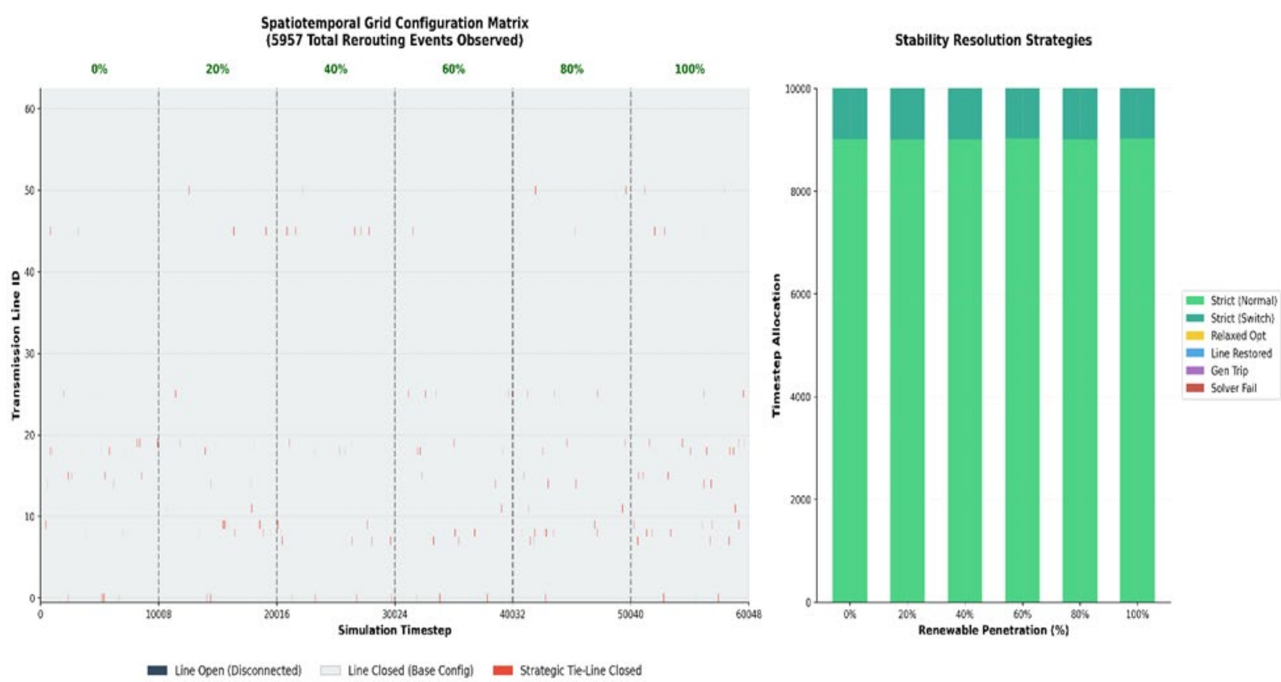

Figure 13 Impact of renewable penetration on grid topology and operational stability strategies for the IEEE 57-bus system

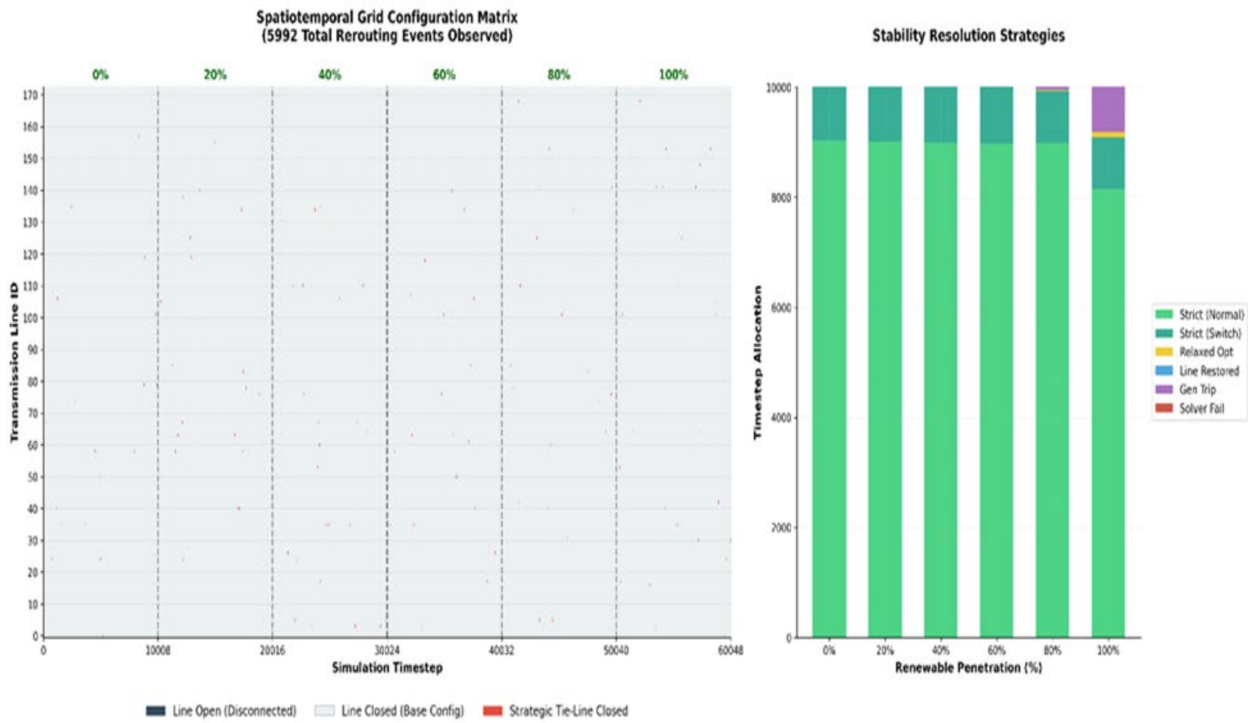

Figure 14 Impact of renewable penetration on grid topology and operational stability strategies for the IEEE 118-bus system

3.1.5 Temporal continuity analysis for the IEEE 33-bus, 57-bus, and 118-bus systems

The temporal continuity analysis is for the chronological integrity of the dataset across training, validation, and test splits. Figure 15-17 illustrate the evolution of two key variables over time. It ensures that the dataset maintains realistic temporal dynamics. It shows mean voltage magnitude deviation from the nominal value (1.0 p.u.) and mean active load (P_load). Figure 15 show the training segment covers the majority of the timeline, followed by shorter validation and test intervals. The mean voltage deviation fluctuates within $-0.06\ to\ 0.04\ p.u.$ The mean load profile exhibits a clear cyclical pattern, and reflect periodic load variations typical of distribution systems. These patterns continue smoothly across the train–validation and validation–test boundaries. This ensures that dataset preserves chronological consistency without abrupt statistical changes. A similar behavior is observed in Figure 16 (IEEE 57 bus system), although the magnitude of variations is slightly larger due to the increased network size and operational complexity. The voltage deviation ranges between $-0.10\ and\ 0.05\ p.u.$ It reflects greater variability in voltage profiles compared with the smaller system. The load profile shows periodic oscillations, consistent with operational demand cycles. The statistical properties of both voltage deviation and load remain consistent across the train, validation, and test segments. This confirms that the dataset splits maintain temporal continuity. The voltage deviation is relatively smaller in Figure 17 (IEEE 118 bus system), remaining mostly within $-0.03\ to\ 0$ p.u. It shows that the transmission network maintains tighter voltage regulation. The load profile displays more pronounced amplitude variations due to the larger aggregate demand of the system. As with the other systems, the splitting strategy preserves the natural time-series characteristics of the data. The persistence of similar statistical patterns across the splits demonstrates that the dataset is representative and suitable for training time-dependent learning models. Overall, the temporal continuity analysis confirms that the data preparation process preserves the physical and temporal characteristics of power system operations.

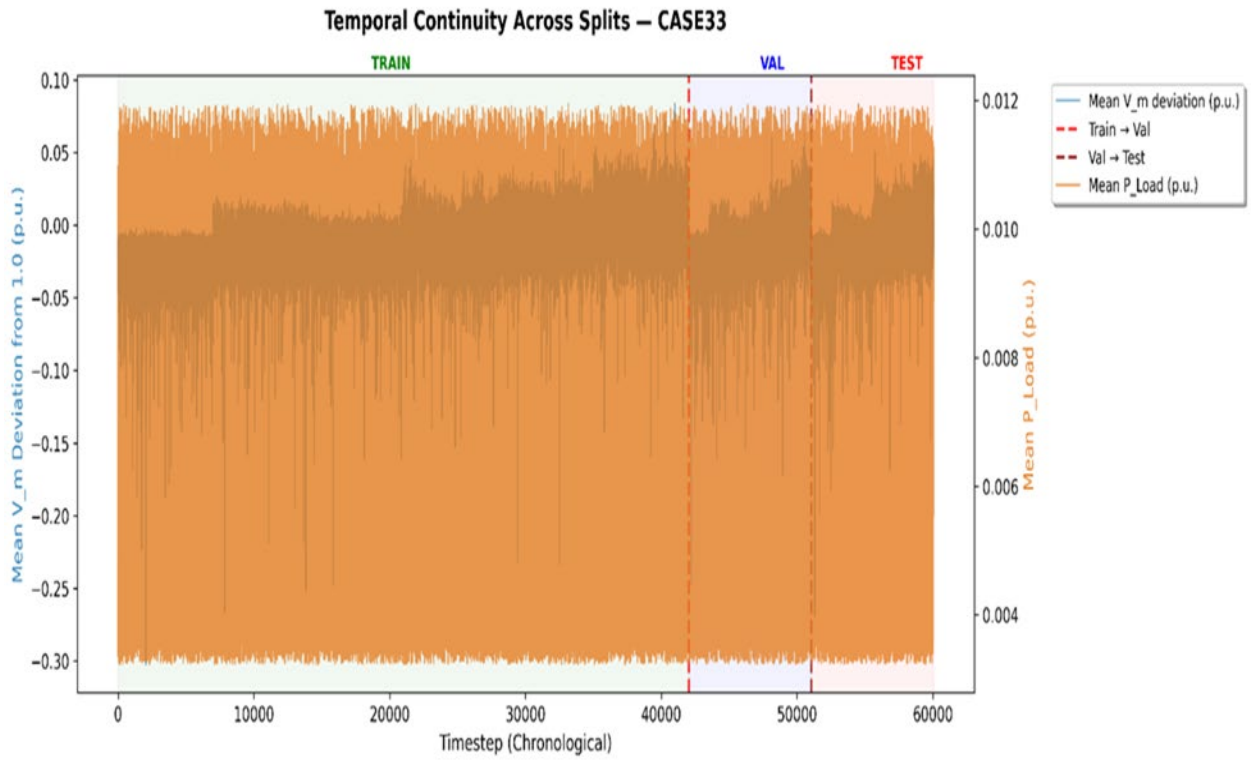

Figure 15 Chronological integrity of the dataset across training, validation, and test splits for IEEE 33-bus system

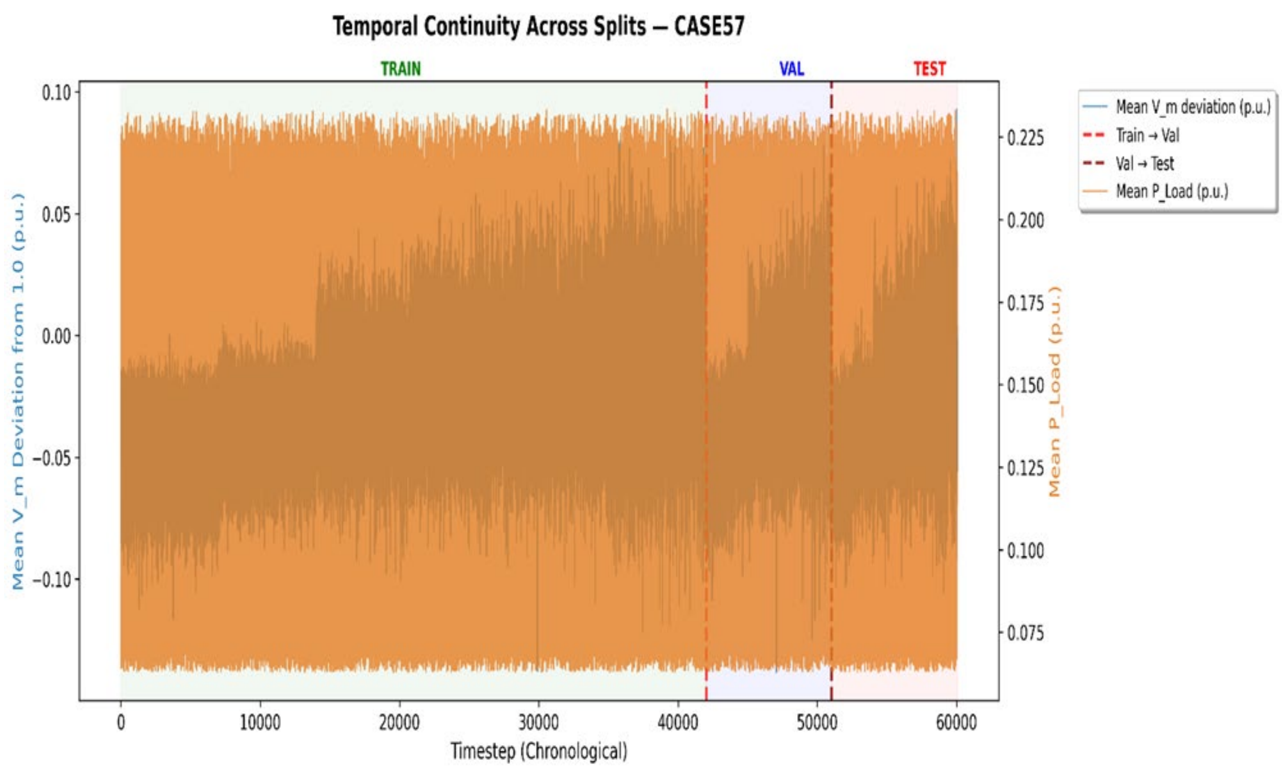

Figure 16 Chronological integrity of the dataset across training, validation, and test splits for IEEE 57-bus system

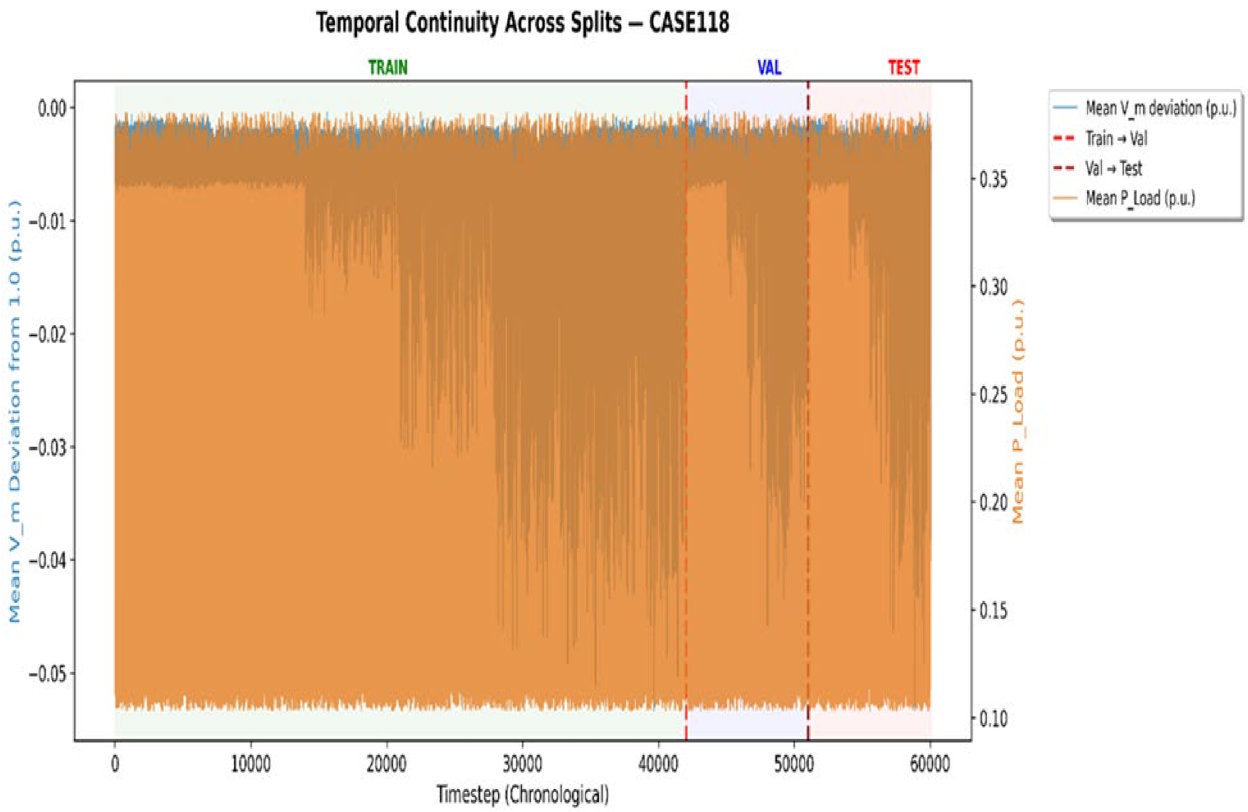

Figure 17 Chronological integrity of the dataset across training, validation, and test splits for IEEE 118-bus system

3.2 Computational performance of the proposed Graph Neural Network

Figure 18-20 presents the computational performance of traditional power flow solvers with the proposed Graph Neural Network (GNN) models. The analysis considers the average inference time per sample on a logarithmic scale. The classical numerical solvers include Newton-Raphson (NR), NR with Iwamoto acceleration, and Gauss-Seidel. The experimental results in Figure 18 exhibit higher computation times for traditional solvers compared with the proposed GNN models. The GNN-based models achieve faster inference speeds. The standardGCN model achieves an inference time of 0.05 ms, which corresponds to a speedup of 10135 times compared with Newton-Raphson. Even the more complex architectures, such as PIResnetGCGRU and PIResnetGCLSTM, maintain inference times below $10\ ms$. It provides 75 to 77 times speedups over the NR solver. A similar trend is observed in Figure 19 where the Newton-Raphson, NR+Iwamoto and Gauss-Seidel solvers requires $122.9\ ms, 15.9\ ms$ and $1175.0\ ms$ respectively. But the GNN models still produce faster inference. The standardGCN architecture achieves a speedup of 3543 times, while dynamicGCN and PIGCN provide speedups of around 206 and 208 times respectively. The physics-informed models (PIGCLSTM and PIGCGRU), maintain inference times below $10\ ms$. It achieves 155 to 154 times speed improvements relative to Newton-Raphson. The IEEE 118 system in Figure 20 show that the traditional solvers is computationally expensive. The standardGCN model achieves an inference speedup of 419 times compared with Newton-Raphson. Both dynamicGCN and PIGCN achieve speedups of around 238 and 262 times respectively. It is also observed physics-informed and residual architectures maintain inference times 50 - 24 times speed improvements over the NR solver. Across all three systems, GNN-based models provide substantial improvements in computational efficiency compared with conventional iterative power flow solvers. The computational time makes them suitable for large-scale and time-sensitive power system operations.

Figure 18 Efficiency analysis the proposed Graph Neural Network models across the IEEE 33-bus systems

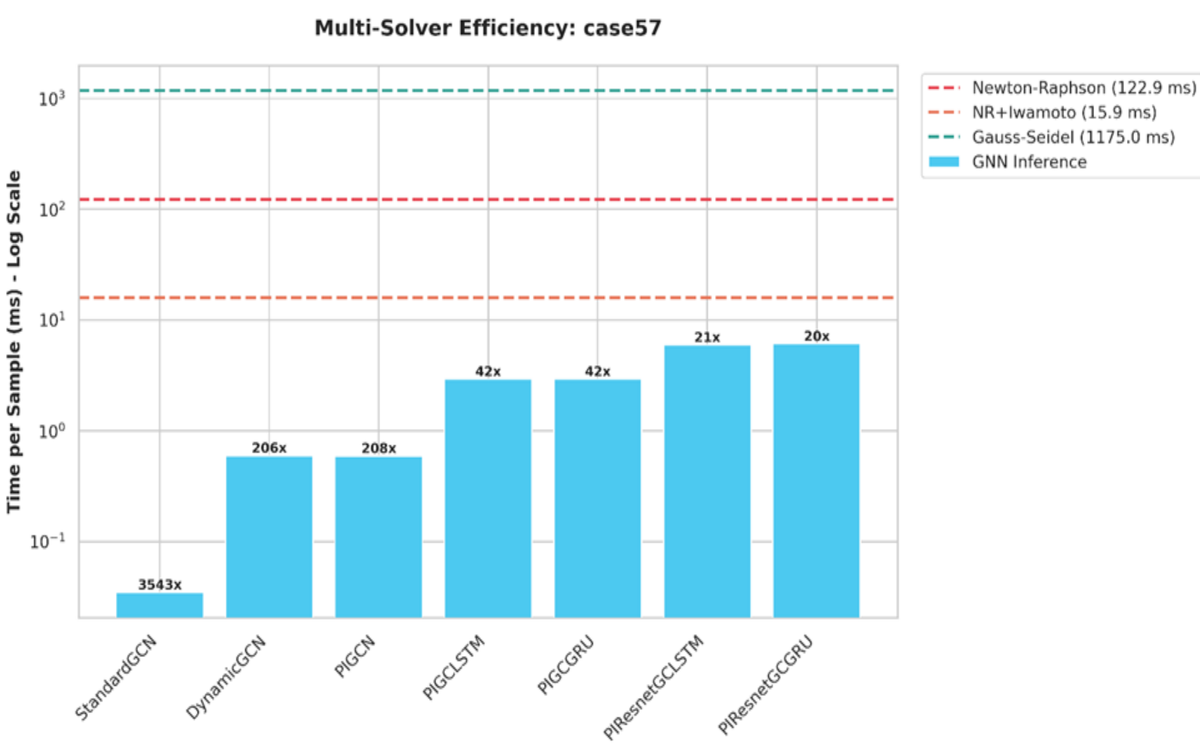

Figure 19 Efficiency analysis the proposed Graph Neural Network models across the IEEE 57-bus systems.

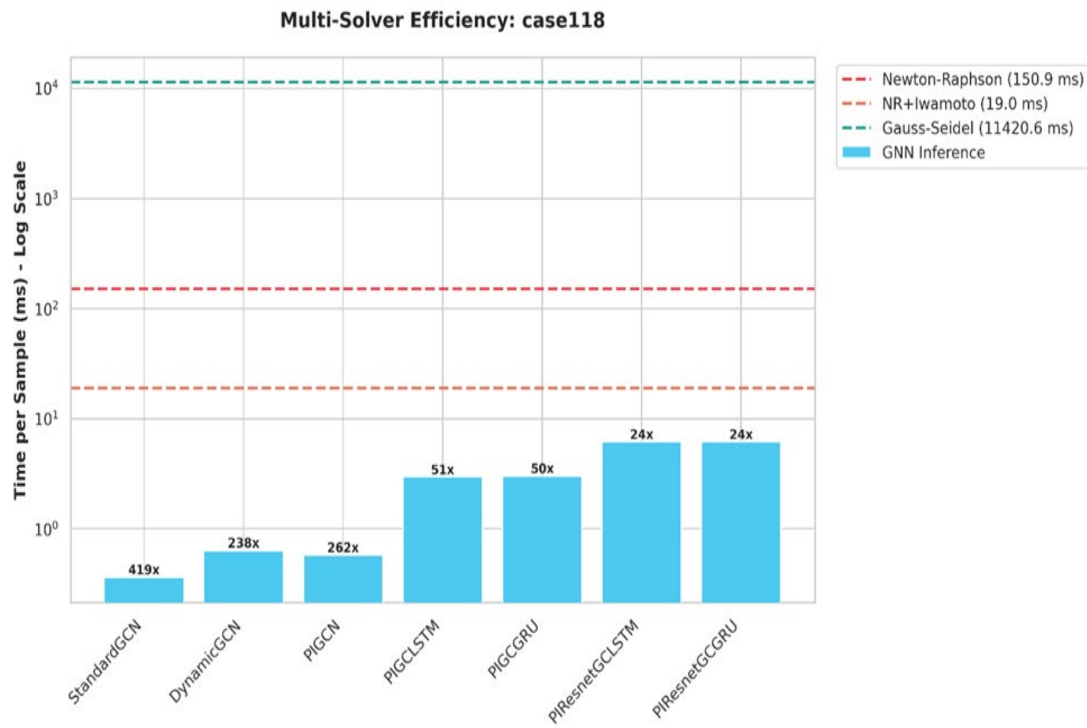

Figure 20 Efficiency analysis of the proposed Graph Neural Network (GNN) models across the IEEE 118-bus systems.

3.3 Physics Gap and Constraint Satisfaction Analysis

The physics gap analysis evaluates how well different GNN architectures satisfy fundamental power system physical constraints across the IEEE 33-bus, 57-bus, and 118-bus systems. The experimental results indicated that the branch constraints exhibit the highest satisfaction rate of 100%. It demonstrates that the proposed GNN models consistently respect transmission line limits. It is also confirmed that branch flows correlate with network topology and are captured by graph-based models. Similarly, the voltage constraint satisfaction remains very high (100%) across different models and power systems. However, a physics gap emerges in power balance constraints, particularly for active and reactive power. Figure 21 depicts that both standardGCN and dynamicGCN achieve almost perfect voltage and branch compliance. However, the power balance satisfaction rate was low with active power constraint satisfaction below 5% and reactive power below 10%. It is therefore collaborated that the non-physics informed models perform well with network structure. But faces challenges to fully capture the nonlinear relationships governing power injections and flows. On the other hand, the physics-informed models (PIGCN, PIGCLSTM, and PIGCGRU) improve these metrics. It increases active power satisfaction from 23 to 40% and reactive power satisfaction from 38 to 62%. In comparison Figure 22 display the same pattern with voltage constraint satisfaction ranging from 73% to 83%. And the reactive power satisfaction from 2 to 18% depending on the architecture. The residual models again outperform purely data-driven and physics informed architectures. However, the IEEE 118-bus system in Figure 23 reveals the strongest physics gap. Although branch and voltage constraints remain nearly perfectly satisfied, reactive power constraint satisfaction drops for most models. For instance, PIGCLSTM and PIGCGRU achieve only around 5% and 4% reactive power satisfaction respectively. This reflects the difficulty of learning high-dimensional nonlinear power flow relationships in large-scale systems. There is therefore the need for adaptive improvement to make the Physics-informed models scalable across different systems. In summary, physics-based regularization and residual learning architectures reduces this physics gap and offer the most practical pathway for accurate power system prediction.

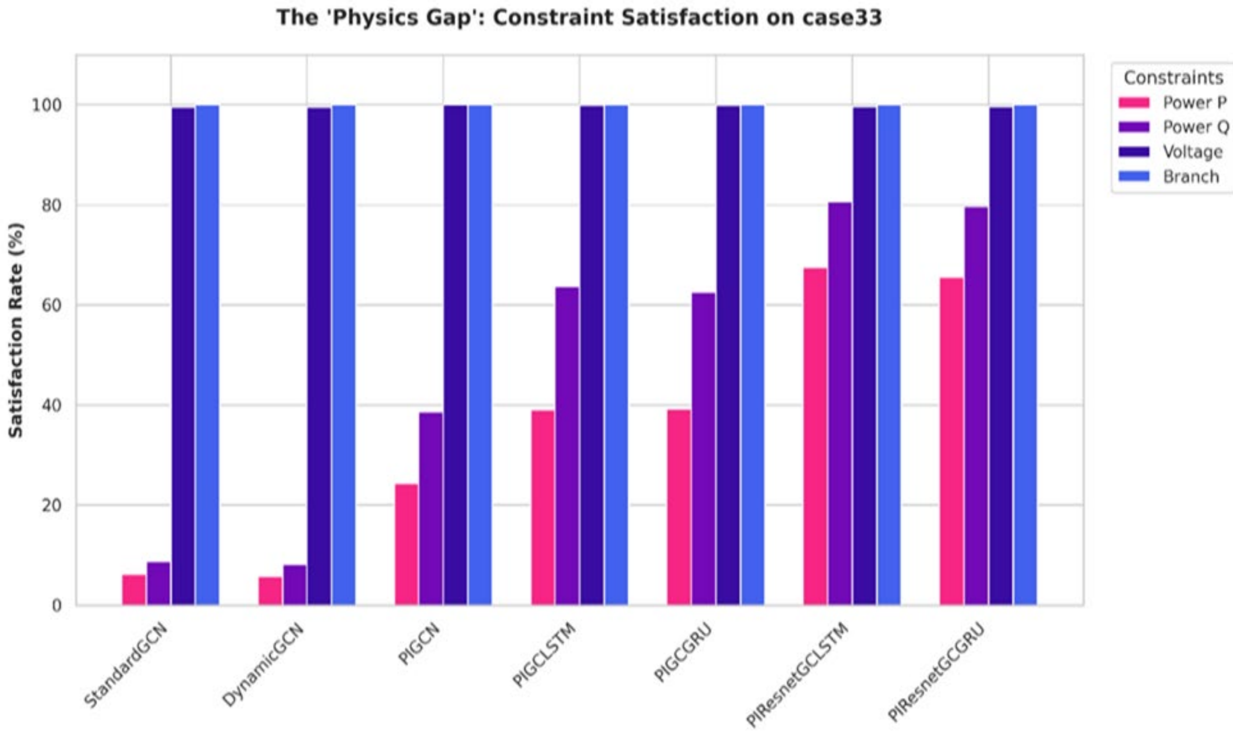

Figure 21 Analysis of satisfaction of power system physical constraints across the IEEE 33-bus-bus systems

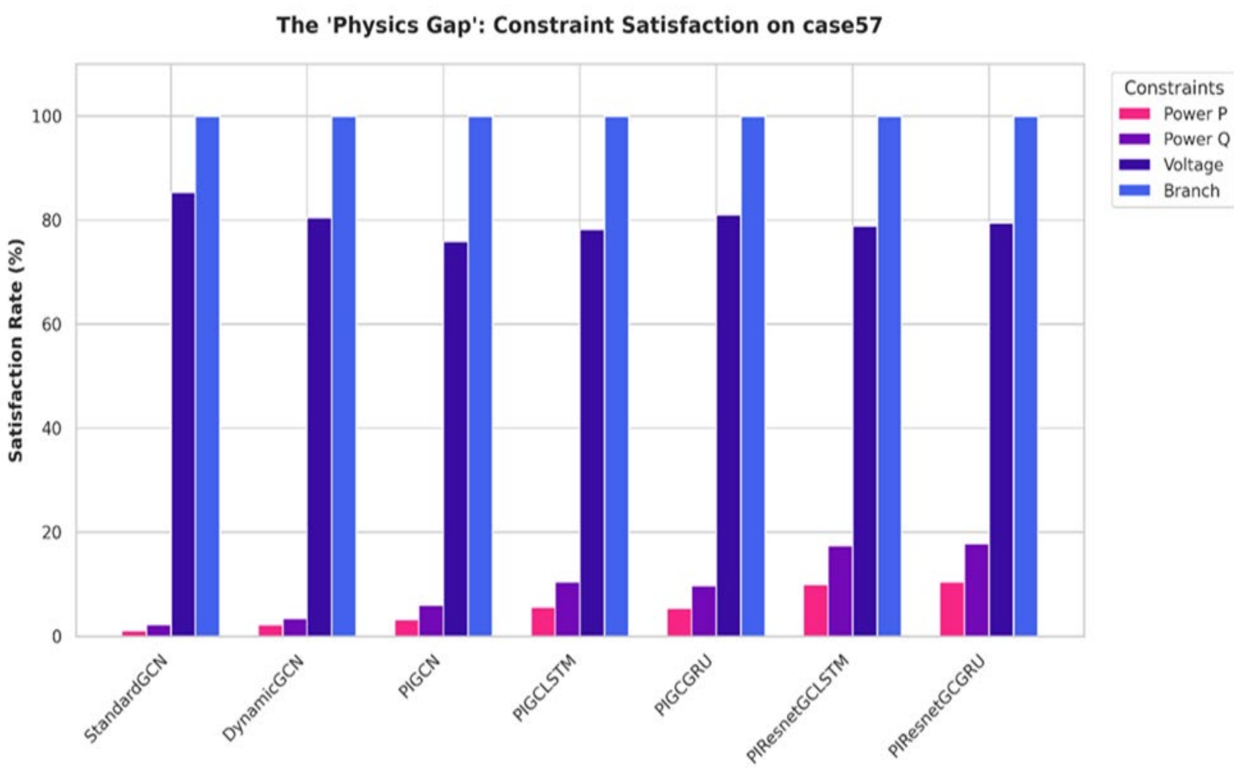

Figure 22 Analysis of satisfaction of power system physical constraints across the IEEE 57-bus-bus systems.

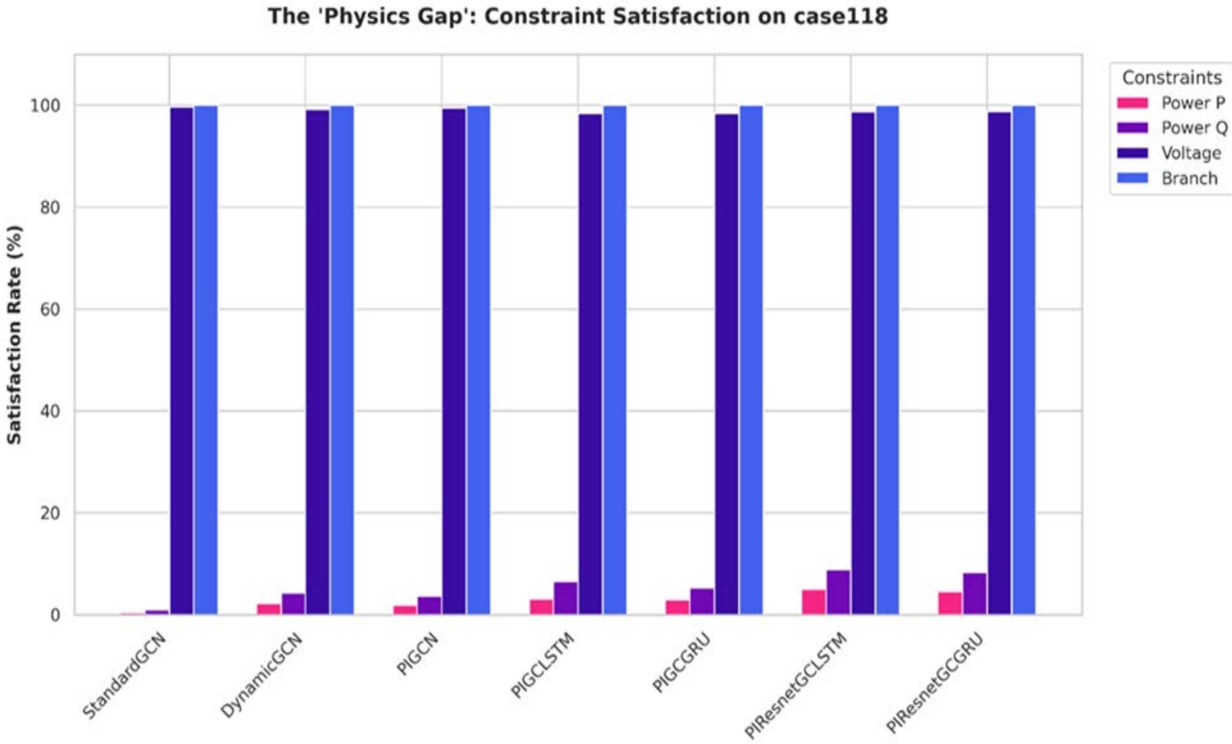

Figure 23 Analysis of satisfaction of power system physical constraints across the IEEE 118-bus-bus systems

3.4 Analysis of the predictive performance of the proposed GNN-based models

The solver comparison evaluates how closely the proposed GNN architectures approximate the Newton-Raphson (NR) power flow solution as the ground truth. Figure 24-26 compares the performance of the models on different bus systems, focusing on the MAE of voltage magnitude and voltage angle. The classical numerical solvers considered include NR with Iwamoto acceleration, Gauss-Seidel, and the backward/forward sweep method (for the radial 33-bus system). The results show extremely small deviations from the NR ground truth on the order of 10^{-9} to 10^{-8} p.u. The results confirm that traditional iterative solvers achieve good accuracy when compared with the GNN-based models. The voltage magnitude is 10^{-2} p.u. and voltage angles of 10^{-2} to 10^{-3} radians. However, the accuracy GNN-based models are acceptable from an operational perspective. It is evident that the physics-informed architectures like PIGCN, PIGCLSTM, and PIGCGRU produce higher MAE compared to non-physics informed GNN models. Figure 24-26 shows that physics informed learning process improves feasibility at the expense of data optimality. The non-physics informed GNN architectures maximize prediction accuracy and feasibility. The residual physics-informed architectures on the other hand show improved performance in both voltage angle and voltage magnitude estimation. Lastly, As the network size increases from the 33-bus system to the 118-bus system, the GNN models show an increase in prediction error. But it maintains good computational efficiency. The results show inference speedups range from tens to several thousand times faster than traditional solvers. The findings confirm the practical value of GNN-based approaches for applications such as real-time grid monitoring, and rapid contingency analysis. It suggests that graph-based learning captures the structural dependencies of power networks effectively.

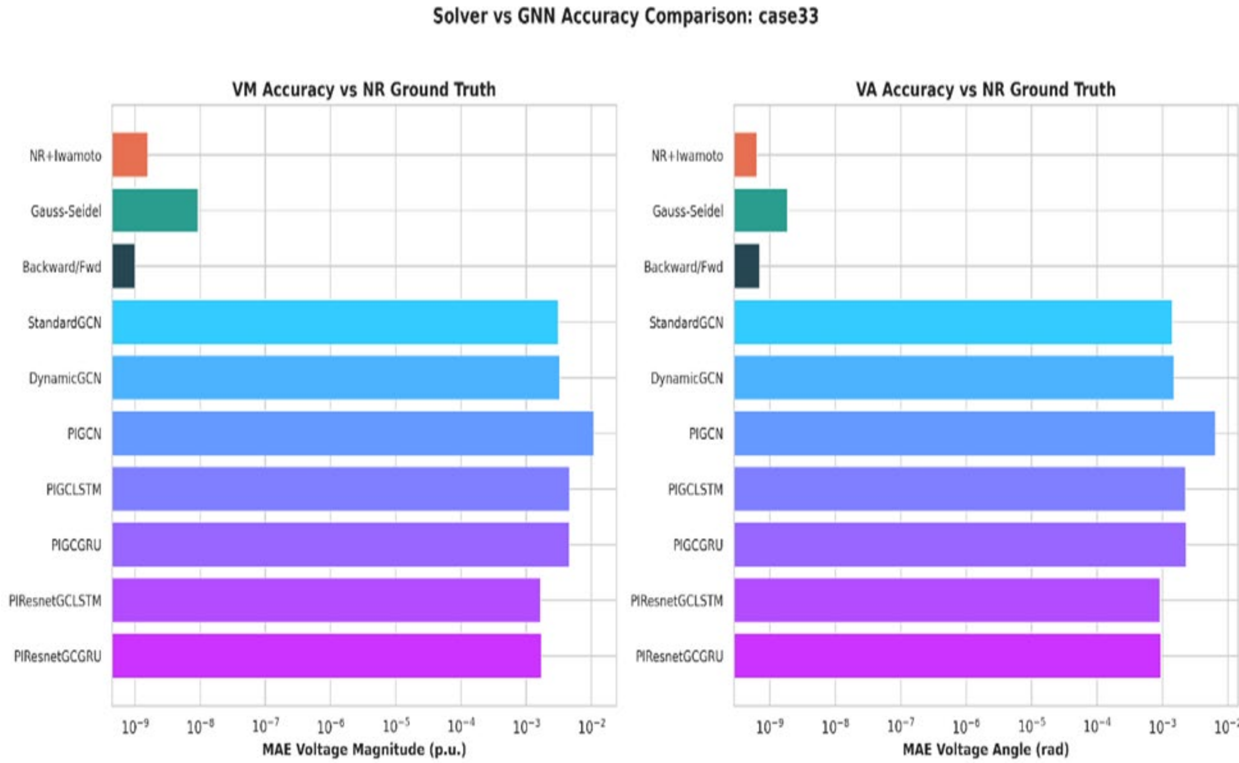

Figure 24 Comparison of the performance of the proposed models with classical numerical solvers for IEEE 33 bus system

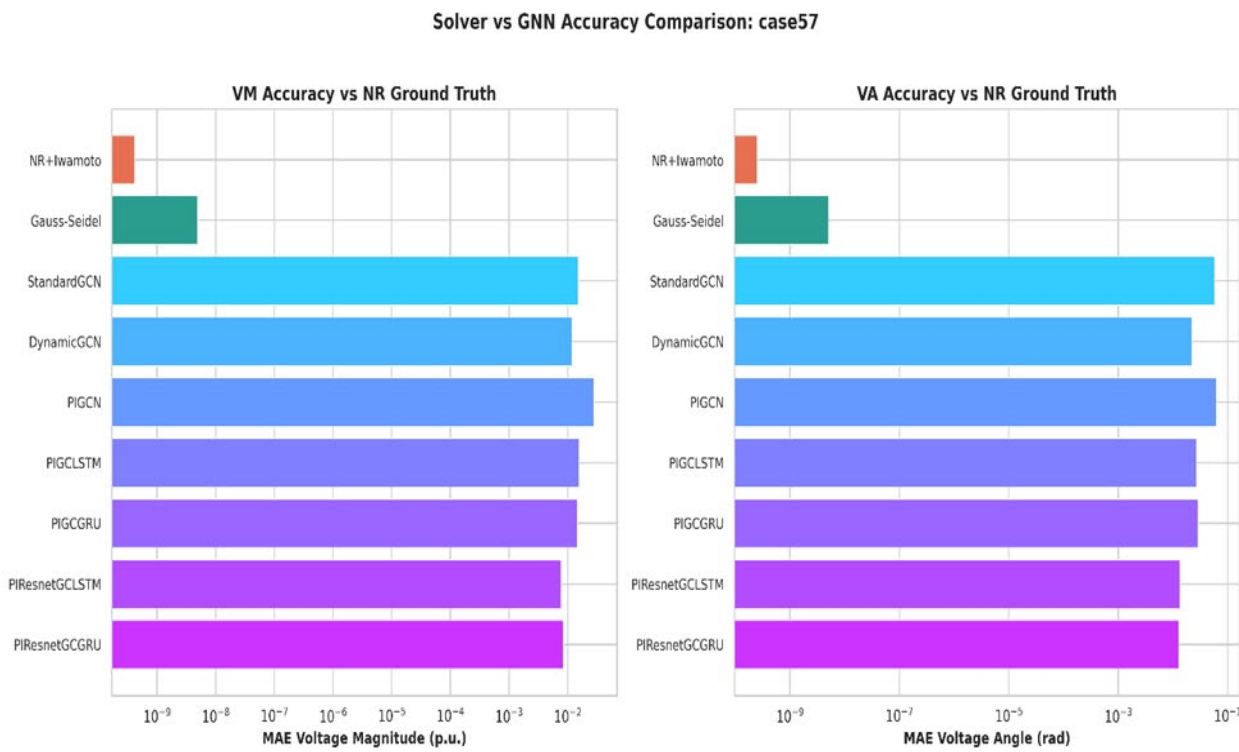

Figure 25 Comparison of the performance of the proposed models with classical numerical solvers for IEEE 57 bus system

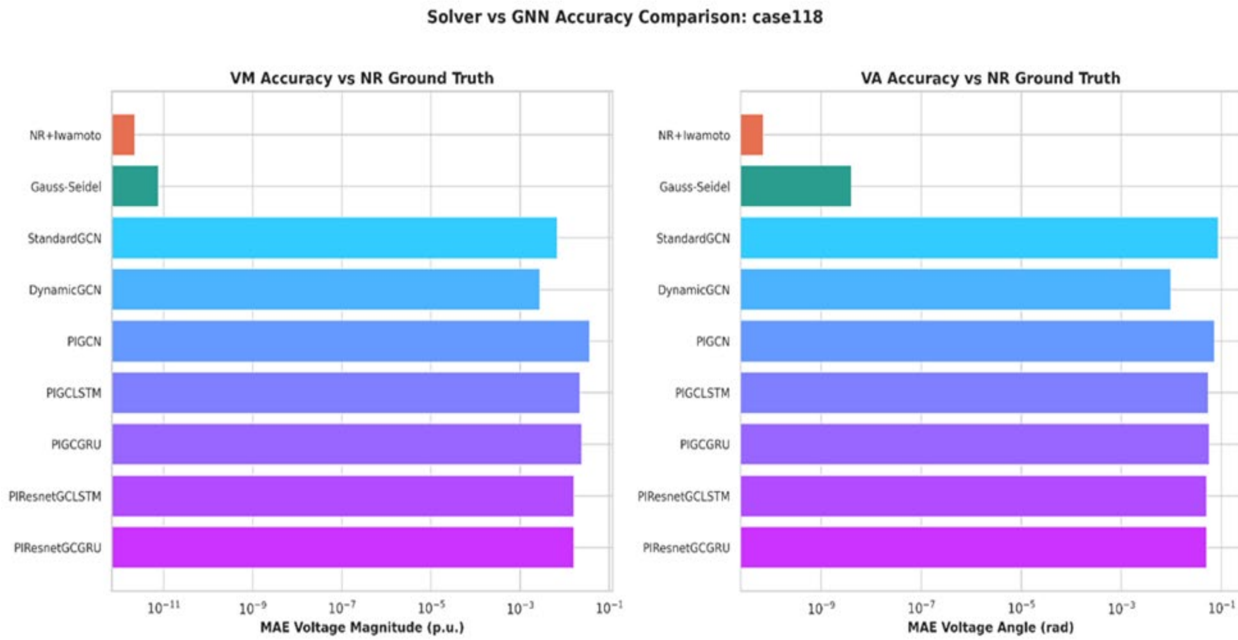

Figure 26 Comparison of the performance of the proposed models with classical numerical solvers for IEEE 118 bus system

3.5 Spatiotemporal uncertainty analysis of different renewable penetration levels for the IEEE bus systems

Tables 1–3 compare the spatiotemporal uncertainty performance of seven models on the IEEE 33-bus, IEEE 57-bus, and IEEE 118-bus systems under renewable penetration levels from 0% to 100%. The results show the comparison of different methods with the proposed PIStGNN. As shown in Table 1, PIResnetGCGRU provides the best performance at all penetration levels. Followed by PIResnetGCLSTM model and the least is the non-spatiotemporal models (StandardGCN, DynamicGCN, and PIGCN). Both PIGCLSTM and PIGCGRU remain in the range of 0.00175–0.00201. The results indicate that the residual graph recurrent structure improves robustness to renewable uncertainty. For the IEEE 57-bus system, the performance changes slightly, with PIResnetGCLSTM emerging as the best model followed by PIResnetGCGRU across all penetration levels. Its uncertainty increases from 0.002876 at 0% penetration to 0.004857 at 100% as shown in Table 2. The predictive uncertainty of PIGCLSTM and PIGCGRU increase from around 0.00410 at 0% to about 0.00635 at 100%. The StandardGCN perform worse with uncertainty value of 0.0197. It is demonstrated that the residual LSTM architecture records the strongest resilience as the system size and penetration increase. The results in Table 3 differ somewhat from the smaller IEEE bus systems. The PIGCGRU model achieves the lowest uncertainty across the penetration range. However, the uncertainty of PIResnetGCLSTM grows with increasing renewable penetration (0% penetration with 0.001190). This suggests that the simpler recurrent graph generalize better than the deeper residual variants in IEEE 118 bus system. In addition, the recurrent graph-based models outperform the pure graph convolution across the three IEEE bus systems. The residual variants achieve the best overall results in both 33-bus and 57-bus cases. In conclusion, the effectiveness of deeper residual spatiotemporal modeling depends on system scale and network complexity. The uncertainty decreases or remains stable as penetration increases for IEEE 33-bus system. For the 57-bus system uncertainty increase with higher penetration. It indicates that larger systems become more sensitive to renewable variability. These patterns demonstrate that penetration level alone does not determine performance; rather, it depends on the interaction between system scale and model architecture.

Table 1 Spatiotemporal uncertainty analysis of different renewable penetration levels for the IEEE 33 bus systems

Model	0%	20%	40%	60%	80%	100%
StandardGCN	0.003032681	0.002785789	0.002597779	0.002731551	0.002577493	0.002648013
DynamicGCN	0.003830805	0.003528686	0.003457917	0.003348494	0.003222336	0.003160034
PIGCN	0.005886518	0.004809505	0.004390091	0.004315754	0.003865049	0.003642302
PIGCLSTM	0.00189661	0.001772686	0.001755097	0.00177819	0.001872644	0.001947111
PIGCGRU	0.002005364	0.001845345	0.001860839	0.001840169	0.001905067	0.001979237
PIResnetGCLSTM	0.001259574	0.001161289	0.001171452	0.001136871	0.001127957	0.001180707
PIResnetGCGRU	0.00119519	0.001104019	0.001080923	0.001063786	0.001056978	0.001125974

Table 2 Spatiotemporal uncertainty analysis of different renewable penetration levels for the IEEE 57 bus systems

Model	0%	20%	40%	60%	80%	100%
StandardGCN	0.019671649	0.019216163	0.018306887	0.018294893	0.018035078	0.018155085
DynamicGCN	0.007793878	0.007181156	0.007718003	0.007857564	0.008339541	0.009399054
PIGCN	0.011133124	0.010479695	0.009013533	0.009420711	0.009617687	0.009662551
PIGCLSTM	0.004101407	0.004054913	0.004767252	0.004914432	0.005216572	0.006147429
PIGCGRU	0.004202176	0.004348957	0.005088652	0.005167923	0.005453737	0.006353899
PIResnetGCLSTM	0.002875693	0.003055863	0.003757378	0.003785996	0.003970475	0.004856808
PIResnetGCGRU	0.003450609	0.003556389	0.004222981	0.004192076	0.004357216	0.005286905

Table 3 Spatiotemporal uncertainty analysis of different renewable penetration levels for the IEEE 118 bus systems

Model	0%	20%	40%	60%	80%	100%
StandardGCN	0.045185547	0.045543607	0.045394797	0.0453988	0.04549436	0.045502752
DynamicGCN	0.001882434	0.001891782	0.001891207	0.001726882	0.001916769	0.001912537
PIGCN	0.002471998	0.002406857	0.002382735	0.00240631	0.002392838	0.002420988
PIGCLSTM	0.001522556	0.001518264	0.001475135	0.00147984	0.001478265	0.001449668
PIGCGRU	0.001684519	0.001540518	0.001472432	0.001440265	0.001423906	0.001392984
PIResnetGCLSTM	0.001190408	0.001388531	0.001519662	0.001610997	0.00176244	0.00180185
PIResnetGCGRU	0.001461807	0.001533965	0.001575716	0.001644865	0.001732303	0.001769893

4. Conclusions

This paper presented a PIStGNN model for end-to-end prediction of ACOPF in sustainable power systems. The proposed method integrates grid topology, nonlinear AC power-flow constraints, and renewable variability. The proposed approach addresses the key limitations of conventional data-driven models; (1) poor physical consistency, (2) limited temporal awareness, and (3) weak scalability under dynamic operating conditions. The study is validated on the IEEE 33-, 57-, and 118-bus systems under varying renewable penetration. The dataset has renewable generation scaled with system size. The generation capacity is 9 MW, 1300 MW, and 3600 MW for the IEEE 33-, 57-, and 118-bus systems, respectively. The voltage profiles set to acceptable operating ranges, of 0.96 and 1.04 p.u. The analysis shows that moderate renewable penetration reduces transmission losses by up to 45%. The excessive penetration increases losses due to reverse power flows and network congestion. The findings also show that the PIStGNN models achieved operationally acceptable MAE on the order of 10^{-2} for voltage magnitude and angle. At same time it delivers inference times as low as 0.05 ms per sample. In addition, the physics-informed and residual spatiotemporal architectures consistently improved predictive uncertainty. The results indicate that recurrent graph models are essential for uncertainty-aware ACOPF prediction. Therefore, the central premise of this work: combining physics-informed learning with spatiotemporal graph modeling yields a scalable, and efficient surrogate for ACOPF prediction is validated. Future work should focus on improving active and reactive power balance satisfaction in large networks. It is also recommended that future research should be extended to warm starting traditional numerical solvers with PIStGNN models to preserve feasibility and optimality.

Declaration of Ethical Standards

As the authors of this study, we declare that he complies with all ethical standards.

Credit Authorship Contribution Statement

A.-M. I. Malori: conceptualization, Validation, Formal analysis, Visualization and methodology
E. A. Frimpong: supervised to ensure technical accuracy of the results
F. E. Boafo: supervised to ensure technical accuracy of the results
E. Twumasi: supervised to ensure technical accuracy of the results
P. Asigri: supervised to ensure technical accuracy of the results
B. M. Adjei: software implementation and experimentation.

Declaration of Competing Interest

The authors declared that they have no conflict of interest.

Funding / Acknowledgements

The authors were supported within the scope of the research by the KNUST Engineering Education Project, Power System Laboratory. The authors declare that generative AI was used only for language editing and not for scientific content generation.

Data Availability

The source code and dataset are publicly available on GitHub repository;
https://github.com/MajidMalori/Optimal-Power-Flow-Prediction-Using-Physics-Informed-Spatiotemporal-Graph-Neural-Network-

References

Abdi, H. (2021). Profit-based unit commitment problem: A review of models, methods, challenges, and future directions. *Renewable and Sustainable Energy Reviews, 138*, 1-28. doi:https://doi.org/10.1016/j.rser.2020.110504

Aravena, I., Molzahn, D., Zhang, S., Petra, C., Curtis, F., Tu, S., . . . Veeramany, A. (2023). Recent Developments in Security-Constrained AC Optimal Power Flow: Overview of Challenge 1 in the ARPA-E Grid Optimization Competition. *Operation Research, 71*(6), 1-43. doi:https://doi.org/10.1287/opre.2022.0315

Colm Lowery and Mark O'Malley. (2012). Impact of Wind Forecast Error Statistics Upon Unit Commitment. *IEEE Transactions on Sustainable Energy*, 760-769. doi:https://doi.org/10.1109/TSTE.2012.2210150

Gao, M., Yu , J., Yang , Z., & Zhao , J. (2024). A Physics-Guided Graph Convolution Neural Network for Optimal Power Flow. *IEEE Transactions on power Systems,, 39*(1), 380-390. doi:https://doi.org/10.1109/TPWRS.2023.3238377

Halilbasic, L., Thams, F., Venzke, A., Chatzivasileiadis, S., & Pinson, P. (2018). Data-driven Security-Constrained AC-OPF for Operations and Markets. *2018 power systems computation conference (PSCC)*. doi:https://doi.org/10.23919/PSCC.2018.8442786

Hasan, F., Kargarian, A., & Mohammadi, A. (2020). A Survey on Applications of Machine Learning for Optimal Power Flow. *2020 IEEE Texas Power and Energy Conference (TPEC)*. College Station, TX, USA. doi:https://doi.org/10.1109/TPEC48276.2020.9042547

Hentenryck, P. V. (2021). Machine Learning for Optimal Power Flows. *Informs Tutorials in Operations Research*, 62-82.

doi:https://doi.org/10.1287/educ.2021.0234

Hmida, J. B., Morshed , M. J., Lee , J., & Chambers , T. (2018). Hybrid Imperialist Competitive and Grey Wolf Algorithm to Solve Multiobjective Optimal Power Flow with Wind and Solar Units. *Energies, 11*(11), 1-23. doi:https://doi.org/10.3390/en11112891

Huang, B., & Wang, J. (2023). Applications of Physics-Informed Neural Networks in Power Systems - A Review. *IEEE Transactions on Power Systems, 38*(1), 572-588. doi:https://doi.org/10.1109/TPWRS.2022.3162473

Jalving, J., Eydenberg, M., Blakely, L., Castillo, A., Kilwein, Z., Skolfield, J., . . . Laird, C. (2024). Physics-informed machine learning with optimization-based guarantees: Applications to AC power flow. *International Journal of Electrical Power and Energy Systems, 157*, 109741. doi:https://doi.org/10.1016/j.ijepes.2023.109741

Jordehi, A. R. (2018). How to deal with uncertainties in electric power systems? A review. *Renewable and Sustainable Energy Reviews, 96*, 145-155. doi:https://doi.org/10.1016/j.rser.2018.07.056

Kaffash, M. (2022). *Data-Driven Uncertainty Representation for Scheduling Flexibility in Energy Systems.* KU Leuven – Faculty of Engineering Science. Retrieved from https://lirias.kuleuven.be/3750267?&lang=en

Kilwein, Z., Boukouvala, F., Laird, C., Castillo, A., Blakely, L., Eydenberg, M., . . . Batsch-Smith, L. (2021). AC-Optimal Power Flow Solutions with Security Constraints from Deep Neural Network Models. *31st European Symposium on Computer Aided Process Engineering.* doi:https://doi.org/10.1016/B978-0-323-88506-5.50142-X

Lei, X., Yang , Z., Yu , J., Zhao , J., Gao , Q., & Yu, H. (2021). Data-Driven Optimal Power Flow: A Physics-Informed Machine Learning Approach. *IEEE Transactions on Power Systems, 36*(1), 346-354. doi:https://doi.org/10.1109/TPWRS.2020.3001919

Lin, Q., Wu,, Q., Huang,, G., & Zhai, M. (2014). An interval parameter optimization model for sustainable power systems planning under uncertainty. *Electrical Power and Energy Systems, 54*, 631–641. doi:https://doi.org/10.1016/j.ijepes.2013.08.018

Roald, L. A., Pozo, D., Papavasiliou, A., Molzahn, D., Kazempour, J., & Conejo, A. (2023). Power systems optimization under uncertainty: A review of methods and applications. *Electric Power Systems Research, 214*, 1-25. doi:https://doi.org/10.1016/j.epsr.2022.108725

Saffari, M., & Khodayar, M. (2024). Spatiotemporal Deep Learning for Power System Applications: A Survey. *IEEE Access, 12,*, 1-35. doi:https://doi.org/10.1109/ACCESS.2024.3424854

Singh, R., & Powar, V. (2024). The Future of Generation, Transmission, and Distribution of Electricity. In A. Lakhtakia, C. M. Furse, & T. G. Mackay, *The Advancing World of Applied Electromagnetics* (pp. 349-383). Springer, Cham. doi:https://doi.org/10.1007/978-3-031-39824-713

Wang, Y. (2023). Advances in spatiotemporal graph neural network. *International Journal of Digital Earth, 16*(1), 2034–2066. doi:https://doi.org/10.1080/17538947.2023.2220610

Xie, R., Wei , W., Li, M., Dong, Z., & Mei, S. (2023). Sizing capacities of renewable generation, transmission, and energy storage for low-carbon power systems: A distributionally robust optimization approach. *Energy, 263*, 1-15. doi:https://doi.org/10.1016/j.energy.2022.125653

Y. Zhou, Li , Y., & Huang, G. (2015). Planning sustainable electric-power system with carbon emission abatement through CDM under uncertainty. *Applied Energy, 140*, 350–364. doi:https://doi.org/10.1016/j.apenergy.2014.11.057

Yang, Z., Zhong, H., Xia,, Q., & Kang, C. (2017). Fundamental Review of the OPF Problem: Challenges, Solutions, and State-of-the-Art Algorithms. *Journal of Energy Engineering, 144*(1), 1-12. doi:https://doi.org/10.1061/(ASCE)EY.1943-7897.0000510

Yaniv, A., Kumar, P., & Beck, Y. (2023). Towards adoption of GNNs for power flow applications in distribution systems. *Electric Power Systems Research, 216*, 1-8. doi:https://doi.org/10.1016/j.epsr.2022.109005

Zhou, D., Zhang, Y., Wang, Q., & Ding, H. (2024). How do uncertain renewable energy induced risks evolve in a two-stage deregulated wholesale power market. *Applied Energy, 353*, 122-140. doi:https://doi.org/10.1016/j.apenergy.2023.122140

Zideh, M. J., Paroma , C., & Srivastava , A. K. (2024). Physics-Informed Machine Learning for Data Anomaly Detection, Classification, Localization, and Mitigation: A Review, Challenges, and Path Forward. *IEEE ACCESS, IEEE Power & Energy Society Section, 12*, 4597-4617. doi:https://doi.org/10.1109/ACCESS.2023.3347989