International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319-9598 (Online), Volume-13 Issue-3, March 2026

Graph Neural Network- Enhanced Power Flow Adjustment

Gowtham I, Perumal B

***Abstract:** Modern power grids are facing unprecedented operational complexity due to the surge in distributed energy resources (DERs), intermittent renewables, and electric vehicle (EV) charging demands. While traditional methods like Newton-Raphson are computationally precise, their iterative nature often fails to meet the sub-second latency requirements of dynamic smart grids. This research proposes a Graph Neural Network (GNN) framework designed to model electrical networks as high-dimensional graphs. By capturing the inherent topological relationships among buses (nodes) and transmission lines (edges), the GNN-based approach provides a scalable, data-driven alternative for real-time power-flow estimation. The framework effectively processes non-linear grid behaviours and uncertainties, ensuring stable and efficient grid management, congestion control, and optimal power dispatch in rapidly evolving electrical environments.*

***Keywords**: Distributed Energy Resources, Electric Vehicle, Graph Neural Network, Non-Linear Grid*

Nomenclature:
EV: Electric Vehicle
GNN: Graph Neural Network
AI: Artificial Intelligence
MSE: Mean Squared Error
MAPE: Mean Absolute Percentage Error
MAX-E: Maximum Error
CNNs: Convolutional Neural Networks
PRD: Percentage Relative Difference
DERs: Distributed Energy Resources

I. INTRODUCTION

Today's power grids are changing fast. In the past, electricity came from stable, predictable sources like coal and gas plants. However, we are now moving toward a "smarter" grid that relies on renewable energy, local generators, and electric vehicle (EV) charging. While this is better for the planet, it makes the grid much harder to manage. Solar and wind power fluctuate with the weather, and thousands of EVs plugging in at once can create unpredictable spikes in demand.

To keep the lights on, engineers use "power flow analysis" to predict how electricity moves through the wires. For decades, the standard approach to this was to use mathematical formulas, such as the Newton-Raphson method. These formulas are very accurate, but they have a major flaw: they are slow. They require constant, repetitive calculations that take too much time when a grid has thousands of connection points. In a modern world where energy levels change in a split second, these old methods can't keep up. To solve this, experts are now turning to Artificial Intelligence (AI). Using machine learning, we can monitor the grid in real time, allowing us to react instantly to changes and keep the entire system stable and efficient. Among these, Graph Neural Networks have gained attention due to their capability to represent electrical networks as graph-structured data. In power systems, buses can be modelled as graph nodes, while transmission lines serve as edges that carry relational information. GNNs utilise message-passing operations to exchange information, and their behaviour is studied using an ML model [4]. ML, an advanced algorithm for predicting renewable generation, has been proposed in deep reinforcement learning as a possible control strategy for power systems with multiple renewable energy sources.

Compared to traditional computational approaches, machine learning algorithms have an intrinsic generalisation capability and greater computational efficiency and scalability. Since machine learning algorithms can learn complex nonlinear input-output relationships and adapt to the data, they have been used to predict the voltage magnitude and phase angle at each bus. Similarly, predictions of initial system variables in ML mode are implemented to reduce the number of solution iterations and the time required by the NR-based ACPF model. However, the spatial information embedded in the predictions for active power flow for each branch was not considered.

A power system is an interconnected network of generators and loads which has embedded graphical information. The graph structure of the power system comprises nodes (buses) and connections between neighbouring nodes, enabling efficient learning of system patterns, voltage correlations, and power-flow dependencies. The Proposed System: A Smarter Grid Approach

The proposed system is a fast and intelligent way to manage modern power grids. Today's grids are under pressure from unpredictable energy sources like solar and wind, as well as sudden power spikes caused by electric vehicle (EV) charging. Instead of relying on outdated, slow mathematical formulas, this system uses Graph Neural Networks (GNNs) [3].

By modelling the power grid like a map of interconnected dots (nodes) and lines (edges), the GNN can "see" how electricity flows across the entire network. It analyses real-time sensor data and weather patterns to estimate

Manuscript received on 30 January 2026 | Revised Manuscript received on 08 February 2026 | Manuscript Accepted on 15 March 2026 | Manuscript published on 30 March 2026.

*Correspondence Author(s)

Gowtham I*, Department of Electrical and Electronics, Engineering, Adhiyamaan College of Engineering, Hosur (Tamil Nadu), India. Email ID: gowthamiyappan950@gmail.com, ORCID ID: 0009-0001-9162-6185

Dr. Perumal B, Assistant Professor, Department of Electrical and Electronics Engineering, Adhiyamaan College of Engineering, Hosur (Tamil Nadu), India. Email ID: balanperumal.b@gmail.com, ORCID ID: 0009-0005-2960-5083

the grid's health in milliseconds. This speed allows operators to spot potential overloads or "hotspots" before they lead to blackouts, making the grid much more reliable and efficient as we move toward greener energy.

A. Preliminaries: How Power Flow is Calculated

To understand why we need AI, we first have to look at how engineers traditionally check the grid. A "Power Flow Calculation" is essentially a health report for the electrical system. It tells us the voltage at each station and the amount of power flowing through each wire.

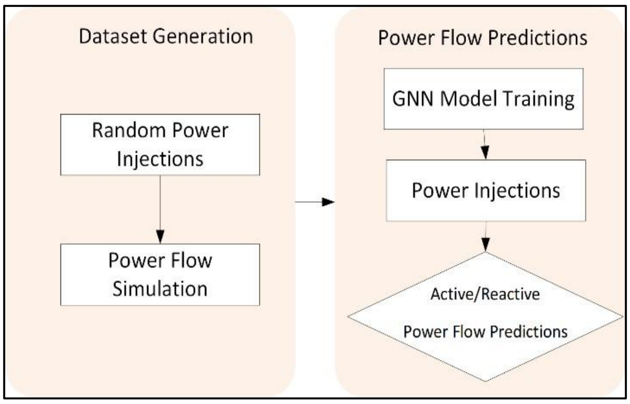
To get these answers, we use Nodal Power Balance Equations. These formulas ensure that the energy going into a point matches the energy coming out. We balance two types of power:

- Real Power: The energy that actually does work, like turning on a motor.
- Reactive Power: The "background" energy needed to maintain voltage and push electricity through the lines.

  While these formulas are very accurate, they are incredibly complex. In a large grid, solving them requires a computer to guess and check the answer thousands of times (an "iterative" process). This takes a lot of time and power, time that grid operators don't always have when things are changing in the blink of an eye, P_i and Q_i are the active and reactive power injections at each node, respectively. The summation terms represent active or reactive power injections or withdrawals at a given node. $i. V_i$ and V_j Is the voltage magnitudes for the two end buses of a transmission line. G_{ij} and B_{ij} are the corresponding conductance and susceptance of a branch. The phase angle θ_{ij} is the difference in voltage phase angles of the two end buses of a branch.

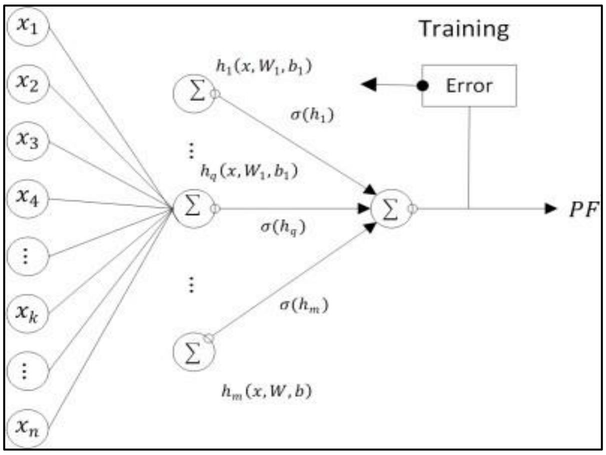
An accurate and comprehensive power flow analysis is achieved by employing conventional ACPF techniques such as the NR and GS methods. This approach accounts for the non-linear properties of power system components, including transformers and transmission lines. However, ACPF solves a set of nonlinear equations representing the power-flow equations based on Kirchhoff's laws and other system constraints using iterative calculations. Due to the inherent, complex, iterative nature of these algorithms, they may diverge. They may not be suitable for some online monitoring applications or for integration into optimisation-based scheduling and dispatching models. A non-iterative method called DC power flow can be used for fast power-flow solutions. DCPF represents the network as an equivalent DC network, thereby reducing calculation complexity. It simplifies the power-flow equations by assuming that voltage magnitudes remain constant and that phase angles are small. With DCPF, the solutions to steady-state active power flow can be found very quickly. However, the approximation error introduced by the earlier assumptions may lead to inaccurate results.

where h^{PF} is the nonlinear power flow prediction model; s_t denotes the system states, and u_t Is the generation dispatch at the period? $t_t d_t$ and r_t denote the load profile and renewable forecast, respectively. The overview of the power flow model is shown in Fig. 1.



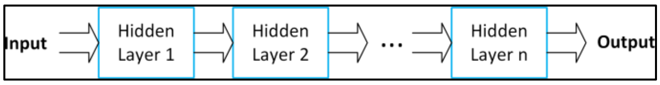
[Fig.1: Overview of Power Flow Model]

For a basic power flow model, training data (injections) are multiplied by the weight vector. W. Then, the results are added by bias b and mapped to an output value after applying an activation function. The choice of activation functions can vary depending on the model selected. Throughout the training process, the weight vector is updated continuously until the error falls below a predetermined threshold or a specified number of epochs is reached.



[Fig.2: Example of Power Flow Model Training Process]

To handle high nonlinearity, a power flow model typically uses multiple nodes (artificial neurons) with several hidden layers, as shown in Fig. 3. The connections between nodes (vector weights) represent the signal strength between neurons. A neural network consisting of multiple layers performs distinct transformations on input data. Each layer or node within the network learns specific features from the input. By leveraging an NN-based power flow model, complex input-output relationships can be learned, which may.



[Fig.3: Neural Network with Multiple Layers]

II. GRAPH NEURAL NETWORK-BASED POWER FLOW MODEL

CNNs have inherent limitations when handling graphical data that contain explicit topological graph correlations. However, recent advances in CNNs have led to a resurgence of GNNs, which are neural networks specialised in processing and extracting knowledge from graph-structured data. A power system can be viewed as a graph comprising nodes (buses) and edges (branches) that denote the connections between nodes. GNNs are specifically designed to capture the intricate dependencies and relationships present in graph data. To achieve this, GNNs have been developed by extending the convolution operation to graphs and, more generally, to non-Euclidean spaces. Previous studies have shown that GNNs achieve state-of-the-art performance on graph analysis tasks. The input vector rcap X concatenates information on the electrical power produced and consumed across the entire grid. The branches in the power system are undirected; such graphs represent buses and their connections. Specifically, each generation $g \in G$ is defined by an active power infeed P_g (in MW) and a reactive power Q_g (in MVar). Therefore, each generation is defined by 2-dimensional information. Similarly, a nodal load $d \in D$ is defined by an active power consumption P_d (in MW) and a reactive power consumption Q_d (in MVar). Thus, the injection vector X is a vector that concatenates all these injection characteristics, including the initial voltage magnitude on each bus.

The convolution operator in the propagation module aggregates information from neighbours. Considering $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as an undirected graph representing a power system, where $\mathcal{V} \in \mathbb{R}^N$ denotes its nodes, and $\mathcal{E} \in \mathbb{R}^K$ denotes its edges. Let $A \in \mathbb{R}^{N \times N}$ be the adjacency matrix of $\mathcal{G}$, we can define a renormalisation equation as,

$$V = \tilde{D}\overset{1}{2}\tilde{A}\tilde{D}\ 1$$

where $\tilde{A} = A + I_N$ represents an adjacency matrix with added self-connections, and I_N is the identity matrix. The adjacency matrix encodes the way injections are connected to edges. Typically, the element at (i, j) of the adjacency matrix A is defined as follows,

$$A_{ij} = \begin{cases} 1; \text{ if } \mathcal{V}i, \mathcal{V}j \in \mathcal{V}, (\mathcal{V}i, \mathcal{V}_j) \in \mathcal{E} \\ 0; \text{ if } \mathcal{V}, \mathcal{V} \in \mathcal{V}, (\mathcal{V}, \mathcal{V}) \notin \mathcal{E}, \end{cases}$$
$$ij\ ij$$

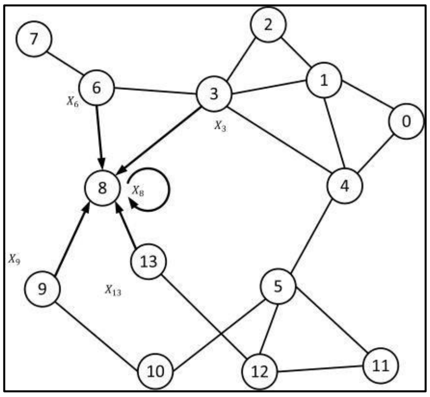
where $(\mathcal{V}_i, \mathcal{V}_j)$ denotes the branches from i to j. The diagonal degree matrix $\tilde{D}$ for $\mathcal{G}$ is defined as $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

GNNs update the representations of nodes in a graph by aggregating information from their neighbouring nodes. This process involves iterative message-passing steps, where each

The node incorporates both local and global information to update its representation. By doing so, GNNs capture both the local structure and the broader context of the graph. The graph convolutional activation is defined as follows,

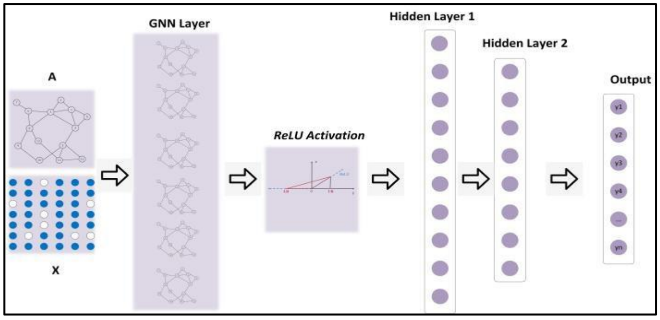
$$F^l(X, A) = \sigma\left(VF^{(l-1)}(X, A)Wl + b_k^l\right),$$

where F^l is the convolutional activations, W^l and b^l are the trainable convolutional weights matrix and bias matrix at the
k -
The layer, $F0 = X$, is the input matrix. In this step, we iteratively update the latent state of each of the n power lines by performing latent leaps that depend on the value of their direct neighbours.



[Fig.4: Example of Message Passing Mechanism in GNN of IEEE 14-bus System]

Fig. 4 illustrates the message-passing mechanism during forward propagation, with the target node (bus 8) receiving information from its neighbouring nodes. The output we want to predict is the flow through each line.



[Fig.5: Illustration of the Proposed GNN Neural Network]

The proposed power flow model, as shown in Fig. 5, has one GNN layer for the embedding step and two hidden layers for the decoding step. The Rectified Linear Unit (ReLU) is chosen as the activation function for forward propagation. It allows a small, non-zero constant gradient to pass through, thereby mitigating the vanishing gradient problem during training. The decoding step decodes the embedded data to the output space, which is the steady-state voltage magnitude. for each bus and the power flow on each branch [2].

Mean squared error (MSE) loss was used to evaluate the model's performance during training. MSE measures the average squared difference between actual and predicted outputs. The MSE loss function used in this paper is defined as where y_i denotes the actual output value while $\tilde{y}_i$ denotes the GNN model estimated output value, and N denotes the number of sample points.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \tilde{y})_i^2$$

III. CASE STUDIES

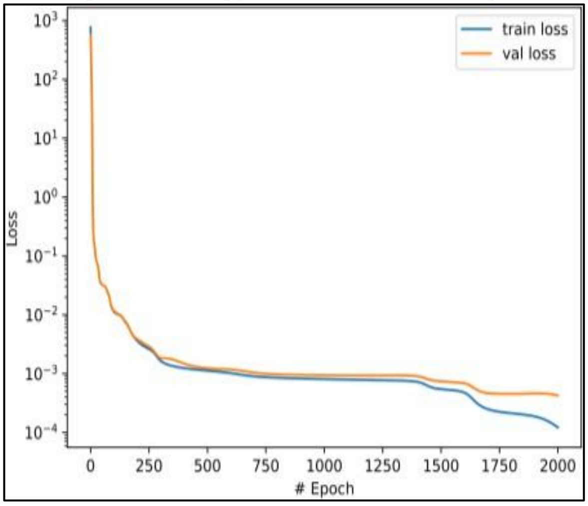
The proposed GNN model was trained to predict voltage magnitudes and active power flows across multiple systems of different sizes: the IEEE 14-bus and IEEE 24-bus test systems. For the data generation process, each load and generator dispatch setting is randomly perturbed by

115% using a uniform distribution, with the load data in the case files used as the base value. For active power control corresponding to variations in the total system load. Power flow data are generated in Python 3.8 using pypower. The GNN-based power flow model is trained using PyTorch on NVIDIA RTX 2070 GPUs. The generated data set was divided into three groups: 80% for training, 20% for validation.

The proposed GNN model is compared with the DNN and CNN models. The following metrics are used to demonstrate the prediction accuracy: (1) maximum error (MAX-E), (2) median absolute error (MED-E), (3) mean absolute percentage error (MAPE), and (4) R2 score, which is defined as follows.

$$R^2(y_i,\hat{y}_i) = 1 - \frac{\sum_i^N (y_i - \tilde{y}_i)\ 2}{\sum^N (y_i - \bar{y})}$$

where $\bar{y}$ is the mean of actual labels. R^2 score indicates the model's goodness of fit. In regression, the R^2 The coefficient of determination is a statistical measure of how well the regression predictions approximate the real data points.



[Fig.6: Training Curve of the GNN Neural Network on the 14-Bus System]

Table I: Prediction Accuracy of Power Flow with Different Tolerances Comparing Different Models

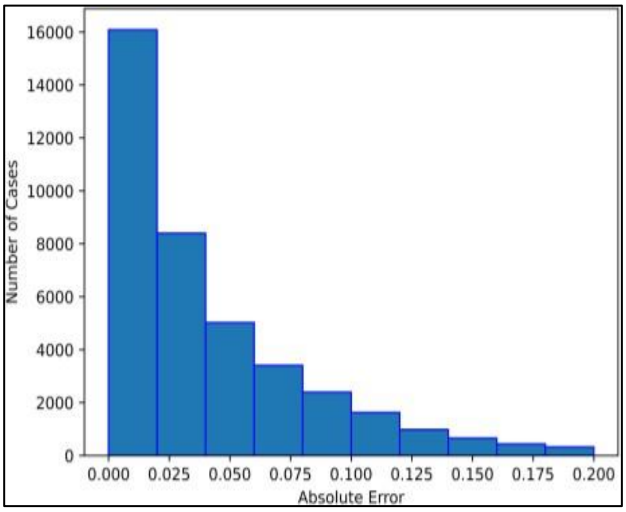
System	Tolerance	1%	2%	3%	4%	5%
14 Bus	DNN	66.65	76.41	82.54	86.48	88.92
	CNN	69.60	79.86	86.00	89.14	91.07
	GNN	89.60	94.24	95.59	96.26	96.66
24 Bus	DNN	71.90	87.95	91.48	93.71	95.56
	CNN	75.56	88.56	91.88	94.46	96.02
	GNN	76.51	90.00	93.11	95.33	96.76

Fig. 6 depicts the loss curve of the proposed GNN model on the 14-bus system during the training process. It can be observed that the proposed GNN model performs well in minimising the MSE loss. The training loss decreases rapidly and then levels off after around 800 epochs. Table I shows the prediction accuracy of different neural network models. Deep Neural networks (DNNs) and convolutional neural networks (CNNs) are used as benchmarks [1]. The proposed GNN model achieves the highest prediction accuracy among the models. For the 14-bus system, the prediction accuracy is 96.66% with a 5% tolerance. Further investigations of the 24-bus system revealed that the GNN model achieves the highest prediction accuracy across varying error tolerances.

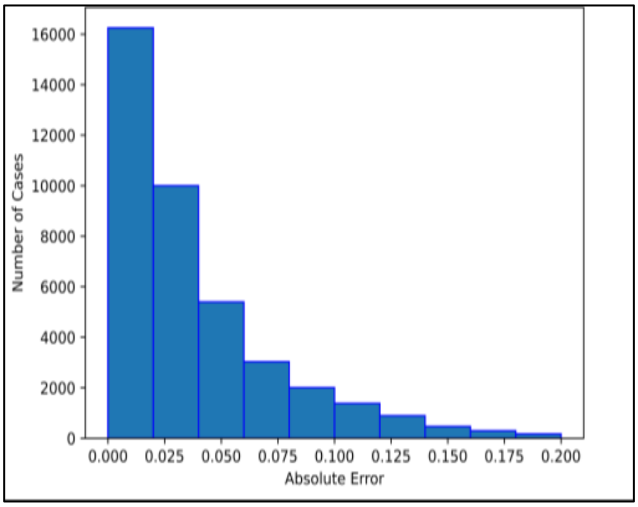
Table II: Comparison of Different Models

System	Model	R^2	MAX-E	MED-E	MAPE
14 Bus	DNN	0.9946	0.53	0.0279	0.0032
	CNN	0.9963	0.28	0.0262	0.0031
	GNN	0.9993	0.25	0.0183	0.0025
24 Bus	DNN	0.9714	4.80	0.2154	0.0059
	CNN	0.9733	4.55	0.1998	0.0055
	GNN	0.9824	4.30	0.1855	0.0053

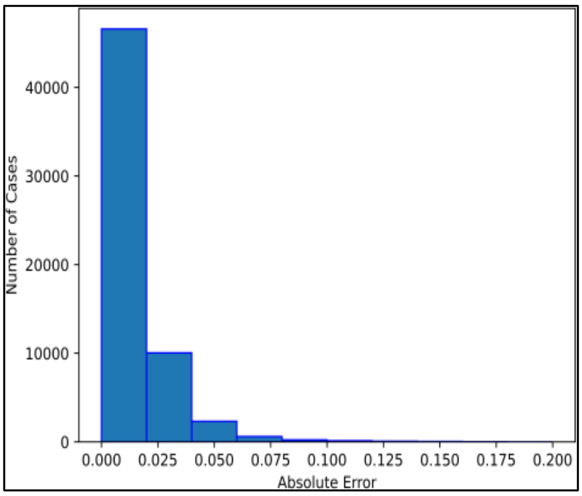
Table II summarises the statistics for all model architectures. The first column shows the R^2 scores for all models on the 14-bus and 24-bus systems. It can be observed that the GNN model has the highest R2 score in both scenarios, which is close to 1, indicating that its predictions more closely approximate the real data points than those of the other machine learning algorithms' power flow models. The GNN model also has the lowest MAX-E, MED-E and MAPE statistical values.



[Fig.7: Absolute Error Distribution of the DNN Neural Network for the 14-Bus System]



[Fig.8: Absolute Error Distribution of the CNN Neural Network for the 14-Bus System]



[Fig.9: Absolute Error Distribution of the GNN Neural Network for the 14-Bus System]

Figs. 7-9 Compare the error distributions of GNN and benchmark models. Combining Table II and Figs. 7-Fig. 9, it is observed that the proposed GNN model outperforms all other benchmarks. Over 95% of power flow prediction errors are located within the range $0.00 - 0.05$ (MW), implying that the predictions with the proposed GNN model are much closer to the reference values. The percentage relative difference (PRD) is calculated using the relative difference formula shown below,

$$PRD = \frac{2|y_i - \tilde{y}_i|}{|y_i + \tilde{y}i|} \times 100\%.$$

Table III presents the PRD of line power flow. Results show that the DCOF model's accuracy is lower than that of the GNN and DNN models. For active power flow on a 24-bus system, the mean PRD of the GNN model is 0.01, lower than that of the DNN model at 0.02. Due to the approximation error introduced by the DCPF model, the maximum PRD of DCPF in both scenarios is much higher than that of data-driven models such as DNN and GNN. Compared with two other models, the proposed GNN achieves much better performance in predicting active power flow on both test systems. By comparing the standard deviations and median PRD across all models, we find that the proposed GNN model outperforms the traditional DCPF model.

Table III: PRD of Active Power Flow [MW] Comparing GNN Model and DCPF Model

System	Model	Mean	Max	Min	Median	Std.Dev.
14-bus	DNN	0.07	15.79	0.00	0.003	0.34
	GNN	0.05	13.45	0.00	0.001	0.26
	DCPF	0.96	17.28	0.00	0.23	0.46
24- bus	DNN	0.02	29.73	0.00	0.016	0.28
	GNN	0.01	24.40	0.00	0.005	0.22
	DCPF	1.36	30.63	0.00	0.53	0.93

IV. CONCLUSIONS

With the potential to shift the entire computational effort to offline training, machine-learning-assisted power flow has become an increasingly interesting research direction. The traditional DCPF model introduces approximation error during power-flow calculations. The GNN model shows great potential for processing power system data with embedded geographical information. The proposed GNN model can be trained offline using historical data, enabling the rapid determination of both line active power flows. It should be noted that the GNN model's architecture needs to be carefully selected, and sufficient resources are required to train it.

Results for the 14- and 24-bus test systems show that the GNN model closely aligns with those obtained from the ACPF method. The GNN model demonstrates high efficiency in power-flow prediction. Compared with the DNN and CNN models, the proposed GNN model achieves higher accuracy. In addition, the proposed GNN model performs faster and yields much better results than the DCPF model. Experts are continually finding new ways to use GNNs in power systems, proving that this technology could be a game-changer for advancing our electricity networks.

DECLARATION STATEMENT

As the article's author, I must verify the accuracy of the following information after aggregating input from all authors.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted objectively and without external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. M. H. Karimi et al., "Graph Neural Networks for Fast Power Flow Computation," Electric Power Systems Research, vol. 215, 109012, 2023. https://arxiv.org/html/2502.05702v1
2. L. Shen et al., "Distributed Graph Neural Networks for Power Flow Optimisation in Smart Grids," IEEE Transactions on Industrial Informatics, vol. 19, no. 12, pp. 12345-12356, 2023.
3. A. Venzke et al., "Deep OPF: A Deep Neural Network Approach for AC Optimal Power Flow," Applied Energy, vol. 357, 122456, 2024. GNN enhanced for grid adjustment. DOI: https://doi.org/10.1109/SmartGridComm47815.2020.9303017
4. R. D. Zimmerman et al., "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education," IEEE Transactions on Power Systems, vol. 26, no. 1, pp. 1219, 2011. Classic solver benchmark for GNN comparisons. DOI: https://doi.org/10.1109/TPWRS.2010.2051168

AUTHOR'S PROFILE

Gowtham.I is currently pursuing his Master of Engineering (M.E.) in Power Systems (2025–2026) at Adhiyamaan College of Engineering, Hosur, Tamil Nadu, India. His academic interests include power system optimisation, energy efficiency, renewable integration, and industrial energy management. He is actively involved in research related to electrical energy conservation and sustainable energy practices. He has presented and contributed to academic projects focused on energy auditing and system performance improvement in industrial facilities.

Dr. B. Perumal is an Assistant Professor in the Department of Electrical and Electronics Engineering at Adhiyamaan College of Engineering, Hosur, Tamil Nadu, India. He has extensive teaching and research experience in power systems, energy management, and control systems. His areas of expertise include electrical system design, energy conservation, and power quality improvement. He has guided numerous postgraduate and undergraduate projects and has published several papers in reputed national and international journals.