Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 9, 1534-1561
2025
Publisher: Learning Gate
DOI: 10.55214/2576-8484.v9i9.10161

Performance evaluation of artificial, physics-informed and graphical neural network models for load flow analysis in smart and resilient power grids: Case study of IEEE and Nigerian power systems

Ibukun Damilola Fajuke[1*], Bolanle Tolulope Abe[2]
[1,2]Department of Electrical Engineering, The Tshwane University of Technology South Africa; fajukeid@tut.ac.za (I.D.) abebt@tut.ac.za (B.T.A.).

Abstract: Load flow analysis is essential for periodic planning, scheduling, and reliable operation of traditional and modern power grids. This study evaluates the effectiveness of Artificial, Physics-Informed and Graph Neural Network (NN) models in performing load flow analysis on standard benchmark systems (IEEE 14- and 30-bus) and practical Nigerian networks (28- and 52-bus) under steady-state, Fault, and Distributed Generation (DG) penetration scenarios. Newton-Raphson (NR) method was used to generate base case voltage magnitudes and phase angles as reference targets for model training. Models were implemented in MATLAB (R2025a) and evaluated using standard statistical metrics (MSE, RMSE, MAE, MAPE) and Line Voltage Stability Index (LVSI). Simulation results showed that ANN achieved MSE values between 0.385-1.079, RMSE 0.62-1.039, MAE 0.037-0.105, MAPE 2.005-5.562%, and LVSI 0.594-0.87. GNN recorded MSE 0.731-1.828, RMSE 0.855-1.352, MAE 0.068-0.18, MAPE 3.073-8.903%, and LVSI 0.524-0.724. PINN showed MSE 1.622-2.552, RMSE 1.274-1.597, MAE 0.158-0.238, MAPE 6.197-10.77%, and LVSI 0.563-0.752. The results demonstrate the suitability of the individual models for rapid and reliable load flow analysis across varying network sizes and operating conditions. The findings of this study will serve as practical guidance for model selection in modern power systems, supporting efficient planning, operation, and integration of DG resources.

Keywords: *Artificial neural networks, Distributed generation, Graph neural networks, Load flow analysis, Physics-informed neural networks, Power system stability, Resilient power grids.*

1. Introduction

Electrical power systems are large-scale, interconnected infrastructures designed to facilitate the generation, transmission, and distribution of electricity across wide geographic regions [1, 2]. Ensuring their stable and economic operation requires continuous monitoring and analysis of electrical parameters at various nodes and branches of the system. One essential analytical tool used in this context is load flow analysis (also known as power flow analysis), which provides critical information on voltage magnitudes, phase angles, real and reactive power flows, and system losses [3]. This analysis forms the backbone of power system planning, operation, and contingency assessment, enabling operators to understand system behavior under both normal and disturbed conditions [3-5].

The growing integration of Renewable Energy (RE) sources, Distributed Generation (DG) units, and advanced power electronic devices has also introduced new complexities into modern grids, making accurate and efficient load flow analysis more crucial than ever [6-8]. Traditionally, numerical methods such as Newton-Raphson (NR), Gauss-Seidel (GS), and Fast Decoupled Load Flow (FDLF) have been widely used for solving load flow problems. For instance, Adejumobi, et al. [9] performed a comparative study of NR, GS, and FDLF methods; T. BS. and K. V. Madhukar, "Deep learning based optimal power

History: Received: 16 July 2025; Revised: 27 August 2025; Accepted: 1 September 2025; Published: 23 September 2025
* Correspondence: fajukeibukundamilola@gmail.com

flow with renewable integration," in Fikri, et al. [3] also utilized NR and GS for power flow studies, while Adebayo, et al. [10] investigated load flow using Load Tap-Changing Transformers (LTCT). Although these methods are effective under many conditions, they often struggle with convergence in highly nonlinear or dynamically complex systems, particularly those impacted by RE and DG integration. These limitations have spurred interest in more adaptive and intelligent approaches [11].

To address these challenges, Artificial Intelligence (AI) techniques have been increasingly explored as alternatives to conventional methods. Metaheuristic algorithms and expert systems such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Sparrow Search Algorithm (SSA), Artificial Bee Colony (ABC) and Fuzzy Logic have demonstrated flexibility in handling the uncertainties and complexities of load flow problems [11-13]. Notable examples include Himakar and Reddy [14] who applied Real-Coded Buragohain and Boruah [5] and Dixit, et al. [15] both applied fuzz logic, while Shazly, et al. [13] implemented PSO for power flow analysis in modern systems. Despite their adaptability, these methods can be computationally intensive and may face issues with convergence in large-scale or highly variable environments [13].

As a result, Machine Learning (ML) techniques have gained attention for their ability to model complex systems using historical data. ML methods like Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), and decision Trees have been used to approximate bus voltages, power flows, and other key indicators with impressive accuracy [16-18]. For example, EFE [19] compared analytical methods with ANN for a 5-bus system; Tiwari, et al. [20] employed ANN for predicting voltage magnitudes and power flows, while Fikri, et al. [3] and Alsulami and Kumar [16] extended this to a Moroccan 24-bus system and Saudi Arabia grid, respectively. Calma and Pacis [21] also demonstrated ANN's ability to assess voltage security and calculate stability indices. These data-driven models can generalize from past system behavior and often eliminate the need for repetitive computations during real-time inference [17].

Researchers have also explored hybrid approaches, combining conventional and AI methods to improve performance. Gnanambal, et al. [22] investigated hybrid PSO-GA for three-phase power flow, while Davoodi, et al. [23] hybridized simplex with PSO to improve convergence. Singh, et al. [24] employed a hybrid PSO-ANN to predict bus voltage magnitudes and angles on the IEEE 14-bus system under various cases. Ting, et al. [25] tested a GA-PSO hybrid method on larger power systems including IEEE 30, 57, and 118-bus systems under heavy loading. Ahiakwo, et al. [26] also utilized a neuro-swarm approach combining ANN and ABC algorithms for load flow computation of a 4-bus system, illustrating the flexibility of hybrid models in practical problems.

While ANNs have shown promise, their reliance on large datasets without embedded physical constraints is limited [17]. To bridge this gap, physics-informed and structure-aware neural networks approaches have emerged. The study of Kaseb, et al. [27] employed Quantum Neural Network for load flow analysis of standard IEEE 4 bus and 30 bus systems. Therefore, this study investigates the effectiveness of three ML models including ANNs, Physics-Informed Neural Networks (PINNs), and Graph Neural Networks (GNNs) in solving load flow problems. Using both IEEE standard systems (14-bus and 30-bus) and practical Nigerian networks (28-bus and 52-bus), the models are evaluated based on their accuracy in predicting bus voltages and phase angles, convergence behavior, and computational efficiency. The aim is to provide insights into the adaptability and reliability of these techniques for both conventional and evolving power grid environments.

2. Theoretical Framework

2.1. Modern Electric Power Grid

The modern electrical power grid is a highly networked system that delivers electricity at high efficiency from the location at which it is generated to where it is needed. It integrates DG resources such as wind and solar at the distribution level as shown in Figure 1. As a result of this integration, the system becomes increasingly dynamic and complicated. Therefore, accurate load flow analysis becomes even more essential to guarantee the efficient and reliable operation of the power grid [28].

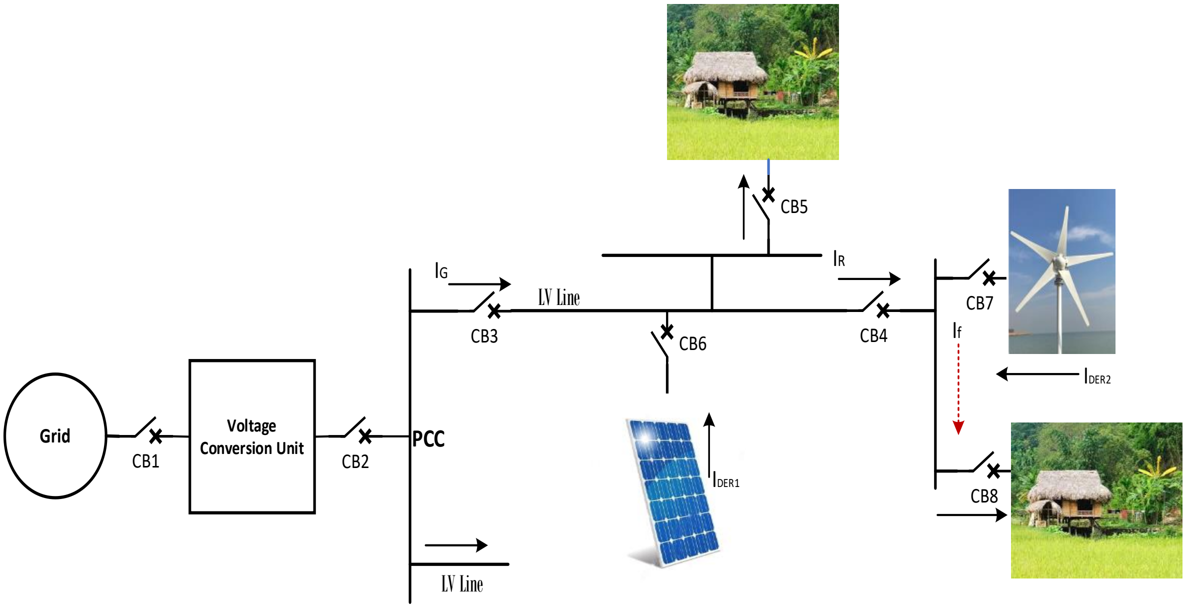

Figure 1.
Modern Electric Power Grid

2.2. Modeling DG for Load Flow Analysis

Distributed Generation are energy sources which are used to compensate for the power losses at distribution level in an electric power system [28]. The mathematical representation of the power contribution from solar and wind DG resources is generally modeled as follows.

For solar PV, the active power contributed is modeled as Ejuh Che, et al. [7]

$$P_{DG}^{PV} = \eta_{PV} \cdot A_{PV} \cdot G_{STC} \cdot \eta_{INV} \qquad (1)$$

For wind turbine generation, the active power contributed is modeled as Ejuh Che, et al. [7]

$$P_{DG}^{WT} = \frac{1}{2}\rho A v^3 C_p \qquad (2)$$

where η_{PV} is the efficiency of the photovoltaic panels; A_{PV} is the area of the photovoltaic panels; G_{STC} is the solar irradiance in W/m^2 (typically obtained from meteorological data); η_{INV} is the efficiency of the inverter system; ρ is the air density; A is the swept area of the wind turbine blades; v is the wind speed at the height of the turbine and C_p is the power coefficient.

The reactive power of DG resources is typically small and can be assumed to be zero unless there are reactive power compensation devices (like capacitors) installed, which is not the case for the current study.

2.3. Fundamentals of Load Flow Analysis

Load flow analysis is a procedure used to determine the known and unknown variables at the different buses of an electric power system for specified operating conditions [5]. These variables are presented in Table 1. The objective of load flow analysis is to solve for bus voltage magnitudes, V_i and the phase angles, δ_i when the power generations and loads are specified.

DOI: 10.55214/2576-8484.v9i9.10161

Table 1.
Load Flow Variables.

Bus Types	Known Variables	Unknown Variables
Generator (PV)	Real power (P)	Voltage angle (θ)
	Voltage magnitude (V)	Reactive power (P)
Load/Generator (PQ)	Real power (P)	Voltage angle (θ)
	Reactive power (Q)	Voltage magnitude (V)
Slack/Reference Bus	Voltage magnitude (V)	Real power (P)
	Voltage angle (θ)	Reactive power (Q)

If the admittance matrix Y_{bus} is known for the power system under consideration. The complex power balance at any bus i of the network is given as Fikri, et al. [3].

$$S_i = V_i I_i^* = S_{Gi} - S_{Li} \tag{3}$$

If V_{bus} and I_{bus} are used to represent the vectors of the bus voltage and bus injection current respectively, the admittance matrix Y_{bus} provides the relationship; $I_{bus} = Y_{bus} \cdot V_{bus}$. Also, if the ith or jth entry has the magnitude, Y_{ij} of the Y_{bus} matrix and the phase γ_{ij}; hence the sending end current I_i can be formulated as Fikri, et al. [3].

$$I_i = \sum_j Y_{ij} V_j = \sum_j Y_{ij} V_j < \left(\delta_j + \gamma_j\right) \tag{4}$$

Therefore, combining equations (1) and (2) we get the complex power balance equations for the network as Fikri, et al. [3].

$$S_i - S_{Gi} - S_{Li} = \sum_{j=1}^{N} Y_{ij} V_i V_j < \left(\delta_i - \delta_j - \gamma_{ij}\right) \tag{5}$$

The base power flow equations, representing both active and reactive power balances, are given as follows for each bus i in the system Fikri, et al. [3].

$$P_i = P_{Gi} - P_{Li} = \sum_{j=1}^{N} Y_{ij} V_i V_j cos\left(\delta_i - \delta_j - \gamma_{ij}\right) \tag{6}$$

$$Q_i = Q_{Gi} - Q_{Li} = \sum_{j=1}^{N} Y_{ij} V_i V_j sin\left(\delta_i - \delta_j - \gamma_{ij}\right) \tag{7}$$

where N represents the number of buses; P_i is the net active power injection at bus i; Q_i is the net reactive power injection at bus i; P_{Gi} is the active power generated at bus i; P_{Li} is the active power load at bus i, Q_{Gi} is the reactive power generated at bus i; Q_{Li} is the reactive power load at bus i; V_i is the voltage magnitude at bus i; V_j is the voltage magnitude at bus j; $\delta_i$is the voltage phase angle at bus i; δ_j is the voltage phase angle at bus j;Y_{ij} magnitude of the admittance between bus i and j and γ_{ij} is the phase angle of the admittance Y_{ij}.

The admittance matrix Y_{bus} can be constructed from the line parameters r, x and b of each branch in the network as Buragohain and Boruah [5].

$$Y_{ij} = \frac{1}{r_{ij} + jx_{ij}} \tag{8}$$

where r_{ij}, x_{ij} and b_{ij} are the resistance, reactance, and shunt susceptance of the branch between buses i and j

For fault scenarios modeling, the system's admittance matrix is modified to reflect line outages or fault conditions. The faulted admittance matrix is given as Buragohain and Boruah [5].

$$Y_{bus}^{fault} = Y_{bus} - \Delta Y_{line} \tag{9}$$

where ΔY_{line} represents the admittance of the faulted line whose modification subsequently affects the power flow equations and system dynamics.

The resulting non algebraic power flow equations is solved using various load analysis techniques to provide valuable information on the performance/state of the power system.

2.4. Load Flow Analysis Techniques

Traditionally, techniques such as NR, GS, and FDLF have been extensively employed due to their proven reliability and mathematical rigor. These conventional approaches are particularly effective for steady-state analysis in conventional power grids, offering acceptable convergence and computational efficiency under normal operating conditions [27]. However, as power systems evolve, these methods often face challenges related to convergence speed, sensitivity to initial conditions, and handling of non-linearities [11]. Nevertheless, since most ML techniques are data-driven models that require a substantial amount of high-quality input-output data to learn complex relationships, the NR method is often employed to generate accurate and reliable load flow solutions under various operating conditions. The NR method is preferred for this task due to its high precision, fast convergence characteristics, and robustness in handling large-scale and non-linear power systems [3]

In the NR technique for load flow analysis, the most important expressions are those that relate the power mismatch at each bus to the Jacobian matrix and the voltage deviations as given in equations (10) to (13). These equations form the core of the iterative update process of the NR technique [19, 29].

$$\begin{bmatrix}\Delta P\\ \Delta Q\end{bmatrix}=\begin{bmatrix}J_1 & J_2\\ J_3 & J_4\end{bmatrix}\begin{bmatrix}\Delta\delta\\ \Delta|V|\end{bmatrix} \tag{10}$$

where ΔP and ΔQ are the mismatch or residual power; $\Delta|V|$ is the change in voltage magnitude; $\Delta\delta$ is the change in voltage angle and J is a matrix of partial derivatives known as a Jacobian.

The Jacobian matrix J is given as EFE [19].

$$J=\begin{bmatrix}\dfrac{\partial P}{\partial\delta} & \dfrac{\partial P}{\partial|V_i|}\\ \dfrac{\partial Q}{\partial\delta} & \dfrac{\partial Q}{\partial|V_i|}\end{bmatrix} \tag{11}$$

The voltage magnitude and phase angles are updated as follows EFE [19].

$$\delta_i^{(K+1)}=\delta_i^K+\Delta\delta_i^K \tag{12}$$

$$\left|V_i^{(K+1)}\right|=\left|V_i^K\right|+\Delta\left|V_i^K\right| \tag{13}$$

where $\Delta\delta_i^{(k)}$ is the change in calculated angle; $\Delta\left|V_i^K\right|$ is the difference between voltage magnitude at bus i; $\left|V_i^{(k)}\right|$ is the most recently voltage magnitude value at bus i; K and $(K+1)$ denote previous and next iteration respectively.

2.5. Overview of Machine Learning Techniques

Machine learning techniques have become such influential tools in contemporary science and engineering fields because they provide data-driven solutions that enhance or even outperform conventional analytic methodologies. ML techniques are capable of learning patterns and correlations from historical data to make correct predictions, classification, and decision-making operations possible without complex mathematical rigor [27, 30, 31].

These techniques have a broad spectrum, ranging from supervised learning methods such as linear regression, support vector machines, and neural networks to unsupervised techniques such as k-means clustering and principal component analysis as illustrated in Figure 2. The adaptability, scalability, and capacity for real-time applications make ML techniques indispensable in addressing emerging challenges in smart grid systems, autonomous vehicles, financial forecasting, and beyond [31]. The ML techniques considered for investigation in this study are ANN, PINN and GNN. A brief description and the fundamental equations of each of these techniques are subsequently presented.

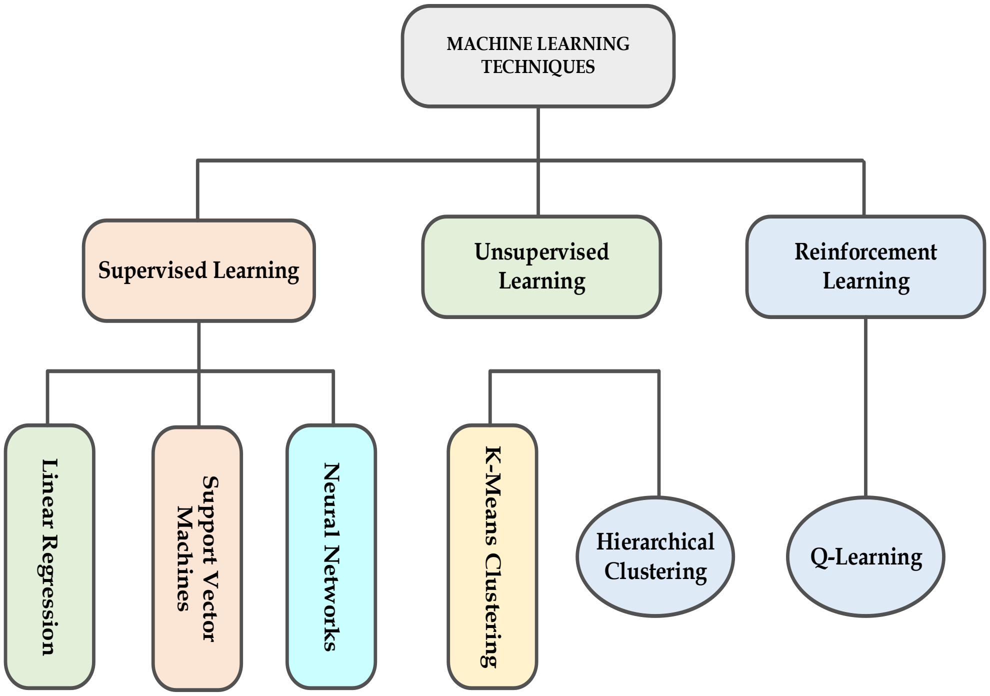

Figure 2.
Classification of Machine Learning Techniques Alnuaimi and Albaldawi [30].

2.5.1. Artificial Neural Networks

Artificial neural network is a common ML technique modeled to process information based on the biological brain. It consists essentially of several interconnected functional units generally referred to as artificial neurons. The neuron receives numerical inputs, processes it internally and produces an output. These neurons are arranged in a layered structure to form a network capable of executing parallel and distributed computations [32-34]. The architecture of a simple ANN with a three-layer network is as shown in Figure 3.

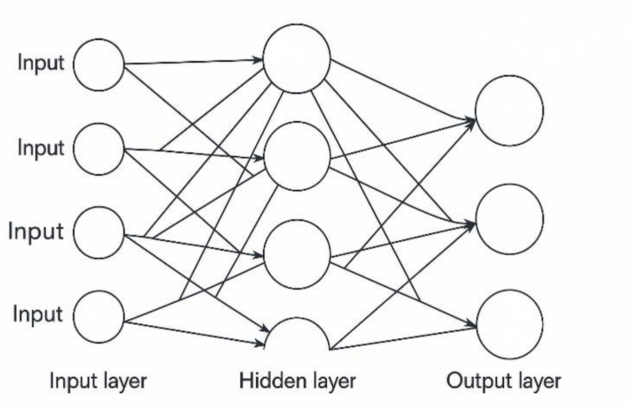

Figure 3.
Simple Architecture of a 3-layered Artificial Neural Network Xue, et al. [17].

For each neuron in l layer, the input is a weighted sum of the outputs from the previous layer, $(l-1)$ Therefore, the weighted sum can be computed using equation (14) [34].

$$z_j^{(l)} = \sum_{i=1}^{m^{(l-1)}} \omega_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)} \tag{14}$$

where $z_j^{(l)}$ represents the input to neuron j in layer l; $\omega_{ji}^{(l)}$ represents the weight from neuron i in layer $(l-1)$ to neuron j in layer l; $a_i^{(l-1)}$ represents the activation of neuron i in layer $(l-1)$ and $b_j^{(l)}$ represents the bias of neuron j in layer l

The activation function transforms the input of the neuron into its non-linear output, and it is computed as Aydin and Gümüş [34].

$$a_j^{(l)} = \sigma\left(z_j^{(l)}\right) \tag{15}$$

The loss function which defines the error between predicted output $\hat{x}$ and actual target, x_i is computed using equation (16).

$$\mathcal{L}_{ANN} = \frac{1}{t_s}\sum_{i=1}^{t_s}|\hat{x} - x_i|^2 \tag{16}$$

Generally, the chain rule is used to compute derivatives of the loss with respect to weights and biases using equation (17) [34].

$$\frac{\partial \mathcal{L}}{\partial \omega_{ji}^{(l)}} = \frac{\partial \mathcal{L}}{\partial a_j^{(l)}} \cdot \frac{\partial a_j^{(l)}}{\partial z_j^{(l)}} \cdot \frac{\partial z_j^{(l)}}{\partial \omega_{ji}^{(l)}} \tag{17}$$

Finally, the weights and biases are updated using equations (18) and (19) respectively [34].

$$\omega_{ji}^{(l)} \leftarrow \omega_{ji}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \omega_{ji}^{(l)}} \tag{18}$$

$$b_j^{(l)} \leftarrow b_j^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial b_j^{(l)}} \tag{19}$$

2.5.2. Physics-Informed Neural Network

Physics-informed neural network is a modern ML technique used in solving differential equations by blending the expressive power of neural networks with the rigor of physical laws [35]. Compared to conventional data-driven models like ANN which rely solely on observed data, the neural network in PINN incorporates governing physical equations such as those derived from conservation laws, fluid dynamics, or electromagnetism directly into the training process. This incorporation is accomplished through the formulation of a loss a loss function that penalizes deviations from both empirical data and the underlying physical principles, as well as the initial and boundary conditions [35-37]. PINNs provide more robust and generalizable solutions while reducing the need for extensive labeled datasets [37]. The architecture of a simple PINN model is depicted in Figure 4.

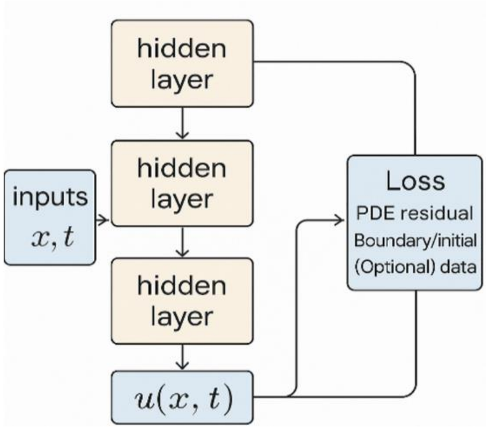

Figure 4.
Basic Model of Physics Informed Neural Network Tipu, et al. [35].

Generally, the PINN model can be constructed by integrating the ANN architecture with the governing physics described by differential equations. It aims to embed the physical laws directly into the loss function as described in equations (20) to (23) Tipu, et al. [35] and Ghalambaz, et al. [37].

$$\mu_\vartheta(x) \approx \mu(x) \qquad (20)$$

where $\mu_\vartheta(x)$ represents the output of the neural network parameterized by weights and biases, ϑ and $\mu(x)$ represents the inputs to the model.

Physical law enforcement, $f_\vartheta(x)$ is expressed as Tipu, et al. [35] and Ghalambaz, et al. [37].

$$f_\vartheta(x) = \aleph[\mu_\vartheta(x)] \qquad (21)$$

where $\aleph[\cdot]$ represents the governing differential operator

As reported in the study of Tipu, et al. [35] it is necessary to ensure the residual is enforced across all collocation points. Hence, the loss function is computed using equation (22) as follows Tipu, et al. [35] and Ghalambaz, et al. [37].

$$\mathcal{L}_{PINN} = \lambda_u \mathcal{L}_{data} + \lambda_f \mathcal{L}_{physics} + \lambda_{bc} \mathcal{L}_{bc} + \lambda_{ic} \mathcal{L}_{ic} \qquad (22)$$

where $\lambda_u \mathcal{L}_{data}$ represents the difference between the predicted and observed data; $\lambda_f \mathcal{L}_{physics}$ represents the governing physical laws; $\lambda_{bc} \mathcal{L}_{bc}$ represents the boundary and initial condition losses and $\lambda_{ic} \mathcal{L}_{ic}$ represents the weighting coefficients.

The model is trained by minimizing the loss function using an optimizer which is given as Ghalambaz, et al. [37].

$$\vartheta^* = arg \begin{bmatrix} min \\ \vartheta \end{bmatrix} \mathcal{L}_{PINN} \qquad (23)$$

2.5.3. Graph neural networks

Graph neural networks belong to the category of deep learning models which are primarily designed to handle data structured as graphs, where the relationships among data points are as critical as the data itself [38-41]. When compared to conventional neural networks that assume grid-like or Euclidean structures, GNNs excel in learning from non-Euclidean data structures such as social networks, power grids, chemical molecules or transportation networks [39]. The basic model of GNN is illustrated in Figure 5. By providing options and preferences to extract, generalize, and represent both non-graph and graph data into network modelling systems, it makes it possible to predict the accurate behavior of electric power system for efficient operational planning and maintenance [38].

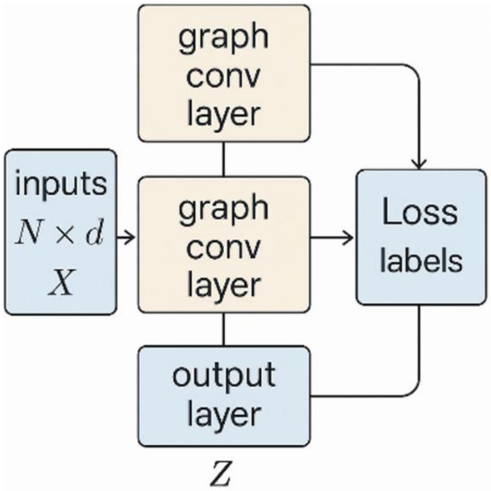

Figure 5.
Simple Model of Graph Neural Network Vrahatis, et al. [38].

The predominant variant of GNNs reported in literature are Graph Convolutional Networks (GCN), which was employed in this study [40]. The fundamental equations are described as follows Jia, et al. [40].

The input node, H^0 which represents the input features of all the nodes in the graph is defined as;

$$H^0 = X \tag{24}$$

The node representation at layer $(l+1)$ is given as;

$$h_i^{(l+1)} = \sigma\left(\sum_{j\epsilon\aleph(i)} f\left(h_i^{(l)}, h_j^{(l)}, e_{ij}\right)\right) \tag{25}$$

where $h_i^{(l)}$ represents the feature vector of node i at layer l; $h_j^{(l)}$represents the feature vector of node j at layer l; $\aleph(i)$represents the set of neighbors of node i; e_{ij} represents the feature of edge from node j to i and f represents the message aggregation function.

The update rule of the GCN is given as;

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\tilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{26}$$

where $\tilde{A} = A + I$ is the adjacent matrix with added self-loops; $\widetilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ is the degree matrix of $\tilde{A}$ and $H^{(0)} = X$ represents the input node features

The attention mechanism of the GCN over neighbors is expressed as;

$$h_i^{(l+1)} = \sigma\left(\sum_n \alpha_{ij}^{(l)} W^{(l)} h_j^{(l)}\right) \tag{27}$$

While the attention coefficient, $\alpha_{ij}^{(l)}$ between node i and neighboring node j at layer l, reflecting the importance of node j's feature to node i is given as;

$$\alpha_{ij}^{(l)} = \frac{exp\left(LeakyReLU\left(a^{\tau}\left[Wh_i^{(l)}\right]\left[Wh_j^{(l)}\right]\right)\right)}{\sum_{k\epsilon\aleph(i)} exp\left(LeakyReLU\left(a^{\tau}\left[Wh_i^{(l)}\right]\left[Wh_k^{(l)}\right]\right)\right)} \tag{28}$$

where W is the trainable weight matrix applied to the input features shared across all nodes and edges in the graph layer; a^{τ} is the trainable weight vector used in computing attention scores; $LeakyReLU$ is an activation function used to introduce non-linearity into the attention computation (usually a small negative slope value) and exp is an exponential function to ensure positive attention scores and enable softmax normalization.

The final output layer is expressed as;

$$Z = softmax\left(H^{(L)}\right) \tag{29}$$

where $H^{(L)}$ represents the output after L layers of the GNN and Z is the predicted labels for each node or graph.

The loss function for the GCN is computed using;

$$\mathcal{L}_{GNN} = -\sum_{i\epsilon y_L} y_i \cdot \log Z_i \tag{30}$$

where y_L is a set of labeled nodes; y_i represents the true label of node i and Z_i is the predicted probability distribution over classes.

3. Description of Test Systems

To assess the performance and generalization capability of the proposed neural network-based load flow models, comprehensive experiments were carried out on both standard IEEE benchmark systems and real-time practical power networks. Specifically, the IEEE 14-bus and 30-bus systems were employed as standard test cases, while the Nigerian 28-bus and 52-bus systems were selected to reflect the operational characteristics of actual grid infrastructures. The IEEE 14-bus system comprises 14 buses, 20 transmission lines, 5 generators, and 11 loads. In contrast, the IEEE 30-bus system features a more intricate network topology with 6 generators, 24 loads, and 41 branches. This increased complexity facilitates a more rigorous assessment of the model's scalability and its ability to manage larger, interconnected systems.

Beyond the standard benchmarks, two practical systems from the Nigerian power grid were also considered. The Nigerian 28-bus system represents a medium-scale network, it comprises of 8 effective generators, 11 loads and 33 transmissions. Lastly, the Nigerian 52-bus system offers a broader and more comprehensive view of the Nigerian transmission network. It comprises of 11 generators, 21 loads and 36 transmissions. With diverse load centers, high-voltage interconnections, and a wide geographic spread, this system presents a highly practical scenario for testing the model's effectiveness under realistic, non-uniform loading and generation conditions.

4. Methodology

4.1. Study Approach

This study evaluates the performance of three neural network models, ANN, PINN and GNN for load flow analysis of power systems. The input dataset is obtained through NR load flow solution by varying load patterns. The dataset is split into training (50%), validation (25%) and testing (25%). As shown in Figure 6, The ANN model was trained via back propagation in MATLAB, while PINN was trained in TensorFlow and the GNN was trained in PyTorch geometric. To evaluate resilience, models are tested under fault conditions such as line outages and retrained to reflect this scenario. DG units such as PV are randomly integrated at weak buses to simulate smart grid enhancements. This approach assesses each model's capability to maintain reliable operation in both normal and disrupted states.

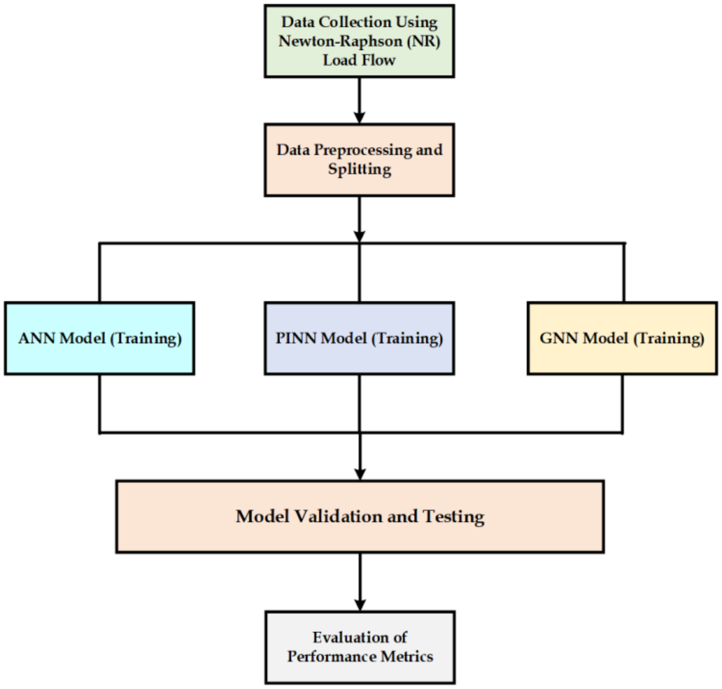
Data Collection Using Newton-Raphson (NR) Load Flow
Data Preprocessing and Splitting
ANN Model (Training)
PINN Model (Training)
GNN Model (Training)
Model Validation and Testing
Evaluation of Performance Metrics

Figure 6.
Block Diagram of the Methodology Flow Process.

4.2. Data Collection

The data utilized in this study were obtained from a combination of standard benchmark repositories and utility-based system records. For the IEEE 14-bus and 30-bus systems, network topology, bus data, and line parameters were sourced from open literature. These datasets include information on bus types, voltage levels, power generation, load demands, and branch admittances. On the other hand, the Nigerian 28-bus and 52-bus systems were constructed using operational data acquired from national control center in Osogbo, Nigeria.

4.3. Implementation of Neural Network Models for Load Flow Analysis

This section presents the implementation of neural network models for load flow analysis. The ANN model was designed as a traditional fully connected network trained on labeled load flow data to approximate the non-linear relationship between power injections and bus voltages. The PINN model incorporated physical power flow equations directly into the loss function. The GNN model leveraged the graph structure of power systems by modeling buses as nodes and transmission lines as edges, capturing the spatial dependencies within the network. The mathematical formulation of each model for load flow analysis is presented in the following sub-sections.

4.3.1. Implementation of ANN

The ANN is used to approximate the nonlinear mapping from bus power injections to bus voltage magnitudes and angles for load flow analysis as illustrated in Figure 7.

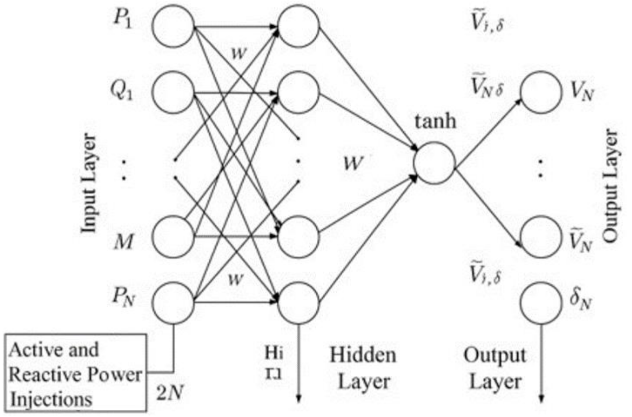

Figure 7.
Architecture of ANN for load flow analysis.

The input vector x comprises active and reactive power injections at all buses:

$$x = [P_1, Q_1, P_2, Q_2, \dots\dots P_N, Q_N]^\tau \in \mathbb{R}^{2N} \tag{31}$$

The output vector, y contains all the predicted bus voltage magnitudes and angles and it is given as;

$$y = [V_1, \delta_1, V_2, \delta_2, \dots .V_N, \delta_N]^\tau \in \mathbb{R}^{2N} \tag{32}$$

where P and Q are the active and reactive power injections, V and δ are the voltage magnitude and angle at the buses.

This mapping is modeled as;

$$y = f_{ANN}(x) \tag{33}$$

where the term, f_{ANN} approximates the power system load flow solution.

The ANN model architecture consists of an input layer, a single hidden layer with nonlinear activation, and an output layer with linear activation. The forward propagation is given by Tipu, et al. [35];

$$z^{(l)} = W^{(1)}(x) + b^{(1)}; \quad a^{(1)} = tanh\left(z^{(1)}\right) \tag{34}$$

$$z^{(2)} = W^{(2)}a^{(1)} + b^{(2)}; \quad \hat{y} = z^{(2)} \tag{35}$$

where $W^{(l)}$ and $b^{(l)}$ represent the weight matrices and bias vectors for layer l, a is the hidden layer activation, and $\hat{y}$ is the estimated output.

Training of the ANN is conducted using supervised learning with a dataset, $\{(X_k, Y_k)\}_{k=1}^{M}$. The objective is to minimize the MSE loss function given as;

$$\mathcal{L}_{ANN}(\pi) = \frac{1}{M}\sum_{k=1}^{M}\|y_k - \hat{y}_k\|^2 = \frac{1}{N}\sum_{i=1}^{N}\left(V_i^p - V_i^{true}\right)^2 + \left(\delta_i^p - \delta_i^{true}\right)^2 \tag{36}$$

where V_i^p and δ_i^p represent the predicted voltage magnitude and angle; V_i^{true} and δ_i^{true} represent the true values of voltage and angle from the power flow solution.

The gradient and the hidden layer error are computed using equations (37) and (38), respectively.

$$\partial^{(2)} = \hat{y} - y \tag{37}$$

$$\partial^1 = \left(W^{(2)^\tau}\partial^{(2)}\right)\circ\left(1 - a^{(1)}\circ a^{(1)}\right) \tag{38}$$

where ° represents element-wise multiplication and $tanh$ activation derivative is applied.

The weights and biases update iteratively using the learning rate, η which is given in equation (39) Tipu, et al. [35].

$$W^{(l)} \longleftarrow W^{(l)} - \eta\partial^{(l)}\left(a^{(l-1)}\right)^\tau; \quad b^{(l)} \longleftarrow b^{(l)} - \eta\partial^{(l)} \tag{39}$$

The training is continued until the MSE converges below a specified threshold or a maximum epoch count has been reached.

4.3.2. Implementation of PINN

PINN embeds the governing nonlinear power balance equations directly into the training loss function. This approach leverages both data-driven learning and physical constraints, enabling the model to approximate bus voltage magnitudes, V_i and voltage angles, δ_i from given power injections while satisfying Kirchhoff's laws and system admittances. Its architecture consists of a fully connected feed-forward neural network with an input layer, multiple hidden layers employing nonlinear activation functions and an output layer with linear activation as shown in Figure 8.

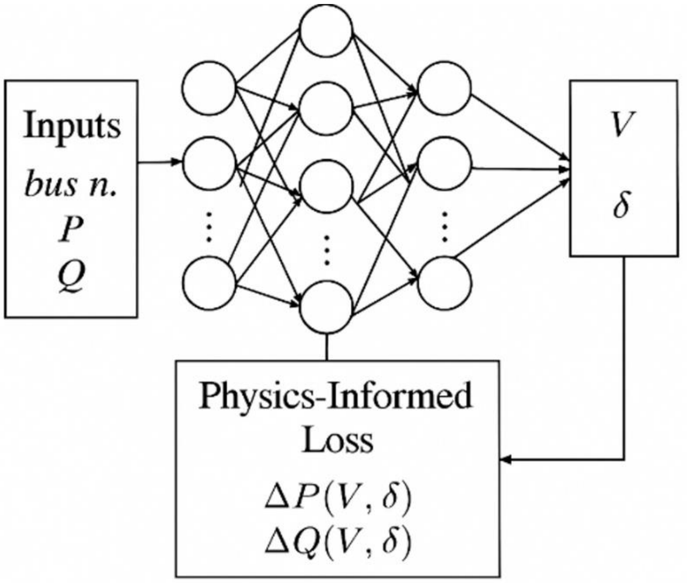

Figure 8.
PINN for Load Flow Analysis.

The input vector, x contains all the active and reactive power injections at all N buses, including load, generator and slack bus injections as modeled in equation (31), while the output vector, y contains all the predicted bus voltage magnitudes and angles as modeled in equation (32). The fundamental load flow equations serve as the physics-based residuals incorporated into the PINN loss function. Hence, for each bus $i \in \{1,2, \ldots . N\}$, the active and reactive power balance residuals, $\mathcal{R}_{Pi}(y)$ and $\mathcal{R}_{Qi}(y)$ are computed using:

$$\mathcal{R}_{Pi}(y) = P_{Gi} - P_{Li} - \sum_{j=1}^{N} Y_{ij}V_iV_j cos\left(\delta_i - \delta_j - \gamma_{ij}\right) = 0 \tag{40}$$

$$\mathcal{R}_{Qi}(y) = Q_{Gi} - Q_{Li} - \sum_{j=1}^{N} Y_{ij} V_i V_j cos\left(\delta_i - \delta_j - \gamma_{ij}\right) = 0 \quad (41)$$

It should be noted that the slack bus voltage magnitude and angle, as well as the line admittances are enforced via fixed boundary conditions, while the PV buses have voltage magnitude constrained to a specified setpoint, with the network estimating reactive power accordingly.

The PINN training minimizes a composite loss function, $\mathcal{L}_{PINN}(\pi)$ which is computed as;

$$\mathcal{L}_{PINN}(\pi) = \mathcal{L}_{data} + \lambda \mathcal{L}_{physics} \quad (42)$$

where π denotes the neural network parameters (weights and biases), and λ is a hyperparameter balancing the two terms.

The data loss function represents the MSE between predicted outputs and the NR load flow solutions and computed using;

$$\mathcal{L}_{data} = \frac{1}{M} \sum_{k=1}^{M} \|y_k - \hat{y}_k\|^2 \quad (43)$$

While the physics loss function represents the MSE of residuals over all buses and samples enforcing power flow constraints and computed using;

$$\mathcal{L}_{physics} = \frac{1}{M} \sum_{k=1}^{M} \sum_{i=1}^{N} \left(\mathcal{R}_{Pi}(\hat{y}_k)^2 + \mathcal{R}_{Qi}(\hat{y}_k)^2\right) \quad (44)$$

The expression in equation (44) is used to penalize any kind of deviations from the physical laws embedded in the network output.

4.3.3. Implementation of GNN

The GNN leverages the inherent graph-structured nature of power systems, enabling spatially aware and scalable load flow solutions by explicitly modeling buses and lines as graph components. It learns by mapping nodal power injections to bus voltage magnitudes and angles by aggregating information along transmission network edges as depicted in Figure 9.

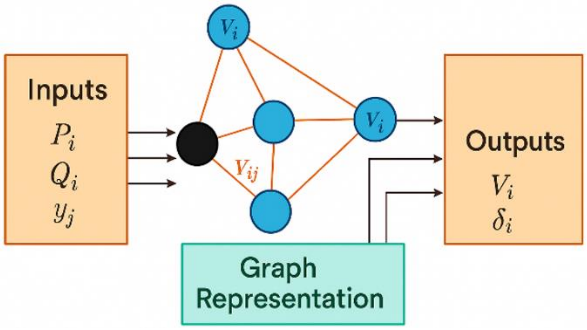

Figure 9.
GNN for Load Flow Analysis.

The power system is modeled as a directionless graph $G_p = (\cup, \varepsilon)$; where $\cup = \{1,2, \ldots, N\}$ represents the set of N nodes (buses) in the power system and $\varepsilon \subseteq \cup \times \cup$ represents the set of edges (transmission lines) connecting the buses. Each node $i \in \cup$ is associated with an input vector given as;

$$h_i^{(0)} = \left[P_i, Q_i, P_G, V_{ref}, B_{ti\ldots\ldots}\right]^{\tau} \quad (45)$$

The input vector given in equation (45) includes active/reactive power injections, generated active power where applicable, reference voltage magnitude for slack or PV buses and bus type indicators, B_{ti}.

Each edge $(i,j) \in \varepsilon$ is characterized by the line admittance parameters representing the magnitude, $|Y_{ij}|$ and angle, γ_{ij} of the transmission as given in equation (46).

$$e_{ij} = \left[|Y_{ij}|, \gamma_{ij}\right]^{\tau} \quad (46)$$

DOI: 10.55214/2576-8484.v9i9.10161

The GNN iteratively updates node embeddings through T message passing layers to capture spatial dependencies [38]. At layer $t \in \{1, \ldots\ldots T\}$, the embedding $h_i^{(t)}$ of node i is updated using equations (47) and (48) [38].

$$m_i^{(t)} = \sum_{j \in \aleph(i)} \varphi_m \left(h_i^{(t-1)}, h_j^{(t-1)}, e_{ij} \right) \tag{47}$$

$$h_i^{(t)} = \varphi_u \left(h_i^{(t-1)}, m_i^{(t)} \right) \tag{48}$$

where φ_m is a learnable message function and φ_u is a s a learnable update function

It should be noted that equations (47) and (48) are used for the passing message and nodes update respectively. After T layers, the final node embeddings $h_i^{(T)}$encode both local features and global network context.

Each node's voltage magnitude and angle are predicted by a readout function, β applied to its final embedding as;

$$\left[\hat{V}, \hat{\delta}\right] = \beta \left(h_i^{(T)} \right) \tag{49}$$

The GNN model is trained in a supervised manner using datasets $\{(X_k, Y_k)\}_{k=1}^{M}$; where X_k contains the nodal power injections and grid topology; and Y_k contains the ground truth voltage profiles obtained from NR load flow solutions. Hence, the training loss minimizes the mean squared error using;

$$\mathcal{L}_{GNN}(\pi) = \frac{1}{M} \sum_{k=1}^{M} \sum_{i=1}^{N} \left(\left(V_i^{(k)} - \hat{V}_i^{(k)} \right)^2 + \left(\delta_i^{(k)} - \hat{\delta}_i^{(k)} \right)^2 \right) \tag{50}$$

The computational efficiency of the GNN is improved by enforcing power balance residuals as in the case of the PINN model to improve generalization and physical consistency.

4.4. Performance Evaluators

The performance of the models was evaluated using Line Voltage Stability Index (LVSI) and a set of statistical evaluators such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as metrics. A brief description and mathematical representation of each of the metrics is subsequently presented.

4.4.1. Line Voltage Stability Index (LVSI)

The LVSI was developed by Kessel and Glavitsch; it gauges how close each transmission line is to voltage collapse by comparing its reactive-power margin with its loading level. It can be computed using [42].

$$L_i = 1 - \left| \sum_{j \in G} F_{ij} \frac{V_j}{V_i} \right| \tag{51}$$

where G is the set of generator buses; F_{ij} is derived from the load flow Jacobian or Y-bus and V_j and V_i are the voltages of generator and load buses.

4.4.2. Mean Squared Error (MSE)

The MSE measures the square-root of the average squared deviation between the neural network's predicted voltage or power values and the reference NR load-flow solutions [43].

$$MSE = \frac{1}{t_s} \sum_{i=1}^{t_s} |\hat{x} - x_i|^2 \tag{52}$$

4.4.3. Root Mean Squared Error (RMSE)

The RMSE is the square-root of the mean of squared differences between the network's predicted voltages and the reference NR load-flow results [43, 44].

$$RMSE = \sqrt{\frac{1}{t_s} \sum_{i=1}^{t_s} |\hat{x} - x_i|^2} \tag{53}$$

4.4.4. Mean Absolute Error (MAE)

The MAE computes the average of the absolute differences between the neural network's predictions and the reference NR load-flow solutions [43, 44].

$$MAE = \frac{1}{t_s}\sum_{i=1}^{t_s}|\hat{x} - x_i| \tag{54}$$

4.4.5. Mean Absolute Percentage Error (MAPE)

The MAPE expresses the mean absolute deviation between predicted load-flow values as a percentage of the reference NR load flow solutions [43, 44].

$$MAPE = \frac{100}{t_s}\sum_{i=1}^{t_s}\frac{|\hat{x}-x_i|}{x_i} \tag{55}$$

where x_i represents true value obtained; $\hat{x}$ represents the predicted value of the sample outcome, and t_s is the total sample number.

5. Results and Discussion

The results of simulation are presented and discussed in this section. It provides a detailed comparative evaluation of the neural network models for load flow analysis, with performance examined under three operating conditions of the test systems.

1. Scenario 1: Base case: - steady-state condition without any system alterations.
2. Scenario 2: Fault condition: - evaluation of robustness when system faults are introduced.
3. Scenario 3: PV DG integration: - assessment under random integration of PV DG.

To ensure comprehensive assessment, the simulation results of each scenario are analyzed and discussed using performance metrics MSE, RMSE, MAE, MAPE, LVSI and Computational Time (CT). The simulation results of each scenario are subsequently presented.

5.1. Simulation Results of Scenario 1 (Steady-State Condition)

For scenario 1, the results of voltage profile comparison across the test systems using the three neural network models with NR method included as reference is illustrated in Figure 10. The voltage profile comparison shows that the performance of ANN, PINN, and GNN varies with system size, with PINN and GNN generally providing more stable voltage magnitudes closer to nominal values (0.95-1.05 p.u.) than ANN. For the 14-bus system, PINN and GNN remained near nominal, while ANN slightly deviated at the upper limit. In the 30-bus system, PINN had slight deviations at the upper limit, while ANN and GNN stayed closer to nominal. However, for the 28-bus system, PINN and GNN performed better, while ANN deviated towards the lower limit. In the 52-bus system, PINN maintained stability, while ANN and GNN deviated toward upper and lower voltage limits.

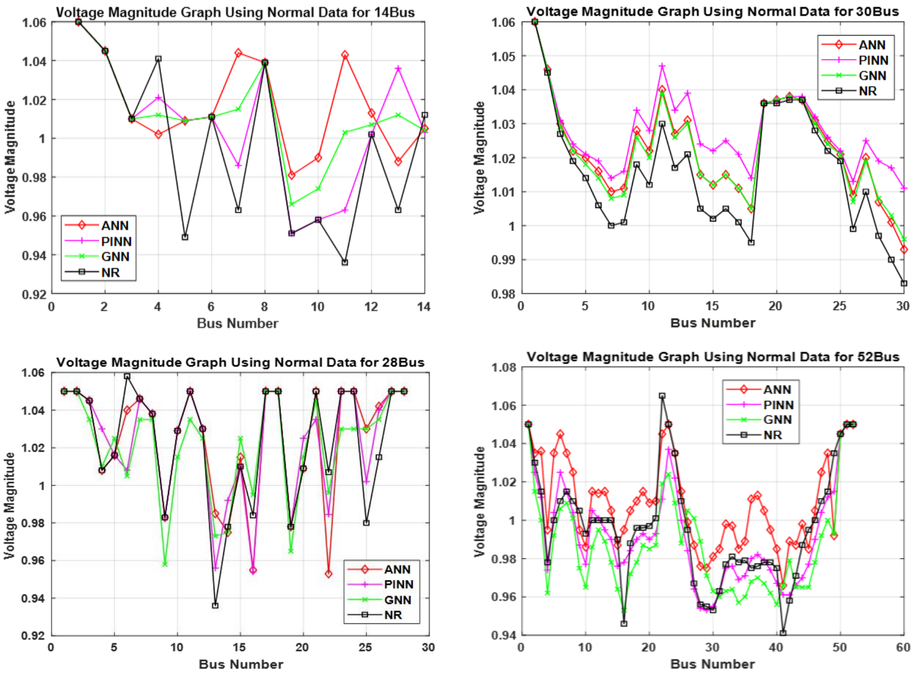

Figure 10.
Comparison of voltage profile across test systems for scenario 1.

The outcomes suggest that PINN, which embeds physical laws into its architecture, tends to maintain voltage magnitudes across larger systems. GNN leverages topological information, making it robust in smaller and moderately sized systems where connectivity strongly impacts performance. ANN, relying purely on data-driven learning, is more prone to voltage deviations as it lacks explicit incorporation of physical or structural constraints. For larger or more complex networks, PINN appears to generalize better, making it more suitable for real-world power system applications where accuracy in voltage stability is critical. This highlights the practical value of physics-informed and graph-based approaches over traditional ANN, particularly in ensuring robust and accurate load flow solutions under varying system sizes.

The performance metrics of the three models across all test systems are summarized in Table 2. The results indicate that ANN consistently achieved the lowest values of MSE, RMSE, MAE, MAPE and CT, alongside the highest LVSI. For the IEEE 14-bus system, ANN achieved the lowest error metrics with an MSE of 1.039, RMSE of 1.019, MAE of 0.091, MAPE of 4.887%, alongside shortest CT of 0.080 minutes and the highest LVSI of 0.778. PINN recorded highest errors (MSE: 2.370; RMSE: 1.540; MAE: 0.190; MAPE: 8.230%), longest CT of 0.170 minutes with a lower LVSI of 0.673, while GNN achieved intermediate error values (MSE: 1.520; RMSE: 1.230; MAE: 0.130; MAPE: 6.480%) longer CT and lowest LVSI of 0.120 minutes and 0.621 respectively.

Table 2.
Summary of performance metrics across test systems for scenario 1.

Test Systems	**Performance Metrics**	**Neural Network Models**		
		ANN	**PINN**	**GNN**
IEEE 14-Bus	MSE	1.039	2.370	1.520
	RMSE	1.019	1.540	1.230
	MAE	0.091	0.190	0.130
	MAPE	4.887	8.230	6.480
	LVSI	**0.778**	**0.673**	**0.621**
	Computation time	0.080	0.170	0.120
IEEE 30-Bus	MSE	0.449	1.870	0.832
	RMSE	0.670	1.368	0.912
	MAE	0.043	0.182	0.078
	MAPE	2.324	7.429	3.579
	LVSI	0.798	0.672	0.724
	Computation time	0.170	0.370	0.260
Nigerian 28-Bus	MSE	0.649	1.974	1.183
	RMSE	0.806	1.405	1.088
	MAE	0.048	0.171	0.105
	MAPE	2.577	6.750	4.872
	LVSI	0.791	0.664	0.716
	Computation time	0.16	0.34	0.24
Nigerian 52-Bus	MSE	0.680	1.935	1.217
	RMSE	0.825	1.391	1.103
	MAE	0.066	0.177	0.120
	MAPE	3.520	6.908	5.651
	LVSI	**0.597**	**0.563**	**0.524**
	Computation time	0.300	0.640	0.450

Similar trends were observed in the other test systems, where ANN outperformed both PINN and GNN models. However, in the 14-bus and Nigerian 52-bus systems, PINN delivered a higher LVSI (0.673 and 0.563) compared to that of GNN (0.621 and 0.524) respectively, indicating a stronger voltage stability margin. The results indicate that minimizing error metrics does not necessarily guarantee stable voltage profiles at individual buses.

While GNN performs well in error reduction, it is less stable in voltage outcomes, PINN on the other hand offers moderate accuracy with slightly higher stability in small and large systems, highlighting a trade-off between predictive accuracy and practical stability. Consequently, while ANN offers overall superior accuracy, PINN and GNN may be more suitable when the need to maintain consistent bus-level voltage stability is prioritized. The convergence characteristics comparison plot of the models for scenario 1 is depicted in Figure 11.

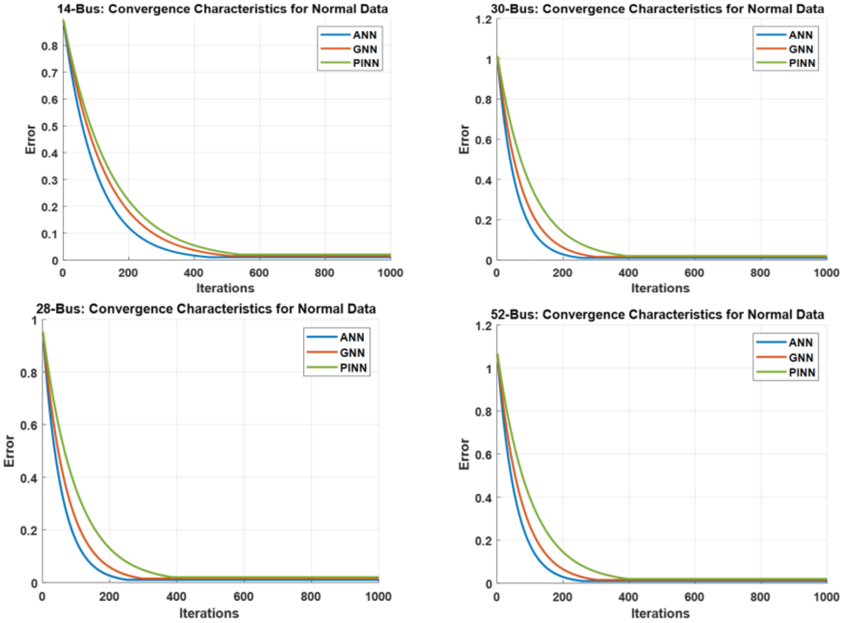

Figure 11.
Comparison of convergence characteristics plot across test systems for scenario 1.

5.2. Simulation Results of Scenario 2 (Fault Condition)

The voltage profile comparison for scenario 2 across test systems is depicted in Figure 12. The results revealed that the performance of the models varies with system disturbances. In the 14-bus system, both PINN and GNN suffered large deviations toward the lower voltage limits, whereas ANN maintained voltage magnitudes closer to the nominal values, indicating better small-system resilience. For the 30-bus system, the performance trends across the models were similar to Scenario 1, suggesting consistent behavior under moderate disturbances.

DOI: 10.55214/2576-8484.v9i9.10161

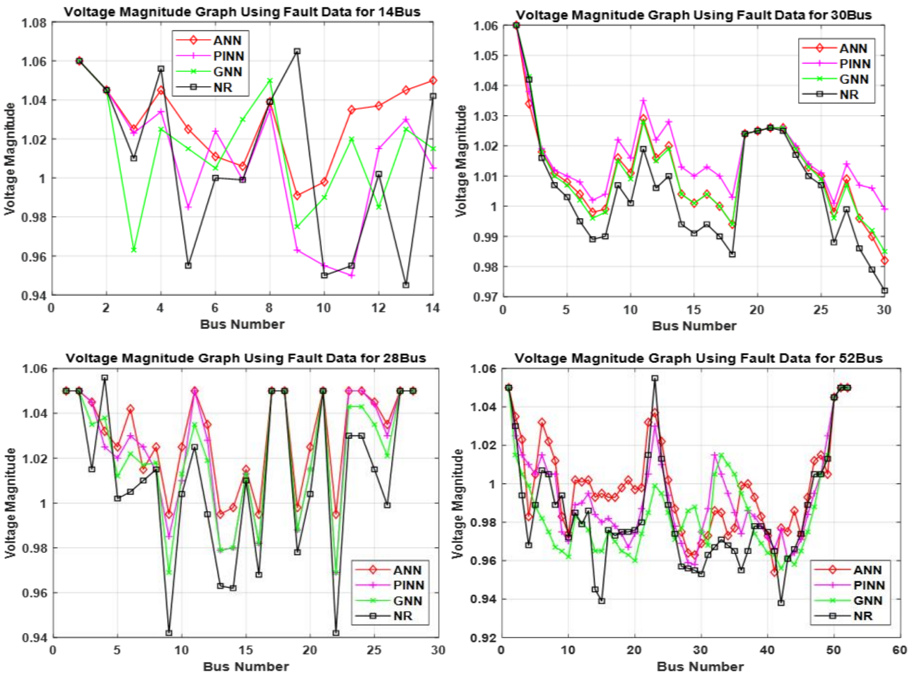

Figure 12.
Comparison of voltage profile across test systems for scenario 2.

Nevertheless, for the Nigerian 28-bus system, PINN showed deviations toward the lower limit but remained within acceptable bounds, while ANN and GNN deviate toward the upper limit, but also remained within nominal operating values. In contrast, for the Nigerian 52-bus system, PINN maintained voltage stability, while ANN and GNN exhibited deviations near the upper voltage limits. These findings suggest that while ANN demonstrates robust voltage control in smaller networks, PINN provides stronger stability in larger and more complex systems, with GNN offering intermediate performance. This implies that the choice of model should not only be informed by the system size as observed in scenario 1, but also disturbance characteristics, as reliance on a single model may compromise stability under certain operating conditions.

The performance of the three models in Scenario 2 across is summarized in Table 3. It is also obvious here that ANN consistently achieved the lowest error metrics, with MSE ranging from 0.415 to 1.079, RMSE from 0.644 to 1.039, MAE from 0.037 to 0.105, and MAPE from 2.019% to 5.562%, shortest CT (0.13-0.5), alongside the highest LVSI values (0.654-0.795). On the other hand, PINN recorded the highest errors (MSE: 1.622-2.305; RMSE: 1.274-1.518; MAE: 0.158-0.208; MAPE: 6.197-8.914%), longest CT (0.25-0.94) and lowest LVSI (0.523-0.611), while GNN showed intermediate performance in both error metrics and LVSI.

Table 3.
Summary of performance metrics across test systems for scenario 2.

Test Systems	Performance Metrics	Neural Network Models		
		ANN	PINN	GNN
IEEE 14-Bus	MSE	1.079	2.305	1.773
	RMSE	1.039	1.518	1.331
	MAE	0.105	0.202	0.168
	MAPE	5.562	8.914	8.316
	LVSI	**0.694**	**0.6111**	**0.563**
	Computation time	0.13	0.25	0.19
IEEE 30-Bus	MSE	0.415	1.622	0.789
	RMSE	0.644	1.274	0.888
	MAE	0.037	0.158	0.072
	MAPE	2.019	6.197	3.255
	LVSI	0.795	0.587	0.694
	Computation time	0.29	0.54	0.4
Nigerian 28-Bus	MSE	0.680	2.118	1.209
	RMSE	0.825	1.455	1.099
	MAE	0.057	0.183	0.106
	MAPE	3.019	7.338	4.918
	LVSI	0.791	0.586	0.690
	Computation time	0.27	0.51	0.37
Nigerian 52-Bus	MSE	0.572	2.179	1.445
	RMSE	0.756	1.476	1.202
	MAE	0.051	0.208	0.146
	MAPE	2.733	8.491	6.932
	LVSI	**0.654**	**0.523**	**0.609**
	Computation time	0.5	0.94	0.7

Nevertheless, a notable trend was observed in the 14-bus and 52-bus systems, where PINN achieved higher LVSI than GNN in scenario 1. In scenario 2, PINN maintained its superiority in LVSI for the 14-bus system, indicating better voltage stability in small networks. However, in the 52-bus system, this trend reversed, with GNN showing higher LVSI than PINN. This suggests that PINN's stability advantage can diminish in larger, stressed networks, highlighting that voltage stability depends on both model architecture and system-specific disturbances. The convergence characteristics comparison plot of the models for scenario 2 is depicted in Figure 13.

DOI: 10.55214/2576-8484.v9i9.10161

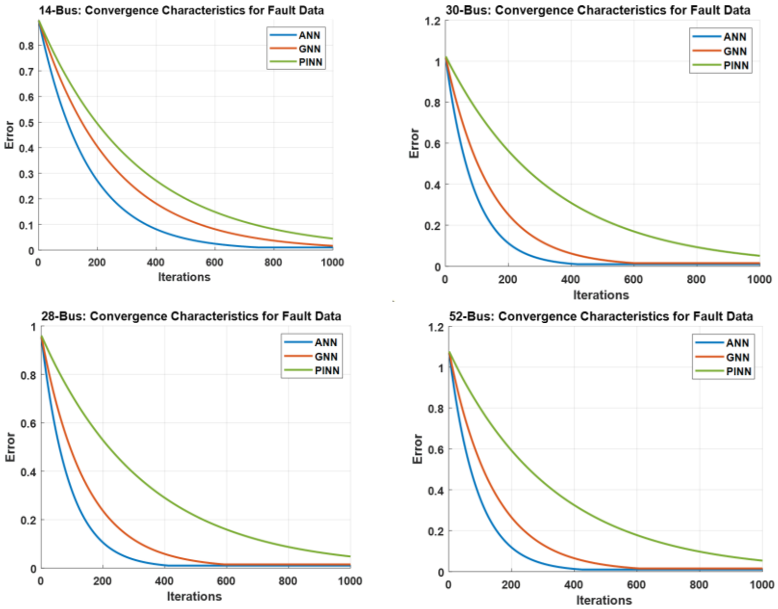

Figure 13.
Comparison of convergence characteristics plot across test systems for scenario 2.

5.3. Simulation Results of Scenario 3 (PV DG Penetration Condition)

The voltage profile comparison for Scenario 3 across the test systems is also illustrated in Figure 14. The results also indicate that the performance of the neural network models varies with the presence of randomly integrated PV-based DG. In the 14-bus system, all three models maintained voltages near nominal values, demonstrating robust performance under low network complexity. For the 30-bus system, ANN and GNN remained close to nominal, while PINN exhibited minor deviations toward the upper voltage limit.

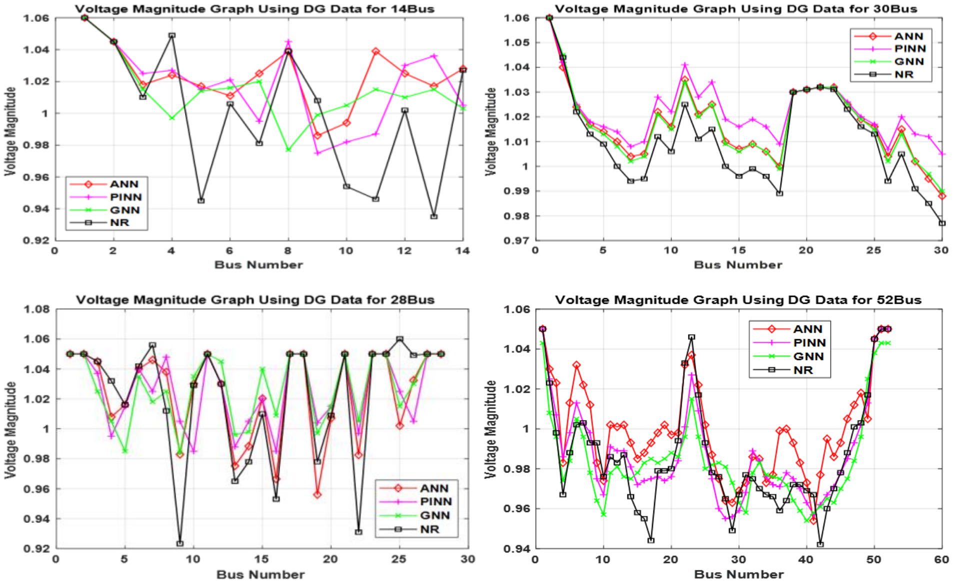

Figure 14.
Comparison of voltage profile across test systems for scenario 3.

DOI: 10.55214/2576-8484.v9i9.10161

For the 28-bus system, PINN and GNN maintained more stable voltages, whereas ANN deviated slightly toward the lower voltage limit. However, in the 52-bus system, both PINN and GNN effectively preserved nominal voltage levels, while ANN showed minor deviations near the upper voltage limit. These findings suggest that while ANN provides reliable voltage control in smaller networks, PINN and GNN offer stronger stability in larger networks under PV-based DG integration. This implies that the choice of neural network model for load flow analysis should consider both network size and level of DG penetration.

The performance metrics of the models across the test systems in Scenario 3 is summarized in Table 4. The results of Table 4 showed that there was a general improvement in the values of LVSI across the test systems compared to scenarios 1 and 2 due to the penetration of PV-based DG. ANN consistently achieved the lowest error metrics, with MSE ranging from 0.385 to 1.079, RMSE from 0.620 to 1.039, MAE from 0.037 to 0.105, and MAPE from 2.005% to 5.562%, and shortest CT ranging from 0.1 to 0.37, along with the highest LVSI values (0.796-0.870), indicating strong predictive accuracy and voltage stability even under PV-based DG fluctuations.

Table 4.
Summary of performance metrics across test systems for scenario 3.

Test Systems	Performance Metrics	Neural Network Models		
		ANN	**PINN**	**GNN**
IEEE 14-Bus	MSE	1.024	2.552	1.828
	RMSE	1.012	1.597	1.352
	MAE	0.096	0.238	0.180
	MAPE	5.146	10.770	8.903
	LVSI	0.87	0.752	0.696
	Computation time	0.1	0.21	0.15
IEEE 30-Bus	MSE	0.385	1.714	0.731
	RMSE	0.620	1.309	0.855
	MAE	0.037	0.166	0.068
	MAPE	2.005	6.612	3.073
	LVSI	0.817	0.705	0.707
	Computation time	0.21	0.44	0.31
Nigerian 28-Bus	MSE	0.539	2.109	1.197
	RMSE	0.734	1.452	1.094
	MAE	0.043	0.187	0.104
	MAPE	2.364	7.571	4.869
	LVSI	0.810	0.703	0.706
	Computation time	0.2	0.41	0.3
Nigerian 52-Bus	MSE	0.569	2.096	1.282
	RMSE	0.754	1.448	1.132
	MAE	0.051	0.198	0.124
	MAPE	2.731	7.956	5.809
	LVSI	0.796	0.601	0.714
	Computation time	0.37	0.77	0.55

PINN recorded the highest errors (MSE: 1.714-2.305; RMSE: 1.309-1.518; MAE: 0.166-0.202; MAPE: 6.612-8.914%), longest CT of 0.21 to 0.77 and lower LVSI (0.601–0.752), while GNN maintained its intermediate performance in error metrics, CT and LVSI. As is the case of scenarios 1 and 2, PINN maintained its superiority in LVSI over GNN for the 14-bus system, indicating better voltage stability in small networks, even in the presence of DG. But the reverse case remains the same for the 52-bus system, with GNN showing higher LVSI than PINN scenarios 2 and 3 as compared to scenario 1.

This pattern suggests that PINN's physics-informed approach consistently enhances voltage stability in small networks, even under the influence of DG. However, in larger networks, its relative advantage diminishes under disturbances or increased system complexity, allowing GNN to occasionally achieve higher LVSI. This highlights that the effectiveness of a neural network model in maintaining voltage stability is both network-size dependent and sensitive to system conditions, implying that model selection should consider not only accuracy but also the scale and operating characteristics of the network. The convergence characteristics comparison plot of the three models for scenario 3 is depicted in Figure 15.

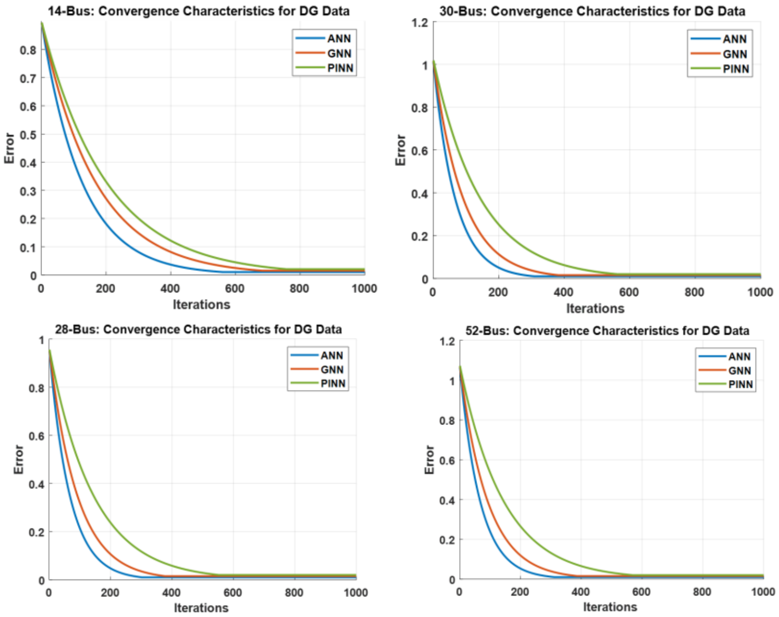

Figure 15.
Comparison of convergence characteristics plot across test systems for scenario 3.

5.4. Performance Comparison of Neural Network Models

This subsection presents a comparative analysis of ANN, PINN, and GNN for load flow prediction across the test systems under each scenario, highlighting trends in accuracy, LVSI and CT.

5.4.1. Performance of the Models on Standard 14-Bus and 30-Bus Systems

The performance comparison of the neural network models across scenarios on the 14-bus and 30-bus systems is illustrated in Figures 16 and 17 respectively. For the standard test systems, ANN consistently achieved the highest predictive accuracy across all operating conditions. For the 14-bus system, ANN recorded the lowest MSE, MAE and MAPE values in scenario 1, highlighting its robustness in smaller networks. Interestingly, while PINN had higher error metrics, its LVSI was greater than GNN in all 14-bus scenarios, indicating stronger voltage stability predictions despite lower overall accuracy.

DOI: 10.55214/2576-8484.v9i9.10161

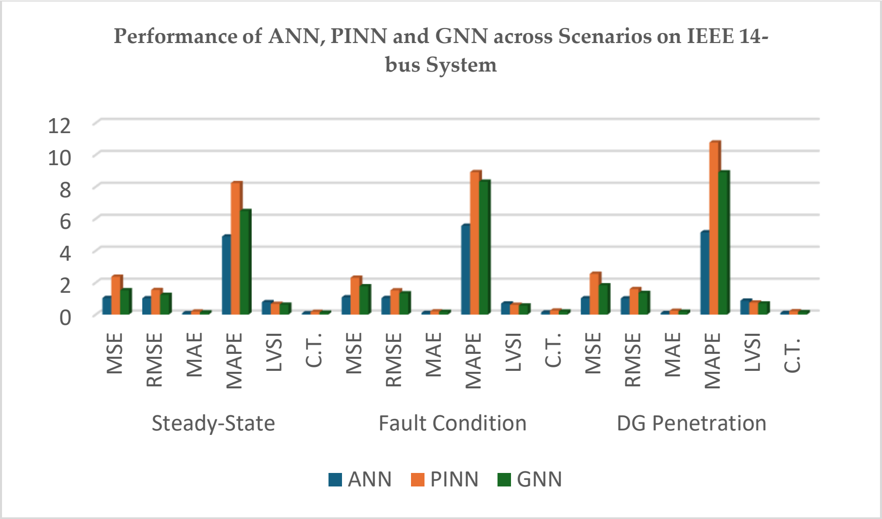

Figure 16.
Performance comparison of the neural network models on IEEE 14-bus system.

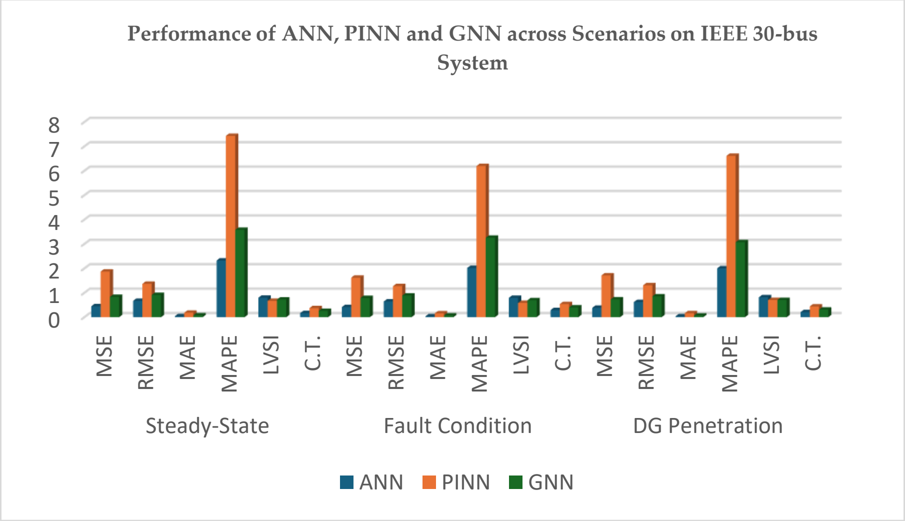

Figure 17.
Performance comparison of the neural network models on IEEE 30-bus system.

It was also observed that in scenarios 2, all models showed slight performance degradation, with increased errors and decreased LVSI, reflecting the increased complexity of stressed conditions. However, in scenario 3, all models showed significant improvement, particularly in the value of LVSI, reflecting the importance of DG integration for improved voltage stability. Generally, GNN maintained intermediate performance, outperforming PINN in accuracy but slightly behind ANN, while computation times were lowest for ANN, moderate for GNN, and highest for PINN.

5.4.2. Performance of the Models on Practical Nigerian 28-Bus and 52-Bus Systems

The performance comparison of the models across scenarios on the Nigerian 28-bus is illustrated in Figure 18. The diagram of figure 18 showed that ANN again demonstrated superior accuracy across all scenarios, with MSE and MAPE consistently lower than both GNN and PINN. The trend of slightly higher errors in scenario 2 and improved LVSI in scenario 3 persisted, indicating the expected challenge posed by stress conditions and benefits DG penetration, respectively. Overall, GNN provided a balance between accuracy and voltage stability, outperforming PINN in error metrics while maintaining reasonably high LVSI. PINN's LVSI was lower than ANN and GNN in this system, although it still contributed to voltage stability prediction. ANN maintained the fastest computation times, confirming its suitability for moderate-size practical systems.

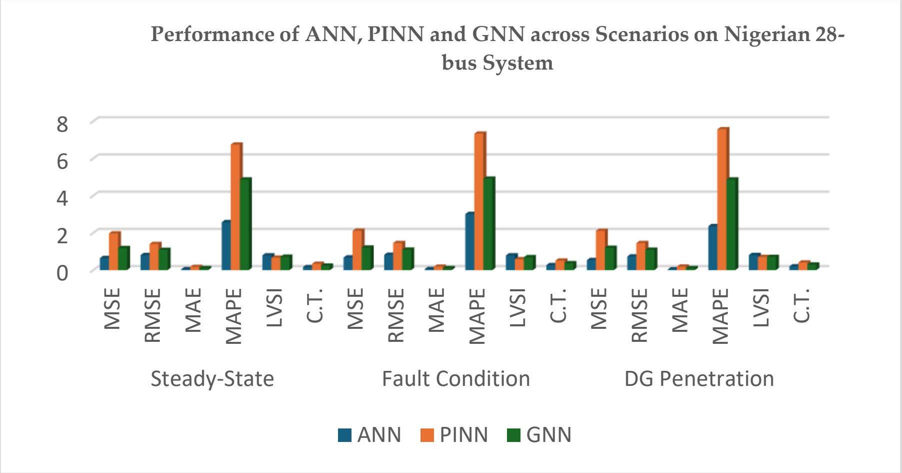

Figure 18.
Performance comparison of the neural network models on Nigerian 28-bus system.

The performance of the models across scenarios on the larger 52-bus Nigerian network is depicted in Figure 19. The diagram of Figure 19 showed that ANN continued to outperform in accuracy metrics. Similarly, LVSI analysis shows that ANN maintained its superiority in all scenarios, while PINN's LVSI exceeded GNN only in scenario 1, but fell below GNN in scenarios 2 and 3. GNN demonstrated better voltage stability performance under stressed conditions, suggesting resilience in larger networks. All models experienced slight performance degradation under fault conditions, and significant improvement in LVSI under DG penetration, highlighting the complexity of high-dimensional systems and benefits of integrating DG. ANN's combination of accuracy and computation efficiency remained superior even in large practical systems.

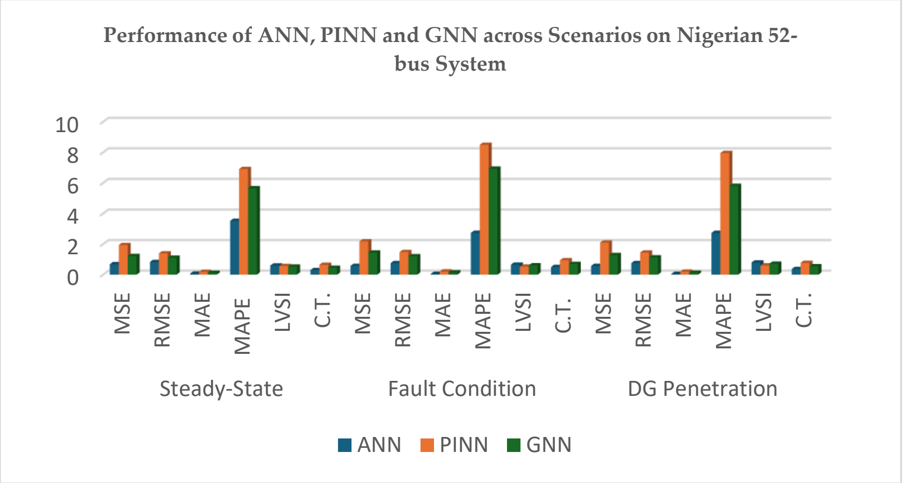

Figure 19.
Performance comparison of the neural network models on Nigerian 28-bus system.

DOI: 10.55214/2576-8484.v9i9.10161

6. Conclusion

The performance comparison of ANN, PINN and GNN models for load flow analysis of power systems under steady-state, fault, and DG penetration scenarios is presented in this study. NR load flow method was used to obtain the necessary dataset (voltage magnitudes and angles) required for training, validation and testing under varied load conditions in each scenario. The dataset was divided into 50:25:25 ratio for training, validation and testing of each model, respectively.

The results demonstrate that across all test systems, ANN consistently delivered the best balance of accuracy, LVSI performance, and computational efficiency, confirming its robustness for load flow analysis under varied operating conditions. GNN generally ranked second, offering a stability-oriented alternative, particularly in larger or stressed networks, while PINN exhibited scenario-dependent voltage stability strengths, performing well in small systems like 14-bus and under Steady-state in 52-bus networks but less so in larger stressed scenarios. The slight degradation of all models under fault conditions underscores the challenges introduced by network stresses. However, the improvement in LVSI across test systems under DG penetration validates the need for integration of DG resources for improved voltage stability. These trends indicate that ANN is best for fast, accurate predictions, GNN is advantageous for stability assessment in complex systems, and PINN can be useful when voltage stability focus outweighs precision.

Overall, this study contributes to the integration of ML into power system analysis by providing a comprehensive comparison of ANN, GNN, and PINN under realistic operational scenarios. The findings of this study offer practical guidance for model selection based on network size, operating condition, and performance priorities, highlighting the trade-offs between accuracy, voltage stability, and computational efficiency.

7. Future Studies

Future studies should consider extending these models to dynamic load flow, real-time operational environments, and hybrid architectures to further enhance grid resilience, smart grid decision-making, and integration of DG resources

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Acknowledgement:

The authors would like to acknowledge the support of Department of Electrical Engineering, Faculty of Engineering and the built environment for providing the necessary resources in carrying out the study. The authors would also love to appreciate the contribution of the research and innovation center of Tshwane University of Technology for their contributions towards the success of the study.

Copyright:

References

[1] X. P. Zhang, "Fundamentals of electric power systems," *Restructured Electric Power Systems: Analysis of Electricity Markets with Equilibrium Models*, pp. 1-52, 2010. https://doi.org/10.1002/9780470608555.ch1

[2] M. M. Rahman, S. H. Dadon, M. He, M. Giesselmann, and M. M. Hasan, "An overview of power system flexibility: High renewable energy penetration scenarios," *Energies*, vol. 17, no. 24, p. 6393, 2024. https://doi.org/10.3390/en17246393

[3] M. Fikri, B. Cheddadi, O. Sabri, T. Haidi, B. Abdelaziz, and M. Majdoub, "Power flow analysis by numerical techniques and artificial neural networks," presented at the 2018 Renewable Energies, Power Systems & Green Inclusive Economy (REPS-GIE), 2018.

[4] P. Bhowmik, S. Bose, D. Rajan, and S. Deb, "Power flow analysis of power system using power perturbation method," *2011 IEEE Power Engineering and Automation Conference*, vol. 3, pp. 380-384, 2011. https://doi.org/10.1109/PEAM.2011.6135117

[5] U. Buragohain and T. Boruah, "Fuzzy logic based load flow analysis," presented at the 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), 2017.

[6] B. Thashmitha and K. V. Madhukar, "Deep Learning based optimal power flow with renewable integration," presented at the 2022 IEEE International Power and Renewable Energy Conference (IPRECON), 2022.

[7] E. Ejuh Che, K. Roland Abeng, C. D. Iweh, G. J. Tsekouras, and A. Fopah-Lele, "The impact of integrating variable renewable energy sources into grid-connected power systems: challenges, mitigation strategies, and prospects," *Energies*, vol. 18, no. 3, p. 689, 2025. https://doi.org/10.3390/en18030689

[8] O. Gül, "Dynamic load flow in modern power systems: Renewables, crypto mining, and electric vehicles," *Sustainability*, vol. 17, no. 6, p. 2515, 2025. https://doi.org/10.3390/su17062515

[9] I. Adejumobi, G. Adepoju, and K. Hamzat, "Iterative techniques for load flow study: A comparative study for nigerian 330kv grid system as a case study," *International Journal of Engineering and Advanced Technology*, vol. 3, no. 1, pp. 4-11, 2013.

[10] I. Adebayo, I. Adejumobi, and G. Adepoju, "Application of load tap–changing transformer (LTCT) to the optimal economic dispatch of generation of the Nigerian 330kV grid system," *International Journal of Emerging Technologies in Sciences and Engineering*, vol. 5, no. 3, pp. 40-50, 2012.

[11] D. Mukherjee, *Application of a new swarm intelligent metaheuristic in solving load flow analysis problem*. Singapore: Springer, 2022. https://doi.org/10.1007/978-981-97-9916-9_7

[12] A. M. Nassef, M. A. Abdelkareem, H. M. Maghrabie, and A. Baroutaji, "Review of metaheuristic optimization algorithms for power systems problems," *Sustainability*, vol. 15, no. 12, p. 9434, 2023. https://doi.org/10.3390/su15129434

[13] A. M. Shazly, N. Anwer, and M. M. Mahmoud, "Solving optimal power flow problem for IEEE-30 bus system using a developed particle swarm optimization method: Towards fuel cost minimization," *International Journal of Modelling and Simulation*, vol. 45, no. 1, pp. 307-320, 2025. https://doi.org/10.1080/02286203.2023.2201043

[14] U. Himakar and M. D. Reddy, "Load flow analysis using real coded genetic algorithm," *International Journal of Engineering Research and Applications*, vol. 4, no. 2, pp. 522-527, 2014.

[15] S. Dixit, L. Srivastava, and G. Agnihotri, "Power flow analysis using fuzzy logic," in *IEEE Power India Conference*, 2006, doi: https://doi.org/10.1109/POWERI.2006.1632605.

[16] W. A. Alsulami and R. S. Kumar, "Artificial neural network based load flow solution of Saudi national grid," in *2017 Saudi Arabia Smart Grid (SASG)*, 2017, doi: https://doi.org/10.1109/SASG.2017.8356516.

[17] T. Xue, U. Karaagac, I. Kocar, M. B. Vavdareh, and M. Ghafouri, "Machine learning basics and potential applications in power systems," in *International Conference on Electrical, Communication and Computer Engineering (ICECCE)*, 2023, doi: https://doi.org/10.1109/ICECCE61019.2023.10441935.

[18] S. N. Okhuegbe, A. A. Ademola, and Y. Liu, "A machine learning initializer for Newton-Raphson AC power flow convergence," in *2024 IEEE Texas Power and Energy Conference (TPEC)*, 2024, doi: https://doi.org/10.1109/TPEC60005.2024.10472261.

[19] S. B. EFE, "Power flow analysis by artificial neural network," *International Journal of Energy and Power Engineering*, vol. 2, no. 6, p. 204, 2013. https://doi.org/10.11648/j.ijepe.20130206.11

[20] S. Tiwari, M. Ansari, K. Kumar, S. Chaturvedi, M. Singh, and S. Kumar, "Load flow analysis of IEEE 14 bus system using ANN technique," presented at the International Conference on Sustainable Energy, Electronics, and Computing Systems (SEEMS), 2018.

[21] E. R. B. Calma and M. C. Pacis, "Artificial neural network-based voltage stability analysis of power transmission networks with distributed generation using phasor measurement unit synthetic data," presented at the IEEE 12th Control and System Graduate Research Colloquium (ICSGRC), 2021.

[22] K. Gnanambal, N. Marimuthu, and C. Babulal, "Three-phase power flow analysis in sequence component frame using hybrid particle swarm optimization," *Applied Soft Computing*, vol. 11, no. 2, pp. 1727-1734, 2011. https://doi.org/10.1016/j.asoc.2010.05.015

[23] E. Davoodi, M. T. Hagh, and S. G. Zadeh, "A hybrid improved quantum-behaved particle swarm optimization–simplex method (IQPSOS) to solve power system load flow problems," *Applied Soft Computing*, vol. 21, pp. 171-179, 2014.

[24] A. Singh, S. Dixit, and L. Srivastava, "Particle swarm optimization-artificial neural network for power system load flow," *International Journal of Power System Operation & Energy Management*, vol. 1, no. 2, pp. 73-82, 2011.

[25] T. Ting, K. Wong, and C. Chung, "Hybrid constrained genetic algorithm/particle swarm optimisation load flow algorithm," *IET Generation, Transmission & Distribution*, vol. 2, no. 6, pp. 800-812, 2008. https://doi.org/10.1049/iet-gtd:20070224

[26] C. O. Ahiakwo, S. Orike, and O. E. Ojuka, "Application of neuro-swarm intelligence technique toload flow analysis," *American Journal of Engineering Research*, vol. 7, no. 8, pp. 94-103, 2018.

[27] Z. Kaseb, M. Möller, G. T. Balducci, P. Palensky, and P. P. Vergara, "Quantum neural networks for power flow analysis," *Electric Power Systems Research*, vol. 235, p. 110677, 2024.

[28] A. M. Saleh, V. István, M. A. Khan, M. Waseem, and A. N. A. Ahmed, "Power system stability in the Era of energy Transition: Importance, Opportunities, Challenges, and future directions," *Energy Conversion and Management: X*, p. 100820, 2024.

[29] S. S. Meel, B. S. Sharma, and M. S. Mohiuddin, "Load flow analysis on STATCOM incorporated interconnected power system networks using newton raphson method," *IOSR Journal of Electrical and Electronics Engineering* vol. 9, no. 4, pp. 61–68, 2014.

[30] A. F. Alnuaimi and T. H. Albaldawi, "An overview of machine learning classification techniques," *BIO Web of Conferences*, vol. 97, p. 00133, 2024.

[31] H. Dahrouj *et al.*, "An overview of machine learning-based techniques for solving optimization problems in communications and signal processing," *IEEE Access*, vol. 9, pp. 74908-74938, 2021. https://doi.org/10.1109/ACCESS.2021.3079639

[32] K. O. Alawode and M. O. Oyedeji, "Comparison of neural network models for load forecasting in nigeria power systems," *International Journal of Engineering Research & Technology*, vol. 2, no. 5, pp. 218–222, 2013.

[33] Z. Cömert and A. Kocamaz, "A study of artificial neural network training algorithms for classification of cardiotocography signals," *Bitlis Eren University Journal of Science and Technology*, vol. 7, no. 2, pp. 93-103, 2017.

[34] F. Aydin and B. Gümüş, "Study of different ANN algorithms for voltage stability analysis," presented at the 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), 2020.

[35] R. K. Tipu, R. Bhakhar, K. S. Pandya, and V. R. Panchal, "Physics-informed neural networks for predicting sediment transport in pressurized pipe flows," *Environmental Earth Sciences*, vol. 84, no. 11, p. 292, 2025. https://doi.org/10.1007/s12665-025-12295-0

[36] Z. K. Lawal, H. Yassin, D. T. C. Lai, and A. Che Idris, "Physics-informed neural network (PINN) evolution and beyond: A systematic literature review and bibliometric analysis," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 140, 2022. https://doi.org/10.3390/bdcc6040140

[37] M. Ghalambaz, M. A. Sheremet, M. A. Khan, Z. Raizah, and J. Shafi, "Physics-informed neural networks (P INNs): Application categories, trends and impact," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 34, no. 8, pp. 3131-3165, 2024. https://doi.org/10.1108/HFF-09-2023-0568

[38] A. G. Vrahatis, K. Lazaros, and S. Kotsiantis, "Graph attention networks: A comprehensive review of methods and applications," *Future Internet*, vol. 16, no. 9, p. 318, 2024. https://doi.org/10.3390/fi16090318

[39] B. Khemani, S. Patil, K. Kotecha, and S. Tanwar, "A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions," *Journal of Big Data*, vol. 11, no. 1, p. 18, 2024. https://doi.org/10.1186/s40537-023-00876-4

[40] Z. Jia *et al.*, "Recent research progress of graph neural networks in computer vision," *Electronics*, vol. 14, no. 9, p. 1742, 2025. https://doi.org/10.3390/electronics14091742

[41] W. Jiang *et al.*, "Graph neural networks for routing optimization: Challenges and opportunities," *Sustainability*, vol. 16, no. 21, p. 9239, 2024. https://doi.org/10.3390/su16219239

[42] B. Ismail *et al.*, "New line voltage stability index (BVSI) for voltage stability assessment in power system: The comparative studies," *IEEE Access*, vol. 10, pp. 103906-103931, 2022. https://doi.org/10.1109/ACCESS.2022.3204792

[43] A. Fayyazbakhsh, T. Kienberger, and J. Vopava-Wrienz, "Comparative analysis of load profile forecasting: LSTM, SVR, and ensemble approaches for singular and cumulative load categories," *Smart Cities*, vol. 8, no. 2, p. 65, 2025. https://doi.org/10.3390/smartcities8020065

[44] Z. Sun, Y. Zhou, K. Gao, Z. Wang, W. Ye, and J. Zeng, "A short-term electricity-load forecasting method integrating empirical mode decomposition and LSTM," *Frontiers in Energy Research*, 2024. https://doi.org/10.3389/fenrg.2024.1423692