

Article

Spatio-Temporal Deep Learning-Assisted Multi-Period AC Optimal Power Flow

Jihun Kim , Sojin Park , Dongwoo Kang *  and Hunyoung Shin * 

School of Electronic and Electrical Engineering, Hongik University, Seoul 04066, Republic of Korea; sebbie36@g.hongik.ac.kr (J.K.); so1220os24@gmail.com (S.P.)

* Correspondence: dkang@hongik.ac.kr (D.K.); hyshin@hongik.ac.kr (H.S.)

Abstract

The increasing penetration of renewable energy resources has amplified variability and uncertainty in power systems, reducing the effectiveness of conventional single-period Optimal Power Flow (OPF) strategies. Multi-period AC-OPF offers a more comprehensive framework by incorporating inter-temporal constraints and resource flexibility, but its high computational complexity and strong temporal coupling make large-scale applications challenging, often causing scalability issues and convergence difficulties in conventional solvers. We address these issues with a spatio-temporal deep learning model that combines a Graph Attention Network (GAT) for topology-aware feature learning with a Temporal Convolutional Network (TCN) for multi-period temporal modeling. The proposed model is trained on large-scale 500-bus and 1354-bus systems under both 8-period and 24-period settings, and it achieves robust scalability with consistently high prediction accuracy. Using the model's predictions, we construct an initial solution and provide it to a conventional OPF solver, which improves convergence performance and demonstrates the model's effectiveness as an auxiliary tool for complex MP-ACOPF problems.

Keywords: multi-period optimal power flow; AC optimal power flow; deep learning; graph neural networks

1. Introduction

The rapid integration of renewable energy resources has significantly increased the variability and uncertainty of net load in power systems. Conventional single-period operational strategies are often inadequate for coping with large and rapid net load changes, which can result in supply–demand imbalances, system security concerns, and increased economic dispatch costs. To address these challenges, multi-period operational strategies have emerged as a promising alternative, enabling operators to account for inter-temporal constraints and resource flexibility [1,2]. In practice, several independent system operators (ISOs), such as California Independent System Operator (CAISO) [3,4] and New York Independent System Operator (NYISO) [5], have already adopted multi-period optimal power flow (MP-OPF) in their real-time market operations.

Compared to its linearized version, the AC-OPF offers a more accurate representation of system physics by explicitly modeling voltage magnitudes, phase angles, reactive power flows, and network losses. However, its inherent non-convexity precludes guarantees of global optimality, and its high-order nonlinear equations impose substantial computational demands, especially in real-time environments. These challenges are further amplified in the multi-period AC optimal power flow (MP-ACOPF) setting, where extending the



Academic Editor: Dorin Petreus

Received: 30 December 2025

Revised: 28 January 2026

Accepted: 9 February 2026

Published: 11 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

optimization horizon introduces exponentially more inter-temporal constraints and state variables, thereby exacerbating computational complexity. Such burdens often hinder the practical deployment of MP-ACOPF in operational settings. This motivates the integration of data-driven machine learning techniques, which can generate near-optimal solutions with reduced computational effort and improve the convergence characteristics of optimization solvers.

To address the computational burden of large-scale OPF, decentralized optimization approaches decompose the overall problem into regional subproblems [6–9]. Early work utilized region-based decomposition, such as the auxiliary problem principle, to enhance scalability with limited coordination [6]. Subsequently, methods based on the alternating direction method of multipliers (ADMM) were applied to non-convex ACOPF, offering improved convergence properties while preserving data locality [7]. Further advancing this, fully decentralized consensus-based ADMM frameworks were explored for DCOPF with demand response to eliminate centralized coordination [8]. Most recently, stochastic recursive gradient methods have been proposed to significantly accelerate convergence in fully decentralized multi-area OPF [9].

While decentralized and distributed optimization efforts improve scalability through problem decomposition, rapid advances in data-driven machine learning have opened new possibilities for achieving scalable OPF solutions with significantly reduced computational effort and improved solver convergence. In recent years, deep neural networks (DNNs) have been actively explored as a means to reduce the computational complexity and enhance the scalability of OPF solvers [10–13]. In [10], a deep neural network was applied to the AC-OPF problem by predicting a subset of independent operating variables, while the remaining variables were computed using power flow equations to satisfy equality constraints. In [11], a DNN-based approach was proposed for solving the SC-DCOPF problem, where generator outputs are predicted from load inputs, and phase angles are reconstructed using power flow equations. In [12], a DNN-OPF framework was introduced to predict OPF solutions based on load demand and renewable energy source (RES) inputs. A post-processing step incorporating network topology was used to identify power congestion and RES curtailment. In [13], a voltage-constrained deep learning method was proposed to predict bus voltages from active and reactive power demands, reconstructing other OPF variables without solving nonlinear power flow equations.

In addition to DNN-based methods, graph-based approaches have also been explored to leverage topological information in OPF problems [14–18]. In [17], Graph Neural Networks (GNNs) were used to approximate AC-OPF solutions and provide warm-starts to traditional solvers, improving convergence reliability and achieving a 3.75× speedup without linear approximations. In [18], a GNN-based model combined with imitation learning was proposed, where the network was trained to approximate optimal solutions by leveraging the structural information of the power grid.

Recurrent neural networks (RNNs) have also been applied to OPF problems. In [19], a data-driven method was proposed to solve the OPF problem using an RNN model trained to map load inputs to optimal generation outputs. Evaluated on a 12-bus system, the approach showed high accuracy and improved computational efficiency compared to conventional solvers. In [20], the multi-period OPF problem was formulated as a sequence-to-sequence task and used a hybrid model combining Long Short-Term Memory (LSTM) and RNNs to predict generator and RES outputs from time-series load and renewable inputs. The model, validated on the IEEE 39-bus system, achieved high accuracy and computational efficiency with post-processing for constraint enforcement.

While recent studies have demonstrated the potential of deep learning in solving OPF problems, most of these approaches have been limited to single-period formulations. Due

to its temporal coupling and increased computational complexity, MP-ACOPF remains relatively underexplored. Furthermore, several works have only validated their models on small-scale systems, leaving the scalability and practicality of such approaches on large-scale power networks unproven. Another limitation is that many existing methods focus solely on either spatial modeling using GNNs or temporal modeling through sequence-based approaches, without integrating both modalities. Studies that jointly address spatio-temporal dependencies in OPF problems are still limited. In addition, most learning-based OPF approaches focus on approximating solutions, while paying little attention to resolving convergence failures commonly encountered in traditional OPF solvers. Thus, unified spatio-temporal learning for MP-ACOPF that integrates topology-aware spatial modeling with multi-period temporal modeling remains insufficient. Moreover, scalability validation on large-scale power networks is still scarce. Overall, deep learning-based approaches for solving multi-period OPF problems are still in their early stages, and there remains a need for more robust and scalable solutions.

To address these limitations, we design a framework that combines Graph Attention Networks (GAT) [21] with Temporal Convolutional Networks (TCN) [22] to jointly model spatial relationships among buses and temporal evolution of system states in multi-period AC-OPF. In particular, GAT learns topology-driven inter-bus interactions, while TCN captures temporal dependencies across periods, enabling joint spatio-temporal learning. The framework is tailored to handle the increased dimensionality and inter-temporal constraints of MP-ACOPF, while producing high-quality initializations that can assist conventional solvers in challenging cases. We evaluate the proposed approach on large-scale systems under 8-period and 24-period scenarios to demonstrate both accuracy and scalability. The main contributions of this work are summarized as follows:

- We propose an accurate, rapid, and scalable deep learning-based framework that combines Graph Attention Networks and Temporal Convolutional Networks to simultaneously learn spatial interactions and temporal demand variations in multi-period AC-OPF problems.
- We demonstrate the high prediction accuracy and scalability of the proposed model through experiments on 500-bus and 1354-bus systems under 8-h and 24-h multi-period scenarios.
- We show that using the predicted outputs of the deep learning model as an initial solution significantly improves the convergence of conventional OPF solvers, highlighting the practical applicability of deep learning as an auxiliary tool for solving multi-period OPF problems.

The remainder of this paper is organized as follows: Section 2 formulates the multi-period AC-OPF problem addressed in this study. Section 3 presents the deep learning model architecture, the performance evaluation strategy, and the dataset generation process. Section 4 describes the experiments and analyses conducted on multiple power systems and multi-period scenarios under various experimental settings. Finally, Section 5 concludes the paper.

2. Problem Formulation

In this section, we present the multi-period AC-OPF problem. The AC-OPF determines the optimal power reference points for generation resources with the objective of minimizing the total generation cost, while ensuring that all system variables—bus angle deviations, bus voltage magnitudes, line power flows, and active and reactive power injections—remain within their operational limits, governed by Ohm’s law and Kirchhoff’s circuit laws. Considering a $\mathcal{N}$ -bus power network, each bus can be a load bus $n \in \mathcal{N}_D$, a generation bus $n \in \mathcal{N}_G$, or both. Here, $\mathcal{N}_D$ and $\mathcal{N}_G$ denote the sets of nodes with load

and generation resources, respectively. The power generation at each node is denoted by $P_n^G + jQ_n^G$, while the power demand is represented by $P_n^D + jQ_n^D$.

The multi-period AC-OPF problem can be formulated as follows:

$$\text{minimize} \quad \sum_{t \in \mathcal{T}} \sum_{n \in \mathcal{N}_G} f_n(P_{n,t}^G) \quad (1)$$

subject to

$$P_{n,t}^G - P_{n,t}^D = \text{Re} \left\{ V_{n,t} \left(\sum_{m=1}^{\mathcal{N}} Y_{nm} V_{m,t} \right)^* \right\}, \quad n \in \mathcal{N}, \quad t \in \mathcal{T} \quad (2)$$

$$Q_{n,t}^G - Q_{n,t}^D = \text{Im} \left\{ V_{n,t} \left(\sum_{m=1}^{\mathcal{N}} Y_{nm} V_{m,t} \right)^* \right\}, \quad n \in \mathcal{N}, \quad t \in \mathcal{T} \quad (3)$$

$$|V_{n,t}((V_{n,t} - V_{m,t})y_{nm} + V_{n,t}y_{sh,n})^*| \leq \bar{S}_{nm}, \quad nm \in \mathcal{L}, \quad t \in \mathcal{T} \quad (4)$$

$$\underline{P}_n^G \leq P_{n,t}^G \leq \bar{P}_n^G, \quad n \in \mathcal{N}_G, \quad t \in \mathcal{T} \quad (5)$$

$$\underline{Q}_n^G \leq Q_{n,t}^G \leq \bar{Q}_n^G, \quad n \in \mathcal{N}_G, \quad t \in \mathcal{T} \quad (6)$$

$$P_{n,t+1}^G - P_{n,t}^G \leq RU_n, \quad n \in \mathcal{N}_G, \quad t \in \mathcal{T} \quad (7)$$

$$P_{n,t}^G - P_{n,t+1}^G \leq RD_n, \quad n \in \mathcal{N}_G, \quad t \in \mathcal{T} \quad (8)$$

$$|\underline{V}_n| \leq |V_{n,t}| \leq |\bar{V}_n|, \quad n \in \mathcal{N}, \quad t \in \mathcal{T} \quad (9)$$

$$\underline{\theta}_n \leq \theta_{n,t} \leq \bar{\theta}_n, \quad n \in \mathcal{N}, \quad t \in \mathcal{T} \quad (10)$$

where $\mathcal{T}$ represents the set of time steps in the multi-period AC-OPF problem, with each step indexed by t . The objective function (1) minimizes the total generation cost, where each cost function $f_n(\cdot)$ is typically modeled as a quadratic function of the generator's active power. Constraints (2) and (3) represent the net active and reactive power injections at each bus n and time t , respectively. In these expressions, $\text{Re}(z)$ and $\text{Im}(z)$ denote the real and imaginary parts of the net complex power injection z . The matrix Y is known as the admittance matrix, whose diagonal elements indicate the total admittances connected to each bus, while off-diagonal elements represent the negative value of line admittances between buses. Power flows in transmission lines must be within line capacity limits, and therefore, (4) limits the apparent power flow on each line $nm \in \mathcal{L}$ to the line capacity $\bar{S}_{nm}$. Here, $\mathcal{L}$ denotes the set of branches, y_{nm} is the line admittance of branch $nm \in \mathcal{L}$, and $y_{sh,n}$ is the shunt admittance of bus $n \in \mathcal{N}$. Constraints (5) and (6) enforce generator active and reactive power limits, respectively. Constraints (7) and (8) impose inter-temporal ramp-rate limits that couple consecutive periods. Finally, (9) and (10) bound the voltage magnitude and phase angle at each bus over all periods. Solving the MP-ACOPF in (1)–(10) repeatedly over many multi-period demand profiles provides paired input–output samples for learning. In this work, we use the multi-period active-load trajectories $P_{n,t}^D$ together with the fixed network topology as inputs, and treat the corresponding optimal voltage magnitude and angle trajectories $V_{n,t}, \theta_{n,t}$ as supervised targets. This casting connects the optimization formulation to a data-driven predictor that can generate high-quality initializations for conventional solvers.

3. Method

To address the multi-period AC-OPF problem described in the previous section, we propose a data-driven framework that uses graph-based deep learning techniques to approximate the OPF solutions. As a first step, grid state data is extracted from benchmark power systems. Using these data, we construct multi-period power demand profiles and solve the resulting problem using a multi-period AC-OPF solver. The obtained solutions are used to construct a graph-structured dataset. We train a GAT-TCN model to learn spatio-temporal patterns in multi-period demand profiles. The predictions are evaluated against the OPF-solver solutions as ground truth. The overall workflow of the proposed framework is illustrated in Figure 1.

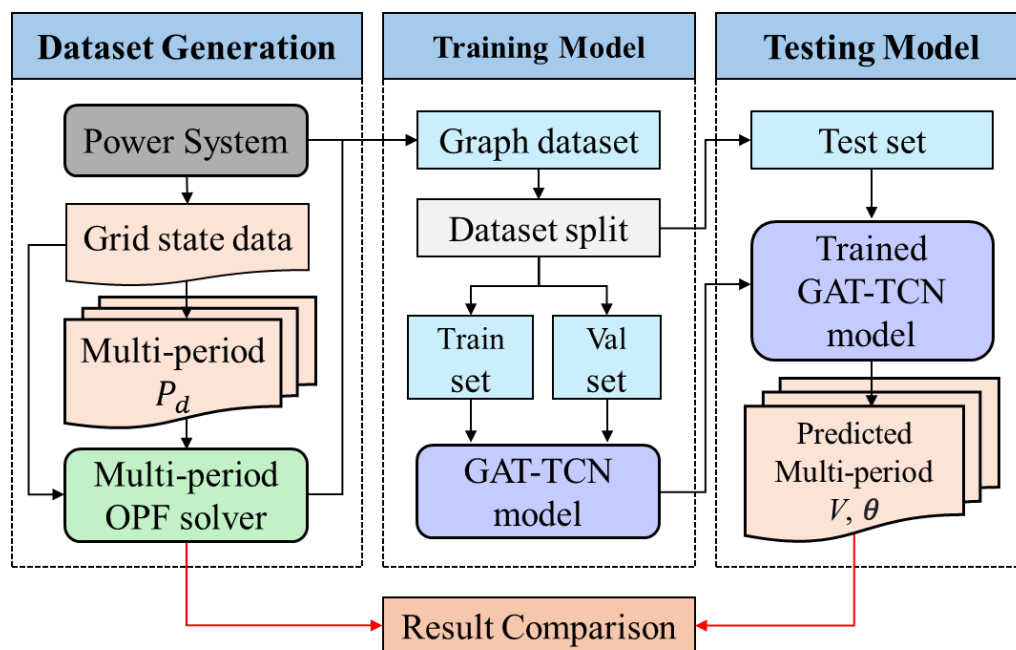


Figure 1. Overall framework of the proposed multi-period AC-OPF prediction approach.

3.1. Model Architecture

The proposed model consists of three main components: a GAT layer, a TCN layer, and a Dense output layer. The overall architecture is illustrated in Figure 2. The GAT layer captures the topology of the power network and learns spatial dependencies among buses. The TCN captures temporal patterns in power demand and is applied to both 8-period and 24-period step inputs. Finally, output values are generated through a Dense layer with a sigmoid activation function.

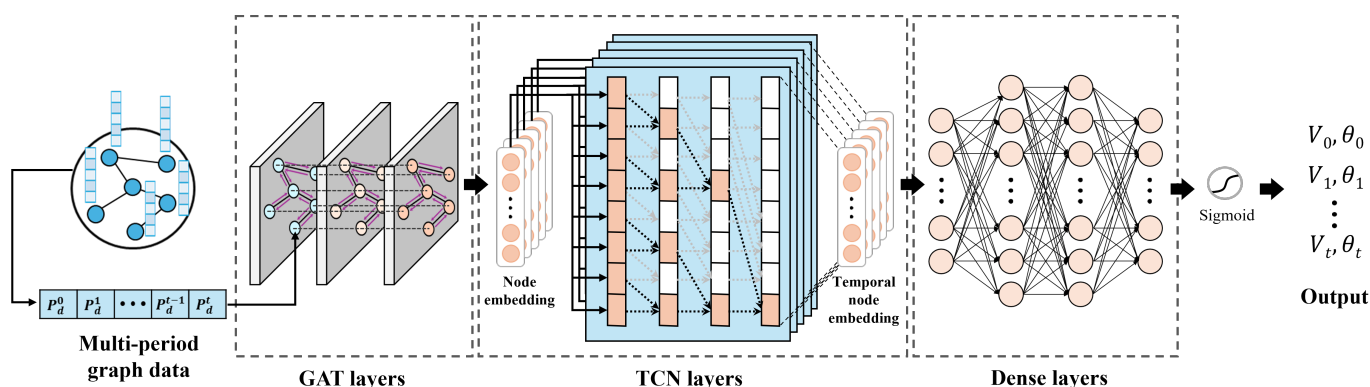


Figure 2. Overall architecture of the GAT-TCN model for multi-period OPF.

Unlike GNN-only or RNN-only models, our GAT-TCN architecture captures both grid topology and multi-period temporal dependencies within a single spatio-temporal model. Compared to an LSTM-RNN hybrid model, the proposed approach explicitly encodes topology-driven inter-bus interactions through GAT. For temporal modeling, TCN enables parallel computation and controls the receptive field through dilation, allowing efficient learning over longer temporal horizons. This design aligns well with the coupled spatio-temporal structure of MP-ACOPF.

3.1.1. Graph Attention Network

Unlike traditional deep learning models that operate on regular 2D data, GNNs utilize graph-structured data to effectively learn spatial and topological information in non-Euclidean domains. This allows buses and branches to be represented as nodes and edges in complex networks such as power systems, enabling the model to learn irregular connections and capture structural characteristics of the system. We adopt GAT as the GNN layer in our model. GAT enhances the basic GCN [23] by applying an attention mechanism, which allows for better capture of important information between nodes in the graph. This attention mechanism learns how much each node should consider the relationships with its neighboring nodes. The attention coefficients e_{ij} obtained from the attention mechanism a represent the importance of node j to node i , and are computed as:

$$e_{ij} = a(Wx_i, Wx_j)$$

where W is a shared weight matrix, and x_i, x_j are the feature vectors of nodes i and j , respectively. This model applies the attention mechanism to the graph structure using only the attention coefficients between nodes and their neighbors, without considering the structural information of the graph itself. To facilitate comparison of coefficients across nodes, the attention coefficients are normalized using the softmax function. As a result, the output features generated are given by:

$$x'_i = \sigma \left(\sum_{j \in N(i)} \alpha_{ij} Wx_j \right)$$

σ is the nonlinear activation function, α is the normalized attention coefficients, and $N(i)$ is the neighborhood of node i in the graph. Additionally, GAT introduces the multi-head attention mechanism, which applies multiple independent attention mechanisms in parallel. After applying K attention mechanisms, their results are concatenated to generate richer features. This approach improves the stability of the model and allows the input data to be processed from multiple perspectives, enabling the model to better understand complex graph data. Through this process, GAT effectively learns the neighboring nodes that have the most significant influence on a particular bus in the power system, capturing the key relationships and patterns within the system and improving prediction performance. The multi-head attention mechanism and the structure of GAT layers are shown in Figure 3. The GNN model is composed of three GAT layers, each utilizing three multi-head attention mechanisms.

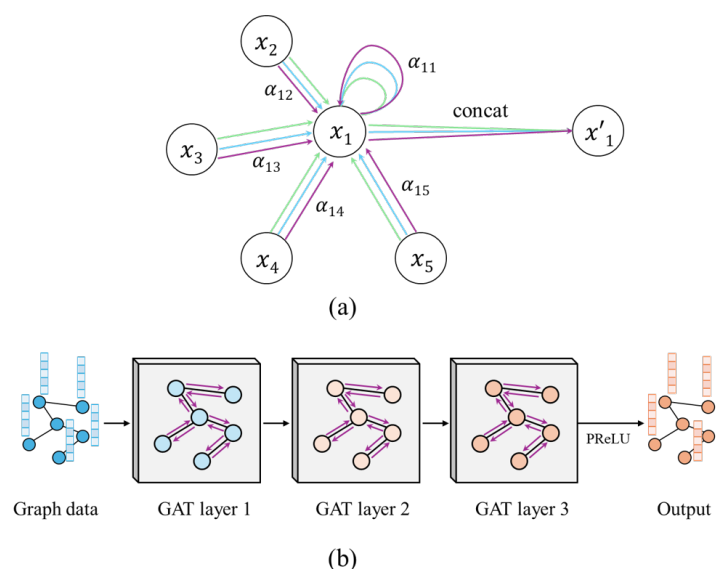


Figure 3. (a) Multi-head Attention mechanism with $K = 3$. Arrows in different colors represent different attentions. The aggregated features from each head are concatenated. (b) Structure of GAT layers.

3.1.2. Temporal Convolutional Network

Models based on recurrent neural networks, such as LSTM [24] and GRU [25], have been widely adopted for learning temporal patterns in spatio-temporal tasks. However, their sequential nature leads to high computational complexity and limits parallelization. In contrast, TCN uses a 1D CNN architecture that enables parallel processing, resulting in faster computation and higher training efficiency to address these issues. Due to its structural advantages, TCN effectively handles time series data and is well-suited for learning complex spatio-temporal patterns. Therefore, TCN was used to capture the temporal relationships of the features extracted by GAT. TCN utilizes causal convolution and dilated convolution to effectively capture temporal patterns. Causal convolution ensures that the output at each period depends only on the current and past inputs, effectively preventing the use of future information during training. To preserve the input sequence length, zero padding is applied within a fully convolutional architecture. This design allows the model to capture the temporal structure of the data while maintaining consistent output dimensions across layers. However, learning long sequences requires large filter sizes and this limitation is addressed by introducing dilated convolution. The dilated convolution operation F for element s is defined as follows:

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i}$$

x is the 1D sequence input ($x \in \mathbb{R}^n$), $*$ is the convolution operator, d is the dilation factor, f is the filter ($f: \{0, 1, \dots, k-1\}$), and k is the filter size. $s - d \cdot i$ represents the direction of the past. Increasing the dilation factor exponentially with network depth allows the receptive field to grow without a proportional increase in computational cost, enabling the model to capture long-range temporal dependencies efficiently. To ensure stable training and prevent the vanishing gradient problem, the TCN model utilizes residual blocks. Each residual block consists of dilated causal convolution, an activation function, and dropout applied twice, along with an additional 1×1 convolution to match the input and output widths. The dilation factor and filter size were set differently according to the dataset. For the 8-period data, the dilation factors were set to $d = 1, 2, 4$ and the filter size to $k = 2$. For

the 24-period data, the dilation factors were set to $d = 1, 3, 9$ and the filter size to $k = 3$. The dilated causal convolution and residual block for the input x and output y are shown in Figure 4.

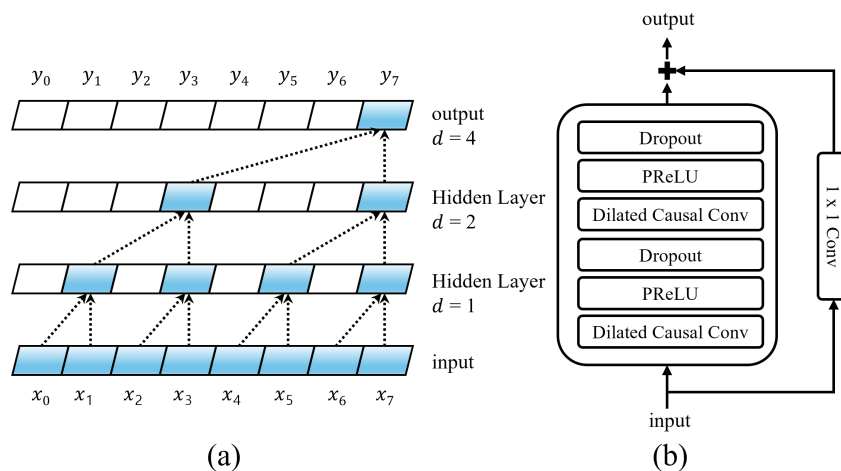


Figure 4. Components of the TCN architecture. (a) Dilated causal convolution for 8-period data with dilation factors $d = 1, 2, 4$ and filter size $k = 2$. (b) Structure of the residual block.

3.1.3. Dense Layer

In the final stage, a dense layer is used to produce the model's predictions. The spatio-temporal features extracted through GAT and TCN layers are passed through four fully connected layers. The first Dense layer has a number of nodes equal to the number of buses, while the second and third layers have twice that number. The final dense layer is set to twice the size of the predicted periods, designed to predict V and θ for each time step. Each layer applies the ReLU activation function, while the final layer uses the sigmoid function to constrain the outputs between 0 and 1, providing numerically stable predictions within a bounded range.

3.2. Dataset Generation

To construct the graph-structured dataset for training, we first generated multi-period power demand profiles and obtained their corresponding grid states by solving the multi-period AC-OPF problem. In this representation, buses were modeled as nodes and branches as edges, with the node features defined as the active power demand P_d over multiple periods. Following the method proposed in [26], parallel branches were combined into a single branch, and an adjacency matrix was constructed to represent the relationships between nodes. The ground-truth outputs of the graph were derived from the voltage magnitude V and angle θ values obtained from the solver for each period. These values were then normalized to the range $[0, 1]$ using min-max normalization. The overall process of generating the multi-period graph dataset is illustrated in Figure 5.

In this study, we conducted experiments using two benchmark power systems. The first is the Synthetic South Carolina 500 bus power system model [27], which consists of 500 buses, 90 generators, and 597 branches, with a total generation capacity of 12,189 MW. The second is the case1354pegase dataset provided by the Pan European Grid Advanced Simulation and State Estimation (PEGASE) project [28], consisting of 1354 buses, 260 generators, and 1991 branches, with a total generation capacity of 128,739 MW. To solve the multi-period OPF problem, multi-period P_d values were generated based on the original P_d values of the two power systems. The demand data from the California ISO website [29], as of 11 December 2024, was referenced to model the P_d values in a duck curve shape. We considered two time frames inspired by real-world market operations from ISOs

like NYISO and CAISO. These ISOs optimize real-time dispatch over 1–2 h (with 5-min intervals) and unit commitment over 75–150 min (with 15-min intervals). The first problem, an 8-period model with 15-min intervals, was specifically tested between 2 PM and 4 PM to capture a rapid increase in net load. The second model was a 24-period model that covered a full day, with data sampled at 1-h intervals. To introduce variation and simulate realistic demand fluctuations, independent uniform noise in the range of $\pm 2.5\%$ was added to the P_d values at each period. The P_d values were adjusted to align with the duck curve modeling, following the appropriate proportions for each power system. Through this process, 1500 multi-period P_d samples were generated for each system and period. To obtain the corresponding OPF solutions, an optimization solver, Knitro 14.2.0 [30], was used.

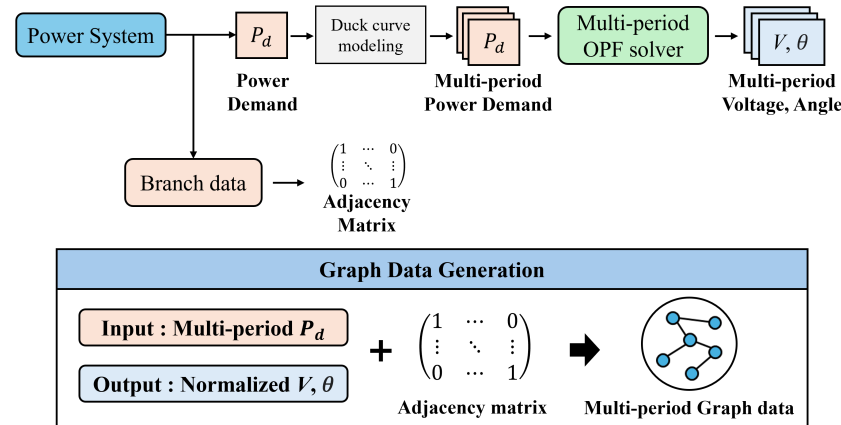


Figure 5. Overall process of generating multi-period graph dataset.

When solving the problems with the solver, the maximum computation time was limited to 300 s. The absolute feasibility tolerance was set to 10^{-6} , and the relative optimality tolerance to 10^{-3} . Among the solved cases, 1000 samples with successfully converged solutions were selected. For each sample, the voltage magnitude V and angle θ at all buses were extracted across all periods and normalized to the range $[0, 1]$. Combined with the corresponding multi-period P_d inputs, each instance was organized into a graph-structured dataset, which was then used to train the proposed deep learning model.

3.3. Training

To approximate the solutions of the multi-period AC-OPF problem, we trained the proposed deep learning model using the constructed dataset. The dataset consists of power demand profiles and their corresponding grid states obtained from an AC-OPF solver. To capture both spatial and temporal dependencies, the model was trained using a unified architecture composed of GAT and TCN. The generated graph dataset consists of 1000 samples per system and period, which is split into a training set (700 samples), a validation set (150 samples), and a test set (150 samples) as summarized in Table 1. For hyperparameter settings, the batch size was set to 32, and the Adam optimizer was used with an initial learning rate of 0.001. An exponential decay schedule was applied to the learning rate, with a decay rate of 0.95 and decay steps set to 5000. The model was trained for up to 500 epochs, with early stopping applied based on the validation loss. The patience was set to 50 epochs to prevent overfitting while allowing sufficient learning progress. To train the model, Mean Squared Error (MSE) was used as the loss function for stable and smooth convergence. MSE calculates the average of the squared differences between the predicted and actual values, allowing effective parameter updates through backpropagation.

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

where y_i and $\hat{y}_i$ represent the ground-truth and predicted values, respectively, and N is the total number of samples.

Table 1. Dataset division.

Dataset Split	Training	Validation	Test
Number of Samples	700	150	150

3.4. Performance Evaluation

For model evaluation, Symmetric Mean Absolute Percentage Error (SMAPE) was employed as a scale-independent metric to enable fair comparison across different systems and operating periods. Unlike MAPE, which can yield excessively large errors when the actual value approaches zero, SMAPE mitigates this issue by normalizing the absolute error with respect to the average of the predicted and actual values. This makes it particularly suitable for our task, where the target variables are normalized between 0 and 1. SMAPE is defined as:

$$\text{SMAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|)/2}.$$

SMAPE takes values between 0% and 200%, and smaller values correspond to better predictive accuracy. The final evaluation metric was computed by averaging the SMAPE of both voltage magnitude and angle.

3.5. Implementation Details

For dataset generation, AMPL was used in conjunction with the Knitro 14.2.0 solver [30]. The deep learning model was implemented in Python 3.8 using the TensorFlow, Keras, and Spektral [31] libraries. To train the network, the hyperparameters were tuned within the predefined search space shown in Table 2. In the GAT layers, a dropout rate of 0.5 was applied to prevent overfitting, and the parametric ReLU (PReLU) activation function [32] was used. The TCN layers were implemented with a dropout rate of 0.3 and also employed the PReLU activation function. All model weights were initialized using the Glorot uniform initializer. Experiments were conducted on a computer with an Intel(R) Core(TM) i9-14900K CPU, 128 GB of RAM, and an NVIDIA GeForce RTX 4090 GPU.

Table 2. Search space and selected hyperparameter.

Hyperparameter	Selected Value	Search Range
GAT layers	3	1, 2, 3, 4, 5
GAT attention heads	3	1, 2, 3, 4, 5
GAT dropout rate	0.5	0, 0.1, 0.3, 0.5, 0.7
TCN dropout rate	0.3	0, 0.1, 0.3, 0.5, 0.7
Dense layers	4	1, 2, 3, 4, 5
Learning rate	0.001	0.0005, 0.001, 0.0015
Loss function	MSE	MSE, MAE
Epochs	500	300, 500, 1000

4. Experiments and Analysis

This section analyzes and discusses the performance of the proposed model through experiments. The experiments use the 8-period and 24-period datasets of the 500-bus and

1354-bus systems described in the Dataset Generation section. We first verify the training stability and generalization of the proposed model through training analysis. We then present an ablation study on the 1354-bus system to evaluate the contribution of the GAT and TCN layers. In addition, we compare the proposed GAT and TCN layers with other representative layers to assess their effectiveness. We further examine whether the solver convergence can be improved by using the trained model's outputs as an AI-assisted initial solution for the failed cases. Finally, we analyze the impact of constraining the output range by comparing models with and without a sigmoid function in the output layer.

4.1. Training Performance and Generalization

To evaluate the training stability and generalization capability of the proposed model, its performance on the training, validation, and test datasets was analyzed using MSE and SMAPE. As summarized in Table 3, both metrics remain highly consistent across datasets, indicating that the model was trained stably without overfitting or underfitting. The absolute difference in SMAPE among the datasets was less than 0.03% in all cases, and the relative deviation remained within 3.5%, demonstrating strong generalization capability. This consistency confirms that the proposed GAT-TCN model achieved stable convergence and maintained reliable predictive accuracy across different system sizes and time horizons.

Table 3. Quantitative performance across training, validation, and test datasets for all configurations.

Case	Metric	Training	Validation	Test
500-bus (8 periods)	MSE ($\times 10^{-4}$)	0.2	0.2	0.1
	SMAPE (%)	0.486	0.489	0.503
500-bus (24 periods)	MSE ($\times 10^{-4}$)	7.3	6.8	7.0
	SMAPE (%)	1.025	0.997	1.017
1354-bus (8 periods)	MSE ($\times 10^{-4}$)	3.8	3.8	3.8
	SMAPE (%)	0.112	0.110	0.113
1354-bus (24 periods)	MSE ($\times 10^{-4}$)	0.4	0.4	0.4
	SMAPE (%)	0.144	0.145	0.143

4.2. Ablation Study

To evaluate the effectiveness of spatio-temporal modeling for MP-ACOPF prediction, we conducted an ablation study on the 1354-bus system under both the 8-period and 24-period settings. In this study, we replaced the GAT and TCN layers with DNN layers to assess how each component contributes to the overall predictive performance. For a fair comparison, when a layer was replaced, we matched the number of layers, hidden dimensionality, and activation functions to those of the original layers. As shown in Table 4, the spatio-temporal model using both GAT and TCN achieves the best accuracy in both the 8-period and 24-period settings. Using only one component (GAT-only or TCN-only) yields inferior performance, and the DNN-only model (without both GAT and TCN) performs better than these single-component models. A GAT-only model focuses on topology-driven interactions across buses, but it does not explicitly model temporal dependencies across periods. Conversely, a TCN-only model captures temporal patterns, but it does not explicitly incorporate the network topology or inter-bus interactions. When only one dependency is structurally emphasized, the model may learn representations that are incomplete for MP-ACOPF and thus generalize poorly. Although the DNN-only model is less structured, it does not focus on only one type of dependency, which may yield more balanced predictions than the single-component models. By contrast, the proposed

GAT–TCN model is designed to separate the roles of spatial and temporal learning. The GAT layer focuses on topology-driven inter-bus interactions, while the TCN layer captures temporal dependencies across multiple periods. This separation allows the model to integrate spatial and temporal information simultaneously. As a result, the proposed architecture is well suited for MP-ACOPF prediction.

Table 4. Ablation results on the 1354 bus system.

1354 Bus: 8 Period			1354 Bus: 24 Period		
GAT	TCN	SMAPE	GAT	TCN	SMAPE
✓	✓	0.14%	✓	✓	0.18%
		0.19%			0.38%
		0.15%			0.24%
✓	✓	0.11%	✓	✓	0.14%

4.3. Comparison of GNN Layers

In this experiment, the performance of the model was compared by changing the GNN layers. The GNN layers compared were APPNP [33], GCN, DCRNN [34], GIN [35], and ARMA [36], and the experiments were conducted under the same conditions as the GAT. The GNN layers were combined with the TCN and the SMAPE values were measured for each dataset for comparisons. The experimental results, as shown in Table 5, show that the GAT layer outperformed the other GNN layers, achieving 0.50% for 8-period setting and 1.02% for 24-period setting on the 500-bus system, and 0.11% for 8-period setting and 0.14% for 24-period setting on the 1354-bus system. This superior performance can be attributed to the GAT layer’s ability to learn the importance of each node through the attention mechanism, allowing it to better capture key relationships within the power system. Additionally, by utilizing multi-head attention, the model could evaluate importance from various perspectives, leading to a better understanding of the complex patterns in the power system and thus achieving excellent performance.

Table 5. Comparison of SMAPE values for different GNN layers.

GNN Layer	500 Bus		1354 Bus	
	8 Period	24 Period	8 Period	24 Period
APPNP	0.51%	1.17%	0.13%	0.17%
GCN	0.51%	1.22%	0.15%	0.15%
DCRNN	0.60%	1.31%	0.17%	0.21%
GIN	0.63%	1.20%	0.28%	0.16%
ARMA	0.51%	1.14%	0.20%	0.18%
GAT	0.50%	1.02%	0.11%	0.14%

4.4. Comparison of TCN and RNN

Following the comparison of GNN layers, we analyzed the time-series modeling performance by comparing TCN with RNN-based models, such as LSTM, GRU, Bi-LSTM, and Bi-GRU. LSTM and GRU are basic forward models, whereas Bi-LSTM and Bi-GRU are bidirectional variants that use both forward and backward information to better capture time-series patterns. The models used in the experiments were combined with the GAT layer under the same conditions as TCN, and the SMAPE values were measured and compared for each dataset. Table 6 shows that TCN outperforms all RNN-based models, achieving lower SMAPE across all datasets. This is because TCN models temporal dependencies through causal and dilated convolutions, which enable parallel computation

across periods and provide a controllable receptive field. Such a design is well matched to MP-ACOPF, where system states across periods are coupled and predictive performance can benefit from capturing dependencies over the multi-period horizon. In contrast, RNN-based models process sequences sequentially, which limits parallelization and can make learning long-range temporal dependencies less stable. Moreover, Bi-LSTM and Bi-GRU do not necessarily yield better performance in our setting, suggesting that increasing recurrent model complexity alone does not guarantee improved generalization. Overall, these results indicate that TCN offers a more reliable and efficient temporal modeling choice for MP-ACOPF prediction.

Table 6. Comparison of SMAPE values for different time-series models.

Sequential Layer	500 Bus		1354 Bus	
	8 Period	24 Period	8 Period	24 Period
LSTM	0.53%	1.15%	0.16%	0.15%
GRU	0.53%	1.21%	0.13%	0.15%
Bi-LSTM	0.63%	1.25%	0.16%	0.16%
Bi-GRU	0.58%	1.10%	0.13%	0.17%
TCN	0.50%	1.02%	0.11%	0.14%

4.5. AI-Assisted Initial Solution for Solver Convergence

In this experiment, we investigated whether using an AI-assisted initialization could improve the convergence rate of multi-period OPF solvers in cases where the solver failed to converge. For each failed case, the trained model predicted the voltage magnitude V and angle θ across all periods. The predicted outputs were de-normalized via inverse min–max normalization and then used as the solver’s initial solution. Out of 1500 total runs, the number of failed cases was as follows: for the 500-bus system, 1 case in the 8-period setting and 2 cases in the 24-period setting; for the 1354-bus system, 43 cases in the 8-period setting and 34 cases in the 24-period setting. This difference mainly reflects the larger problem size of the 1354-bus system, which increases numerical difficulty and makes it more likely for the solver to reach the time limit under the same settings. With the AI-assisted initialization, all previously failed cases in the 500-bus system successfully converged. For the 1354-bus system, the convergence rate improved from 97.13% to 99.67% in the 8-period setting, and from 97.73% to 99.73% in the 24-period setting, as summarized in Table 7.

To assess the runtime impact of the proposed AI-assisted initialization on the 1354-bus system, we compared solver runtimes for the failed cases. Without AI-assisted initialization, the solver spent 300 s for these cases in both the 8-period and 24-period settings. With AI-assisted initialization, the mean solve time decreased to 17.54 s in the 8-period setting and 204.02 s in the 24-period setting. This corresponds to a 94.15% reduction for the 8-period setting and a 31.99% reduction for the 24-period setting.

Table 7. Comparison of solver convergence with and without AI assistance.

Method	500 Bus		1354 Bus	
	8 Period	24 Period	8 Period	24 Period
Solver only	99.93%	99.87%	97.13%	97.73%
Solver + AI assist	100%	100%	99.67%	99.73%

Figure 6 presents a representative case of the optimality gap evolution when the solver fails to converge. As shown by the red curve for without AI-assisted initialization, the optimality gap ceases to decrease and continues to oscillate as the iterations proceed.

In contrast, with the proposed AI-assisted initialization, the optimality gap decreases rapidly after approximately 60 iterations, ultimately enabling the algorithm to converge. This suggests that the AI-assisted initialization can significantly improve the algorithmic convergence for large-scale multi-period AC-OPF problems by providing a high-quality initial point.

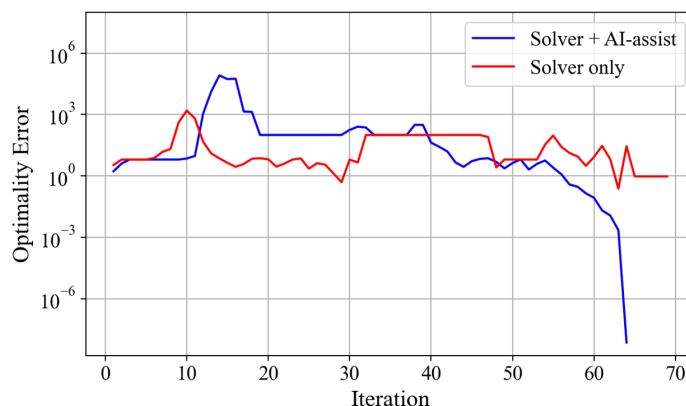


Figure 6. Evolution of the optimality gap over iterations with (blue) and without (red) AI assistance.

4.6. Effect of Output Range Constraints on Performance

In this experiment, we analyzed the impact of output range limitation on performance by comparing the convergence rates when applying the initial solution using SMAPE values and failed convergence cases in the 1354-bus system. The original model applied a sigmoid function to the output layer, limiting the output range and guiding the model to satisfy the constraints. In contrast, the comparison model did not use a sigmoid function in the output layer, allowing the output values to have no such limitation. The results, as shown in Table 8, showed that the model with the sigmoid function had lower SMAPE values and higher convergence rates for failed convergence cases. Specifically, SMAPE decreased from 0.19% to 0.11% for 8-period setting and from 0.54% to 0.14% for 24-period setting. Additionally, the convergence rate increased from 99.20% to 99.67% for 8-period setting and from 99.20% to 99.73% for 24-period setting. This confirms that using a sigmoid function to limit the output range improves the learning performance of the AI model. It also provides a better initial point when used as the initial solution, positively influencing the solver's stable convergence.

Table 8. Comparison of SMAPE and solver convergence rates with and without output range constraints.

Model	SMAPE		Convergence Rate	
	8 Period	24 Period	8 Period	24 Period
Unconstrained	0.19%	0.54%	99.20%	99.20%
Constrained	0.11%	0.14%	99.67%	99.73%

5. Conclusions

In this study, we proposed a spatio-temporal deep learning framework that combines Graph Attention Networks (GAT) and Temporal Convolutional Networks (TCN) for MP-ACOPF. Experiments on 500-bus and 1354-bus systems under 8-period and 24-period horizons demonstrated consistently high predictive accuracy and robust scalability across system sizes and time resolutions. A key advantage of the proposed architecture is the role separation between spatial and temporal learning, which aligns well with the coupled

spatio-temporal structure of MP-ACOPF. This design enables stable learning and consistent generalization across different system sizes and time horizons. Beyond prediction accuracy, the proposed method provides practical value as an auxiliary tool for conventional solvers. By using the predicted voltage magnitude and angle trajectories as AI-assisted initializations, we significantly improved solver convergence rates in failed cases, demonstrating enhanced reliability for large-scale MP-ACOPF. In addition, constraining the output range with a sigmoid activation improved numerical stability and solution quality. Future work will investigate constraint-aware learning to further promote feasibility by incorporating AC-OPF constraints into the learning process, and extend the framework to explicitly handle uncertainty from renewable generation and other stochastic operating conditions.

Author Contributions: Conceptualization, J.K., S.P., H.S. and D.K.; Methodology, J.K., S.P., H.S. and D.K.; Software, J.K., S.P. and H.S.; Validation, J.K. and S.P.; Formal analysis, J.K., S.P. and H.S.; Investigation, J.K. and S.P.; Resources, D.K. and H.S.; Data curation, J.K. and S.P.; Writing—original draft preparation, J.K., S.P. and H.S.; Writing—review and editing, J.K., S.P., H.S. and D.K.; Visualization, J.K.; Supervision, H.S. and D.K.; Project administration, J.K., H.S. and D.K.; Funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Korea Electric Power Corporation (Grant No. R23XO05-09) and also by 2025 Hongik University Research Fund.

Data Availability Statement: The data are derived from publicly available sources cited in this article, and all procedures for dataset construction are described in the manuscript. Further information is available from the corresponding authors upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Harrison, J.; Harvard Electricity Policy Group. Price Formation and Multi-Period Dispatch. 2020. Available online: <https://hepg.hks.harvard.edu/publications/price-formation-and-multi-period-dispatch> (accessed on 8 February 2026).
2. Zhao, J.; Zheng, T.; Litvinov, E. A Multi-Period Market Design for Markets with Intertemporal Constraints. *IEEE Trans. Power Syst.* **2019**, *35*, 3015–3025. [\[CrossRef\]](#)
3. CAISO. *Business Practice Manual for the Real-Time Market, Section 34*; CAISO: Folsom, CA, USA, 2024.
4. CAISO. *2024 Special Report on Battery Storage*; Technical Report; CAISO: Folsom, CA, USA, 2025.
5. NYISO. *Transmission and Dispatch Operations Manual*; NYISO: Rensselaer, NY, USA, 2024.
6. Baldick, R.; Kim, B.H.; Chase, C.; Luo, Y. A Fast Distributed Implementation of Optimal Power Flow. *IEEE Trans. Power Syst.* **1999**, *14*, 858–864. [\[CrossRef\]](#)
7. Erseghe, T. Distributed Optimal Power Flow Using ADMM. *IEEE Trans. Power Syst.* **2014**, *29*, 2370–2380. [\[CrossRef\]](#)
8. Wang, Y.; Wu, L.; Wang, S. A Fully-Decentralized Consensus-Based ADMM Approach for DC-OPF with Demand Response. *IEEE Trans. Smart Grid* **2017**, *8*, 2637–2647. [\[CrossRef\]](#)
9. Hussan, U.; Wang, H.; Ayub, M.A.; Rasheed, H.; Majeed, M.A.; Peng, J.; Jiang, H. Decentralized Stochastic Recursive Gradient Method for Fully Decentralized OPF in Multi-Area Power Systems. *Mathematics* **2024**, *12*, 3064. [\[CrossRef\]](#)
10. Pan, X.; Chen, M.; Zhao, T.; Low, S.H. DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems. *IEEE Syst. J.* **2022**, *17*, 673–683. [\[CrossRef\]](#)
11. Pan, X.; Zhao, T.; Chen, M.; Zhang, S. DeepOPF: A deep neural network approach for security-constrained DC optimal power flow. *IEEE Trans. Power Syst.* **2020**, *36*, 1725–1735. [\[CrossRef\]](#)
12. Zafar, R.; Vu, B.H.; Chung, I.Y. A deep neural network-based optimal power flow approach for identifying network congestion and renewable energy generation curtailment. *IEEE Access* **2022**, *10*, 95647–95657. [\[CrossRef\]](#)
13. Huang, W.; Pan, X.; Chen, M.; Low, S.H. DeepOPF-V: Solving ac-opf problems efficiently. *IEEE Trans. Power Syst.* **2021**, *37*, 800–803. [\[CrossRef\]](#)
14. Liu, S.; Wu, C.; Zhu, H. Topology-aware graph neural networks for learning feasible and adaptive AC-OPF solutions. *IEEE Trans. Power Syst.* **2022**, *38*, 5660–5670. [\[CrossRef\]](#)
15. Gao, M.; Yu, J.; Yang, Z.; Zhao, J. A physics-guided graph convolution neural network for optimal power flow. *IEEE Trans. Power Syst.* **2023**, *39*, 380–390. [\[CrossRef\]](#)

16. Yang, M.; Liu, J.; Qiu, G.; Liu, T.; Tang, Z. AC optimal power flows: Combining lagrangian dual and physics-guided graph neural networks. *TechRxiv* **2023**. [CrossRef]
17. Diehl, F. Warm-starting AC optimal power flow with graph neural networks. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2019; pp. 1–6.
18. Owerko, D.; Gama, F.; Ribeiro, A. Optimal power flow using graph neural networks. In *Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: New York, NY, USA, 2020; pp. 5930–5934.
19. Zhang, Z.; Yang, S.; Jia, Y.; Xiao, J.; Bai, X. Data-driven rnn method for distribution network optimal power flow problems. In *Proceedings of the 2023 3rd Power System and Green Energy Conference (PSGEC)*; IEEE: New York, NY, USA, 2023; pp. 382–387.
20. Zafar, R.; Chung, I.Y. Data-driven multiperiod optimal power flow for power system scheduling considering renewable energy integration. *IEEE Access* **2024**, *12*, 95278–95290. [CrossRef]
21. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, BC, Canada, 30 April–3 May 2018.
22. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271. [CrossRef]
23. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, Toulon, France, 24–26 April 2017.
24. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
25. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555. [CrossRef]
26. Ramesh, A.V.; Li, X. Spatio-temporal deep learning-assisted reduced security-constrained unit commitment. *IEEE Trans. Power Syst.* **2023**, *39*, 4735–4746. [CrossRef]
27. Birchfield, A.B.; Xu, T.; Gegner, K.M.; Shetye, K.S.; Overbye, T.J. Grid structural characteristics as validation criteria for synthetic networks. *IEEE Trans. Power Syst.* **2016**, *32*, 3258–3265. [CrossRef]
28. Jozs, C.; Fliscounakis, S.; Maeght, J.; Panciatici, P. AC power flow data in MATPOWER and QCQP format: iTesla, RTE snapshots, and PEGASE. *arXiv* **2016**, arXiv:1603.01533. [CrossRef]
29. California Independent System Operator. California ISO—Today’s Outlook. 2024. Available online: <https://www.caiso.com/todays-outlook> (accessed on 11 December 2024).
30. Byrd, R.H.; Nocedal, J.; Waltz, R.A. Knitro: An integrated package for nonlinear optimization. In *Large-Scale Nonlinear Optimization*; Springer: Boston, MA, USA, 2006; pp. 35–59.
31. Grattarola, D.; Alippi, C. Graph neural networks in tensorflow and keras with spektral [application notes]. *IEEE Comput. Intell. Mag.* **2021**, *16*, 99–106. [CrossRef]
32. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
33. Gasteiger, J.; Bojchevski, A.; Günnemann, S. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, 6–9 May 2019.
34. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR ’18)*, Vancouver, BC, Canada, 30 April–3 May 2018.
35. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? In *Proceedings of the International Conference on Learning Representations*, New Orleans, LA, USA, 6–9 May 2019.
36. Bianchi, F.M.; Grattarola, D.; Livi, L.; Alippi, C. Graph neural networks with convolutional arma filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3496–3507. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.