TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Master's Thesis in Informatics

Optimal Power Flow Using Graph Neural Networks

Can Bayraktaroglu

TECHNICAL UNIVERSITY OF MUNICH

DEPARTMENT OF INFORMATICS

Master's Thesis in Informatics

Optimal Power Flow Using Graph Neural Networks

Optimaler Leistungsfluss mittels Graph-Neuronaler-Netzwerke

Author:	Can Bayraktaroglu
Supervisor:	Prof. Dr.-Ing. Matthias Althoff
Advisor:	Michael Eichelbeck, M.Sc.
Submission Date:	20.11.2023

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Ich versichere, dass ich diese Master's Thesis selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

Munich, 20.11.2023 Can Bayraktaroglu

Acknowledgments

I extend my sincere gratitude to my advisor, Mr. Michael Eichelbeck, for his invaluable guidance and unwavering support throughout my research. His expertise and encouragement have been instrumental in shaping this research.

Abstract

This thesis investigates the Optimal Power Flow (OPF) problem in electrical networks, specifically focusing on the Alternating Current Optimal Power Flow (ACOPF) scenario. It explores the application of both Homogeneous and Heterogeneous Graph Neural Networks (GNNs) to optimize power generation and distribution, thereby enhancing network efficiency with a focus on network bus features. Motivated by the imperative of efficient energy management, the performance of these models is evaluated and compared against traditional numerical solvers. The research demonstrates the efficacy of GNNs in solving real-world power system challenges, offering valuable insights into their application in the energy optimization landscape. This work significantly contributes to the discourse on energy optimization techniques, emphasizing the feasibility and efficiency of GNNs in addressing intricate power system problems.

Contents

List of Figures

List of Tables

1 Introduction

In the realm of modern power distribution networks, optimizing power flow remains a pivotal task. This task is not only technical in nature, involving the effective management of electrical grids, but also has far-reaching implications for energy efficiency and sustainability. The underlying motivation for optimizing power flow in power distribution networks is multifaceted, centering on the need to enhance the reliability, efficiency, and adaptability of these networks.

The importance of this task is underscored by the evolving landscape of power systems, characterized by increasing integration of renewable energy sources, fluctuating demand patterns, and the complexity of maintaining stable operations across diverse grid topologies. In such a dynamic environment, the ability to optimize power flow can lead to more effective utilization of resources, reduced operational costs, and a more resilient power infrastructure. Additionally, in the context of global sustainability goals and the transition towards cleaner energy sources, optimizing power flow in power distribution networks becomes crucial in managing the variability and intermittency associated with renewable energy sources.

The pursuit of solutions for this task aligns with broader objectives in the field of energy management, where the focus extends beyond mere technical efficiency to encompass aspects like environmental impact, economic viability, and social implications. The integration of advanced technological solutions in power systems, therefore, holds the promise of not only improving operational efficiency but also contributing to a more sustainable and reliable energy future. This motivation is rooted in the ever-growing need for innovative approaches to manage and balance the complex interplay of generation, distribution, and consumption within power grids, especially in the face of escalating global energy demands and the shifting paradigms of energy production and consumption. Building upon the foundational understanding of the task's importance, it becomes clear that more efficient solutions, particularly those enabled by the latest machine learning approaches, are instrumental in propelling our infrastructures and society towards overarching goals of sustainability, economic viability, and socio-economic progress.

Machine learning, with its ability to analyze large datasets, recognize complex patterns, and make predictive insights, offers a transformative potential for power distribution networks. In terms of sustainability, machine learning algorithms can optimize the integration and utilization of renewable energy sources, thereby reducing reliance on fossil fuels and minimizing environmental impact. This optimization is crucial in managing the intermittent nature of renewable energy, ensuring that supply consistently meets demand while minimizing waste and inefficiency.

From an economic standpoint, machine learning-driven solutions in power systems can significantly reduce operational and maintenance costs. By accurately predicting demand patterns and potential system failures, these technologies enable proactive management of the grid, thus preventing costly downtimes and extending the lifespan of infrastructure components. Such cost efficiencies not only benefit utility providers, but can also translate into lower energy costs for consumers, contributing to broader economic stability and growth.

On a socio-economic level, the application of advanced machine learning in power distribution can lead to more equitable energy access. By improving grid efficiency and reliability, especially in underserved or remote areas, machine learning can play a key role in bridging the energy divide. Furthermore, as these technologies foster the adoption of renewable energy sources, they contribute to creating a more sustainable energy landscape, which is vital for long-term socio-economic development.

In summary, the application of machine learning in power flow optimization is not just a technical enhancement but a catalyst for sustainable, economically viable, and socially beneficial advancements in energy infrastructure. These technologies hold the key to navigating the complexities of modern power systems while aligning with the global imperative of creating a more sustainable and equitable energy future.

1.1 Recent Works and Research Scope

GNNs have emerged as a powerful tool for power flow analysis in energy grids [1][2][3]. They provide a localized, scalable parametrization of network data, making them well-suited for large and complex power system networks [4][5].

In terms of homogeneous approaches, GNNs have been used to approximate OPF solutions. Tuo et. al [4] trained a homogeneous GNN model using historical power system data to predict power flow outcomes. This approach enables rapid estimation of line flows, providing more accurate solutions with high efficiency compared to traditional models. Owerko et. al [6] proposed a GNN for power flow analysis using unsupervised learning approaches.

As for heterogeneous approaches, research is still in its early stages. However, the potential of Typed Graph Neural Networks (TGN) and other advanced GNN architectures for power flow analysis is being explored. Böttcher et. al [1] proposed a GNN architecture to solve the Alternating Current (AC) power flow problem under realistic constraints. The authors demonstrated the development of a framework that uses GNNs to learn the physical constraints of the power flow. They performed unsupervised training to learn a general solution of the AC power flow formulation independent of the specific topologies and supply tasks used for training. Their approach showed a high increase in computation time and good accuracy compared to state-of-the-art solvers. Garcia et. al [7] provided heterogeneous TGNs, which effectively handle different node types in power grids. This approach distinguished itself by accurately representing the unique characteristics of various node types like generators, loads,

and slack nodes. A specialized framework, incorporating a unique message-passing mechanism, facilitates effective interaction among these heterogeneous nodes, enhancing the model's accuracy and practicality. The research demonstrated the TGNs' adaptability to different grid sizes and configurations, showing potential for real-world application.

Despite their promising results, GNNs also have some shortcomings. They can be computationally expensive in large graphs and may struggle with unseen nodes. There is also no guarantee that the predicted optimal point is a feasible point, and violation of important constraints could lead to severe security issues for the grid [4][8]. Furthermore, in some cases, the graph structure used by GNNs may not be informative for the predictive task.

In conclusion, while GNNs offer a promising approach for power flow analysis, further research is needed to address their limitations and fully realize their potential.

The comprehensive scope of this research is intricately focused on utilizing GNNs to address the complexities of the ACOPF problem within electrical power distribution networks. This study delves into the development and application of both homogeneous and heterogeneous GNN models, each specifically tailored to adapt to the unique dynamics of power flow in these networks. The methodology employed in this research is multifaceted, encompassing a structured approach towards data preparation, modeling of energy grids, and detailed architectural development of GNNs.

The initial phase of this research includes an in-depth review of related works and studies, providing a foundational understanding of the existing methodologies and their limitations in solving the ACOPF problem. This review serves as a springboard for the development of the unique approaches adopted in this study. Following this, the research methodically addresses the process of data preparation, emphasizing the significance of accurately defining edge and node features within the network. This process is crucial for constructing a comprehensive representation of the network that the GNN models can interpret and analyze effectively.

Subsequently, the research outlines the additional processing steps required for both homogeneous and heterogeneous approaches, elucidating the distinctive strategies employed for each. This involves a detailed explanation of the architectures and operations of the GNNs, highlighting how these models are specifically designed and optimized to address the ACOPF problem. The homogeneous approach focuses on learning node representations on fixed and homogeneous graphs, while the heterogeneous approach caters to the complexity and diversity of real-world networks by generating new graph structures and learning effective node representations on these graphs. In terms of methodology, the research is grounded in multifaceted data preparation, which involves the collection, cleaning, and structuring of data from power systems in a format conducive to analysis by GNNs. This is followed by a phase of comprehensive experimentation, where these models are applied to the prepared data, and their performance is evaluated. A crucial aspect of this experimentation is the comparative analysis against traditional numerical solvers, which serves as a benchmark to gauge the effectiveness and efficiency of GNNs in this new application area.

The purpose of this research is to significantly advance the field of energy optimization by leveraging the latest advancements in machine learning, specifically GNNs. By testing these models against conventional numerical solvers, the study aims to demonstrate the superior potential of GNNs in providing efficient and scalable solutions to the complex challenges inherent in power system management. The goal is to establish GNNs as a robust and effective tool for enhancing operational efficiency and reliability in power distribution networks, thereby contributing valuable insights and innovative methodologies to the domain of power systems engineering.

1.2 Overview

In the introduction, the importance of optimizing power flow in power distribution networks for energy efficiency and sustainability is emphasized. Then, existing GNN applications in power flow analysis and the stage for exploring both homogeneous and heterogeneous GNNs for the ACOPF problem are explained, focusing on model development, data preparation, and comparative analysis. Finally, the scope and purpose of the research are highlighted to the reader.

In the background chapter, the foundational concepts of power systems and GNNs relevant to the study of ACOPF are covered. It begins by discussing power system modeling, essential for understanding and managing energy grids, and introduces Direct Current Optimal Power Flow (DCOPF) as a simplified model for power flow analysis. The chapter then delves into the complexities of ACOPF, a critical model for efficient and reliable power grid operation. Subsequently, it shifts focus to GNNs, explaining their role in processing graph-structured data through feature propagation and aggregation.

The third chapter details the approach used in the research to tackle the ACOPF problem using GNNs. It begins with insights from related works, then explains data preparation for energy grids, focusing on edge and node features. Both homogeneous and heterogeneous approaches for GNN modeling are discussed, emphasizing systematic preprocessing of network data and leveraging deep learning to optimize power flow. Homogeneous GNNs are explored through DCOPF-based data preparation and a transformer-based architecture, while heterogeneous GNNs consider different node types and employ edge and node type-based models.

In the fourth chapter, the implementation and numerical experiments of approaches to solve the ACOPF problem using GNNs are detailed, expanding on the formulation presented in the previous chapter. Numerical experiments involve training on datasets with varied load profiles, evaluating models using various numerical performance metrics. Subsequently, the results are shared and discussed.

Finally, the conclusion is shared based on the key findings and future outlook of the research.

2 Background

2.1 ACOPF

2.1.1 Power System Modeling

Power system modeling is a critical aspect of energy grid management and planning. It involves the use of mathematical models and simulations to analyze and predict the behavior of the electric power system [9]. These models help researchers understand the impacts of variability and uncertainty on power system operations.

The structure of power system modeling typically includes various components of the energy system, such as power plants, storage systems, and energy networks [10]. It also considers a multitude of factors; including future electricity demand, fuel prices, technology cost, performance, policy and regulation [9].

One of the key aspects of power system modeling is the simulation of generation and transmission capacity investment. This involves planning a resource portfolio for the future, determining what type of generation should be built to meet demand, and assessing whether it will necessitate the development of new transmission capacity [10].

As for the electrical characteristics, power system modeling takes into account various factors such as voltage levels, current flows, power factor, and phase angles. It also considers the physical properties of the system components, including the impedance of transmission lines and the reactive power capabilities of generators [10].

Several research studies have been conducted in the field of power system modeling. For instance, the Advanced Grid Modeling Program under the United States' Department of Energy's Office of Electricity supports the nation's foundational capacity to analyze the electric power system using big data, advanced mathematical theory, and high-performance computing. In the context of renewable energy integration, energy storage systems are proving to be indispensable. They facilitate the integration of renewable energy sources, which are being widely deployed in both microgrids and bulk power systems, and thus will be the hallmark of the clean electrical grids of the future [10].

In the context of graph theory, a power system can be represented as a graph where nodes represent buses; while edges represent transmission lines. This graph representation is particularly useful for analyzing the topological properties of the power grid, such as connectivity and centrality [11].

GNNs have been proposed as a novel method for power reliability studies. A GNN model is trained using historical power system data to predict power flow outcomes [12]. The GNN model enables rapid estimation of line flows. This implies that the agent

can solve the problem independently of the power system grid structure [13].

In conclusion, power system modeling plays a crucial role in the efficient operation and management of energy grids. It provides valuable insights into the current state of the grid and helps in making informed decisions for future needs. As the energy sector continues to evolve, power system modeling will remain a vital tool for addressing the challenges and opportunities that lie ahead.

2.1.2 DCOPF

One of the common approaches to solve OPF is to use a linear approximation of the power flow equations, known as the Direct Current (DC) power flow model.

DCOPF is a mathematical model used in power systems to determine the optimal dispatch of power generation while minimizing the total generation cost [14]. It is widely used in electricity markets for market simulation and planning due to its robustness and speed [15].

The DCOPF model is based on several assumptions. It linearizes and approximates the ACOPF, ignoring the complexities of AC [14]. It assumes that power losses in transmission lines are negligible, leading to a lossless power flow model. It also assumes that the voltage magnitude at all buses is constant and that the phase angle differences between buses are small.

Under these assumptions, the OPF problem reduces to a Linear Programming (LP) problem, which can be solved efficiently by standard LP solvers. However, the DC power flow model also introduces some errors and limitations, such as ignoring the power losses.

The DCOPF equations for active power are given by:

$$\min_{P_G} \sum_{i=1}^{N_G} c_i P_{G_i} \tag{2.1}$$

subject to:

$$P_{G_i} - P_{D_i} = \sum_{j=1}^{N_B} B_{ij}\theta_j, \quad i = 1, ..., N_B \tag{2.2}$$

$$P_{G_i}^{\min} \leq P_{G_i} \leq P_{G_i}^{\max}, \quad i = 1, ..., N_G \tag{2.3}$$

$$\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}, \quad i = 1, ..., N_B \tag{2.4}$$

$$-f_{ij}^{\max} \leq B_{ij}(\theta_i - \theta_j) \leq f_{ij}^{\max}, \quad (i, j) \in E \tag{2.5}$$

where P_G is the vector of generator active power outputs, c_i is the cost coefficient of generator i, P_D is the vector of load active power demands, B is the bus susceptance matrix. N_G is the number of generators in Eq. 2.3 and N_B is the number of buses in Eq. 2.2 and 2.4.

θ is the vector of bus voltage angles, E is the set of transmission lines and f_{ij} is the active power flow limit on line (i, j) in Eq. 2.5. As the optimization problem is restricted to network loss minimization in this study, cost coefficients are neglected further on.

The DCOPF solutions serve as valuable suboptimal initial vectors for the subsequent ACOPF problem, enabling a more efficient convergence toward optimal states. Mathematically, this is exemplified by considering the OPF problem as minimizing a cost function, represented in Eq. 2.1, subject to various constraints including active power balance equations, generator output limits, bus voltage angle constraints, and transmission line flow limits.

The DCOPF solutions, derived under the DC power flow model, approximate this complex problem within these constraints. These solutions, however, exhibit inherent inaccuracies due to their simplified nature, as they neglect reactive power and voltage magnitudes and assume zero line resistances. Despite these limitations, these solutions form a robust foundation for the ACOPF problem.

Utilizing these DCOPF results as initial vectors in the ACOPF context can be mathematically expressed as incorporating these solutions $(P_i, \theta_i)\ \forall i$ within the constraints of the ACOPF equations, thereby jump-starting the iterative optimization process.

Employing these approximate solutions aids in narrowing down the solution space, facilitating a more efficient exploration for the ACOPF problem to converge towards the true optimal solution. This strategic utilization of DCOPF solutions minimizes computational effort and accelerates the convergence of iterative algorithms in ACOPF.

2.1.3 ACOPF Problem Formulation

The ACOPF is a fundamental mathematical model used in power systems for determining the optimal dispatch of power generation [16][17].

The ACOPF problem aims to find the most economical generation dispatch that satisfies operational boundaries, such as network flow constraints, generator capacity constraints, line capacity constraints, and bus voltage constraints [16].

The ACOPF problem subject to this study can be formulated $\forall i \in \mathcal{N}$; with $\mathcal{N}$ being the number of buses in a network, as follows:

$$\min_{P,Q,V,\theta} \sum_{i=1}^{N} c_i(P_i^2 + Q_i^2) \tag{2.6}$$

subject to:

$$P_i = \sum_{j=1}^{N} |V_i||V_j|(G_{ij}\cos(\theta_i - \theta_j) + B_{ij}\sin(\theta_i - \theta_j)) \tag{2.7}$$

$$Q_i = \sum_{j=1}^{N} |V_i||V_j|(G_{ij}\sin(\theta_i - \theta_j) - B_{ij}\cos(\theta_i - \theta_j)) \tag{2.8}$$

$$Pi^{\min} \leq P_i \leq P_i^{\max} \tag{2.9}$$

$$Q_i^{\min} \leq Q_i \leq Q_i^{\max} \tag{2.10}$$

$$V_i^{\min} \leq V_i \leq V_i^{\max} \tag{2.11}$$

$$\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max} \tag{2.12}$$

$$P_{ij}^2 + Q_{ij}^2 \leq S_{ij}^{\max}, \quad (i,j) \in E \tag{2.13}$$

where P and Q are the vectors of active and reactive power outputs in equation, c_i is the cost coefficient in Eq. 2.6. V and θ are the vectors of bus voltage magnitudes and angles, G and B are the bus conductance and susceptance matrices in Eq. 2.7 and 2.8. E is the set of transmission lines, S_{ij} is the maximum apparent power flow limit on line (i,j) w.r.t the connected busses.

P_{ij} and Q_{ij} are the active and reactive power flows on line (i,j) in Eq. 2.13.

The ACOPF formulation is a non-convex optimization problem, making it computationally complex (NP-hard). Despite this, it has been one of the most active research areas in the power system community.

Therefore, various numerical solvers have been developed to solve the ACOPF problem, such as gradient-based methods, Newton-based methods, interior point methods, convex relaxation methods, etc [18]. However, these solvers may not guarantee global optimality or scalability for large-scale power systems.

The ACOPF finds its greatest practical application in electric power systems markets where electric power is sold at the minimum cost while still securing the reliability of the electric power grid.

Specific to this study, main goal is thus to approximate the ACOPF results of numerical solvers faster and more efficiently; using various GNN model architectures.

2.2 Graph Neural Networks

GNNs are a class of deep learning models designed to perform inference on data described by graphs. They are capable of processing structured data inputs, such as acyclic graphs, cyclic graphs, and directed or undirected graphs [19]. GNNs have been widely applied in various domains, including social networks, sensor networks, functional networks in brain imaging, regulatory networks in genetics, and meshed surfaces in computer graphics [20][21].

GNNs operate by implementing a function that maps a graph and one of its nodes onto an m-dimensional Euclidean space [19]. This is achieved by combining the feature information and the graph structure to learn better representations on graphs via feature propagation and aggregation. The GNN model can be seen as an extension of the classic feedforward neural networks, with the added capability of handling non-Euclidean data [20][21].

In terms of terminology, the fundamental components of GNNs are nodes and edges, which represent entities and their relationships, respectively. GNNs operate by performing message passing between the nodes of graphs, recursively aggregating the representations of each node's neighbors and itself [11].

The architecture of GNNs typically involves several layers of transformation. Each layer applies a propagation rule that transforms the features of a node based on its own features and the features of its neighbors [11]. This process is repeated for a fixed number of steps or until the node representations converge.

Several variants of GNNs have been proposed, including Graph Convolutional Networks (GCNs), Graph Recurrent Networks, Graph Attention Networks, and Graph Residual Networks [22]. These variants have demonstrated groundbreaking performances on many deep learning tasks [21][23].

Furthermore, a graph transformer operator is a generalization of the transformer neural network architecture for arbitrary graphs. It uses self-attention to encode the structural and positional features of each node in the graph, and can be applied to various graph representation learning tasks, such as semi-supervised classification, graph generation, and video prediction [24]. A graph transformer operator can be stacked to form a deep Graph Transformer Network (GTN), which can capture complex graph patterns and dynamics [25].

All models designed, implemented and used in this study utilize a specific layer type named TransformerConv. TransformerConv layers are built on a graph transformer operator introduced by Shi et al. [26].

The field has also seen the development of models that focus on specific aspects of GNNs. For instance, the study of adversarial attacks and defenses on graph data has gained attention [27]. Similarly, research on the computational capabilities of GNNs has been conducted, characterizing the functions that can be approximated by GNNs [19].

In the context of quantum chemistry, Neural Message Passing has been introduced as a method for predicting the quantum mechanical properties of small organic molecules [28]. Non-local Neural Networks have also been proposed as a building block for

capturing long-range dependencies [29].

Recent surveys have provided comprehensive overviews of the methods and applications of GNNs [21][23], including discussions on the challenges and future research directions in this rapidly growing field. Topics such as the explainability in GNNs, benchmarking of GNNs, and the use of GNNs for dynamic network modeling have been explored [20]. Furthermore, research has been conducted to bridge the gap between spatial and spectral domains in GNNs [30].

In conclusion, GNNs represent a significant advancement in the field of deep learning, offering a powerful tool for handling graph-structured data. The ongoing research and development in this area continue to push the boundaries of what is possible with GNNs, opening up new avenues for exploration and application.

3 Approach

This chapter provides a comprehensive guide on data preparation, modeling of energy grids, applied methodologies and the architecture of homogeneous and heterogeneous GNNs to tackle the ACOPF problem as defined in this study.

Firstly, insights from related works and studies are shared to build a foundation for the approaches employed in the study.

Secondly, the methods on data preparation regarding edge and node features are explained to the reader.

Following this, the additional processing steps for both homogeneous and heterogeneous approaches are given in detail, followed by a detailed guide on the architectures and operations of GNNs.

In summary, the approaches employed in the study involve systematic preprocessing of network data to construct appropriate graph structures, enabling the GNNs to capture relationships and dependencies. Emphasis is placed on leveraging the power of deep learning to learn complex patterns from the data, enabling the models to optimize power flow effectively.

3.1 ACOPF with GNNs

3.1.1 Related Works and Studies

ACOPF is a critical problem in power systems, and GNNs have been increasingly utilized to address it. The application of GNNs to ACOPF has been explored in both homogeneous and heterogeneous approaches for node types and edge types.

In the homogeneous approach, GNNs have been designed to learn node representations on fixed and homogeneous graphs [31][32]. For instance, Gopinath et. al [33] conducted a comprehensive benchmarking effort to evaluate and compare state-of-the-art open-source tools for solving the ACOPF problem. They used numerical experiments that included all instances found in the public library PGLIB with network sizes up to 30,000 nodes.

Zhang et. al [34] used deep neural networks to predict the solutions of ACOPF problems. They proposed a Lagrangian-based approach to overcome the challenge, regarding the training data may contain suboptimal solutions; where the model predicted voltage magnitude and voltage phase angle given power inputs.

On the other hand, the heterogeneous approach considers the complexity and diversity of real-world networks. Yun et al. [32] proposed GTNs that are capable of generating new graph structures while learning effective node representations on the new graphs.

They demonstrated that GTNs and FastGTNs with non-local operations achieve state-of-the-art performance for node classification tasks.

Several further research studies have been conducted, which have made significant contributions to the application of GNNs in power systems and further improvements to tackle the ACOPF challenge via GNNs; employing different approaches for various domains of the problem. A quasi Monte-Carlo method based on GCN was proposed to calculate distribution characteristics of power flow system [35]. This novel approach aimed to boost computational accuracy by leveraging the breakthroughs of deep learning.

Donon et. al [12] proposed a neural network architecture that emulates the behavior of a physics solver; which solves electricity differential equations to compute electricity flow in power grids. This innovative approach was designed to overcome the limitations of current methods, which are slow and do not scale with the size of the power system network.

Kim et. al [2] applied a GCN model for predicting an optimal load-shedding ratio that prevents transmission lines from being overloaded under line contingency. They demonstrated that the GCN model outperforms other models by an order of magnitude in terms of accuracy and computational speed.

Liao et. al [11] provided a comprehensive overview of GNNs in power systems, where they additionally summarized several classical paradigms of GNN structures and reviewed key applications in power systems in detail.

Lastly, Owerko et. al [36] proposed using GNNs trained under the imitation learning framework to approximate a given optimal solution, demonstrating the viability of their approach on standard Institute of Electrical and Electronics Engineers (IEEE) power grids, both in terms of accuracy and computational time.

In conclusion, GNNs have shown promising results in solving the ACOPF problem, providing efficient and scalable solutions. However, further research is needed to improve the robustness and generalizability of these models, especially in the face of increasingly complex and dynamic power systems.

Throughout the scope of this study, datasets for both homogeneous and heterogeneous GNNs are collected, sampled and processed respective to the model architectures.

The data model employed adopts a structured approach to represent the dynamics of the electrical transmission grid. Each bus, akin to a node within the graph representation, is characterized by four distinct features essential for comprehensive analysis.

These features include v_n, denoting the voltage at the node; δ_n, representing the voltage angle at the node relative to the slack bus; p_n, signifying the net active power flowing into the node; and q_n, indicating the net reactive power flowing into the node. These parameters collectively encapsulate the state of the n^{th} node, expressed as real scalars, and are integral to understanding the grid's operational dynamics.

Furthermore, the physical attributes of the network are encapsulated within the power flow equations, $p = P(v, \delta, W)$ and $q = Q(v, \delta, W)$. These equations establish the relationship between local net power generation and the global state of the network,

with W representing the topology of the grid.

This grid, essentially an electrical network, is abstracted into a graph, wherein nodes symbolize power-producing, consuming and neutral entities; while edges denote the electrical connections between these nodes. The energy grid is then converted step by step to a network graph consisting of nodes and edges with inherent features extracted from the grid itself.

To facilitate analysis, the data is structured into two fundamental matrices: the State Matrix $X \in \mathbb{R}^{N\times 4}$ and the Adjacency Matrix A. X encapsulates the graph's features. Each row in X corresponds to the state of the respective node, incorporating the four aforementioned features.

Additionally, the Adjacency Matrix A, a sparse matrix belonging to $\mathbb{R}^{N\times N}$, captures the network connections. Specifically, $A_{ij} = 1$ if node i is connected to node j, providing a concise and efficient representation of the grid's topology.

This structuring of the data forms the foundation upon which the subsequent analyses and computations in this study are based, ensuring a rigorous and methodical approach to the research.

3.1.2 Edge Features

The processing step mutual for both homogeneous and heterogeneous approaches begins with the extraction of edge features based on individual bus connections and transmission line properties.

Within the framework of this research, an algorithm is deployed for the precise computation of edge attributes, denoting the electrical line parameters in the grid.

The algorithm's systematic approach involves a step-by-step analysis of each line connecting buses; excluding external grid bus and converted generator buses, which are connected to the network through transformers, ensuring a focused evaluation.

Edge Attributes

For every line, the algorithm calculates the resistance $r_{\text{ohm per km}}$ and reactance $x_{\text{ohm per km}}$ values in relation to the specific source and destination buses, taking into account the line's length $length_{\text{km}}$. Subsequently, the total resistance R_i and reactance X_i for each line are derived, encapsulating the essential electrical properties of the lines.

These derived parameters are fundamental attributes characterizing the edges within the network graph, facilitating subsequent objective analyses.

The algorithm further takes the undirected nature of the edges into consideration, enabling symmetric evaluations, which are pivotal for rigorous graph-based analyses.

Edge attributes representing short-circuit impedance values are calculated using precise formulas for the connections of external grid bus and the buses converted from external grid buses.

For each line connecting these nodes, the short circuit impedance values (Z, R, X) are computed as follows:

$$Z_k = V_k \times \frac{S_N}{100 \times S} \tag{3.1}$$

$$R = V_{kr} \times \frac{S_N}{100 \times S} \tag{3.2}$$

$$Z_N = \frac{V_N^2}{S_N} \tag{3.3}$$

$$Z_{\text{ref,trafo}} = V_N^2 \times 10^6 \times \frac{S_N}{S} \tag{3.4}$$

$$Z = \frac{Z_k \times Z_{\text{ref,trafo}}}{Z_N} \tag{3.5}$$

$$X = \sqrt{Z^2 - R^2} \tag{3.6}$$

In the context of the provided equations, V_k signifies the transformer's relative short-circuit voltage percentage; V_{kr} represents the real part of relative voltage short-circuit percentage.

The variable S represents the apparent power rating of the transformer, measured in mega-volt-amperes MVA. It encompasses both real (active) and reactive power components in an AC circuit. V_N signifies the nominal voltage at the low voltage level side of the transformer, expressed in kilovolts kV.

The term Z_K in Eq. 3.1 corresponds to the short circuit impedance, calculated as the product of V_k (in percentage), system apparent power S_N, and a scaling factor involving 100 times the transformer's apparent power S. The variable R denotes the resistance, computed using V_{kr}, system apparent power S_N, and a scaling factor involving 100 times the transformer's apparent power S, subsequently transformed into a tensor.

Z_N represents the squared nominal voltage V_N of the transformer divided by the system apparent power S_N. The term $Z_{ref,trafo}$ in Eq. 3.4 stands for the squared nominal voltage of the transformer multiplied by 10^6 and then scaled by the system apparent power S_N and the transformer's apparent power S. Z signifies the short circuit impedance, calculated as the product of Z_K and $Z_{ref,trafo}$, divided by Z_N.

Lastly, X represents the reactance, determined using the Pythagorean theorem in Eq. 3.6, ensuring compliance with the fundamental relationship between resistance, reactance, and impedance in electrical circuits. Thus, the transformers connecting the external grid bus and from-external-grid-bus-converted generator buses are modeled as lines; providing the edge features similar to the rest of the bus-to-bus connections in the network.

For the transmission lines connecting the remainder of buses, the edge attributes are incorporated into the network graph as follows: for each line, the corresponding edge

is established with attributes (R, X), symbolizing the resistance and reactance values; extracted from the pre-set values of the grid.

These edges, representing the electrical connections, are vital components of the graph representation, allowing for comprehensive analysis of the network's structural and operational characteristics.

After collecting the edge attributes from the respective lines and transformers modeled as lines; the next step involves the incorporation of the edge indices; which display the topology of the network graph.

Edge Indices

In the context of the graph representation used in this research, the extraction of edge indices is a fundamental step. These edge indices are structured as a tensor of shape $2 \times num\_edges$, where the first row corresponds to the bus indices of source buses, and the second row represents the bus indices of target buses; playing a crucial role in defining the connectivity of the graph.

The process of extracting the edge indices from the adjacency matrix A involves mapping the electrical connections between nodes onto the graph structure. Each entry A_{ij} in the adjacency matrix signifies the existence of an electrical connection between bus i and bus j.

By iterating over the rows and columns of A, the non-zero entries indicate the presence of edges in the graph. For each non-zero entry, the corresponding bus indices i and j are recorded in the edge_index tensor.

This extraction process ensures that the resulting graph representation accurately reflects the underlying electrical connections within the power grid, enabling precise analysis and computations in subsequent stages of the research.

By collecting all edge attributes and edge indices of the network graph, the topology and inter-node connectivity are defined. The upcoming step of processing involves the node features; which together represent the state of the graph.

3.1.3 Node Features

After systematically extracting the edge attributes and edge indices between individual buses for the base graph structure, the computation and extraction of node features follow. For active and reactive power, the sign convention is parallel to the consumer perspective; such that a positive active or reactive power value of a node indicates the consumption of power.

All approaches for edge and node features shared are the same for both homogeneous and heterogeneous datasets until this section, for the respective model structure. However, heterogeneous models require further processing steps; as explained in the following part.

In the process of generating the input and target datasets for this study, two distinct methodologies were employed.

Firstly, inputs were derived from the results of DCOPF computations performed by numerical solvers for the homogeneous approach.

Secondly, a comprehensive methodology for the heterogeneous approach was followed by employing inputs based on the constraints w.r.t each node type.

In both approaches, ACOPF results of numerical solvers from the respective grids are utilized as targets.

By combining these two methodologies, the input and target datasets for grids were curated, forming the foundation upon which the study's analyses and conclusions were based.

Following the uttered steps, input and output datasets are standardized to have 0 mean and unit variance.

For a dataset consisting of inputs X and targets Y consisting of N nodes, the mean vector $\mu \in \mathbb{R}^{4\times 1}$ is calculated as:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{3.7}$$

whereas the standard deviation vector $\sigma \in \mathbb{R}^{4\times 1}$ is computed as follows:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \mu)^2} \tag{3.8}$$

then, each feature vector in both the input and target datasets are standardized using the following equation:

$$z_i' = \frac{z_i - \mu}{\sigma} \tag{3.9}$$

3.2 Homogeneous GNN

3.2.1 Data Preparation for Homogeneous GNN

In terms of the homogeneous datasets, the node features and the constraints are generated according to the DCOPF results of the numerical solvers.

Each input X for the homogeneous datasets thus has the shape $\mathbb{R}^{N\times 4}$, consisting of the node features. The processed networks are inputted to the numerical solvers and the resulting data is processed in the following steps.

These results are thought to serve as suboptimal initialization values for each node, helping the homogeneous model converge to a solution faster and more efficiently.

The input and output generation for the homogeneous approach is described by the following pseudo-algorithm:

Algorithm 1 Homogeneous Input and Output Generation

1: Sample reference loads uniformly.
2: Set upper and lower limits of active-reactive powers of loads.
3: Replace all external grids but the first one with generators and set the generators to non-slack.
4: Maximize the branch limits.
5: Maximize line, trafo, and trafo3w loading percents.
6: Run DCOPF and ACOPF numerical solvers and handle exceptions if no convergence.
7: Store the resulting data points in a dataframe and drop unnecessary columns.
8: Set the initial voltage magnitudes and reactive powers at each bus index to that given in network.
9: Store the resulting grid datasets locally in CSV format.

After collecting the datasets for each grid, these are stored locally w.r.t the grid name for further processing.

The implementation of homogeneous data preparation for training, validation and test datasets including the edge attributes can be summarized by the following pseudo-algorithm:

Algorithm 2 Homogeneous Data Preparation

1: Process the network graph of the grid and map indices of buses to ascending correct order.
2: Calculate edge attributes and indices for the graph.
3: Convert edge connections to undirected and add self loops to edge connections.
4: Store path to the supervised datasets directory of the specified grid.
5: Read all the CSV files in the directory of the grid and process all the data according to 85 - 10 - 5 split.
6: For each data point in training, validation, and test sets, process the data and append to respective lists.
7: **return** Training, Validation and Test Datasets & Edge features

The approaches described in both pseudo-algorithms display the overview of the steps taken from data generation to actual training and evaluation phase of the homogeneous approach.

The following training and evaluation steps of the homogeneous model is trivial. The training and validation datasets, consisting of the inputs X and targets Y with the corresponding edge attributes for each network graph are utilized; followed by the final testing.

3.2.2 Homogeneous Model

The model described in this research employs a homogeneous GNN architecture based on the TransformerConv layer. This architecture is structured as a series of layers, each utilizing the TransformerConv layer to process input data.

The node transformation equation of a TransformerConv layer is formulated as follows:

$$\mathbf{x}_i' = \mathbf{W}_1\mathbf{x}_i + \sum_{j\in\mathcal{N}(i)} \alpha_{i,j}\left(\mathbf{W}_2\mathbf{x}_j + \mathbf{W}_6\mathbf{e}_{ij}\right) \tag{3.10}$$

where e denotes the edge attributes between nodes i and j and α denotes the attention coefficients; computed according to the following equation:

$$\alpha_{i,j} = \text{softmax}\left(\frac{(\mathbf{W}_3\mathbf{x}_i)^\top(\mathbf{W}_4\mathbf{x}_j + \mathbf{W}_6\mathbf{e}_{ij})}{\sqrt{d}}\right) \tag{3.11}$$

with d being the multi-head count.

Eq. 3.10 describes the transformation of node embeddings $\mathbf{x}_i'$ in the graph. Here, $\mathbf{x}_i$ represents the input node features, $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_6$ are learnable weight matrices, and $\mathbf{e}_{ij}$ signifies the edge attributes between nodes i and j. The summation is taken over neighboring nodes $j \in \mathcal{N}(i)$, where $\mathcal{N}(i)$ represents the neighborhood of node i in the graph.

Eq. 3.11 outlines the calculation of attention coefficients $\alpha_{i,j}$ utilizing a softmax function. The attention mechanism computes the relevance of neighboring nodes' features concerning the current node i. The attention scores are determined based on the dot product of transformed node features $\mathbf{W}_3\mathbf{x}_i$ and $\mathbf{W}_4\mathbf{x}_j$, along with the edge attributes $\mathbf{W}_6\mathbf{e}_{ij}$. The division by $\sqrt{d}$ normalizes the attention coefficients, where d represents the dimensionality of the input features.

The reason for the selection of TransformerConv layers resides in the regarded similarity and resemblance of the ACOPF equations for active and reactive power given in Eq. 2.7 and 2.8; with an addition of self-loop concept introduced by the term $\mathbf{W}_1\mathbf{x}_i$ and a non-linear activation function; Exponential Linear Unit (ELU), allowing additional non-linearity for the expressiveness of the model aiming to tackle the challenge of ACOPF approximation based on the DCOPF results of the numerical solvers.

The model's core components include these layers, specialized for graph-structured data processing. These layers handle node features represented as tensors and edge indices denoting relationships between nodes. The utilization of graph topology is aimed to enhance the model's ability to capture intricate power system dynamics.

A crucial design feature is the incorporation of multiple layers, allowing the information processing. Each layer refines node embeddings iteratively, enabling the model

to discern local and global patterns within the power system. The iterative nature of the layers facilitates the extraction of increasingly abstract features as data propagates through the network.

In the forward pass of the homogeneous GNN model, the node embeddings are updated iteratively through a series of TransformerConv layers. The process involves fundamental equations, denoted as Eq. 3.10 and Eq. 3.11, which govern the transformation of node features within the model.

In summary, the study employs a specialized GNN architecture based on the TransformerConv layer for tackling the described ACOPF problem. This architecture leverages the graph transformer operator, a versatile framework capable of capturing intricate graph patterns and dynamics. The core components of the model include TransformerConv layers, designed specifically for processing graph-structured data. These layers handle node features and edge features, enhancing the model's ability to understand complex power system dynamics.

Incorporating these architectural elements and equations, the model aims to demonstrate an understanding of the power system topology and its dynamic behavior in the homogeneous approach. This nuanced representation forms the foundation for accurate ACOPF estimation, showcasing the proposed approach.

3.3 Heterogeneous GNNs

3.3.1 Data Preparation for Typed Heterogeneous GNNs

In contrast to the homogeneous approach, the need to take additional steps within the heterogeneous approach for data preparation becomes apparent, particularly when considering different node types like slack nodes, generator nodes, load nodes, and neutral nodes.

Each of these node categories presents unique characteristics, demanding tailored processing techniques to accurately represent their behaviors in the overall grid model. This diversity underscores the significance of adopting a comprehensive approach to data processing in power systems.

The strategic classification of buses into discrete categories constitutes a pivotal aspect of research endeavors. Far beyond organizational taxonomy, this categorization is rooted in the nuanced and multifaceted nature of modern electrical grids.

Each category embodies distinct functional roles, encapsulating unique electrical attributes and operational significance.

The slack bus, for instance, stands as a reference point for voltage phase angle. Generator buses, on the other hand, emerge as primary sources of electrical power, their outputs intricately shaping the grid's overall generation capacity.

Load buses, conversely, signify focal points of electrical consumption, representing the demand side of the network dynamics. In this intricate tapestry, neutral buses, although often overlooked, represent grounding points crucial for ensuring both safety

and system integrity.

By methodically categorizing buses based on their roles, engineers and researchers can construct targeted models, conduct simulations, and perform in-depth analyses.

This precision enables the accurate representation of grid behaviors under diverse operational scenarios, fostering comprehensive investigations into stability, efficiency, and reliability aspects.

Consequently, this systematic categorization serves as the cornerstone of scholarly research efforts, ensuring a profound understanding of the intricacies inherent in modern power systems and facilitating their adept management and optimization.

The intricacies of the characteristics, categorization criteria and the formulation of the constraints for each node type are explained in more detail in the following.

Slack Node

As previously mentioned in section 3.1.2, all external grid buses, but one present in the network, are converted to generator buses as a step of network processing on each transmission grid. The remaining external grid bus is regarded as the slack node further on.

This specific node in the network graph presents itself as the reference point for all voltage angle calculations.

Additionally, in line with the properties of an external grid bus; slack node is also responsible for the balancing of power surplus and deficit within the context of network loss minimization process.

The voltage angle of every other node in the network graph shall be calculated w.r.t the voltage angle of the slack node, thus leading to the slack node having zero as initial voltage angle.

The lower and upper limits on voltage magnitude V_{trafo}^{min} and V_{trafo}^{max} represent the voltage magnitude limits of the corresponding transformer.

Analogously, the limits on the active power P_{SN}^{min} and P_{SN}^{max}; as well as the limits on the reactive power Q_{SN}^{min} and Q_{SN}^{max} are bounded by the maximum apparent power limit of the respective transformer.

Including the slack node, each node type has its constraints; which are extracted from the network itself.

In this case, the constraints on the node features of the slack node are given in the following equations:

$$V_{trafo}^{min} \leq V_{SN} \leq V_{trafo}^{max} \tag{3.12}$$

$$\theta_{SN} = 0.0 \tag{3.13}$$

$$\sqrt{P_{SN}^2 + Q_{SN}^2} \leq S_{trafo} \tag{3.14}$$

where V_{SN}, θ_{SN}, P_{SN} and Q_{SN} denote the node features of the slack node. Eq. 3.13 formulate the constraint on the voltage angle θ. Eq. 3.12 and 3.14 present the lower and upper boundary constraints on the voltage magnitude, active and reactive power.

Generator Node

In the process of categorizing buses within the power grid, the identification of generator buses is a critical step, influencing subsequent analyses and simulations. The categorization algorithm operates methodically, taking into account the interconnections between buses and their associated elements.

Firstly, the algorithm establishes a mapping between given bus indices and real indices to ensure accurate representation within the computational framework and graph structure. The slack bus, denoted as the reference point for the entire network, is initially identified and removed from the list of bus indices.

The algorithm then systematically evaluates each remaining bus. For generator nodes, the algorithm looks for buses with attached dynamic generators and static generators. The previously external grids in the network, which are converted to buses with linked dynamic generators are classified as generator nodes.

For the remainder of buses, the criterion for classification is trivial. If the maximum apparent power of a static generator surmounts the power demand from a load in that specific bus, it is unequivocally classified as a generator node.

This analysis ensures that all buses designated as generator nodes are accurately identified, incorporating both the presence of generators and the absence of significant active power demands from loads.

The lower and upper boundaries for active and reactive power, excluding the from-external-grid-converted-buses; $\forall i \in \mathcal{N}_{\mathcal{G}}$, with N_G being the number of generator nodes in the graph are derived as follows:

$$\sqrt{P_i^2 + Q_i^2} \leq S_{sgen,i} \tag{3.15}$$

where S_{sgen} denotes the maximum apparent power of the static generator in the respective bus, which is extracted from the pre-set value provided by the framework being used for network processing.

Regarding the sign convention adhered by throughout the study, power generation is characterized by negative power values. Thus, the boundaries for the active and reactive power are set accordingly.

However, the constraints of from-external-grid-converted generator nodes are derived analogous to the slack node.

By aligning these boundaries with the limits imposed by static generators, the study aims to encourage optimal utilization of locally available resources, paving the way for sustainable and resilient power distribution networks.

This systematic approach is followed to guarantee a precise classification of generator nodes, forming a fundamental component of the broader categorization process.

By accurately identifying these buses, the subsequent modeling and simulation processes can capture the dynamic behaviors and contributions of generators within the power network, enabling the utilization of machine learning techniques for this specific task.

Load Node

The accurate delineation of load nodes within the power grid is of paramount importance for comprehensive analysis. For this node type, a similar algorithm to generator nodes is applied, focusing exclusively on nodes designated as load buses.

This systematic process, devoid of complexities introduced by other bus types, ensures precise categorization and subsequent modeling.

A load node within the power grid is characterized by the presence of an active electrical load connected to the corresponding bus. In power systems, electrical loads represent the devices or components that consume electrical energy, such as residential homes, industrial machinery, or commercial establishments.

The algorithm initiates by evaluating each bus's characteristics, specifically concentrating on active and reactive power parameters. For load nodes, the algorithm identifies the connected static generators and loads, crucial components influencing the power balance.

When a bus in the power grid is linked to one or more of these consuming devices and the power demand from these devices surmount the maximum apparent power limit of a static generator in that corresponding bus, it is classified as a load node.

The identification of load nodes is fundamental in power system analysis, as they signify points where electrical energy is drawn from the grid to power various applications.

Through careful examination, the algorithm calculates the minimum and maximum active and reactive power boundaries for these nodes, aligning with the study's overarching goal of promoting self-sufficiency and grid robustness.

The lower and upper boundaries for active and reactive power for load nodes $\forall i \in \mathcal{N}_{\mathcal{L}}$, with N_L being the number of load nodes in the graph are derived as follows

$$\sqrt{P_i^2 + Q_i^2} \leq S_{load,i} - S_{sgen,i} \tag{3.16}$$

where $S_{load,i}$ denotes the apparent power demand limit of the load at ith bus. The reasoning behind this approach is to enforce self-sustenance through maximal utilization of local power generation resources, in case of an existing static generator.

By balancing these reactive power boundaries, the aim is to ensure not only stability but also optimal utilization of locally available resources.

This approach, akin to the strategy applied to generator nodes, ensures that load nodes, all of which are inherently connected to loads, operate within defined power boundaries.

Such precise delineation of power limits enables the utilization of available resources, paving the way for the establishment of sustainable and resilient power distribution networks.

Neutral Node

The last node type subject to the data post-processing method in this study is the neutral node.

Neutral nodes emerge as a distinct entity, marked by their unique absence of any attached loads or generators. Devoid of any active generators or loads, these nodes are identified based on the absence of connections to both dynamic and static generators as well as loads of any kind.

This rigorous criterion ensures that neutral nodes remain unburdened by any localized power generation or consumption, emphasizing their fundamental role as stabilizing entities within the network.

The presence of neutral nodes holds strategic importance for the understanding of network graphs for the task in this study.

Through the introduction of this node type, it is aimed for the neutral nodes to serve as balancing points within the network; considering their absence of active and reactive power-related components.

Therefore, they should be regarded as power transmitting, binding components in the network, neither generating nor consuming power.

Following this approach, the lower and upper boundaries for active and reactive power are set to zero $\forall i \in N_N$; with N_N being the number of neutral nodes in the network graph.

3.3.2 Heterogeneous Processing

For all node types, computation of input node features is identical. $\forall i \in \mathcal{N}$, the voltage magnitude V_i is set to the initial voltage magnitude of the respective bus of the grid. Then, voltage angle θ_i is initialized flat. Finally, the active and reactive powers P_i and Q_i are set to the mean of respective lower and upper boundaries; calculated according to the constraint formulations shared in the preceding section w.r.t the node type.

In addition to the difference in constraints of the approach for the node features, the extraction, classification and storage of edge features are also achieved differently.

The edge feature extraction step can be summarized as follows:

Algorithm 3 Extraction of Edge Attributes for Heterogeneous Datasets

Initialization: Initialize dictionaries for edge types and edge attributes.
Extract node types as a dictionary.
for each line in the network **do**
    Calculate Resistance R_i and Reactance X_i.
    Determine the edge type based on the types of from and to buses.
    Add edge features to the corresponding edge type in the dictionaries.
    if from bus and to bus node types differ **then**
        Add edge features to the corresponding reversed edge type.
    end if
end for
Add the external grid connections.
Add generator bus connections.
for each stored edge type in the processed grid **do**
    Initialize a set for unique edge pairs.
    Get the edge index and edge attribute for the current key.
    Remove duplicate edge pairs from the edge index and edge attribute.
    Store the edge index and edge attribute in the dictionaries.
end for
return Indices-per-edge-type as dict, Attributes-per-edge-type as dict

After collecting the edge indices and attributes, generation of heterogeneous dataset can be summarized as the following pseudoalgorithm:

Algorithm 4 Heterogeneous Data Generation

Initialization: Initialize a Hetero Data Instance.
Process the network and extract node types as a dictionary.
for each node type **do**
    Compute core features
    Compute constraint features.
    Initialize features in line with shared approaches.
    Store the computed features in the Hetero Data Instance.
    Store the ACOPF results of numerical solvers as targets per node type
end for
for each edge type **do**
    Store edge indices and attributes in the Hetero Data Instance.
end for
return Processed Network and Heterogeneous Dataset

As the final step, the voltage magnitudes of targets for each node, which are given

in p.u w.r.t the initial voltage; are multiplied by the respective initial voltage to reverse relativization.

3.3.3 Heterogeneous Models

The heterogeneous models conceptualized and implemented in this study aim at solving the ACOPF approximation task through utilization of the diversity of graph topology, structure and components in a heterogeneous manner.

Edge Type-Based Heterogeneous Model

The model is designed for solving the ACOPF problem in power systems using edge type-based heterogeneous GNNs. The model operates on a graph structure composed of various node types; as shared in the preceding section, and various edge types representing connectivity patterns.

Each connection is categorized and assigned to a specific from node - to node edge type and the layers are stored according to these unique edge types. The edge indices and edge attributes are processed accordingly, building an entire data structure based on the unique edge types consisting of connections of node types.

The model's parameters include the dimensionality of hidden features, output feature dimensionality and the number of layers. It utilizes activation functions, defaulting to ELU; and edge attributes such as conductance and susceptance in graph convolution operations. Same as all, this model is also initialized lazily via a forward pass of a randomly generated dataset.

During the forward pass, the input and constraint features of each node are concatenated to form $\mathbb{R}^{N\times 11}$ vectors. Then, the model applies TransformerConv layers specific to each edge type, leveraging self-attention mechanisms for node feature aggregation.

Aggregation functions like summation are used to combine information from neighboring nodes. Linear transformations are applied to aggregated features, followed by non-linear activations.

Following the node transformations iteratively w.r.t number of layers, the outputs are aggregated using summation. On top of that, the aggregated results are passed to linear layers, which transform the results linearly to actual output shapes of the respective node types; followed by non-linear activations.

Lastly, a final aggregation is done on each vector passed; resulting in the finalized $\mathbb{R}^{N\times 4}$ output vector consisting of $(V_i, \theta_i, P_i, Q_i) \quad i = 1, ..., N$ respective to the corresponding node types.

Forward Pass

The forward pass of the edge-based heterogeneous GNN is highlighted in the following pseudo-algorithm:

Algorithm 5 Forward Pass of Edge Type - Based Heterogeneous GNN

Initialization: Initialize dictionaries and variables.
for each layer in the model **do**
    for each edge type in the input data **do**
        Convolute node features using a TransformerConv layer per edge type.
        Append the convolution result to the output dictionary.
    end for
end for
for each node type in the output dictionary **do**
    Aggregate the results for each node type.
    Apply a fully connected layer based on the node type and a non-linear activation.
    Rearrange and reshape the features to the desired shape.
    Store the processed features in the output dictionary.
end for
return Output dictionary containing processed node features.

Node Type-Based Heterogeneous Model

This specific model designed and implemented in this study represents a node type-based heterogeneous GNN for solving the ACOPF problem in power systems. The model operates on a heterogeneous graph structure with different node types, being slack node, generator node, load node and neutral node.

It consists of transformer-based graph convolutional layers tailored for various edge types in the input graph. The model's parameters include the dimensionality of hidden features, output feature dimensionality, number of graph convolutional layers, activation function with ELU as default, optional normalization layer type and initial data for lazy initialization.

In the forward pass, the model processes nodes of each type separately, considering their neighboring nodes and edge attributes.

Multiple transformer-based graph convolutional layers are applied, each specific to edge type. Features from neighboring nodes are aggregated using self-attention mechanisms and specific aggregation functions, with different schemes applied for different node types. Normalization is then applied to aggregated features based on node types, optionally.

Finally, linear transformations are performed on the normalized features, producing output features for each node type.

The resulting features capture complex relationships and patterns among different node types, which can be utilized for solving the ACOPF problem in power systems.

Forward Pass

A summary of the forward pass for the node type-based heterogeneous model is

given as follows:

Algorithm 6 Forward Pass of Node Type-Based Heterogeneous GNN

```
Initialization: Initialize an empty output dictionary.
for each layer in the model do
    for each node type in input data do
        for each neighbor of node do
            Collect node features of connected nodes
            Collect the corresponding edge attributes and indices.
        end for
        Concatenate edge attributes and node features of neighbor nodes.
        Perform convolution through TransformerConv layer per node-type.
        Append the result to the output dictionary.
    end for
    Group and aggregate convolution results for each node type.
end for
for each node type in output dictionary do
    Apply fully connected layer and reshape the result.
    Store the processed features in the output dictionary.
end for
Return Output dictionary containing processed node features.
```

For this model, in contrast to the edge type-based approach; the forward pass is designed in a way that follows each connection between nodes per node type and thus applies graph convolution on the node features according to individual bus connections.

In comparing the forward passes of the edge type-based heterogeneous GNN and the node type-based heterogeneous GNN, both models share a fundamental structure of iterating through layers and processing information based on distinct node or edge types. However, nuanced differences exist in their respective methodologies.

In summary, the edge type-based heterogeneous GNN primarily operates based on edge types, convoluting node features, and subsequently aggregating and processing them.

On the other hand, the node type-based heterogeneous GNN takes a node type-centric approach, collecting information from connected nodes for each node type, convoluting, aggregating, and processing features accordingly. The choice between these approaches may depend on the specific requirements of the heterogeneous graph data and the desired focus on edges or nodes in the learning process.

Node Type-Based Heterogeneous Model with Embedded Constraint Enforcement

Within the context of this study, a self-supervised approach is employed finally; where an enhanced version of the node type-based heterogeneous model is designed.

This enhancement aims to embed the constraint enforcement approach explained in the following, through a customized tangent hyberbolic function. The equation this customized constraint enforcement method implements is given as follows:

$$w_{x,i} = C_{x,i}^{max} - C_{x,i}^{min} \quad i = 1, ..., N \tag{3.17}$$

where $w_{x,i}$ represents the scaling coefficient for the forward pass of the activation function. C_i^{min} and C_i^{max} are the lower and upper bounds of constraints on the passed node feature.

The forward pass of the activation function is formulated as the following equation:

$$x'_i = 0.5 * w_{x,i} * tanh(x_i) + 0.5 * w_{x,i} \quad i = 1, ..., N \tag{3.18}$$

This activation ensures that the passed node feature is mapped to a value within the boundaries C_i^{min} and C_i^{max} passed to it, given as follows:

$$C_{x,i}^{min} \leq x'_i \leq C_{x,i}^{max} \quad i = 1, ..., N \tag{3.19}$$

thus ensuring the model output does not violate any constraint boundaries. Hence, the last model architecture introduced in this study is a node type-based heterogeneous model with embedded constraint enforcement; where the constraints are given as additional parameters to the forward pass; in contrast to the base architecture.

4 Implementation and Numerical Experiments

In the preceding chapter, the problem formulation was presented. The approach of this study towards problem formulation was methodically explained, providing readers with a comprehensive understanding of the research context.

The systematic steps involved in tackling the challenge were transparently shared, allowing readers to grasp the logical progression and rationale behind each decision made. Through this exposition, the chapter effectively communicated the research objectives and the structured methodology employed, fostering a deep understanding of the problem domain and the study's strategic approach; backed by the mathematical formulations of the ideas that build the foundation of this study.

In this chapter, the details on the implementations of the uttered approaches are highlighted to the reader, followed by the numerical experiments carried out.

4.1 Implementation

4.1.1 Network Processing

Pandapower is a Python-based, Berkeley Software Distribution (BSD)-licensed power system analysis tool that aims to automate static and quasi-static analysis and optimization of balanced power systems. It provides power flow, OPF, state estimation, topological graph searches, and short circuit calculations according to IEC 609091 [18].

This framework is based on electric elements such as lines, two and three-winding transformers, or ideal switches. All elements can be defined with nameplate parameters and are internally processed with equivalent circuit models, which have been validated against industry-standard software tools [18]. The tabular data structure used to define networks is based on the Python library pandas, which allows the handling of input and output parameters.

Pandapower includes a Newton-Raphson power flow solver formerly based on PYPOWER, which has been accelerated with just-in-time compilation [18]. Additional enhancements to the solver include the capability to model constant current loads, grids with multiple reference nodes, and a connectivity check. The implementation in Python makes Pandapower easy to use and allows comfortable extension with third-party libraries [18].

Throughout the entire study, Pandapower framework is used for data preparation and the processing of transmission grids. Additionally, Newton-Raphson numerical

solvers provided by PYPOWER are utilized for DCOPF and ACOPF results on the grid datasets, which are provided by SimBench.

SimBench is a publicly available benchmark dataset of electrical distribution grids that provides time series of power load, storage, and generation for the simulation of electrical distribution grids [37]. The dataset contains a grid level spanning model from high to low voltage, enriched by generation, load, and storage time series.

The project aims to develop a benchmark dataset for solutions in the field of analysis, planning, and operational management of distribution networks as well as transmission networks. The dataset is designed for a wide range of use cases, ensuring versatile applicability [37].

In context of homogeneous datasets, the following 4 different grids provided by SimBench have been utilized for network graphs:

- a 64 bus High Voltage (HV) grid with mixed generator types
- an 82 bus HV grid from an urban district
- a 103 bus Medium Voltage (MV) grid from a commercial district
- a 115 bus MV grid from a semi-urban district

According to the convergence of Newton-Raphson solvers of PYPOWER and the necessity for diversity in grid structure and nature for generalization; these 4 grids are selected for the homogeneous scenario in the study.

For each of these grids, the active and reactive power demands of loads are sampled uniformly with $\pm 10\%$ tolerance from the nominal values in the grid dataset. Following this step, the network graph is processed, DCOPF inputs and ACOPF targets are generated according to the algorithms explained in the section 3.2.1.

The homogeneous dataset generated from each grid is then split according to 85-10-5 ratio. Then, the training datasets, validation datasets and the test datasets from each grid are concatenated in random order; such that the final training datasets consist of 340, the validation datasets consist of 40 and lastly the test datasets are comprised of 20 inputs and targets. Thus, the homogeneous dataset is ready for employment.

Same as the homogeneous process, the generation of heterogeneous datasets starts with the sampling of nominal active and reactive power demands of the loads and proceeds with the creation of network graph based on the respective grid. Yet, due to the grid topology specific architecture of the heterogeneous models; only the 64 bus grid is utilized for all heterogeneous processing.

Heterogeneous dataset generation based on the given grid is then carried out according to the approaches and algorithms given in the section 3.3.2. In total, according to 80-15-5 split ratio; the training dataset consists of 240, the validation dataset consists of 45 and the test dataset is composed of 15 constraint-based inputs and ACOPF targets.

As the final step in both homogeneous and heterogeneous processing steps, input node features, constraints and targets are then standardized together to have 0 mean

and unit variance based on the distribution of targets; according to the Eq. 3.7, 3.8 and 3.9.

4.2 Numerical Experiments

After the conceptualization and implementation of shared approaches on data preparation and the models of the study, this chapter shares the results of numerical experiments done on the homogeneous and heterogeneous models in a supervised approach.

4.2.1 Homogeneous GNN

The numerical experiment carried out on the homogeneous GNN is trivial. After collecting the homogeneous datasets from all 4 grids given in 4.1.1, being a 64 bus HV grid, an 82 bus HV grid, a 103 bus MV grid, and a 115 bus MV grid; the stored DCOPF datasets are processed according to the Algorithm 2 of the preceding chapter. Followed by the extraction of training, validation and test datasets; these datasets are further used for the model training and evaluation.

In terms of the training and validation phase, Mean Average Error (MAE) and Root Mean Squared Error (RMSE) are chosen as the main metrics for evaluation; where the backpropagation is achieved through RMSE via supervised training. The objective function for RMSE is given by the following equation:

$$RMSE(Y_{ACOPF}, \hat{Y}_{GNN}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4.1}$$

where y_i represents the ACOPF result of the numerical solver for the ith node and $\hat{y}_i$ is the model output for the ith node, respectively. Just as has been throughout the study, N denotes the number of nodes of the respective grid. Similarly, the MAE calculation is formulated as follows:

$$MAE(Y_{ACOPF}, \hat{Y}_{GNN}) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.2}$$

RMSE's sensitivity to outliers is particularly advantageous in scenarios where larger errors carry more significant consequences and should be heavily penalized. This characteristic ensures that the GNN pays more attention to minimizing these larger errors, which are crucial for ACOPF approximation objective of the study.

Additionally, the smooth and continuous nature of RMSE, being differentiable everywhere, aligns well with the requirements of gradient-based optimization through ADAM optimizer. This differentiability facilitates a more efficient and stable training process, as

the gradients provide consistent directions for the GNN's updates. Moreover, RMSE's consistency with variance and standard deviation offers an intuitive interpretability, especially when relating the model's performance to other statistical measures.

The squaring of errors in RMSE, which disproportionately emphasizes larger errors, aligns with the objective; where such errors are particularly undesirable, as the cost of large errors is much higher than that of smaller errors. Finally, the loss landscape of RMSE is generally smoother compared to that of MAE.

This smoother landscape aids in a more stable and reliable optimization process, as it reduces the chances of erratic updates during training.

These factors collectively led to the choice of RMSE over MAE for backpropagation, coupled with a preference for a smooth optimization process. The decision also indicates a focus on achieving a balance between model interpretability and performance, especially considering that larger errors have a disproportionately higher impact for power flow optimization.

The main objective of this study, as emphasized, is the approximation of ACOPF results of numerical solvers provided by Pandapower.

In line with this objective, a third metric Mean Relative Error (MRE) is used for evaluation of the approximation of the targets in terms of the generalization capability of the homogeneous model. The MRE is interpreted as mean percentual deviation from the target ACOPF results of the solvers, providing a more realistic benchmark for the capability of the model(s).

The evaluation function w.r.t MRE is formulated as follows:

$$MRE(Y_{ACOPF}, \hat{Y}_{GNN}) = \frac{1}{N} \sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{y_i + \epsilon} \tag{4.3}$$

where ϵ represents a small value to avoid division by 0. The results are shared in table 4.1 with the most successful heterogeneous model of the study, providing comparison of both approaches w.r.t generalization ability.

Training and Validation

After evaluating the early training results w.r.t given metrics of the model for different parameter combinations, a homogeneous GNN with 256 hidden channels and 5 layers in total is selected for further experimenting. The homogeneous model was hence trained for 1000 epochs.

The plot of training losses for both MAE and RMSE are illustrated in the following figure 4.1:

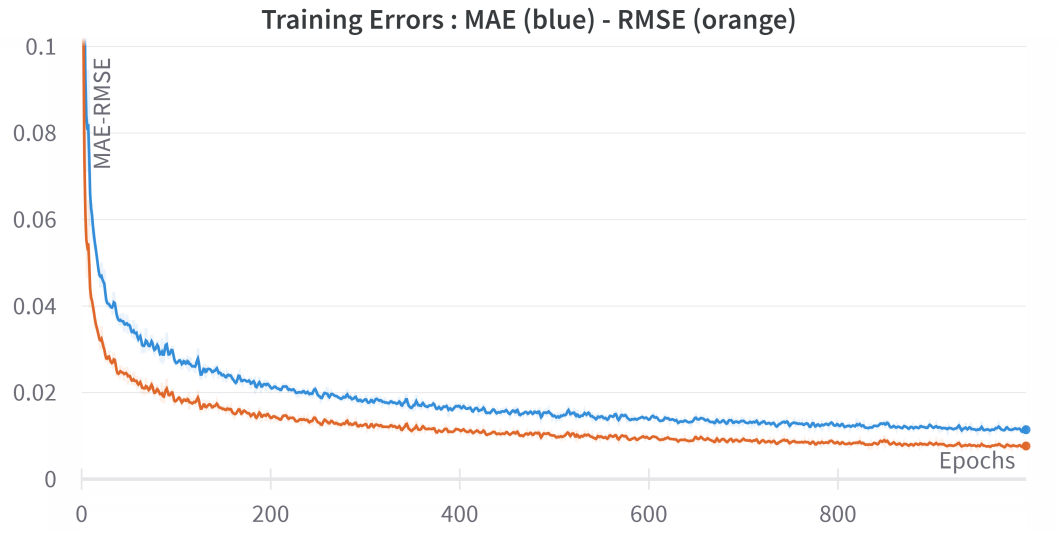

Figure 4.1: Training Errors: MAE (blue) - RMSE (orange)

Accordingly, the plot of validation losses for both metrics is given in the following figure 4.2:

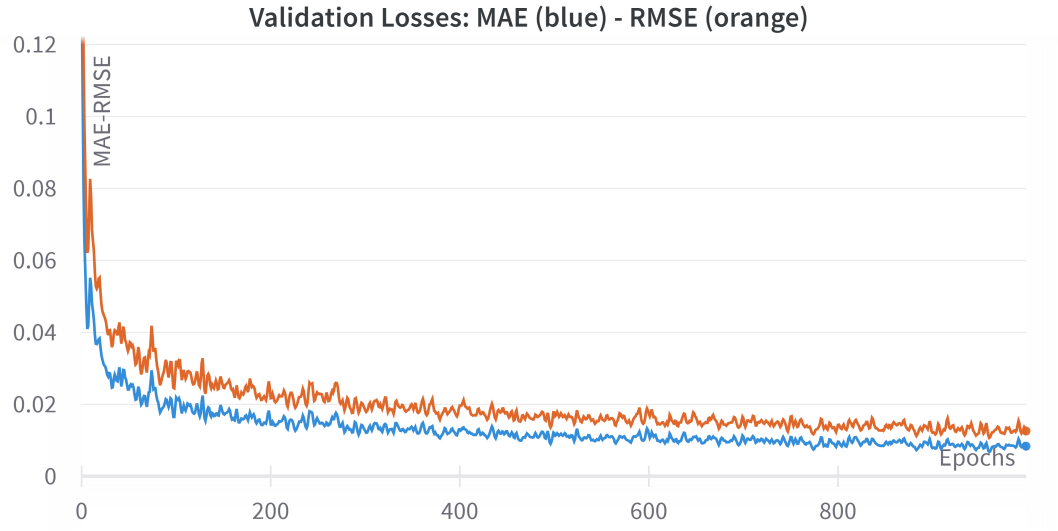

Figure 4.2: Validation Losses: MAE (blue) - RMSE (orange)

Upon examining the training and validation plots in figure 4.1 and 4.2 for the ho-

mogeneous model, a rapid decline is observed in both MAE and RMSE during the initial epochs. This indicates that the model is effectively learning from the training data. As the number of epochs increases, the rate of error reduction slows, suggesting the model is approaching convergence. Notably, the errors continue to decrease at a subtle rate beyond the 200-epoch mark, which indicates ongoing model improvement without stagnation.

The training and validation error trends exhibit a high degree of similarity, which implies that the model generalizes well to unseen data and is not overfitting. Overfitting is further dispelled by the observation that the validation errors do not increase as training progresses. The RMSE values remain consistently higher than the MAE values throughout both training and validation, which is an expected outcome, given that RMSE tends to amplify the impact of larger errors.

Overall, the model demonstrates solid performance with stable learning characteristics. The close alignment of training and validation errors throughout the training process is a positive sign of the model's capability to generalize. The final assessment of the model's performance would, however, require a consideration of the absolute error values at the end of training, the complexity of the data being modeled, and the specific requirements of the application at hand.

4.2.2 Heterogeneous GNNs

The training, validation and testing of heterogeneous models are executed similar to the homogeneous model. Yet, in contrast to the numerical experiment on the homogeneous model; it has been observed that the addition of the MAE for training and validation does not necessarily deliver any improvement on the interpretation of results w.r.t the performance of the model in this phase; thus this metric was disregarded further on for the evaluation of heterogeneous models. However, the objective function given in Eq. 4.1 for RMSE is further used explicitly for the training and validation of both edge type-based and node type-based heterogeneous models; whereas MRE formulated in Eq 4.3 is used for the testing of both models.

Edge Type-Based Heterogeneous Model

After evaluating the early training results w.r.t RMSE of the model for different parameter combinations, an edge-based heterogeneous GNN with 16 hidden channels and 1 layer in total is selected for further experimenting; as the increase in further complexity in terms of expressiveness of the model has not led to any improvement for the model performance. A numerical experiment of the edge type-based heterogeneous model is first conducted with only the node features; ignoring the collected constraint features for the model for 150 epochs; in order to see, if the model is able to tackle the challenge.

The plot of training of the edge type-based heterogeneous GNN is given in the figure 4.3:

Figure 4.3: Training Losses Without Constraint Features: RMSE

whereas, the plot of validation losses for this GNN is illustrated as follows:

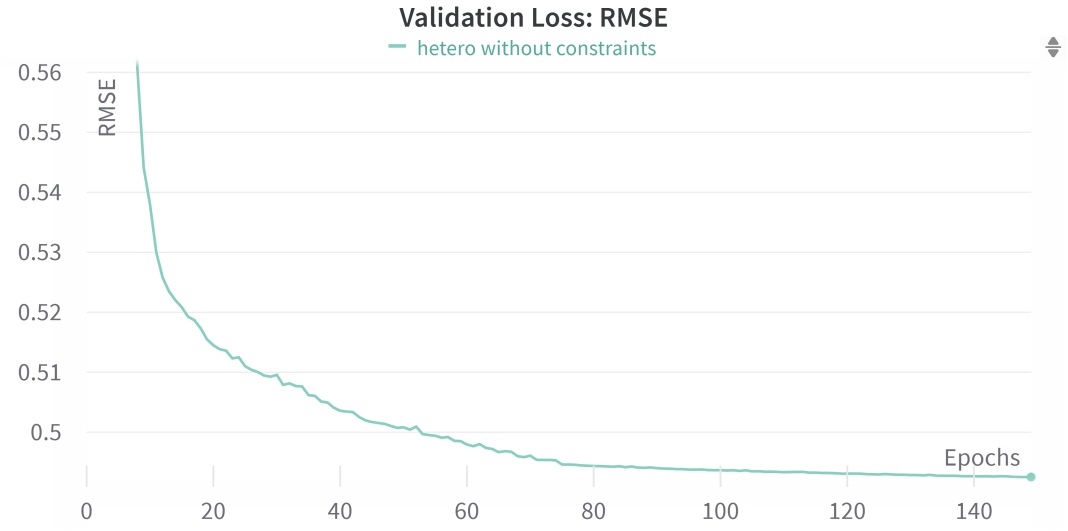

Figure 4.4: Validation Losses Without Constraint Features: RMSE

The plots of training and validation losses illustrated in figure 4.3 and 4.4 of the edge type-based heterogeneous GNN without constraints show a trend towards convergence.

However, the nature of this convergence raises questions about the optimality of the solution reached by the GNN.

Both training and validation loss curves exhibit a rapid decline initially, which is a common and expected behavior as the model begins to learn from the data. Yet, as the epochs progress, the rate of decrease in loss diminishes, with the curves approaching a plateau.

This plateau suggests that the GNN is not making significant learning gains after a certain point, which could imply that it has reached its capacity in terms of learning from the provided data with the current architecture and hyperparameters. The relatively high loss values at the plateau in both training and validation indicate that the GNN may have converged to a non-optimal solution.

Moreover, the absence of overfitting, as evidenced by the close mirroring of training and validation losses, while generally positive, also suggests that the GNN's ability to fit the training data is fundamentally limited.

This limitation could be due to several factors, such as an insufficiently expressive model architecture, lack of proper regularization, or suboptimal learning rate, which prevent the GNN from capturing the underlying patterns in the data effectively.

In summary, the GNN's current design, as evaluated by the provided loss curves, does not demonstrate a convincing performance; in terms of being able to tackle the main challenge of the study,

The convergence to a non-optimal solution with relatively high error rates indicates that the model, in its current form, is not a viable design for the task at hand. Further investigation and modification of the model's architecture, training procedure, or data representation may be required to improve its performance.

In addition to the utilization of the edge type-based heterogeneous GNN without constraints, an additional numerical experiment has been carried out; where the constraint features for each node are concatenated to the node features at the beginning of each forward pass. This further enhancement aims to observe, if additional information through the introduction of the constraint features would be able to improve model performance in terms of learning and generalization.

The incorporation of constraint features into the edge type-based heterogeneous GNN architecture represents a strategic modification aimed at augmenting the model's expression capabilities. By concatenating constraint features to the node features at the onset of each forward pass, the model is endowed with a richer representation of each node, potentially enabling it to capture more complex dependencies and patterns inherent in the data. This methodological adjustment is predicated on the hypothesis that supplementing node information with constraint-related attributes could bolster the GNN's ability to learn more effectively and generalize better to unseen data. The outcome of this numerical experiment is anticipated to shed light on the impact of integrating constraint features on the overall performance of the GNN model.

The edge type-based heterogeneous GNN is hence trained again for 150 epochs with exact settings including constraint features for further evaluation. The plot for training

losses of the model is given in the following figure 4.5:

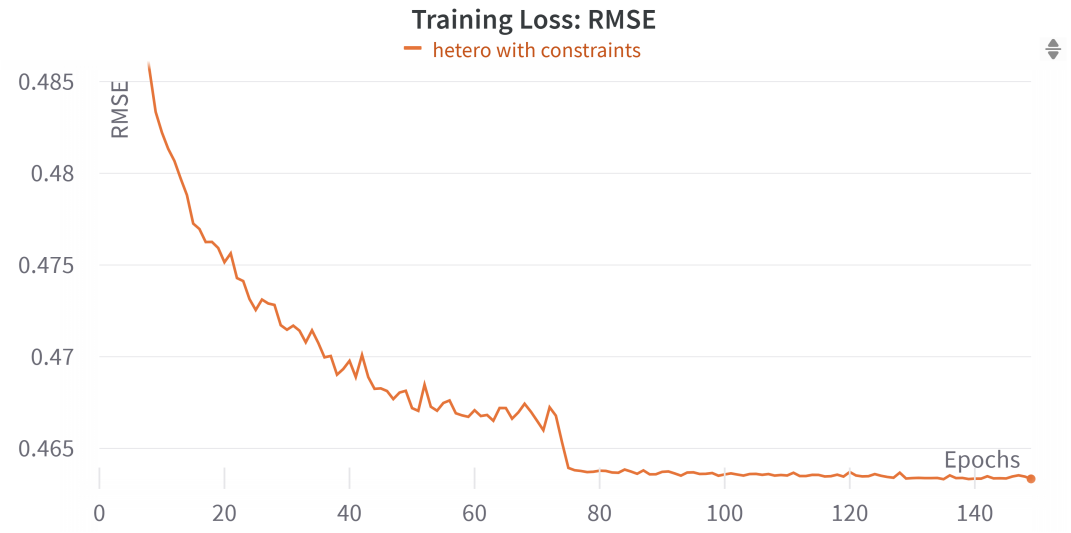

Figure 4.5: Training Losses With Constraint Features Edge-Based GNN: RMSE

whereas, the plot of validation losses for this GNN with provided constraint features is illustrated as follows:

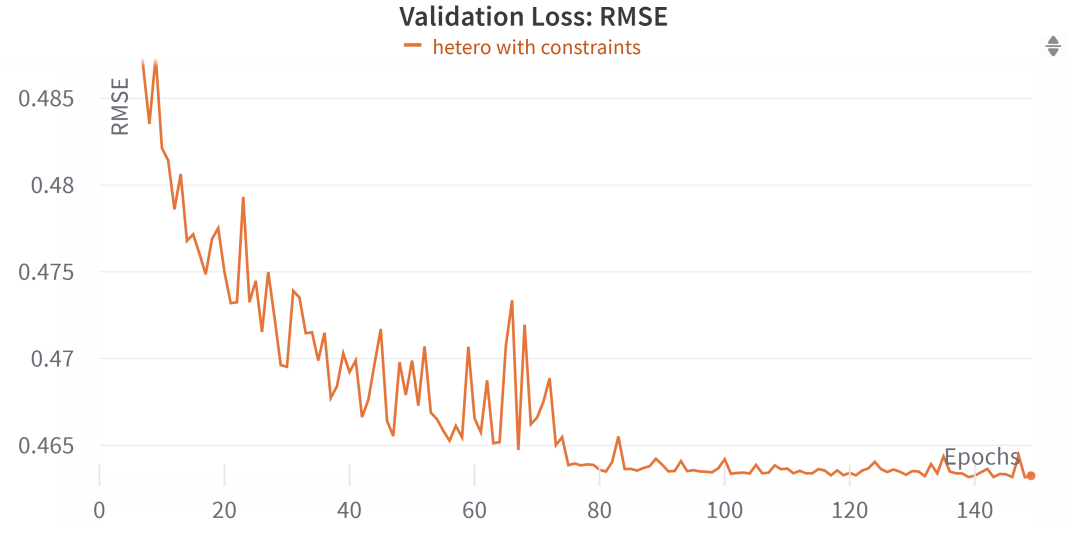

Figure 4.6: Validation Losses With Constraint Features Edge-Based GNN: RMSE

As observed in both plots of training losses in figure 4.5 and validation losses in figure 4.6, the introduction of constraint features has not led to any improvement in terms of the expressiveness and the capability of the model to tackle the challenge at hand. Thus, it can be stated that the edge type-based heterogeneous model in its current architecture fails to capture the intricacies of the complex objective of approximating the ACOPF results of numerical solvers.

Node Type-Based Heterogeneous Model

After evaluating the early training results w.r.t RMSE of the model for different parameter combinations, a node type-based heterogeneous GNN with 16 hidden channels and 1 layer in total is selected for further experimenting; as the increase in further complexity in terms of expressiveness of the model has not led to any improvement for the model performance.

The plot of the training losses after 150 epochs for the node type-based heterogeneous model is illustrated in the figure 4.7:

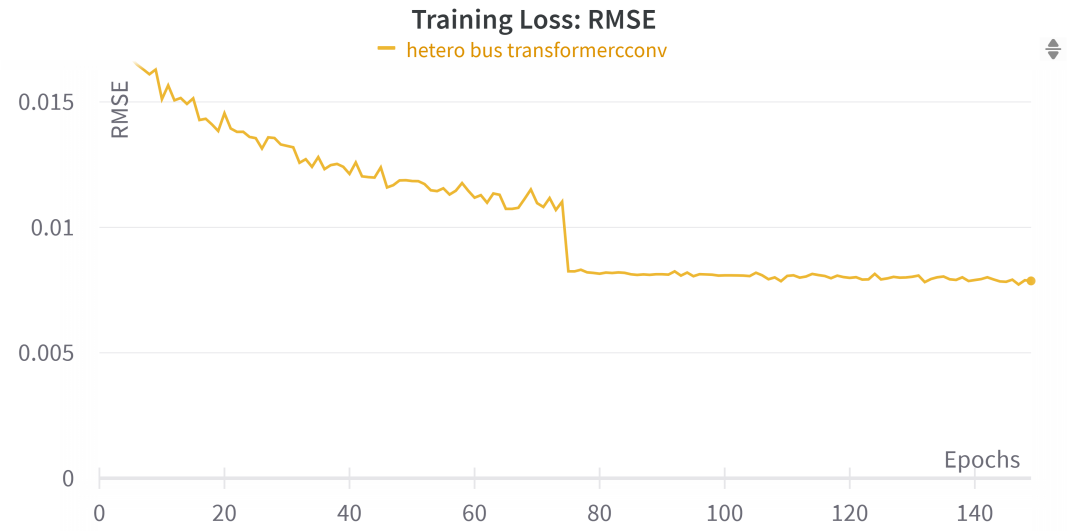

Figure 4.7: Training Losses Node Type-Based GNN: RMSE

whereas the plot of validation losses for this GNN with provided constraint features is illustrated as follows:

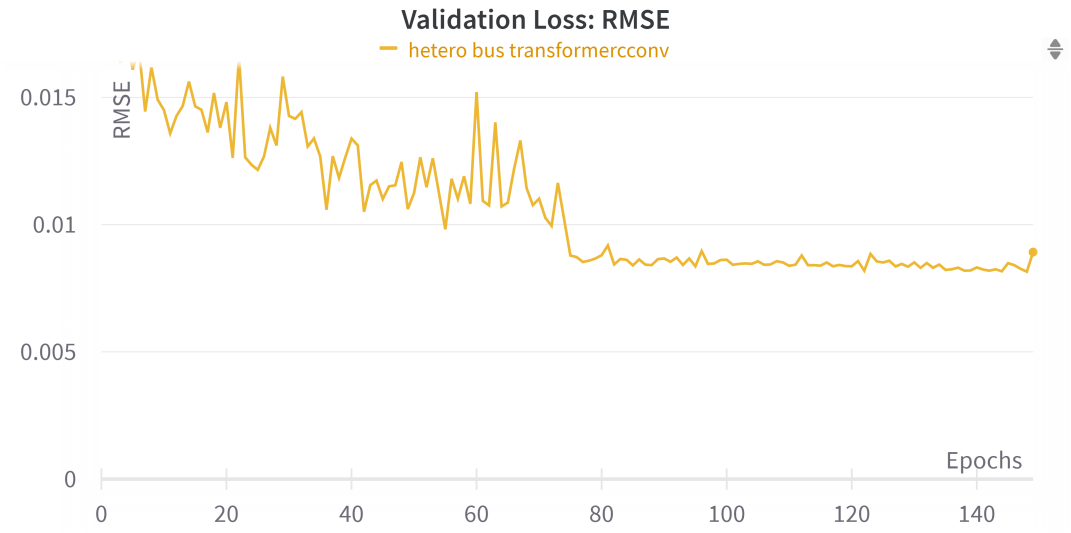

Figure 4.8: Validation Losses Node Type-Based GNN: RMSE

The training and validation loss plots illustrated in figure 4.7 and 4.8 for the node type-based heterogeneous GNN reflect a model that has effectively converged to an optimal solution. Throughout the training process, the RMSE steadily decreases, indicating that the model is learning and improving its predictions of the underlying patterns within the data. This consistent reduction in RMSE, without significant volatility, suggests that the model is capturing the complexities of the task effectively.

Moreover, the validation loss closely tracks the training loss and settles at a low value, implying that the GNN generalizes well to unseen data. This is a critical aspect of model performance, as it demonstrates the model's ability to not only fit the training data, but also to maintain accuracy on data it has not encountered before.

The stability of the loss values in the later epochs, along with their low magnitude, supports the conclusion that the node type-based heterogeneous GNN is a viable design for approximating the results of ACOPF numerical solvers.

The model's ability to converge to an optimal solution is indicative of its potential as a reliable tool in the study's objective, reinforcing its value in the approximation of complex power flow calculations. The convergence of the GNN, as depicted in the loss plots, highlights its success in meeting the main goal of the study, proving its efficacy in tackling the intricacies of ACOPF result approximation.

Thus, the node type-based heterogeneous GNN, in its current architecture mimicking the connections of the individual busses in the network; proves itself to be worth for further training, evaluation and comparison with regard to the main objective of the study.

For all models until this step of the study, it is worth noting that the training and vali-

dation phases have been accommodated by an adaptive manual learning rate decrease; based on the course of validation losses. This approach explains the sudden decrease in both training and validation losses of the node-based heterogeneous model at the 75th epoch.

Node Type-Based Heterogeneous Model with Embedded Constraint Enforcement

After evaluating the early training results w.r.t RMSE of the model for different parameter combinations, a node type-based heterogeneous GNN with embedded constraint enforcement with 16 hidden channels and 1 layer in total is selected for further experimenting; as the increase in further complexity in terms of expressiveness of the model has not led to any improvement for the model performance.

The plot of the training losses after 150 epochs for the node type-based heterogeneous model with embedded constraint enforcement is illustrated in the figure 4.9:

Figure 4.9: Training Losses Node Type-Based GNN with Embedded Constraint Enforcement: RMSE

whereas the plot of validation losses for this GNN is illustrated as follows:

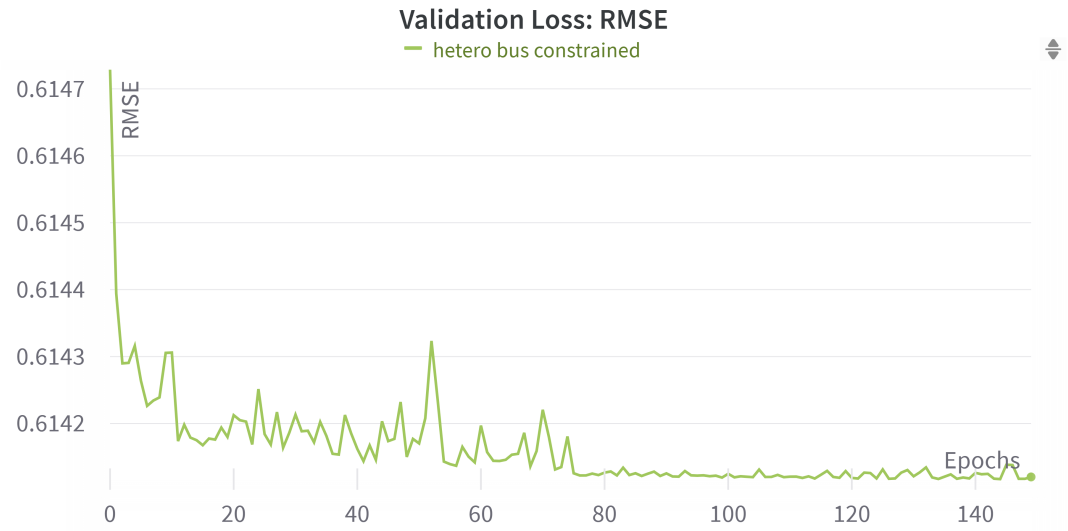

Figure 4.10: Validation Losses Node Type-Based GNN with Embedded Constraint Enforcement: RMSE

Similar to the edge type-based heterogeneous GNN, the training and validation loss plots illustrated in figure 4.9 and 4.10 for the node type-based heterogeneous GNN with embedded constraint enforcement reveal a pattern of convergence that is indicative of a non-optimal solution. The losses, represented by RMSE, display a decline that plateaus prematurely, suggesting that the GNN's learning has stagnated. This levelling off of the loss, occurring early in the training process and persisting through subsequent epochs, points towards a saturation effect in the network's learning capacity.

This saturation effect can be attributed to the use of a customized tanh activation function for embedded constraint enforcement. While such functions are known for their efficacy in certain contexts, they are also prone to saturating, especially when the output values are driven to the limits of the tanh function's range. In this saturation regime, the gradients can become vanishingly small, severely hampering the network's ability to continue learning and refining its parameters. This phenomenon, known as the vanishing gradient problem, is likely impeding further reduction of the RMSE, thereby preventing the GNN from achieving a more optimal fit to the data.

As a result, despite the introduction of embedded constraints aimed at enhancing the network's performance w.r.t avoidance of constraint violations, the choice of activation function seems to have introduced an unintended limitation. The GNN's inability to significantly improve beyond a certain point suggests that it has not succeeded in capturing the full complexity of the ACOPF results it aims to approximate. Consequently, the current implementation of the GNN, with the customized tanh-based constraint enforcement and self-supervised training approach; has not proven to be a viable design

for the intended application. Further investigation into alternative activation functions or additional architectural adjustments may be necessary to overcome this limitation and achieve a solution that meets the study's objectives.

4.3 Results

After the numerical experiments on all models have been carried out, the node type-based heterogeneous GNN was selected for further training and evaluation. Accordingly, this heterogeneous GNN was trained for 1000 epochs equivalent to the homogeneous model.

In this section, a comparative analysis of the test performance between two GNN architectures, namely the homogeneous GNN and the heterogeneous GNN; is presented. The homogeneous GNN, trained on a diverse dataset comprising network graphs with unique load profiles derived from four different grid topologies, demonstrated superior performance. It achieved an RMSE of 0.0025, an MAE of 0.0016, and an MRE of 0.0324.

These results suggest that the homogeneous GNN has effectively captured the variability inherent in the multiple grid topologies, as evidenced by its low prediction errors and its ability to generalize across different network configurations.

In contrast, the heterogeneous GNN was trained exclusively on network graphs stemming from a single grid topology, yet its test performance was found wanting. It reported an RMSE of 0.0064, an MAE of 0.002, and an MRE of 0.047. Despite being exposed solely to a single grid topology during training, the model did not exhibit a marked advantage in predictive accuracy on this topology. This raises concerns regarding its capability to generalize, even within the grid topology it was trained on.

Furthermore, the inferior performance metrics relative to the homogeneous model suggest that the heterogeneous GNN might have overfitted to the nuances of the single grid topology. Overfitting is typically characterized by a model's inability to maintain performance on new, unseen datasets, and in this instance, it implies that the model may not adapt well to variations beyond the training scope.

The performance of both models is summarized in the table below:

Table 4.1: Test Results of GNN Models

Model	RMSE	MAE	MRE
Homogeneous GNN	0.0025	0.0016	0.0324
Node Type-Based Heterogeneous GNN	0.0064	0.002	0.047

In summary, the homogeneous GNN has shown robustness and adaptability across various grid topologies, affirming its efficacy in generalizing from a heterogeneous training regime.

On the other hand, the heterogeneous GNN's performance raises questions about its practicality and reliability, as it fails to convincingly outperform on the grid topology it was tailored to. These findings underscore the importance of diversity in training

datasets to foster generalization and discourage overfitting, thereby enhancing the model's utility in real-world applications.

5 Conclusion

This research has delved into the exploration of the application of GNNs in optimizing power flow within complex power distribution networks.

5.1 Key Findings and Contributions

Through detailed experimentation and analysis, it has been demonstrated that GNNs, particularly the homogeneous model, can effectively approximate ACOPF results of Newton-Raphson solvers within a certain tolerance interval on network topologies with unseen load profiles.

The study investigated both homogeneous and heterogeneous GNN models. The homogeneous GNN, trained on diverse datasets from multiple grid topologies, showed robust performance, evidencing its ability to generalize across different network configurations. This adaptability is crucial for real-world applications where grid structures vary widely.

In contrast, the heterogeneous GNN, focused on a single grid topology, failed to exceed the performance of the homogeneous model. It highlighted the challenges in training models that are overly specialized and the importance of exposure to diverse data for generalization.

In addition to that, the node type-based heterogeneous model; which outperformed all remaining heterogeneous model architectures, has also been unsuccessful, relative to the homogeneous model; in being wired up on a single grid topology. Unable to overfit on the same grid topology with unseen load profiles, the model underachieved in terms of expressiveness and learning capability relative to the homogeneous GNN.

Yet, both models, employed utilizing different approaches, managed to approximate ACOPF results from grid topologies with unseen load profiles within an average of 5% deviation from the targets. Although these results indicate promising performance; the necessity for further improvement and evaluation regarding violations of network limits remains.

The research underscores the potential of GNNs in power system analysis, particularly in scenarios involving complex and dynamic grid structures. The ability of these models to learn from and adapt to varying grid configurations can be instrumental in enhancing the efficiency and reliability of power distribution, especially considering increasing energy costs and sustainability targets on a global scale. However, the study also brings to light the limitations of current GNN architectures, especially when dealing with highly specific or constrained networks.

In conclusion, while this thesis has successfully demonstrated the potential of GNNs in power system analysis, it also highlights the need for continuous improvement in model development and training methodologies. The insights gained from this research contribute significantly to the field of power systems engineering and pave the way for more resilient, efficient, and intelligent power distribution networks.

5.2 Future Research and Outlook

The findings from this study open several avenues for future research, which can be categorized as follows:

Model Enhancement

The findings of this thesis suggest several specific areas for model enhancement in the application of GNNs to power systems:

Improved Learning Algorithms: Current GNN models could benefit from advanced learning algorithms that can more effectively capture the non-linearities and complex dependencies inherent in power system networks. Advancing and customizing attention mechanisms or incorporating time-series analysis can provide deeper insights; taking the transient behaviors into account. These additions can lead to an improvement for the GNNs to capture highly-complex relations within power systems.

Utilizing Hybrid Approaches: Combining GNNs with other machine learning approaches, such as reinforcement learning or unsupervised learning models, could yield more robust and adaptable systems. Such hybrid models could leverage the strengths of each approach, improving predictive accuracy and generalization across different network conditions.

Scalability and Efficiency: Enhancing the scalability and computational efficiency of GNNs is crucial for their application in real-time power system analysis. Research should focus on developing lightweight models that maintain high accuracy while reducing computational overhead, making them viable for real-world, large-scale power grid applications.

Incorporation of Physical Laws: Embedding the physical laws governing power systems directly into the GNN architecture could improve the model's accuracy and reliability. This approach ensures that the model's outputs are not just data-driven but also conform to fundamental power system principles.

Uncertainty Quantification: Incorporating mechanisms for uncertainty quantification in GNN predictions can provide a more comprehensive understanding of the model's

confidence in its outputs, w.r.t bus and line violations. This is particularly important in power system applications where decisions based on model predictions can have significant implications.

Exploring More Complex Architectures: Investigating new GNN architectures, including those with embedded constraint enforcement, could provide breakthroughs in accurately modeling complex power systems.

Broader Data Exposure and Improved Processing

Expanding the diversity, range and quality of data used to train GNNs in power systems is crucial for enhancing their effectiveness and reliability. The following are key areas for improving data exposure:

Variety of Grid Topologies: Training GNNs on a wide range of grid topologies, including different scales (from small microgrids to large national grids), configurations (radial, meshed, hybrid), and types (urban, rural, industrial) will significantly improve the model's ability to generalize. Each topology presents unique characteristics and challenges, and exposure to this variety ensures that the model can adapt to and accurately represent different real-world scenarios. Although SimBench already provides a broad spectrum of grid topologies, the inability to process and use them for the task at hand leads to the study not being able to utilize this variety; which is shared in more detail in the following.

Improved Numerical Solvers: For the major portion of SimBench grids, the numerical solvers provided by Pandapower did not converge for any load profile, extremely shrinking down the variety of grid topologies, which could be utilized to broaden data versatility for all models. This being said, more precise and able solvers can significantly enhance the performance of all models; including precision and generalization.

Detailed Evaluation of Model Outputs: Pandapower framework does not provide any feature, which can be used to gather voltage magnitudes and voltage angles based on set generator points. Thus, other than the computed constraints and metrics like MRE w.r.t ACOPF targets; it is not possible to evaluate the model outputs in terms of bus and line violations. A feature enabling precise violation checks would provide a more detailed evaluation of model performance and thus lead to better and stable results.

This study has unveiled numerous promising directions for future research in the application of GNNs to power systems. In summary, the findings of this study provide a comprehensive roadmap for advancing the application of GNNs in power systems, encompassing model enhancement and the broadening of data exposure and processing. These avenues for future research hold the potential to transform and elevate the effectiveness and reliability of GNN-based power system analysis.

Bibliography

[1] L. Böttcher, H. Wolf, B. Jung, P. Lutat, M. Trageser, O. Pohl, X. Tao, A. Ulbig, and M. Grohe. "Solving AC Power Flow with Graph Neural Networks under Realistic Constraints." In: *arXiv preprint arXiv:2204.07000* (2022).

[2] C. Kim, K. Kim, P. Balaprakash, and M. Anitescu. "Graph Convolutional Neural Networks for Optimal Load Shedding under Line Contingency." In: *2019 IEEE Power Energy Society General Meeting (PESGM)*. 2019, pp. 1–5.

[3] M. Bechler-Speicher, I. Amos, R. Gilad-Bachrach, and A. Globerson. *Graph Neural Networks Use Graphs When They Shouldn't*. 2023. arXiv: 2309.04332 [cs.LG].

[4] M. Tuo, X. Li, and T. Zhao. "Graph Neural Network-based Power Flow Model." In: *arXiv preprint arXiv:2307.02049* (2023).

[5] F. M. B. Jonas Berg Hansen Stian Normann Anfinsen. "Power Flow Balancing With Decentralized Graph Neural Networks." In: *IEEE Transactions on Power Systems* (2022).

[6] D. Owerko, F. Gama, and A. Ribeiro. *Unsupervised Optimal Power Flow Using Graph Neural Networks*. 2022. arXiv: 2210.09277 [eess.SY].

[7] T. B. Lopez-Garcia and J. A. Domínguez-Navarro. "Power flow analysis via typed graph neural networks." In: *Engineering Applications of Artificial Intelligence* 117 (2023), p. 105567. ISSN: 0952-1976.

[8] T. Pham and X. Li. "Reduced Optimal Power Flow Using Graph Neural Network." In: *arXiv preprint arXiv:2206.13591* (2022).

[9] F. Calero et al. "A Review of Modeling and Applications of Energy Storage Systems in Power Grids." In: *Proceedings of the IEEE* 111.7 (2023), pp. 806–831.

[10] V. A. Lacerda, J. Girona-Badia, E. Prieto-Araujo, and O. Gomis-Bellmunt. "Modelling Approaches of Power Systems Considering Grid-Connected Converters and Renewable Generation Dynamics." In: *arXiv preprint arXiv:2112.00867* (2021).

[11] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang. "A Review of Graph Neural Networks and Their Applications in Power Systems." In: *Journal of Modern Power Systems and Clean Energy* 10.2 (2022), pp. 345–360.

[12] B. Donon, B. Donnot, I. Guyon, and A. Marot. "Graph Neural Solver for Power Systems." In: *2019 International Joint Conference on Neural Networks (IJCNN)*. 2019, pp. 1–8.

[13] R. Solheim, B. A. Høverstad, and M. Korp as. "Using Graph Neural Networks in Reinforcement Learning with application to Monte Carlo simulations in Power System Reliability Analysis." In: *arXiv preprint arXiv:2307.02049* (2023).

[14] B. Eldridge, R. P. O'Neill, A. Castillo, and A. E. Castillo. "An Improved Method for the DCOPF With Losses." In: *IEEE Transactions on Power Systems* 33.4 (2017), pp. 3779–3788.

[15] Y. Fu and M. Shahidehpour. "DCOPF-Based LMP Simulation: Algorithm, Comparison With ACOPF, and Sensitivity." In: *IEEE Transactions on Power Systems* 22.4 (2007), pp. 1475–1485.

[16] L. Zhang and B. Zhang. *An Iterative Approach to Finding Global Solutions of AC Optimal Power Flow Problems*. 2021. arXiv: `2102.12075 [eess.SY]`.

[17] H. D. Nguyen, K. Dvijotham, and K. Turitsyn. "Constructing Convex Inner Approximations of Steady-State Security Regions." In: *IEEE Transactions on Power Systems* 34.1 (2019), pp. 257–267.

[18] L. Thurner, A. Scheidler, F. Schafer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, and M. Braun. "Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems." In: *IEEE Transactions on Power Systems* 33.6 (2018), pp. 6510–6521.

[19] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. "Computational Capabilities of Graph Neural Networks." In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 81–102.

[20] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. "Geometric Deep Learning: Going beyond Euclidean data." In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42.

[21] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. "A Comprehensive Survey on Graph Neural Networks." In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2021), pp. 4–24.

[22] A. Javaloy, P. Sanchez-Martin, A. Levi, and I. Valera. *Learnable Graph Convolutional Attention Networks*. 2023. arXiv: `2211.11853 [cs.LG]`.

[23] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. "Graph Neural Networks: A Review of Methods and Applications." In: *arXiv preprint arXiv:1812.08434* (2018).

[24] Z. Liu, S. Luo, W. Li, J. Lu, Y. Wu, S. Sun, C. Li, and L. Yang. *ConvTransformer: A Convolutional Transformer Network for Video Frame Synthesis*. 2021. arXiv: `2011.10185 [cs.CV]`.

[25] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini. *Recipe for a General, Powerful, Scalable Graph Transformer*. 2023. arXiv: `2205.12454 [cs.LG]`.

[26] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun. *Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification*. 2021. arXiv: 2009.03509 [cs.LG].

[27] L. Sun, Y. Dou, C. Yang, K. Zhang, J. Wang, P. S. Yu, L. He, and B. Li. "Adversarial Attack and Defense on Graph Data: A Survey." In: *IEEE Transactions on Knowledge and Data Engineering* 35.8 (2023), pp. 7693–7711.

[28] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. *Neural Message Passing for Quantum Chemistry*. 2017. arXiv: 1704.01212 [cs.LG].

[29] X. Wang, R. Girshick, A. Gupta, and K. He. *Non-local Neural Networks*. 2018. arXiv: 1711.07971 [cs.CV].

[30] L. Waikhom and R. Patgiri. "Graph Neural Networks: Methods, Applications, and Opportunities." In: *arXiv preprint arXiv:2108.10733* (2021).

[31] X. Yang, M. Yan, S. Pan, X. Ye, and D. Fan. *Simple and Efficient Heterogeneous Graph Neural Network*. 2023. arXiv: 2207.02547 [cs.LG].

[32] S. Yun, M. Jeong, S. Yoo, S. Lee, S. S. Yi, R. Kim, J. Kang, and H. J. Kim. "Graph Transformer Networks: Learning meta-path graphs to improve GNNs." In: *Neural Networks* 153 (2022), pp. 104–119. ISSN: 0893-6080.

[33] S. Gopinath and H. L. Hijazi. *Benchmarking Large-Scale ACOPF Solutions and Optimality Bounds*. 2022. arXiv: 2203.11328 [math.OC].

[34] L. Zhang and B. Zhang. *Learning to Solve the AC Optimal Power Flow via a Lagrangian Approach*. 2021. arXiv: 2110.01653 [eess.SY].

[35] D. Wang, K. Zheng, Q. Chen, G. Luo, and X. Zhang. "Probabilistic Power Flow Solution with Graph Convolutional Network." In: *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. 2020, pp. 650–654.

[36] D. Owerko, F. Gama, and A. Ribeiro. *Optimal Power Flow Using Graph Neural Networks*. 2019. arXiv: 1910.09658 [eess.SY].

[37] C. Spalthoff, D. Sarajlic, C. Kittl, S. Drauz, T. Kneiske, C. Rehtanz, and M. Braun. "SimBench: Open source time series of power load, storage and generation for the simulation of electrical distribution grids." In: *International ETG-Congress 2019; ETG Symposium*. 2019, pp. 1–6.