Tania Beatriz López García

Power Flow and Optimal Power Flow via Physics-Informed Typed Graph Neural Networks

Director/es

Domínguez Navarro, José Antonio

ISSN 2254-7606

ISSN 2254-7606

Tesis Doctoral

POWER FLOW AND OPTIMAL POWER FLOW VIA PHYSICS-INFORMED TYPED GRAPH NEURAL NETWORKS

Autor

Tania Beatriz López García

Director/es

Domínguez Navarro, José Antonio

UNIVERSIDAD DE ZARAGOZA
Escuela de Doctorado

Programa de Doctorado en Energías Renovables y Eficiencia Energética

2023

Universidad Zaragoza

1542

PhD Program in Renewable Energies and Energy Efficiency
UNIVERSITY OF ZARAGOZA

Power Flow and Optimal Power Flow via Physics-Informed Typed Graph Neural Networks

PhD Thesis submitted by

Tania Beatriz LOPEZ GARCIA

to obtain the degree of

Doctor of Philosophy

Supervisor
Dr. José A. DOMÍNGUEZ-NAVARRO

February 1, 2023

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mi director de tesis, José Antonio Domínguez Navarro, por su apoyo y respeto a mis sugerencias e ideas, y por su guía que ha facilitado las mismas. Su experiencia y orientación han sido de gran valor para mí y estoy muy agradecida por el tiempo y esfuerzo que ha dedicado a mi investigación. Muchas gracias, José Antonio.

También quiero agradecer a todos los que se han mantenido a mi lado durante estos años, especialmente a mi madre Teresa, a mi abuela Beatriz, a mi padre Isaías, a mis hermanos Isaías Jr. y Nathalia, y a César; sin su apoyo incondicional no podría haber completado esta tarea. Igualmente agradezco el amor infinito de mi hijo, Cédric; sin él los estudios doctorales posiblemente habrían sido expeditos pero mucho menos satisfactorios. Gracias a Fabian por su valiosa ayuda y aportaciones.

Por último, pero no menos importante, quiero agradecer a mis compañeros de laboratorio que me han acompañado y apoyado durante todo este tiempo: Yassine, Natalia, Víctor, Nacho y David; gracias por hacer este tiempo más llevadero.

Muchas gracias a todos.

Resumen

Esta tesis explora las redes neuronales de grafos tipados informadas por la física aplicadas al modelado de redes de transporte de energía eléctrica, concretamente a los problemas de flujo de potencia y flujo de potencia óptimo. Las redes de transporte de energía eléctrica son complejos sistemas interconectados, cruciales para garantizar un suministro estable de electricidad. Para lograr la transición hacia sistemas energéticos asequibles, fiables y sostenibles, la electrificación de diversos sectores económicos y la integración de fuentes de energía renovable en la red de transmisión han aumentado significativamente. La mayoría de las tecnologías para la generación de energía renovable añaden fluctuación e incertidumbre a la generación de electricidad, por lo que, para garantizar un suministro eléctrico eficiente y estable en todo momento, los operadores de la red de transporte deben realizar frecuentes simulaciones de flujo de potencia y de flujo de potencia óptimo para evaluar el estado de la red. Por estas razones es necesario investigar técnicas nuevas, flexibles y más eficientes para resolver estos análisis.

Los recientes avances en el aprendizaje automático, y en particular en las redes neuronales artificiales, indican que estos métodos tienen potencial para resolver problemas de análisis de redes eléctricas de forma rápida y fiable. Hasta la fecha, pocos trabajos han intentado aprovechar las capacidades de aprendizaje de las redes neuronales artificiales para abordar estos temas. Sin embargo, la mayoría de los trabajos publicados no resuelven dos grandes retos: en primer lugar, la necesidad de grandes cantidades de datos de entrenamiento y, en segundo lugar, la falta de capacidad de generalización para analizar redes de transporte realistas con topología variable. En esta tesis, se superan estos inconvenientes introduciendo redes neuronales de grafos tipados, que están especializados para procesar datos estructurados en forma de grafos con distintos tipos de elementos.

La red de transmisión puede representarse directamente como un grafo, y al asignar distintos tipos de nodos para representar los diferentes elementos de la red de transmisión, se incrementa la precisión y la interpretabilidad del modelo propuesto. El modelo resultante es un modelo de red de transporte adaptable que puede aplicarse a diversos problemas, como las aplicaciones de flujo de potencia y flujo de potencia óptimo que se presentan en esta tesis. El esquema de aprendizaje presentado está informado por la física, de forma que el entrenamiento no está supervisado, sino que incorpora información de las leyes físicas del sistema subyacente en la función de costo. Además, el modelo resultante puede probarse en redes eléctricas con diferentes configuraciones y, en el caso del flujo de potencia, con redes de diferentes tamaños. Se demuestra que el método propuesto, con las aplicaciones consideradas, consigue resultados similares a los obtenidos con un método convencional pero hasta cuatro órdenes de magnitud más rápido, sin necesidad de datos de entrenamiento y con capacidad de generalización a diferentes redes de transporte. Se puede concluir, por tanto, que el trabajo presentado en esta tesis ofrece un método basado en redes neuronales para agilizar la resolución del complejo sistema de ecuaciones no lineales presente en el problema de flujo de potencia, así como el problema de optimización con restricciones presente en el problema de flujo de potencia óptimo. Estos resultados proporcionan un valioso paso hacia el desarrollo de un sistema general para ayudar a los operadores de sistemas de transmisión a optimizar la integración de nuevas tecnologías en la red convencional, y mejorar la fiabilidad y sostenibilidad de los sistemas eléctricos.

Abstract

This thesis explores physics-informed typed graph neural networks applied to the modeling of electricity transmission grids, specifically to the power flow and optimal power flow problems. Transmission grids are complex interconnected systems crucial in ensuring a stable supply of electricity. To achieve the transition towards affordable, reliable and sustainable energy systems, the electrification of varied economic sectors and the integration of large shares of renewable energy sources in the utility grid have both seen a significant increase. Most renewable energy generation technologies add fluctuation and uncertainty to electricity generation, thus to ensure an efficient and stable power supply at all times, transmission grid operators must perform frequent power flow and optimal power flow simulations to evaluate the state of the grid. This calls for new, flexible and more efficient techniques for solving these different analyses.

Recent advances in machine learning, in particular artificial neural networks, indicate that these methods have the potential to solve operational analysis problems in a fast, reliable way. Until today few works have attempted to take advantage of the learning abilities of artificial neural networks to tackle these issues. However, most of the published work fails to solve two big challenges: first is the need for large amounts of training data and second is the lack of generalization ability to analyze realistic transmission grids with varying topology. In this thesis, we overcome such drawbacks by introducing typed graph neural networks, which are specially useful for processing graph-structured data with different element types.

The transmission grid can be directly represented as a graph, and by assigning different node types to represent the different elements of the transmission grid, the accuracy and interpretability of the proposed model is increased. The resulting model is a customizable transmission grid model that can be applied to various problems, such as the power flow and optimal power flow applications which are presented in this thesis. The presented learning scheme is physics-informed, such that the training is unsupervised, instead incorporating information of the underlying physical laws in the loss function. Additionally, the resulting model can be tested on grids with different configurations, and in the power flow case, with grids of different sizes. It is shown that the proposed method, with the considered applications, achieves results similar to those obtained with a conventional solver but up to four orders of magnitude faster, without the need of training data and with the ability to generalize to different transmission grids. It can be thus concluded that the work presented in this thesis offers a neural network based method to expedite solving the complex, nonlinear equation system present in the power flow problem, as well as the constrained optimization problem present in the optimal power flow problem. These results provide a valuable step towards developing a general system for aiding transmission system operators in optimizing the integration of new technologies into the conventional grid, and improving the reliability and sustainability of power systems.

Contents

List of Figures

List of Tables

List of Abbreviations

AC	**A**lternating **C**urrent
AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
DC	**D**irect **C**urrent
FFR	**F**ast **F**requency **R**esponse
GN	**G**raph **N**etwork
GNN	**G**raph **N**eural **N**etwork
GPU	**G**raphics **P**rocessing **U**nit
IPOPT	**I**nterior **P**oint **OPT**imizer
KKT	**K**arush-**K**uhn-**T**ucker
MC	**M**onte **C**arlo
ML	**M**achine **L**earning
MLP	**M**ulti**L**ayer **P**erceptron
MVAr	**M**ega**V**olt-**A**mpere **r**eactive
MW	**M**ega **W**att
NN	Artificial **N**eural **N**etwork
N-R	**N**ewton-**R**aphson
RE	**R**enewable **E**nergy
OPF	**O**ptimal **P**ower **F**low
PF	**P**ower **F**low
PI	**P**hysics-**I**nformed
p.u.	**P**er **U**nit
SCADA	**S**upervisory **C**ontrol **A**nd **D**ata **A**cquisition
SDG	**S**ustainable **D**evelopment **G**oals
SE	**S**tate **E**stimation
TGN	**T**yped **G**raph **N**etwork
V	**V**olt
VD	**V**iolation **D**egree

List of Symbols

Symbols regarding neural networks

$\mathbb{A}$	adjacency matrix
$\mathbf{b}$	bias of NN
d	embedding dimension
f	input dimension
$\mathcal{G}$	graph structure
H	number of samples in the training/testing batch
h	index of number of samples
L	number of TGN layers
l	index of number of TGN layers
$\mathcal{N}(\cdot)$	neighboring (adjacent) element
N_B	total number of bus type nodes (generator + load)
N_D	total number of load type nodes
N_E	total number of branch type nodes
N_G	total number of generator type nodes
N_K	total number of PV type nodes
o	output dimension
T	number of message-passing steps
t	index of number of message-passing steps
$\mathcal{V}_B$	set of all bus type nodes
$\mathcal{V}_D$	set of load (PQ) type nodes
$\mathcal{V}_E$	set of branch type nodes
$\mathcal{V}_G$	set of generator (PV + slack) type nodes
$\mathcal{V}_K$	set of PV type nodes
$\mathcal{V}_S$	set of slack type nodes
$\mathbf{W}$	kernel of NN
w	all trainable parameters of NN
$\mathbf{x}$	input feature matrix
$\mathbf{y}$	inferred output matrix
$\mathbf{z}$	embedded state matrix
α	learning rate
γ	encoding function (NN)
ζ	a linear activation function
η	ramp function for violation degree
Λ	loss function element
λ	loss function element weight
μ	message-passing function (NN)
ν_d	a node in the load type node set
ν_e	a node in the branch type node set
ν_g	a node in the generator type node set
ν_k	a node in the PV type node set
ν_s	the only node in the slack type node set

σ	update function (NN)
φ	decoding function (NN)
ψ	a nonlinear activation function

Symbols regarding transmission grid

B	branch charging susceptance (PU)
Bs	bus shunt susceptance (MVAr)
Gs	bus shunt conductance (MW)
J_{loss}	branch power loss
$\tilde{P}$	active power imbalance (PU)
Pd	active power load (MW)
Pf	active power flowing from sender side of branch
Pg	active power generation (MW)
Pg^{max}	upper active power generation limit (MW)
Pg^{min}	lower active power generation limit (MW)
Pt	active power flowing from receiver side of branch
$\tilde{Q}$	reactive power imbalance (PU)
Qd	reactive power load (MVAr)
Qf	reactive power flowing from sender side of branch
Qg	reactive power generation (MVAr)
Qg^{max}	upper reactive power generation limit (MVAr)
Qg^{min}	lower reactive power generation limit (MVAr)
Qt	reactive power flowing from receiver side of branch
R	branch resistance (PU)
V	voltage magnitude (PU)
Vf	voltage magnitude on sending side of branch
Vt	voltage magnitude on receiving side of branch
V^{max}	upper voltage magnitude limit (MW)
V^{min}	lower voltage magnitude limit (MW)
X	branch reactance (PU)
Y	branch admittance (complex value)
Y_{br}	branch admittance matrix
Z	branch series impedance (complex value)
Γ	concatenated sending and receiving bus characteristics
δ	branch series admittance phase (rad)
θ	voltage phase (rad)
θf	voltage phase on sending side of branch
θt	voltage phase on receiving side of branch
ξ	branch series impedance magnitude (PU)
ϖ	concatenated branch characteristics
ρ	branch series admittance magnitude (PU)
τ	transformer off-nominal turns ratio
ϕ	branch series impedance phase (rad)
ω	transformer phase shift angle (rad)

Chapter 1

Introduction

1.1 Uncertainty in power system modeling

Power grids are complex networks characterized by distinctive topological properties. One way to define the power network is by considering the producers and consumers as the nodes, and the transformers and power lines as the edges. In contrast to other types of networks, in this case the interaction of nodes and edges can be described by underlying physical properties. On the other hand, power grids may behave in a non-intuitive manner and thus require appropriate modeling to adequately consider the interactions between the elements of the power system. As an example, the work in Schäfer et al. (2022) describes Braess' paradox in power grids: a phenomenon in which increasing the capacity of existing lines or adding new lines may reduce the overall system performance and stability.

Power system modeling is a vital tool for system operators to remain informed of the current condition of the grid, enabling efficient and secure electrical energy management. The modeling of power systems relies on expressing the physical state of the grid through a set of power flow equations that are based on physical laws that describe the way electricity flows through the grid, and the modeling of other elements such as generators, transformers, loads and their control systems (Glover, Sarma, and Overbye, 2012; Sauer, Pai, and Chow, 2017). The parameters of the power flow equations must be updated regularly with estimated and updated power and voltage values; in the past it was not needed to make very frequent corrections because the power system was overbuilt. However, in the last couple of decades, important infrastructure modernization changes have contributed to increasing uncertainty and complexity of the power grid; these changes include the integration of renewable energy sources, advanced sensors, demand response technologies, and energy storage systems. With the increasing complexity, robust and reliable power flow (PF) and optimal power flow (OPF) analyses are becoming simultaneously increasingly difficult to model and vital for the operation, control and optimization of power systems at all levels, and particularly at transmission level (Schweppe and Wildes, 1970; Gomez-Exposito, Conejo, and Canizares, 2009).

PF simulations are vital for reliability planning and congestion analyses (Glover, Sarma, and Overbye, 2012). In a power transmission system, congestion occurs when the demand for electricity exceeds the capacity of the transmission lines to deliver it. This can be caused by a variety of factors, including an increase in electricity demand, a decrease in the capacity of transmission lines due to maintenance or repair work, or a natural disaster that damages transmission infrastructure. On the other hand, the AC-OPF is a non-convex nonlinear optimization problem and the building bock of many applications, including minimizing the total operating cost of the power system while ensuring that the system is stable and secure under all operating conditions (security constrained OPF), optimal transmission switching

(Fisher, O'Neill, and Ferris, 2008), capacitor placement (Baran and Wu, 1989), expansion planning (Niharika, Verma, and Mukherjee, 2016), and security-constrained unit commitment (Jianhui Wang, Shahidehpour, and Zuyi Li, 2008).

There are several reasons why the current transmission grid is presenting challenges for conventional PF and OPF solvers in finding solutions. Power grids are characterized by nonlinear dynamics with variable behavior, this is partly due to the complex dynamics of the active and reactive load profiles which can vary in fast and chaotic ways (Wildberger, 1994). The variability of load profiles may be a consequence of several factors, such as: the economic activities and population in the area, the weather conditions, the price of electricity and of competing energy sources, and the end-use technology advancements. In recent years, increased interconnection and complexity of power systems has raised concerns about the performance, security, and control of transmission and distribution networks.

Thus, conventional PF and OPF solvers are facing several challenges when attempting to find simulate the current transmission grid. These challenges are largely due to the following factors:

- Increased electrical demand.
- Increased renewable energy penetration.
- Increased distributed power generation.
- Model-based uncertainty.
- Varying electricity market

In recent years both the importance and complexity of software modelling of the electrical grid have increased due to higher electrical demand and the need to increase the sustainability of the conventional power grid (Smith et al., 2022). These issues arise from the ambitious, yet necessary, goals for emission reduction. Electrification of diverse energy sectors, such as heating and transport, has become a strategy for decarbonisation, this phenomenon has consequently increased the demand and the dependence on electricity (Xie et al., 2021).

With the ultimate goal of mitigating climate change and developing a sustainable power infrastructure, the last few years have seen a growth in the integration of renewable energies (REs). As more RE is integrated with the existing power systems, more fluctuations in power and uncertainty have been introduced. Renewable energy is highly stochastic and less controlable than conventional sources, the dynamics introduced both on demand and supply sides make load profiles harder to forecast and introduce significant alterations in loads and generation profiles. It is becoming more common for transmission system operators to be forced to modify the generators setpoints to ensure that the electrical grid can meet the power demands. However, to ensure that the grid is working in an efficient and stable state, PF and OPF analyses must be frequently carried out. Similarly, for expansion planning, a large number of PF and OPF simulations must be performed, and the stochasticity introduced by RE sources increases the number of scenarios to consider, which may take an excessive amount of time. The frequency in which these operations can be performed depends on the computational complexity; OPF solvers may be prohibitively slow for large power grids. A typical way to tackle this problem is to solve OPF approximations such as the DC-OPF model; these approaches are computationally more efficient, but the solutions obtained may be fail to satisfy the given constraints, and it has been shown that the DC OPF voltages and phases are not AC feasible (Baker, 2021). As system loading increases, the difference between the AC and DC OPF solutions increases as well.

REs require power systems to become more flexible as they may cause loading in sections of the grid where it is usually not expected and cause instability (Babatunde, Munda, and Hamam, 2020). Modern power networks are encountering enormous difficulties relating to previously unheard-of amounts of load peaks and voltage swings as a result of the expanding deployment of distributed renewable generators, electric vehicles, and demand response programs. Additionally, inverter-based resources (such as wind and solar systems) lack inertia. In power systems, inertia refers to the kinetic energy stored in rotating generators which gives them the tendency to remain in motion and thus temporarily make up for power losses from failures in power plants. In the past, inertia from conventional generators running on fossil fuels, nuclear energy and hydropower was abundant. However, the reliance on inertia is largely based on primary frequency response grid services which is derived from relatively slow-responding mechanical systems, i.e. a grid with slower generators needs more inertia to maintain reliability than a grid that can respond quickly. Using electronics and smart technologies, inverter-based resources (such as REs) can quickly detect frequency deviations and respond to system imbalances (fast frequency response), thus reducing the need for inertia in the first place. Thus, REs inherently reduce the need for inertia but introduce the need to speed up optimization simulations to be able to integrate these faster fluctuating RE sources. Real-time monitoring of the smart power grid is therefore becoming more and more important for energy management as well as for the early diagnosis of system instability (Wang et al., 2019; Giannakis et al., 2013).

The penetration of RE sources relate to the expansion of decentralized power generation and storage, where orchestration is facilitated through new, digital technologies (Heymann et al., 2023). The current electric system was not designed to accommodate diversified and distributed power generation sources, particularly renewable ones characterized by their variable behavior (Afonso, Marques, and Fuinhas, 2019). Decentralized energy systems are characterized by geographically locating energy production close to the sites of energy consumption, this is an additional source of uncertainty since it implies an increased number of small dispersed generators connected to the power system. Decentralized energy systems are considered as one of the most important developing areas of electric power to be able to meet the increasing electric power demand under critical environmental and social constraints. However, the growth in the number of dispersed generators has caused a rise in complexity in controlling, protecting and maintaining existing power systems. Furthermore, many times the distributed generators are based on RE sources, which are intermittent and not controllable. These non-programmable energy sources produce generation profiles that vary in time both over long and short term, the latter instances being harder to predict (e.g. minute to minute and hourly changes) (Wan and Parsons, 1993). As a result of these trends, power systems are already becoming increasingly stressed.

Additional to the changes in the power grid infrastructure, model-based uncertainty is also a factor contributing to the challenge of adequately modeling these grids. All model-based methods face model inaccuracies to a certain degree, this is especially true for complex systems like power grids, as there is a lack of complete understanding of the physical phenomenons and their relationships. The customary way to solve the power flow analysis is the deterministic power flow, this method produces precise network voltages and flows through each line, but requires precise values for each input variable. The specified values rely on assumptions about the operating condition of the grid, derived from historical measurements or forecasting and thus, many times are not accurate. Even in the case of state estimations, where the inputs are based on measurements, there are inaccuracies arising from issues such as time-skew problems, three-phase unbalance, static modeling of dynamic components, e.g. approximate representations of transmission lines and transformers through equivalent models, and on top of that assumptions about the parameters in these models, such as branch characteristics (namely resistance, reactance and shunt capacitance) that may be lacking, among others (Al-Othman and Irving, 2006; Dimitrovski, Tomsovic, and Vaccaro, 2011).

On top of the difficulty in modeling the generation, an additional source of uncertainty in PF analysis stems from the challenge in modeling the dynamics of the electricity market. All of these uncertainties can affect the model-based PF solutions to a significant extent and emphasize the need for researching different options to obtain reliable solution algorithms that are robust to the effect of data uncertainties (Vaccaro and Villacci, 2009).

1.2 Artificial intelligence for power system analysis

Many of the methods currently used for power system modelling were developed before widespread integration of variable renewable generation energy and the electrification of transport and heating, furthermore, most are computationally intensive. The increasing complexity of the electricity system may become too convoluted to describe in a timely and precise manner with conventional PF analysis approaches, thus requiring the investigation of new tools for power system modelling (Tovar-Facio, Martín, and Ponce-Ortega, 2021).

To account for the nonlinearity, variability and stochastic nature of power grids, diverse methodologies to carry out PF analyses have been investigated, such as probabilistic methods and sample based approaches. In particular uncertainty propagation using sampling based methods (e.g. Monte Carlo) may require a large number of model runs to sample various combinations of input values, resulting in prohibitively expensive computations. On the other hand, probabilistic methods encounter problems in dealing with the non-normal distribution and the dependence on possessing large amounts of input data, as well as the difficulty arising from accurately identifying probability distributions for some types of input data, such as the power generated by REs; these issues limit the applicability of the previously mentioned methods.

Artificial Intelligence (AI) encompasses various technologies e.g., expert systems, machine learning, deep learning, reinforcement learning, computer vision, natural language processing that can complement, and extend, human cognitive capabilities (Goodfellow, Bengio, and Courville, 2016). It is widely believed that AI can positively contribute to achieving a majority of the United Nations “Sustainable Development Goals” (SDG). A recent study showed how AI may enable 79% of all targets across all SDGs, mainly through technological improvements (Vinuesa et al., 2020).

The increasing complexity of planning and operating electrical grids and electricity markets due to the higher number of actors and services could potentially be addressed through AI. The application of AI in the energy sector, and specifically for assisting power system operators and energy management systems, is not new (Zhang, Hope, and Malik, 1989); the first instance of AI applied to power system operation dates as far back as 1983, when many AI systems were referred to as "expert systems" (Sakaguchi and Matsumoto, 1983). Neural networks and expert systems have been continuously used for more than 30 years in the energy sector with different applications such as:

- Enhancing energy system operation and optimization (Köhnen et al., 2022).
- Electricity network planning (Borges and Martins, 2012).
- Consumer behavior and tariff analysis (Bregere and Bessa, 2020).
- Electric load or renewable power forecasting (Sweeney et al., 2020; Hong and Fan, 2016; Heymann et al., 2021).
- Policy making (Zuiderwijk, Chen, and Salem, 2021).

With the growing availability of data and computing power, deep learning capabilities have grown to a great extent, however some challenges are still present in complex language and scene understanding, learning from small amounts of experience and transferring learning past training conditions, and reasoning about structured data. In order to benefit as much as possible from machine learning methods and address the previously mentioned challenges, efficient and well thought-out learning frameworks must be investigated. A framework that has received increasing attention in the last years is graph networks. Due to their inherent incorporation of the underlying structure in their architecture, graph networks are able to build more faithful and accurate models of data or phenomena that can naturally be represented as graphs, such as transmission systems.

To obtain interpretable PF and OPF solvers with tolerance to parameter uncertainties, acceptable computational cost and run-times, machine learning approaches deserve to be investigated. The main difficulties in applying AI and machine learning approaches to the OPF problems result from the presence of the complex, nonlinear feasibility physical and engineering constraints that regulate power flows; self supervised learning methods could be an interesting research direction to counter these challenges. Analyzing and operating the electrical grid more efficiently can various positive impacts: helping to accommodate REs on the existing power grid, positively impacting energy quality and consumer prices, and ultimately lead to a sustainable and reliable energy system that will help mitigate the climate crisis.

1.3 Research gap and research questions

In this section several works that are focused on solving state estimation, PF or OPF problems with neural networks are reviewed and the research gaps are discussed. The background analysis is presented with Tables 1.1 and 1.2. The columns of Table 1.1 are explained in the following points:

- The work presented in this thesis deals with solving the PF and a form of OPF of an electrical grid, therefore, the applications considered in the following tables are based on these topics. Some works that deal with state estimation (SE) are also considered in cases where the framing of the objective is similar to the PF or OPF challenges.
- It is shown that all works (except Donon et al. (2020)) need training data and carry out either exclusive or partial supervised learning. The acquisition of enough learning targets is time consuming, additionally these points must

cover a large enough learning space to generalize well to all possible circumstances that could be presented in inference time.

- A major part of the physical properties of a power grid depend on the characteristics of the transmission lines; these characteristics influence energy losses and the overall efficiency of the grid. Most of the presented works do not consider branch characteristics even though it seems natural to include branch features which has such important influence in the behavior of the power grid when analyzing or optimizing these.
- In Table 1.1 the column "scalable" refers to the model size being independent of the size of the electrical grid to be analyzed, making it scalable in both time and memory space to deal with larger grids.
- To be useful in close to real-world conditions, it would be beneficial for the methodologies to be robust to changes in topology in some degree or other, however must presented works do not currently deal with these situations.

TABLE 1.1: Literature review (first part).

Publication	**General application**	**Needs training data**	**Branch characteristics**	**Scalable**	**Changes in topology**
L. Zhang et al., ’19	SE	yes	✗	✗	✗
L. Wang et al., ’20	SE	yes	✗	✗	✗
Q. Yang et al., ’20	SE	yes	✗	✓	✗
M. Tran et al., ’21	SE	yes	✗	✗	✗
V. Bolz et al., ’19	PF	yes	✗	✓	✗
B. Donon et al., ’19	PF	yes	✗	✓	✓/ ✗
B. Donon et al., ’20	PF	no	✓	✓	✓/ ✗
D. Wang et al., ’20	PF	yes	✗	✓	✗
X. Hu et al., ’21	PF	yes	✗	✗	✗
T. Pham & X. Li, ’21	PF	yes	✗	✗	✗
J. Hansen et al., ’22	PF	yes	✓	✓	✓
D. Owerko et al., ’19	OPF	yes	✗	✓	✗
F. Fioretto et al., ’20	OPF	yes	✗	✗	✗
Z. Yan & Y. Xu, ’20	OPF	yes	✗	✗	✗
R. Nellikkath & S. Chatzivasileiadis, ’21	OPF	yes	✗	✗	✗
L. Zhang et al., ’20	DC OPF	yes	✗	✗	✗

The columns of Table 1.2 (which is simply a continuation of Table 1.1) are explained in the following points:

- The first column indicates time improvement with respect to a specified benchmark method. The symbol $\sim$ is meant to mean "approximately". The following abbreviations are used: "NS" indicates that it is not specified, "NR" indicates the Newton-Raphson method, "WLS" indicates the weighted least squares method, "MC" indicates the Monte-Carlo method, "LR" and "KNN" indicate the linear regression and k-nearest neighbors algorithms, respectively, and "IPOPT" indicates the interior point optimizer; at times the solver method is not specifically mentioned.
- In the third column, a very brief description of the data pre-processing is presented. In this context the abbreviation "inj" indicates all power injections, including load and generation, while "gen" indicates only power generation, and

"V mag" indicates the voltage magnitude. The term "noise" simply indicates that some amount of noise was added to the input data. Down-sampling refers to obtaining the training or testing input samples from a few chosen samples of a larger grid (usually from a real world data-set). Smoothing refers to removing noise from a given data set. The state estimation cases usually deal with missing or corrupt data since they usually have to deal with data from supervisory control and data acquisition (SCADA) systems. The hot start mentioned in Fioretto, Mak, and Hentenryck (2020) refers to the proposed model learning an OPF solution from a previously solved state and inferring the next state. In most cases presented in the table, some idea of the percentage of noise introduced in the inputs with respect to the benchmark cases is presented, but when the noise is especially varied among the different inputs, it is not specified, such as with Donon et al. (2020) and Hansen, Anfinsen, and Bianchi (2022).

- The column that indicates physics-informed topology points out whether information of the electrical grid configuration guides in some way the architecture of the proposed models by imposing some physical meaning on some of the outputs in the neural network layers. In traditional neural networks, there are numerous redundant connections and parameters that hinder their out-of-sample performance, which is one reason for which models with physics-informed topology are more efficient; most of the works that include this aspect are based on graph networks.

- By including a physics-informed loss function several advantages are introduced, including adding interpretability to the model, encouraging the neural networks to generate physically consistent solutions, reducing the search space of the learnable parameters, thus reducing the amount of training samples needed to correctly train the models. Usually, when a physics-informed loss function is applied in conjunction with the need of training data, it is the case that a regularization term is added to the loss function to reduce the amount of training data needed. These regularization terms have been found to be especially useful to deal with the constraints present in the OPF problem, many times adopting Lagrangian based learning by considering the Karush-Kuhn-Tucker (KKT) conditions.

As can be noted, most works depend heavily on the availability of training data. This makes the learning space quite large and takes time and computational resources to obtain target values. Another important point is that not many works consider branch characteristics or changes in topology, which are interesting and important factors in the PF and OPF problems (especially for planning-related services). Thus, the contributions of this thesis are guided by the following research questions:

Research question 1: *Can we develop a computationally efficient, flexible, neural network based model of the transmission grid that does not depend on training data?*

The main contribution of this thesis that addresses this research question is presented in Chapter 3. Here a physics-informed graph network based model is presented; it is scalable, able to incorporates transmission branch characteristics and changes in electrical grid topology, and is trained in a non-supervised fashion (i.e. it does not need training data). This model is customizable and able to be used for

TABLE 1.2: Literature review (second part).

Publication	Time improvement and benchmark	Data pre-processing	Physics-informed topology	Physics-informed loss function
L. Zhang et al., '19	500x faster, NR	Noise in P & Q inj.: 20%, V mag : 10%	✗	✗
L. Wang et al., '20	NS	For up to 3% of data: missing values, noise	✗	✗
Q. Yang et al., '20	NS	Noise in P & Q inj.: 20%, V mag : 10%	✓	✗
M. Tran et al., '21	160x faster, WLS	Noise in P & Q loads: 50%	✓	✗
V. Bolz et al., '19	NS	Noise in P & Q load, and P gen: 50-150%	✗	✗
B. Donon et al., '19	2x faster, NR	Noise in P & Q inj, V mag	✓	✓
B. Donon et al., '20	∼10X faster, NR	Noise in P & Q inj., V mag and line characteristics	✓	✓
D. Wang et al., '20	∼25 x faster, MC	Noise in P & Q inj.:10%	✗	✗
X. Hu et al., '21	Slower than LR and KNN	P & Q inj. scaling and down-sampling, noise	✓	✓
T. Pham & X. Li, '21	NS	Noise in V mag: 10%	✗	✗
J. Hansen et al., '22	Faster than NR	Noise in P & Q load, P gen, line characteristics	✓	✗
D. Owerko et al., '19	10^5 x faster, IPOPT	Noise in P & Q load: 90-110%	✓	✗
F. Fioretto et al., '20	10^4 x faster, AC solver	Noise in P & Q load: 20%; hot start	✗	✓
Z. Yan & Y. Xu, '20	2 x faster, IPOPT	Noise in P & Q load: 5%; smoothing.	✗	✓
R. Nellikkath & S. Chatzivasileiadis, '21	Slower to train, NS test	Noise in P & Q load: 60-100%	✗	✓
L. Zhang et al., '20	Faster than DC-OPF	NS	✗	✓

different applications.

Research question 2: *Can the resulting neural network based model be utilized for solving the system of equations necessary for solving the power flow problem?*

To address this research question, the model described in Chapter 3 is assigned the application of PF solver. A description of the customizations necessary to carry out this task are described in Chapter 4 and in the work Lopez-Garcia and Domínguez-Navarro (2023). The described model is successfully tested on grids of different size from the ones seen during training, proving great generalzation capabilities.

Research question 3: *Can the resulting neural network based model be utilized to minimize a cost function subject to constraints for solving the optimal power flow problem?*

To address this last research question, the model described in Chapter 3 is assigned the application of OPF solver, described in Chapter 5. The proposed OPF solver is tested both on randomly generated test cases and on time series load data. It successfully minimizes the constraints included in the determined loss function and significantly reduces calculation time compared to the traditional methods.

Chapter 2

Background

This chapter is dedicated to describing some key concepts related to the power flow applications addressed, and the machine learning frameworks that were used to develop the proposed model in this thesis. In the first part, a description of both power flow and optimal power flow analyses is presented, and an attempt is made to explain the technical complexities behind solving these simulations. In the second part, the key concepts of traditional multilayer perceptrons and graph networks are presented, as they are instrumental to constructing the typed graph network based model which is the main focus of this work.

2.1 Power systems side

For correct power system operation, power flow (PF) analyses must be executed frequently, as they are necessary for many procedures such as power systems planning, security assessment, management and optimization (Glover, Sarma, and Overbye, 2012). Conventionally, the PF analysis is carried out by determining and solving a set of nonlinear algebraic equations with iterative numerical analysis methods; most known methods of this type have been tested at some point to solve the PF problem (Stott, 1974; Amerongen, 1989).

While the PF problem consists of solving a system of nonlinear equations, the optimal power flow (OPF) problem minimizes an objective function subject to constraints that include the PF equations. The OPF problem helps to optimize electrical power systems by determining the appropriate generator setpoints for power and voltage, in response to specific load demands. It is a nonlinear, non-convex problem over complex values and variables, and it is often necessary to solve it repeatedly in different scenarios, either in real-time or as part of larger studies. This is especially important due to the increasing unpredictable nature of power systems, particularly due to the increasing use of renewable energy sources.

Many variations of the AC-OPF problem are relevant to power system analysis, in all cases the solution to the objective function much satisfy the physical and engineering constraints of the power system (Chowdhury and Rahman, 1990). In fact, one of the key difficulties of this learning task is the presence of the complex nonlinear feasibility constraints in the OPF. The work in this thesis focuses on a specific version of the AC-OPF problem, with the main purpose of minimizing power loss over transmission lines.

2.1.1 Power flow formulation

Transmission grids are basically formed by substations (also called buses), transmission lines, loads and generators. The buses are the nodes to which the other elements are connected, and the branches connect two buses. The objective of the PF analysis is to determine the voltage magnitude and phase at all buses for a given load, generation, and grid configuration under balanced three-phase, steady-state conditions. Fig. (2.1) illustrates a power grid and the transmission grid related to it.

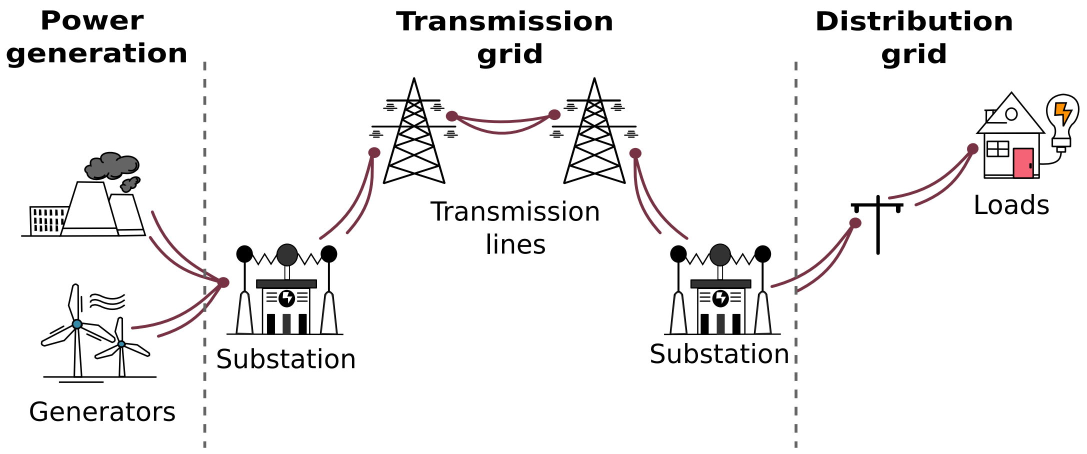


FIGURE 2.1: Example power grid.

The branches are modelled internally using the standard π transmission line model; each of the two buses that are connected to each branch are assigned either a sender ("from") or receiver ("to") node identification. The branch series impedance vector has the complex value: $Z = R + \mathbf{i}X := \xi\angle\phi \in \mathbb{C}^{N_E}$; where the resistance of the branch model constitutes the real part, and the equivalent reactance the imaginary part; N_E is the total number of transmission lines. The admittance of the branch is given by: $Y = \frac{1}{Z} = \frac{1}{\xi}\angle - \phi := \rho\angle\delta$; defining the magnitude and phase of the series admittance of each branch as ρ and δ, respectively. The line charging susceptance of the equivalent branch model is represented by B; this general model can include an ideal phase shifting transformer located at the sender end of the branch. The angle of the transmission line is the angle between the two voltage vectors to which the branch is connected, it is represented with ω; in the case of transformers, this value is the phase shift angle of the transformer. The off nominal turns ratio of the transformer is represented with τ (for lines it equals zero).

The power that flows through the grid depends on the power imbalances at the buses and the impedance of the branches, while the power balance at each bus is determined by the possible loads, generators and branches attached to it. Loads and generators constitute the external injections of the power grid, loads as a specified power demand, and generators as a specified power source. Given the grid topology, the specified values of the injections and line characteristics, the proposed PF solver computes the resulting voltages in the buses and hence the current flow through the branches can be determined. The relationship between the branch characteristics, the voltages of the buses and the currents is given by:

$$\begin{bmatrix} If \\ It \end{bmatrix} = Y_{br} \begin{bmatrix} Vf \\ Vt \end{bmatrix} \tag{2.1}$$

where If and It represent the vector of complex current injections at the sender and receiver sides of all transmission lines; Vf and Vt represent the vector of complex voltage values at the sender and receiver sides of all branches; branch admittance matrix of the transmission grid Y_{br} given by:

$$Y_{br} = \begin{bmatrix} \left(Y + \mathbf{i}\frac{B}{2}\right)\frac{1}{\tau^2} & -Y\frac{1}{\tau e^{-\mathrm{i}\omega}} \\ -Y\frac{1}{\tau e^{-\mathrm{i}\omega}} & Y + \mathbf{i}\frac{B}{2} \end{bmatrix} \tag{2.2}$$

where the series admittance vector Y is defined as: $Y = \frac{1}{Z}$.

It should be noted that before the load flow is solved, the network losses are unknown, thus, a generator bus called the slack bus, is designated to compensate for these losses. The voltage magnitude and phase are given beforehand for the chosen slack bus, and the power needed to compensate for the total grid losses must be determined. In addition to the slack bus, two other types of buses can be defined: PV buses constitute the set of buses directly connected to a generator (that are not the slack bus); the remaining non-generation buses are classified as load or PQ buses. For each PV bus the voltage magnitude V and active power generation Pg is given; the voltage phase θ and the generated reactive power Qg, must be determined. For each PQ bus the active P and reactive powers Q at the bus are given; the voltage phase θ and magnitude V must be determined.

For a solution to be obtained, the power balance in all nodes must be achieved by solving a nonlinear equation system of the form $\tilde{S} = 0$ (Glover, Sarma, and Overbye, 2012), which is deconstructed into nodal power balance equations as functions of unknown voltage values, as shown below for a bus n:

$$\tilde{P}_n(V_n, \theta_n) = Pg_n - Pd_n - Gs_n V_n^2 - \mathrm{Re}\left[\left(\sum_{\substack{e\in\mathcal{N}(n)\\ n=\mathrm{from}_e}} If_e + \sum_{\substack{e\in\mathcal{N}(n)\\ n=\mathrm{to}_e}} It_e\right)\cdot V_n\right] \tag{2.3}$$

$$\tilde{Q}_n(V_n, \theta_n) = Qg_n - Qd_n + Bs_n V_n^2 - \mathrm{Im}\left[\left(\sum_{\substack{e\in\mathcal{N}(n)\\ n=\mathrm{from}_e}} If_e + \sum_{\substack{e\in\mathcal{N}(n)\\ n=\mathrm{to}_e}} It_e\right)\cdot V_n\right] \tag{2.4}$$

$$\forall n \in \{1, \ldots, N_B\}$$

where N_B represents the total number of buses, $e \in \mathcal{N}(n)$ represents all the branches that are connected to bus n, and the complex value $(Gs - \mathbf{i}Bs)\, V^2 \in \mathbb{C}^{N_E}$ represents the effect of the shunt conductance on the transmission lines.

With N_K and N_D representing the total number of PV and PQ buses, respectively, there are $N_K + 2N_D$ voltage values that must be found (only phase for PV buses and both phase and magnitude for PQ buses). Afterwards, $N_K + 1$ power balance equations are solved to find the reactive power injections of PV buses and the active power injection of the slack bus; this way all unknown variables are found.

After finding all voltage magnitude and phase values, the remaining power unknown values are found with the following, simple power balance equations:

$$Qg_k = Q_k(V_k, \theta_k) + Qd_k \tag{2.5}$$

$$Qg_s = Q_s(V_s, \theta_s) + Qd_s \tag{2.6}$$

$$Pg_s = P_s(V_s, \theta_s) + Pd_s \tag{2.7}$$

$$\forall\, k \in \{1, \ldots, N_K\}$$

2.1.2 Electric power loss

Considering the definitions from the previous subsections, the active and reactive power flowing from each end of all transmission lines are calculated as (Glover, Sarma, and Overbye, 2012):

$$Pf_e = Vf_e Vt_e \frac{\rho_e}{\tau_e} \sin\left(\theta f_e - \theta t_e - \delta_e - \omega_e\right) + \frac{Vf_e^2}{\tau_e^2} \rho_e \sin\left(\delta_e\right) \tag{2.8}$$

$$Pt_e = Vt_e Vf_e \frac{\rho_e}{\tau_e} \sin\left(\theta t_e - \theta f_e - \delta_e + \omega_e\right) + Vt_e^2 \rho_e \sin\left(\delta_e\right) \tag{2.9}$$

$$Qf_e = Vf_e Vt_e \frac{\rho_e}{\tau_e} \cos\left(\theta f_e - \theta t_e - \delta_e - \omega_e\right) + \frac{Vf_e^2}{\tau_e^2} \left(\rho_e \cos\left(\delta_e\right) - \frac{B_e}{2}\right) \tag{2.10}$$

$$Qt_e = Vt_e Vf_e \frac{\rho_e}{\tau_e} \cos\left(\theta t_e - \theta f_e - \delta_e + \omega_e\right) + Vt_e^2 \left(\rho_e \cos\left(\delta_e\right) - \frac{B_e}{2}\right) \tag{2.11}$$

where Pf and Pt represent the power flowing from the "sending" end and from the "receiving" end of all lines, respectively; θf and θt, and Qf and Qt are similarly defined, but with the reactive power and voltage phase, respectively.

In this work, only real power line losses under steady-state operating conditions are considered. The electric power loss through each transmission line is calculated simply as the difference between the power that leaves the sending end of a line, and the actual power that reaches the receiving end of a line, as shown in the following equation where the power loss is represented by $P_{loss} \in \mathbb{R}^{N_E}$:

$$P_{loss,e} = ||Pf_e| - |Pt_e||, \ \forall e \in \{1, \ldots, N_E\} \tag{2.12}$$

It should be noted that the last part of equations (2.3) and (2.4) that represent the real and imaginary parts of the power flowing from and to every node can be calculated by aggregating the equations (2.8) - (2.11). It should be noted that these values in these equations are per transmission line, i.e. $Pf, Pt, Qf, Qt \in \mathbb{R}^{N_E}$ and that the sending and receiving buses can be repeated, since a bus can be connected to more than one branch. Thus, to calculate the power imbalance at each node, the total power flowing from and to each node must is added, as shown below:

$$Pf_n = \sum_{\substack{e \in \mathcal{N}(n) \\ n = from_e}} Pf_e, \quad Pt_n = \sum_{\substack{e \in \mathcal{N}(n) \\ n = to_e}} Pt_e \tag{2.13}$$

$$Qf_n = \sum_{\substack{e \in \mathcal{N}(n) \\ i = from_e}} Qf_e, \quad Qt_n = \sum_{\substack{e \in \mathcal{N}(n) \\ i = to_e}} Qt_e \tag{2.14}$$

$$\forall n \in \{1, \ldots, N_B\}$$

2.1.3 Optimal power flow formulation

The AC-OPF problem tackled in this work focuses on finding the active and reactive power generation, and voltage magnitude and phase of generator buses, and the voltage magnitude and phase of load buses, such that the total power loss in the transmission grid is minimized. The problem also considers voltage magnitude and power generation constraints. The complete OPF formulation is described in model 1.

Model 1 AC Optimal Power Flow Problem

```
**Variables:**
$V_g,\ \theta_g,\ Pg_g,\ Qg_g\ \forall g \in \{1, \dots, N_G\}$
$V_d,\ \theta_d\ \forall d \in \{1, \dots, N_D\}$
**minimize :** $\sum_{e=1}^{N_E} ||Pf_e| - |Pt_e||$
**subject to:**
$\theta_s = 0$
$\tilde{P}_n(V_n, \theta_n) = 0$                                  ▷ eq. (2.3)
$\tilde{Q}_n(V_n, \theta_n) = 0$                                  ▷ eq. (2.4)
$Pg_g^{min} \leq Pg_g \leq Pg_g^{max}\ \forall g \in \{1, \dots, N_G\}$
$Qg_g^{min} \leq Qg_g \leq Qg_g^{max}\ \forall g \in \{1, \dots, N_G\}$
$V_n^{min} \leq V_n \leq V_n^{max}\ \forall n \in \{1, \dots, N_B\}$
```

In this work, the OPF problem is solved by inferring the variables included in model 1, such that the objective function minimizes the total electrical power loss of the transmission lines. The objective function is subject to several constraints, the first one fixes the slack phase at zero (the slack bus is typically taken as the reference bus). The second and third constraints capture the need to maintain power balance. The last three constraints set the generator injection limits and the bus voltage magnitude limits.

It is important to note that solving AC-OPF problems is NP-Hard in general (Bienstock and Verma, 2019). Consequently, the different solution methods are expected to display a wide variety of quality-runtime trade-offs and are likely to be specialized to specific classes of inputs.

2.2 Machine learning side

In this second section of the chapter, focus is shifted to the machine learning aspects necessary for understanding the typed graph neural networks based model proposed in this thesis for solving the PF and OPF problems. Some basic concepts of the classic multilayer perceptron are introduced, these are the functions applied in the graph networks which are later described. Graph neural networks are, in turn, the basis of the final typed graph networks framework that is applied throughout this work.

2.2.1 Multilayer perceptron building blocks

A multilayer perceptron (MLP) is a class of the feedforward artificial neural networks consisting of multiple fully connected layers. In the context of feedforward NNs, the connections between the units do not form a cycle and the information flows through the network in only one direction, from the input layer to the output layer. Being fully connected, the relations are all-to-all (all units in a layer are connected to all units of the following layer), which means that all input units can interact to determine any output unit's value, independently across outputs. Each fully connected layer is composed of artificial neurons (also called units), and performs an affine transformation on these, followed by a nonlinearity (Goodfellow, Bengio, and Courville, 2016). This way, each layer output vector is the product between a weight vector and the input vector, followed by an added bias term, and finally an activation function.

For an MLP with L hidden layers, the following operations are implemented:

$$\mathbf{z}^{(0)} = \mathbf{x} \tag{2.15}$$

$$\mathbf{z}^{(l)} = \psi\left(\mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}\right), \; l = 1, \ldots, L-1 \tag{2.16}$$

$$\mathbf{y} = \psi\left(\mathbf{W}^{(L)}\mathbf{z}^{(L-1)} + \mathbf{b}^{(L)}\right) \tag{2.17}$$

where $\mathbf{x} \in \mathbb{R}^f$ and $\mathbf{y} \in \mathbb{R}^o$ represents the input and output of the layer, respectively; $\mathbf{z}^{(l)}$ represents the hidden state of layer l; $\mathbf{W}^{(l)} \in \mathbb{R}^{f_l \times f_{l-1}}$ and $\mathbf{b}^{(l)} \in \mathbb{R}^{f_l}$ represent the trainable parameters of layer l (kernel and bias, respectively); ψ is an activation function (typically nonlinear, e.g. hyperbolic tangent or rectified linear unit); f_l denotes the number of units of layer l. Typically, for regression problems, the last layer will be a linear activation function.

MLPs are trained using an optimization algorithm, such as stochastic gradient descent, to adjust the weights of the connections between the nodes in order to minimize the error between the predicted output and the true output.

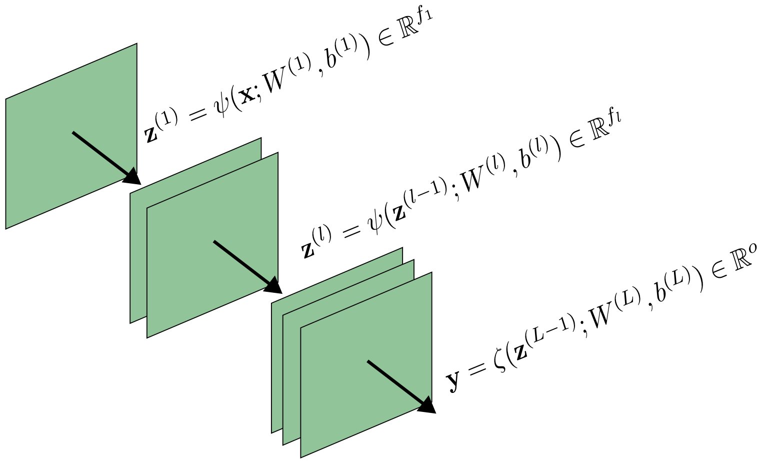

$\mathbf{z}^{(1)} = \psi(\mathbf{x}; W^{(1)}, b^{(1)}) \in \mathbb{R}^{f_1}$
$\mathbf{z}^{(l)} = \psi(\mathbf{z}^{(l-1)}; W^{(l)}, b^{(l)}) \in \mathbb{R}^{f_l}$
$\mathbf{y} = \zeta(\mathbf{z}^{(L-1)}; W^{(L)}, b^{(L)}) \in \mathbb{R}^{o}$

FIGURE 2.2: Multilayer perceptron layers scheme.

The recursive process is illustrated in Fig. 2.2; where the nonlinear activation function ψ is defined as $\psi(\mathbf{z}; \mathbf{W}, \mathbf{b}) = \psi(\mathbf{W}\mathbf{z} + \mathbf{b})$, meanwhile ζ represents a linear function defined as: $\zeta(\mathbf{x}; \mathbf{W}, \mathbf{b}) = \mathbf{W}\mathbf{x} + \mathbf{b}$. It should be noted that to define these MLPs, four hyperparameters need to be previously specified: number of layers, hidden and output dimension sizes, and activation function to be used.

In this work, MLPs are used as the basic building blocks of the graph neural network (GNN) model of the transmission grid.

2.2.2 Graph networks

For more than ten years, researchers have been studying and developing neural networks that can process and analyze graph data, organizing their calculations in a way that reflects the structure of the graph. These types of neural networks are referred to as GNNs (Gori, Monfardini, and Scarselli, 2005; Scarselli et al., 2009). Graph networks (GNs) have gained popularity in the last years due to their proven ability to learn patterns and relationships between entities of the underlying physical phenomenon. Commonly, real world phenomena and objects do not inherently possess a relevant order, instead, these entities can be ordered based on the characteristics of the relationships they have with each other. The problem with MLPs to capture these relationships is based on the fact that an MLP trained to predict an output based on a particular input, would not necessarily make an accurate prediction for the same inputs under a different ordering. In cases when the function inputs can be permutated, a basic MLP may consider each ordering as fundamentally different and thus require an exponential number of training examples to learn an approximating function.

Graph networks provide a way to handle such inputs in which the order is not important and the inputs can be permutated. To consider the relationships between elements, the state of each object i can be generally calculated as $\mathbf{x}_i^{(t+1)} = f\left(\mathbf{x}_i^{(t)}, \sum_j \varrho(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})\right)$, where function ϱ describes the relationship between object i and neighboring object j, and the next state of object i is computed from function f which depends on the current state of object i and the output of the relationship function ϱ. Function ϱ, which captures the relationship between neighboring elements, is the same throughout the entire graph model, so that it is capable of dealing with input permutation. Graph networks are capable of handling structures other than the pairwise relations described in the previous example; the interactions between elements depends on the topology of the graph and the resulting neighborhoods of each element (which are not dependant on a specific order).

Since the architecture of graph network models is directly related to the underlying phenomenon, they explicitly represent the sparsity of the relationships between different entities. The graph structured data applied makes it possible to use efficient algorithms for inference and reasoning, such as message-passing. Message-passing involves applying a standard procedure for exchanging information among the parts of a graphical model, allowing for the combination and partial parallelization of reasoning processes which can be applied to models of different sizes and configurations. Additionally, graph structured data, which consists of entities (also called nodes) and the relationships between them (also called edges) guides the learning process to prioritize learning about patterns and relationships, rather than just about independent elements.

Constraining the connections and interactions of the elements of graph network models such that they depend on the underlying structure induces a relational inductive bias and supports efficient algorithms for learning and reasoning. The work in Battaglia et al. (2018) presents a general framework for defining GNs, and the methods which operate on these. The general definition of graph networks defines a "class of functions for relational reasoning over graph-structured representations". It should be noted that the functions implemented do not need to be neural networks, but in this thesis these functions are always small MLPs.

Within the established GN framework, a graph is defined as list $\mathcal{G}$ of three elements: $\mathcal{G} = (\mathrm{u}; \mathcal{V}; \mathcal{E})$; where u represents a global attribute, $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ is the set of edges. Two methods are defined: update and aggregation functions. There is one update function defined for each element of the graph (applied per-node, per-edge and a once for the global update). The aggregation functions must be invariant to input permutations, and should take a variable number of arguments (e.g. elementwise summation, mean, maximu, etc.). The three aggregation functions are applied in a specific way: 1.) from updated edges to nodes, 2.)from updated edges to the global attribute, and finally 3.) from updated nodes to the global attribute.

These functions are executed in a latent space, according to the graph structure which is typically represented in matrix form by an adjacency matrix. This way, the embedded graph elements iteratively add information from their neighborhood over numerous message-passing steps, in which nodes are updated as a function of the embedded values of neighboring nodes and their own previous state. The broadcasting of signals on the graph is computed as a series of local operations, commonly matrix multiplications with adjacency matrices.

Even though the focus of this thesis is placed on nodal outputs, the output of GNs can be of different types:

- Edge focused : uses edges as output, e.g. to make decisions about interactions among elements (Kipf et al., 2018).
- Node focused : uses nodes as output, e.g. to reason about physical systems (Battaglia et al., 2016).
- Graph focused : uses the global attributes as output, e.g. to predict global attributes of a physical system or to answer questions about a visual scene (Gilmer et al., 2017; Santoro et al., 2017).

2.2.3 Typed graph neural networks

As was mentioned in the previous section, GNs have gained popularity for learning structured data and transferring learned information beyond training conditions (Battaglia et al., 2018). Using GNs as the basis for a neural network model for transmission systems, several benefits are hoped to be introduced. One advantage is that the model can scale well to handle larger power grids. Another advantage is that the model can be tested and generalize to grids of different size, or configuration from the ones seen during training.

However, to learn the interactions between the different elements of the transmission grid, the model needs access to all relevant bus and branch characteristics. Taking the transmission lines as edges, then it would be important to be able to process edge features effectively; there are few GNN models in the literature that are able to process edge features, such as the edge-condtioned convolutional layer proposed by Simonovsky and Komodakis (2017). Edge-conditioned convolutional layers can be used to embed branch features unto node representations. Some efforts were made during this doctoral work to apply these GNNs, however these layers have a significant computational cost and complexity in comparison with models that only consider node features. Besides, the relationship between nodes change as the features are propagated, so that the original edge information is lost after a few layers. Additionally, to account for the difference in operation of load, generation and slack substations, it was desirable to treat these as different types of nodes in the proposed graph model. Finally, in the case of transmission system modeling, accumulating local information into a global attribute does not intuitively make sense.

For the previous reasons, the typed graph networks (TGNs) (Prates et al., 2019) were adopted as the basis of this work. The basis of these types of models is that any node can be of a given type, and thus can be associated with its own embedding, update and output functions, and parameter set. The concept of node types is a generalization of the conventional GNs, essentially removing the forceful relationships between edges and nodes, edges and global attributes, and nodes and global attributes. These considerations give the TGN model increased versatility, and allow training on domains with richer structure than regular graphs. Node types are instantiated according to the ontology of the problem that is to be solved.

In general, the different node types can be embedded into (possibly) different latent spaces, and different types of nodes can share information with each other by computing messages from one embedding dimension to the other. This way, messages to be computed from type ν_i nodes to type ν_j nodes are implemented with a trainable function $\mu_{\nu_i \to \nu_j}$.

A TGN network can be defined with the following elements:

- A set of node types: $\mathcal{V} = \{\nu_i \in \mathbb{R}^{d_i} \mid i = 1, \dots, N\}$.
- A set of embedding dimension sizes: $\{d_1, \dots, d_N\}$.
- A set of message functions: $\mathcal{M} = \{\mu : \nu_i \to \nu_j \mid \nu_i, \nu_j \in \mathcal{V}\}$.
- A set of update functions: $\sigma_i : \mathbb{R}^{d_i + |\mathcal{N}(\nu_i)|} \to \mathbb{R}^{d_i} \mid \forall i \in \{1, \dots, N\}$.

where $|\mathcal{N}(\nu_i)|$ is the number of adjacent node types for a node type ν_i.

In the proposed TGN-based model, two sets of functions are defined additional the message-passing and update functions:

- A set of encoding functions: $\gamma_i : \mathbb{R}^{f_i} \to \mathbb{R}^{d_i} \mid \forall i \in \{1, \dots, N\}$.
- A set of decoding functions: $\varphi_i : \mathbb{R}^{d_i} \to \mathbb{R}^{o_i} \mid \forall i \in \{1, \dots, N\}$.

where f_i and o_i represent the input feature and output dimensions for a type i node, respectively. In this work, all of these functions are implemented with small MLPs. This way, the TGN-based model takes as input a feature vector for every node of every node type. All nodes are then projected to their corresponding embedding space through a linear encoding function γ, afterwards T message-passing (μ) and update (σ) steps are iteratively repeated. Finally nodal outputs are obtained from a final decode mapping with φ_i. The process of the whole layer is described in algorithm 2.

The outputs of the message-passing functions are aggregated via matrix multiplication with the adjacency matrices, which indicate the sparsity pattern between different node types. In this context, an adjacency matrix $\mathbb{A}_{i,j} \in \mathbb{R}^{N_i \times N_j}$ is non-zero if any of the ν_i type nodes is adjacent to any of the ν_j type nodes, where N_i and N_j represent the number of type ν_i and type ν_j nodes, respectively. The update function input for each node consists of the aggregation of all messages, concatenated with their previous embedded node state.

It should be noted that the messages are aggregated within the local one-hop neighborhood; by performing several steps, the receptive field of each node is extended, thereby allowing distant propagation of information. At the final update step, each node has shared information with neighboring nodes $T-$hops away as explained in more detail in Battaglia et al., 2018. This iterative message-passing process is illustrated in Fig. 2.3.

This way, as is evident, the amount of parameters to be trained does not depend on the size of the grid; instead, the number of parameters depends on the number of node types, the input dimension of the node types, the embedding space sizes, and the output dimensions.

2.2.4 Physics informed machine learning

Deep neural networks have shown promise as a machine learning approach, but they require a large amount of data in order to be trained effectively. In cases where

Algorithm 2 Basic typed graph network layer

```
▷ A TGN layer $l$ is a graph partitioned into $N$ node types.
$\text{TGN} : \mathcal{G} = \left(\mathcal{V} = \bigcup_{i=1}^{N} \mathcal{V}_i\right)$

▷ Calculate adjacency matrices for all adjacent node types.
$\mathbb{A}_{i,j},\ \forall \nu_j \in \mathcal{N}(\nu_i),\ \forall i = \{1, \ldots, N\}$

▷ Embed input features of all node types.
$\mathbf{z}_i^{(0)} = \gamma_i^{(l)}(\mathbf{x}_i),\ \forall i = \{1, \ldots, N\}$

▷ Perform $T$ message-passing, aggregation and update steps.
for $t = 1, \ldots, T$ do

    ▷ Compute messages from adjacent nodes of all node types.
    $m_i^{(t)} = \{\mu_{j \to i}^{(l)}(\nu_j) \mid \forall \nu_j \in \mathcal{N}(\nu_i)\},\ \forall i = \{1, \ldots, N\}$

    ▷ Aggregate all received messages of all node types.
    $\bar{m}_i^{(t)} = \{A_{i,j} \times \mu_{j \to i}^{(l)}(\nu_j) \mid \forall \nu_j \in \mathcal{N}(\nu_i)\},\ \forall i = \{1, \ldots, N\}$

    ▷ Compute updated embedding for every node type.
    $\mathbf{z}_i^{(t+1)} = \sigma_i^{(l)}\left(\mathbf{z}_i^{(t)}, \bar{m}_i^{(t)}\right),\ \forall i = \{1, \ldots, N\}$
end for

▷ Decode refined embeddings unto desired output dimension.
$\mathbf{y}_i = \varphi_i^{(l)}\left(\mathbf{z}_i^{(T)}\right),\ \forall i = \{1, \ldots, N\}$
```

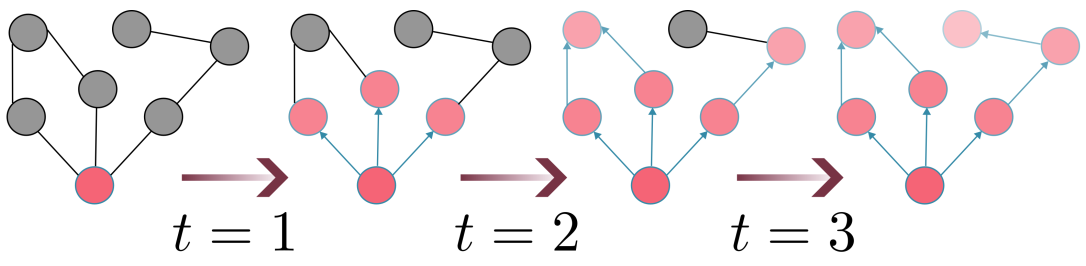


FIGURE 2.3: Propagation of information through the message-passing steps.

this data is not available, physics-informed approaches use physical laws to generate additional information that can be used to train these networks. In the context of transmission system modeling, for even moderately large power systems, covering the whole load space with labeled data is intractable due to the curse of dimensionality. Additionally, the physical laws that govern transmission systems are well known; for these reasons it seems adequate to apply physics-informed neural networks to the problem at hand.

There are different ways to generate physics-informed machine learning algorithms, three general ways which are based on the introduction of biases are listed below (Karniadakis et al., 2021):

- Observational bias: Introduced directly through data that embodies the underlying process.
- Inductive bias: Considers prior assumptions that are incorporated into the machine learning model architecture, such that the predictions are guaranteed to implicitly satisfy a set of given physical laws.
- Learning bias: Introduced by appropriate choice of loss functions, constraints and inference algorithms that can modulate the training phase of a machine learning model to favour convergence towards solutions that adhere to the underlying physics.

These different modes of biasing a learning algorithm towards physically consistent solutions are not mutually exclusive and can be effectively combined to yield a very broad class of hybrid approaches for building physics-informed learning machines.

In this work, by basing the model on graph networks inductive bias is incorporated. Furthermore, the loss functions used in the following application chapters (chapter 4 and chapter 5) integrate power equilibrium equations, and several constraints regarding the operation of elements in the transmission grid. This way, both inductive and learning biases are included in the basic model presented in this thesis.

Chapter 3

Model Setup

3.1 Introduction

This chapter describes the fundamental elements of the physics-informed typed graph network (PI-TGN) model developed in this work. A description is made of the different elements, their purpose and their role in the overall purpose of developing an efficient, robust model for power systems analysis.

Some key aspects of the proposed model are that the parameters scale linearly with respect to the size of the underlying electrical grid, it supports testing on electrical grids of different sizes and configurations, and it does not need training data.

3.2 Basic PI-TGN model of transmission grid

Representing a transmission system as a graph is straightforward since the different elements of the grid are related through the grid topology; the state of the grid depends on both external injections and this configuration. The approach taken in this thesis is to represent the transmission grid state as graph structured data, then to learn the relationships between the different elements present in the grid, and finally output nodal predictions to estimate one or more variables related to the node. In this work, each node can represent a substation, a transformer or a transmission line. This concept is illustrated with a simple example in Fig. 3.1, the left part represents the transmission grid and the right part shows the corresponding model architecture, which is directly related to the electrical grid structure. The example transmission grid includes PV, PQ and $slack$ buses, the branches that connect them, and external injections; each of the buses and branches are represented by a node type. The different grid elements are fundamentally different in that each has different characteristics and a different number of expected outputs, e.g. the branch characteristics are important factors in the resulting grid state, such that they cannot be treated as simple edges (nor even weighted edges), however, in the present work they are not expected to produce an output. In essence, the different grid elements should be treated as distinct node types. Thus, the basic model in this work is based on typed-graph networks (Prates et al., 2019) which allow different types of elements to be defined, instead of the usual nodes and edges (described in more detail in subsection 2.2.3). The use of TGNs allows to obtain faithful representations of the different elements present in electrical grids, such as the branches and the different types of buses, by considering each of them as different node types in the corresponding graph representation.

The basic architecture of the proposed PI-TGN model consists of a fixed number L of TGN layers, executed sequentially. The first layer inputs depend on the transmission grid element characteristics and initial states, external injections (loads

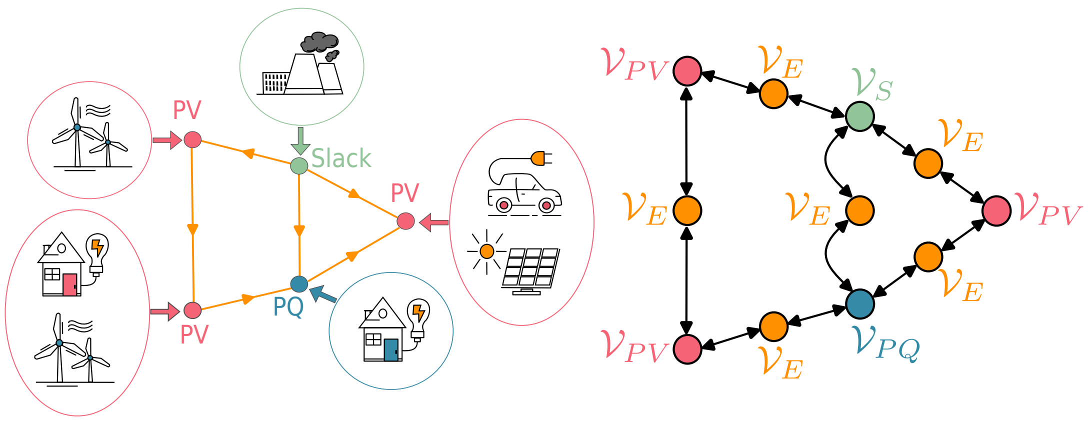


FIGURE 3.1: Small example of transmission grid and corresponding TGN architecture.

and generators), and topology. Each layer generates outputs for the predetermined nodes, which leads to an updated graph state. Each TGN layer (except the first) forms their input features from the updated graph state, along with the previously established external injections, electrical grid characteristics and configuration. The process is repeated until the the final inference values are obtained from the final grid state, from the last TGN layer.

In this work, the recursion of the TGN layers is achieved in two ways: either stacking independently parameterized TGN layers or recursively updating the state of a single TGN layer, as illustrated in Fig. 3.2. By using shared parameters, the total model size is L times smaller than with independent parameters, as the same TGN layer is recursively used for each approximation.

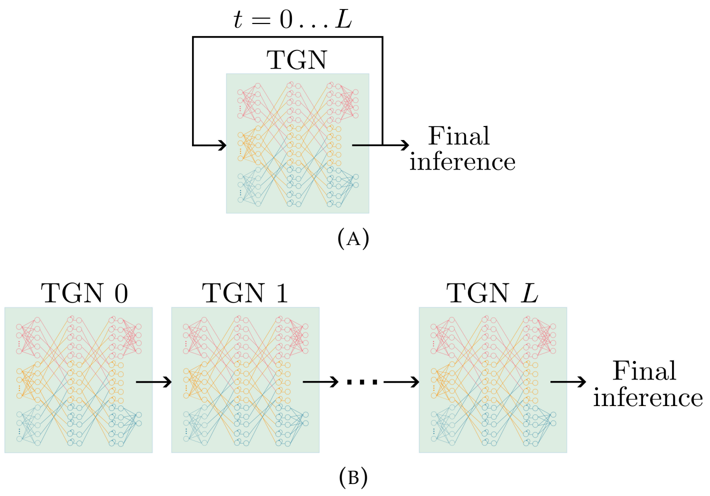


FIGURE 3.2: Shared (top) and independent (bottom) parameters of the TGN layers.

The output of each TGN layer could be regarded as analogous to an iteration in the conventional Newton-Raphson method, however an important difference to note is that the proposed TGN-based method does not act directly on the variables to be estimated (voltage values), but instead modifies the parameters of the NNs used in each layer according to the established loss function. This way, during training, the TGN-based model learns, not just the characteristics of the different elements, but also the relationships and dynamics between them, which allows it to minimize the loss function at inference time without needing to solve an optimization problem.

The number of TGN layers L is empirically chosen to produce a precise enough solution while maintaining the overall size of the solver relatively small. Furthermore, to avoid the distortion of messages being propagated from distant nodes Topping et al., 2022, the number of message-passing and update steps T is also small.

3.2.1 TGN layers

Each TGN layer takes different input features for each type of node, embeds these inputs into a latent space, performs a fixed number of message-passing and embedded update steps, and finally the predefined nodes are decoded to obtain a layer output. The encoding, message-passing, update and decoding functions are small fully connected neural networks; their parameters are the only ones that must be learned.

All TGN layers have the same graph configuration, which is fully defined by the node types and the way they are connected. The architecture of each TGN layer is defined by a graph $\mathcal{G} = (\mathcal{V}_{v_1}, \dots, \mathcal{V}_{v_N}, \mathcal{V}_E,)$, where N types of bus nodes $\mathcal{V}_v$ are defined, in addition to branch type nodes $\mathcal{V}_E$. The topology of the graph is defined by N adjacency matrices, since bus type nodes cannot be directly connected to each other but must be connected to at least one branch type node, and branch type nodes cannot be directly connected to each other either. The elements and organization of each TGN layer is shown in Fig. (3.3); each part is further explained in the following subsections.

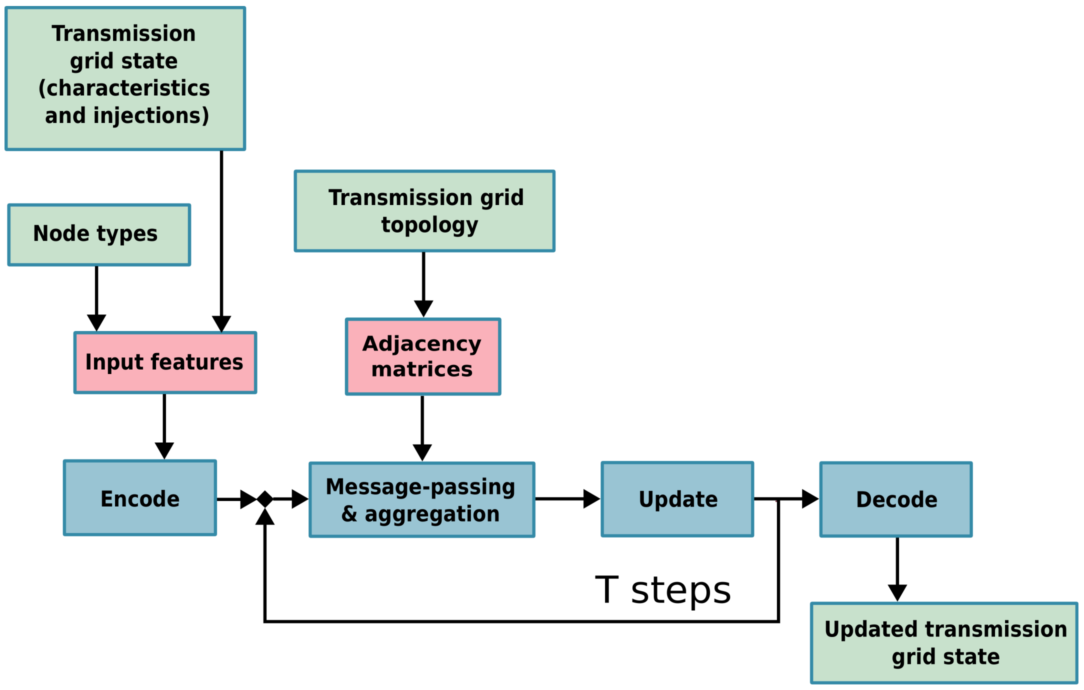

Transmission grid state (characteristics and injections)
Node types
Transmission grid topology
Input features
Adjacency matrices
Encode
Message-passing & aggregation
Update
Decode
T steps
Updated transmission grid state

FIGURE 3.3: TGN layer elements and organization.

3.2.2 Node types

To model the transmission grid, two general node types are defined: bus nodes and branch nodes (which include transformers). Branch type nodes are defined as:

$$\nu_e \in \mathcal{V}_E, \quad e = 1, \dots, N_E, \quad N_E = |\mathcal{V}_E|$$

The determination of the different bus type nodes depends on the application, the different types of bus type node sets defined in this work are listed below:

- Load / PQ nodes: $\nu_d \in \mathcal{V}_D, \quad d = 1, \dots, N_D, \quad N_D = |\mathcal{V}_D|$.
- PV nodes : $\nu_k \in \mathcal{V}_K, \quad k = 1, \dots, N_K, \quad N_K = |\mathcal{V}_K|$.
- *slack* nodes : $\nu_s \in \mathcal{V}_S, \quad |\mathcal{V}_S| = 1$.
- Generator nodes : $\nu_g \in \mathcal{V}_G, \quad g = 1, \dots, N_G, \quad N_G = |\mathcal{V}_G|$.
- Bus nodes : $\nu_n \in \mathcal{V}_B, \quad n = 1, \dots, N_B, \quad N_B = |\mathcal{V}_B|$.

Note that generator type nodes include PV buses and the slack bus, i.e. $\mathcal{V}_G = \mathcal{V}_K \bigcup \mathcal{V}_S$. In a similar way, the bus node type includes all generator and load buses. Bus type nodes are not used in any of the applications, as this would reduce the model to a conventional graph network without types, but it is used to describe some general characteristics of both applications.

It should be noted that all substations of the transmission grid must be included in one and only one of the following node types: PQ, PV and *slack*, or alternatively in load and generator node types. All the transformers and transmission lines in the transmission grid must be contained in the branch node types.

3.2.3 Input features

The input features ($\mathbf{x}$) for each node type differ; for substation nodes they can include information about the voltage, power injections at the substation, and functions of these values; for branch nodes they can include branch and transformer characteristics, and functions of these values:

$$\mathbf{x}_{\mathcal{V}_B} \subseteq \{V_n,\ \theta_n,\ Pd_n,\ Qd_n,\ Pg_n,\ Qg_n,\ Pg_n^{max},\ Pg_n^{min},\ Q_n^{max},\ Q_n^{min}\} \tag{3.1}$$
$$\mathbf{x}_{\mathcal{V}_E} \subseteq \{R_e,\ X_e,\ B_e,\ \tau_e,\ \omega_e\} \tag{3.2}$$

Each node type will have a fixed number of input features f, i.e. all the input features for a χ type node is defined by:

$$\mathbf{x} \in \mathbb{R}^{N_\chi \times f_\chi} \tag{3.3}$$

The specific input features will depend on the application of the model, which in this thesis may be the power flow (chapter 4) and an optimal power flow (chapter 5) analyses. In any case, the input features depend on external injections, established operation limits and an initial state of the inference variables.

3.2.4 Encode step

As within the typed graph network paradigm described in subsection 2.2.3, the input features of every node of all types are embedded into a hidden dimension space through an encoding function γ; an arbitrary χ type node is embedded into a d_χ-dimensional hidden space:

$$\gamma_\chi : \mathbf{x}_\chi \in \mathbb{R}^{N_\chi \times f_\chi} \to \mathbf{z}_\chi \in \mathbb{R}^{N_\chi \times d_\chi} \tag{3.4}$$

It is in this embedded space that the internal operations of the TGN layer are performed, and according to the configuration of the electrical grid.

3.2.5 Adjacency matrices

The topology of the TGN-based model is determined by the transmission grid configuration, i.e. by the way the buses are connected via the transmission lines. The information about the configuration of the grid is obtained from the indices of the sending and receiving buses of each transmission branch. Considering that the substations are assigned to N number of bus type nodes, the topology of the grid is captured through N adjacency matrices:

$$\mathbb{A}_{v_1, v_e} \in \mathbb{R}^{N_{v_1} \times N_E} \tag{3.5}$$

$$\vdots$$

$$\mathbb{A}_{v_N, v_e} \in \mathbb{R}^{N_{v_N} \times N_E} \tag{3.6}$$

An adjacency matrix $\mathbb{A}_{\chi, e}$ defines the connectivity between type χ nodes and branch nodes, i.e. it defines to which branch is each type χ node connected. In this thesis, only cases in which all buses are connected to the rest of the grid through at least one transmission line are considered (multiple islands are not considered). These adjacency matrices will determine the way the messages are shared within the TGN layer. The same adjacency matrices are used for every layer of the TGN based solver since the configuration of the power grid is not altered between layers.

3.2.6 Message-passing, aggregation and update step

A fixed number T of message passing and state update steps is set. During the message-passing portion of the calculations, every node will receive information from all their adjacent nodes. In the case of modeling the transmission grid, all adjacent nodes are of a different type. Bus type nodes are only adjacent to branch type nodes, and vice versa. For this, the message-passing functions μ project the messages from the sender node embedding space to the receiver node embedding space (as mentioned in subsection 2.2.3). Thus, $2N$ message-passing functions are defined, one for sending information from each type of bus node to adjacent branch nodes, and one for sending information from branch nodes to each of the types of bus nodes. The message-passing functions of a step t are defined as:

$$\mu_{v_1 \to v_e}(\mathbf{z}_{v_1}^{(t)}) : \mathbb{R}^{N_{v_1} \times d_{v_1}} \to \mathbb{R}^{N_{v_1} \times d_{v_e}} \tag{3.7}$$

$$\vdots$$

$$\mu_{v_N \to v_e}(\mathbf{z}_{v_N}^{(t)}) : \mathbb{R}^{N_{v_N} \times d_{v_N}} \to \mathbb{R}^{N_{v_N} \times d_{v_e}} \tag{3.8}$$

$$\mu_{v_e \to v_1}(\mathbf{z}_{v_e}^{(t)}) : \mathbb{R}^{N_e \times d_{v_e}} \to \mathbb{R}^{N_e \times d_{v_1}} \tag{3.9}$$

$$\vdots$$

$$\mu_{v_e \to v_N}(\mathbf{z}_{v_e}^{(t)}) : \mathbb{R}^{N_e \times d_{v_e}} \to \mathbb{R}^{N_e \times d_{v_N}} \tag{3.10}$$

As described in subsection 2.2.3, at the final update step, each node type has shared information with neighboring nodes $T-$hops away.

In this case, the aggregation function is computed as matrix multiplications with the corresponding adjacency matrix. Once the propagation and aggregation steps have been performed, an update function σ defined for every type of node takes as input their current embedded state, and the aggregated messages received from adjacent nodes. The message-passing, aggregation and update iteration at step $t+1$ of all bus type nodes and of branch type nodes is illustrated in the following equations:

$$\mathbf{z}_{v_1}^{(t+1)} = \sigma_{v_1}\left(\mathbf{z}_{v_1}^{(t)},\ \mathbb{A}_{v_1,v_e} \cdot \mu_{v_e \to v_1}(\mathbf{z}_{v_e}^{(t)})\right) \tag{3.11}$$

$$\vdots$$

$$\mathbf{z}_{v_N}^{(t+1)} = \sigma_{v_N}\left(\mathbf{z}_{v_N}^{(t)},\ \mathbb{A}_{v_N,v_e} \cdot \mu_{v_e \to v_N}(\mathbf{z}_{v_e}^{(t)})\right) \tag{3.12}$$

$$\mathbf{z}_{v_e}^{(t+1)} = \sigma_{v_e}\left(\mathbf{z}_{v_e}^{(t)},\ \mathbb{A}_{v_1,v_e}^{T} \cdot \mu_{v_1 \to v_e}(\mathbf{z}_{v_1}^{(t)}),\ \ldots,\ \mathbb{A}_{v_N,v_e}^{T} \cdot \mu_{v_N \to v_e}(\mathbf{z}_{v_N}^{(t)})\right) \tag{3.13}$$

The message passing and update functions are the same for all nodes of the same type in the same layer, supporting the concept of combinatorial generalization. This way, the same TGN architecture can operate with input graphs of different sizes and shapes.

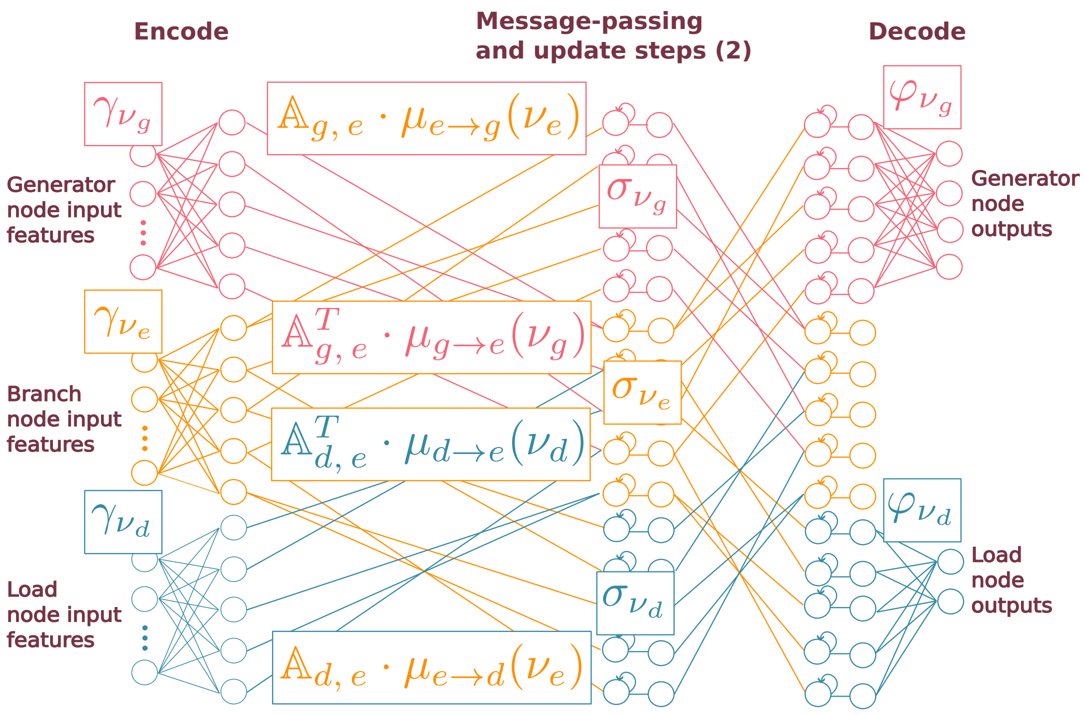
3.2.7 Decode step

After T propagation, aggregation and update steps, the final embeded state of the desired nodes is decoded to obtain the output of the TGN layer. The nodes that will be decoded depend on the application. The decoding functions will map the final embedding state into the corresponding output dimension g, for each node type, so that a node of arbitrary type χ:

$$\varphi_\chi : \mathbf{z}_\chi^{(T)} \in \mathbb{R}^{N_\chi \times d_\chi} \to \mathbf{y}_\chi \in \mathbb{R}^{N_\chi \times o_\chi} \tag{3.14}$$

The combination of encoding, message-passing, updating and decoding steps constitute a single TGN layer. An example scheme of a TGN layer, showing the four TGN layer functions, for three types of nodes (generator, load and branch) and two message passing and embedded update steps is shown in Fig. (3.4).

With the desired nodal outputs, a new transmission grid state inference is made, and this new grid state is either used to calculate the input features of the next layer, or is taken as the final inference if the last TGN layer has been reached.



Encode
Message-passing and update steps (2)
Decode
γ_{ν_g}
Generator node input features
γ_{ν_e}
Branch node input features
γ_{ν_d}
Load node input features
$\mathbb{A}_{g,e} \cdot \mu_{e \to g}(\nu_e)$
$\mathbb{A}^T_{g,e} \cdot \mu_{g \to e}(\nu_g)$
$\mathbb{A}^T_{d,e} \cdot \mu_{d \to e}(\nu_d)$
$\mathbb{A}_{d,e} \cdot \mu_{e \to d}(\nu_e)$
σ_{ν_g}
σ_{ν_e}
σ_{ν_d}
φ_{ν_g}
Generator node outputs
φ_{ν_d}
Load node outputs

FIGURE 3.4: Example TGN layer with two message-passing and update steps.

3.3 Complexity Analysis

By basing the proposed solver on TGNs, it scales well to larger power grids, since the amount of parameters to be trained does not depend on the size of the grid but on the dimension of the input features, the embedding dimension and output dimension of each type of node.

Each TGN layer is composed of four main functions: encoding, message passing, updating and decoding. The four functions are defined by small MLPs, with either one or two layers. The message passing and update MLPs have two layers, one with a nonlinear activation function (either hyperbolic tangent or leaky rectified linear unit) and a linear layer. The encoding and decoding MLPs have a single linear layer, and different instances of each are defined for each type of node. The first layer of the message passing MLPs if of size $d_{IN} \times d_{IN}$, and the second of size $d_{IN} \times d_{OUT}$. When a type χ node casts information unto a type κ node ($\mathcal{V}_\chi \to \mathcal{V}_\kappa$), d_{IN} and d_{OUT} represent the embedding size of the type χ and type κ nodes, respectively. The first layer of the update functions for a type χ node is of size $(|\mathcal{N}(\chi)|+1)d_\chi \times d_\chi$, and the other of size $d_\chi \times d_\chi$; where $|\mathcal{N}(\chi)|$ is the number of neighboring node types. The decoding functions consist of a single layer of size $d_\chi \times o_\chi$. Thus, the total number of trainable parameters of each TGN layer is independent of the size of the grid.

The operations necessary for a node type χ are :

$$\text{Encode}: [N_\chi \times f_\chi] \cdot [f_\chi \times d_\chi] + d_\chi \tag{3.15}$$
$$\text{Message 1}: [N_\kappa \times d_\kappa] \cdot [d_\kappa \times d_\chi] + d_\chi \quad \forall \kappa \in \mathcal{N}(\chi) \tag{3.16}$$
$$\text{Message 2}: [N_\kappa \times d_\chi] \cdot [d_\chi \times d_\chi] + d_\chi \quad \forall \kappa \in \mathcal{N}(\chi) \tag{3.17}$$
$$\text{Aggregation}: [N_\chi \times N_\kappa] \cdot [N_\kappa \times d_\chi] \quad \forall \kappa \in \mathcal{N}(\chi) \tag{3.18}$$
$$\text{Update 1: } [N_\chi \times (\sum_{\kappa \in \mathcal{N}(\chi)} 1) d_\chi] \cdot [(\sum_{\kappa \in \mathcal{N}(\chi)} 1) d_\chi, d_\chi] + d_\chi \tag{3.19}$$
$$\text{Update 2}: [N_\chi \times d_\chi] \cdot [d_\chi \times d_\chi] + d_\chi \tag{3.20}$$
$$\text{Output}: [N_\chi \times d_\chi] \cdot [d_\chi \times o_\chi] + o_\chi \tag{3.21}$$

With N_x representing the cardinality of a type x node set, the encoding function for a type x node has complexity $\mathcal{O}(f_x N_x d_x)$, and since f_x and d_x are predefined constants, this translates to a $\mathcal{O}(N_x)$ complexity. Similarly, each decoding and message passing function has complexity $\mathcal{O}(N_x)$. The update function includes an aggregation procedure which involves the multiplication of sparse adjacency matrices, with dense matrices that represent messages passed from one type of node to another. The total amount of values in all sparse matrices is $2N_E$, the complexity of the aggregation multiplications for a type x node is at most $\mathcal{O}(2N_E d_x)$. The total amount of nodes $N = N_K + N_D + N_S + N_E$ depends on the particular case of electrical grid, but as all operations have at most $\mathcal{O}(DN) = \mathcal{O}(N)$ complexity (with D being some constant dependant on the chosen hyperparameters), the time complexity of the proposed solver is linear with respect to the size of the electrical grid.

3.4 Physics informed and modular nature

The training of the proposed model is carried out in an unsupervised manner, applying physics-informed neural networks by incorporating information of the physical system in the loss function and aiming to minimize the violation of the physical laws and constraints that govern the system, thus eliminating the time consuming need of solving the training cases beforehand with other solvers to produce targets.

Additionally, since the model is based on graph networks which are inherently modular in nature, different node types can be connected or added in any desired configuration. This way, the different node types represent different types of elements in the transmission grid, thus comprehensively performing subtasks relevant to the overall task (transmission grid modeling). This way, it is possible to analyze different electrical grid configurations; adding, modifying or removing elements will simply change the adjacency matrices and the number of nodes of a certain type.

3.5 Power grid considerations and data

The proposed model takes information from different data structures that describe the transmission system, curated in the Matpower data format, these are listed below:

- Bus data structure:
  - Bus ID : identifier for the substation.
  - *Pd* and *Qd* : Real (MW) and reactive (MVAr) power demand, respectively.

  - Gs and Bs : Bus shunt conductance (MW demanded at V = 1.0 p.u.) and susceptance (MVAr injected at V = 1.0 p.u.).
  - V and θ : Voltage magnitude(p.u.) and phase (radians), respectively.
  - V^{max} and V^{min} : Maximum and minimum voltage magnitude (p.u.), respectively.

- Generator data structure:
  - Bus ID : identifier for the generating substation.
  - Pg and Qg : Real (MW) and reactive (MVAr) power generation, respectively.
  - Qg^{max} and Qg^{min} : Maximum and minimum reactive power output (MVAr), respectively.
  - Pg^{max} and Pg^{min} : Maximum and minimum active power output (MW), respectively.

- Branch data structure:
  - "to" and "from" bus IDs : the transmission lines are identified by the receiving and sending bus IDs.
  - R, X, B : Resistance (p.u.), reactance (p.u.), and total line charging susceptance (p.u.), respectively.
  - τ : Transformer off nominal turns ratio.
  - ω : Transformer phase shift angle (radians).

The transformer off nominal turns ratio is simply zero for transmission lines, for the transformers it represents the number of winding turns at the bus on the sending side, divided by the number of winding turns at the receiving side. Disregarding resistance and reactance, i.e. $R = X = 0$, then $\tau = \frac{V_{from}}{V_{to}}$.

As was mentioned in section 2.1.1, the branch series impedance is given by the complex value: $Z_e = R_e + iX_e = \xi\angle\phi$. However, making reference to the conventional admittance matrix, the branches admittance magnitude and phase are chosen as features in the following application chapters. The admittance is given by: $Y_e = \frac{1}{Z_e} = \frac{1}{\xi_e}\angle - \phi_e = \rho_e\angle\delta_e$; defining the magnitude and phase of the series admittance of each branch as ρ and δ, respectively. Thus, two of the branch node features are defined as:

$$\rho_{\nu_e} = \frac{1}{\sqrt{R_{\nu_e}^2 + X_{\nu_e}^2}}, \quad \forall \nu_e \in \mathcal{V}_E \tag{3.22}$$

$$\delta_{\nu_e} = -\arctan\left(\frac{X_{\nu_e}}{R_{\nu_e}}\right), \quad \forall \nu_e \in \mathcal{V}_E \tag{3.23}$$

As was previously mentioned, some considerations must be made regarding the types of power grids that can be analyzed with the proposed model. First of all, isolated buses (or multiple islands) are not considered. Second of all, the sum of all types of nodes must cover all the buses, transformers and branches in the transmission grid, and the sets of these node types do not overlap i.e. $\mathcal{V}^* = \mathcal{V}_G \cup \mathcal{V}_D \cup \mathcal{V}_E$ and $\mathcal{V}_\chi \cap \mathcal{V}_\kappa = \emptyset$, $\forall \chi \neq \kappa$, where $\mathcal{V}^*$ is the set of all buses, transformers and branches. To bound the particular problem that is to be solved with the proposed model, certain considerations were taken and applied through a preprocessing step that includes:

- For the bus data:
  - It is considered that in this case all buses are in the same load zone and have the same base unit (kV).
  - Area number is not considered.
  - Unless time series data is used, the total active and reactive power load must not be so different from the total active and reactive generation (details can be found in section 5.2.3).
- For the generation data:
  - All generators must have the same base unit (in MW and MVAr for active and reactive power, respectively).
  - The area participation factor is not considered.
  - In grids with many reactive generators, it should be made certain that there is enough reactive load to consume it.
- For transmission line data:
  - Repeated lines are eliminated (same "to" and "from" buses).
  - All lines are activated (no branch in the system is deactivated, except the one that is randomly chosen to add topology changes in every sample).
  - Ensure long term, short term and emergency ratings are all set to zero (this constraint is not being considered in the current problem solution).
  - All branches should have valid resistance (non-negative) and reactance values (non-zero).

After the preprocessing step, for each application, noise is added to the original case values to obtain the training dataset. All values that are considered in p.u. are varied between 90% and 110% of the original value, this includes voltage magnitudes and the transmission branch characteristics. The noise added to the power loads vary between 50% and 150% from the nominal values, which allows for load demands that may become congested and are harder computationally than the original case. For generator buses, the active power generation and voltage magnitude nominal values are initialized to a value between 25% and 75% of the total range. All of these noise ranges were chosen to allow for diverse enough values to increase the generalization ability of the model, while trying to avoid ill-conditioned power system cases.

Chapter 4

Application to Power Flow

4.1 Introduction

As was mentioned in the introduction of this thesis, contemporary power grids are being challenged by various changes to their infrastructure, especially by the large-scale deployment of renewable energies, electric vehicles, and demand response programs. In this context, frequent monitoring of the operating conditions of the transmission grid becomes increasingly important. Existing conventional power system state estimation methods are computationally expensive or yield sub-optimal or infeasible results.

With the emergence of artificial intelligence, many systems such as decision trees, neural networks and fuzzy logic methods have been applied to power system problems (Vankayala and Rao, 1993; Lopez-Garcia, Coronado-Mendoza, and Domínguez-Navarro, 2020). Amongst these approaches, NNs have shown promise to a certain extent, due to their ability to synthesize complex mappings accurately and rapidly, along with the possibility to continuously learn. However, most NNs used for power system modelling, such as in Hu et al. (2021) and Fikri et al. (2018), implement MLPs. MLPs usually suffer from local minima, over-fitting issues and lack of interpretability, in addition to not scaling well to larger power grids and not being able to be used for grids of different size or configuration from the ones they are trained on. A pioneering work by Donon et al. (2020) also applies GNNs, and training is carried out in an unsupervised manner, however the model is quite complicated, needs many layers and does not consider changes in grid topology during training.

The proposed model presents a simple TGN based solver to calculate the AC PF in a way that allows to solve many different scenarios in parallel, considering the continuously changing balance between energy supply and demand, and does this in linear time as opposed to the exponential time needed for conventional methods that solve Jacobian matrices. By using different types of nodes that represent unique elements of the transmission grid, a higher degree of interpretability is achieved. The presented solver can generalize to electrical grids of different sizes and parameters, and it additionally considers changes in the grid topology during training, making it specially adequate for analyzing different scenarios of possible line outages, which is essential for security assessments and other transmission system services.

4.2 Methodology

In this section a description of how the TGN framework is applied to solve the PF problem is presented. The goal of the proposed PF solver is to infer all of the bus voltage phase values, and the voltage magnitude values in the load buses. It is shown that by training four small MLPs, the PF problem can be solved in linear time, robust to changes in topology (in particular to single branch line outages) and different branch characteristics. The training is not supervised, but physics-informed, and the solver architecture is modular in nature.

4.2.1 PI-TGN based PF solver

For the proposed PI-TGN based PF solver, three types of bus nodes are considered in addition to the branch type nodes. The graph for every TGN layer is defined by: $\mathcal{G} = (\mathcal{V}_S, \mathcal{V}_K, \mathcal{V}_D, \mathcal{V}_E)$, where $\mathcal{V}_S$, $\mathcal{V}_K$, $\mathcal{V}_D$ and $\mathcal{V}_E$ represent slack, PV, PQ, and branch nodes, respectively. The scheme of the TGN based OPF solver proposed in this work is shown in Fig. 4.1. The solid blue squares represent the main parts of each TGN layer. The orange squares constitute either the initialization or the update of the inference variables, including the final predicted values (the orange line indicates that the values are used only for the first layer, and the purple line indicates that the values are the final inference of the last layer). The pink squares represent calculations done outside of the TGN model to obtain the necessary inputs, this includes using the power equilibrium equations to find the active power needed by the slack bus, and the reactive power needed in every generation bus (which is locally compensated). The green square defines the steady state of the electrical grid to be analyzed. In the following subsections the different elements of the scheme are explained in more detail.

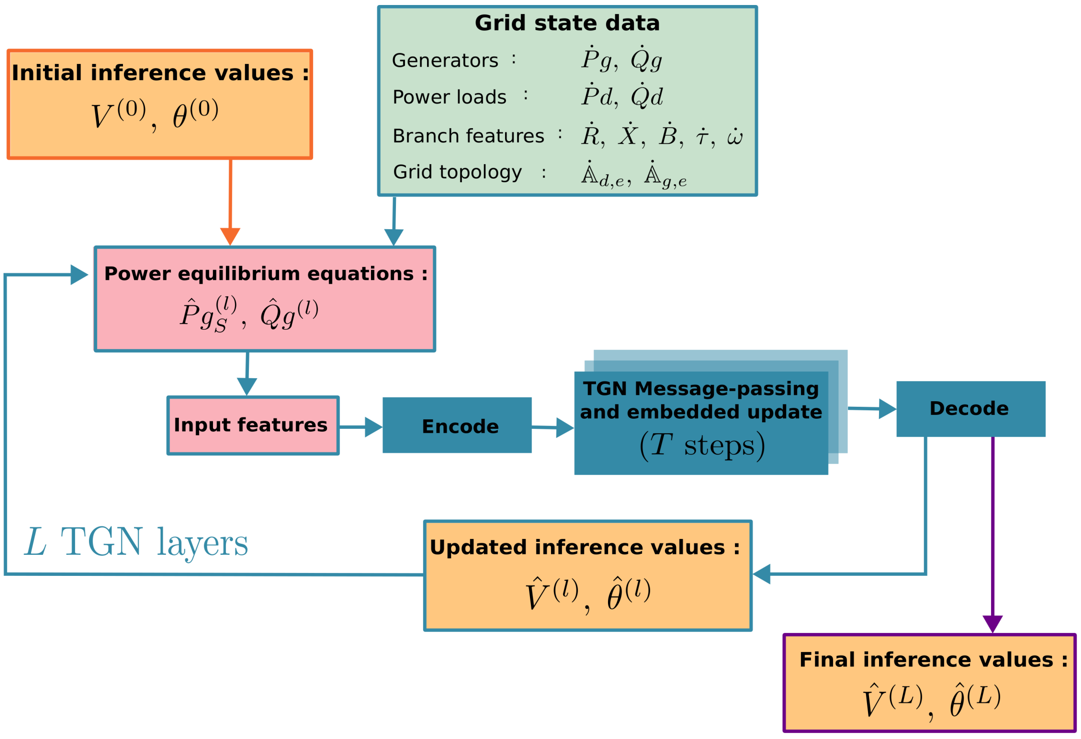

Grid state data
Generators : $\dot{P}g,\ \dot{Q}g$
Power loads : $\dot{P}d,\ \dot{Q}d$
Branch features : $\dot{R},\ \dot{X},\ \dot{B},\ \dot{\tau},\ \dot{\omega}$
Grid topology : $\dot{\mathbb{A}}_{d,e},\ \dot{\mathbb{A}}_{g,e}$
Initial inference values : $V^{(0)},\ \theta^{(0)}$
Power equilibrium equations : $\hat{P}g_S^{(l)},\ \hat{Q}g^{(l)}$
Input features
Encode
TGN Message-passing and embedded update (T steps)
Decode
L TGN layers
Updated inference values : $\hat{V}^{(l)},\ \hat{\theta}^{(l)}$
Final inference values : $\hat{V}^{(L)},\ \hat{\theta}^{(L)}$

FIGURE 4.1: PI-TGN based PF solver scheme.

4.2.2 Grid state data and initial inference values

The data for all experiments is based on benchmark IEEE test cases, similar to the default test cases available for Matpower (Zimmerman, Murillo-Sanchez, and Thomas, 2011); perturbations are added to the injections, branch characteristics and grid topology for each sample. The information of the IEEE test cases is imported through three 2-dimensional arrays, with information for the buses, generators, and branches.

To generate a batch of data for each training or inference iteration, two steps are taken: (1) adding random uniform noise to certain elements of the benchmark data structures or loading a pre-defined time series, and (2) setting initial values for the voltage magnitude and phase values.

The following list presents the values to which noise is added, and how it is applied. As mentioned in section 3.5, all of noise ranges were chosen to allow for diverse enough values to increase the generalization ability of the model, while trying to avoid ill-conditioned power system cases. In the following, the prime symbol is used to indicate original benchmark case values.

- The active and reactive power load of every bus is uniformly distributed between 50 and 150% of the original benchmark case value:

$$\dot{P}d_n = \mathrm{U}_P \cdot Pd'_n, \quad n = 1, \dots, N_B \tag{4.1}$$
$$\dot{Q}d_n = \mathrm{U}_Q \cdot Qd'_n, \quad n = 1, \dots, N_B \tag{4.2}$$
$$\mathrm{U}_P,\, \mathrm{U}_Q \sim \mathcal{U}(0.5, 1.5)$$

  It should be noted that independent noise values are applied to the active and reactive power loads, and that it is only applied when time series data is not available. The power load is limited in a fixed way to prevent the rare occurrence of the total load demand exceeding the total maximum power generation capacity.

- Uniformly distributed noise is added to the voltage magnitude of PV buses, such that the resulting values are between 90% and 110% of the nominal value:

$$\dot{V}_k = \mathrm{U}_V \cdot V'_k, \quad k = 1, \dots, N_K \tag{4.3}$$
$$\mathrm{U}_V \sim \mathcal{U}(0.9, 1.1)$$

- Uniformly distributed noise is also added to the nominal power generation of PV nodes, such that the values are between 25% and 75% of the allowed range :

$$\dot{P}g_k = \mathrm{U}_{Pg} \cdot (Pg_k^{max'} - Pg_k^{min'}) + Pg_k^{min'}, \quad k = 1, \dots, N_K \tag{4.4}$$
$$\mathrm{U}_{Pg} \sim \mathcal{U}(0.25, 0.75)$$

- For the branch data structure, noise is added to the branch resistance, reactance and susceptance, as well as to the transformer tap ratio values. The noise values are generated independently, but in the same manner:

$$\dot{R}_e = \mathrm{U}_R \cdot R'_e, \quad e = 1, \dots, N_E \tag{4.5}$$
$$\dot{X}_e = \mathrm{U}_X \cdot X'_e, \quad e = 1, \dots, N_E \tag{4.6}$$
$$\dot{B}_e = \mathrm{U}_B \cdot B'_e, \quad e = 1, \dots, N_E \tag{4.7}$$
$$\dot{\tau}_e = \mathrm{U}_\tau \cdot \tau'_e, \quad e = 1, \dots, N_E \tag{4.8}$$
$$\mathrm{U}_R,\, \mathrm{U}_X,\, \mathrm{U}_B,\, \mathrm{U}_\tau \sim \mathcal{U}(0.9, 1.1)$$

- After noise has been added to the injections and transmission line characteristics, the topology of the grid is changed by randomly deleting a transmission line for each sample. This way both injection and topology changes are involved during training.

N_B, N_K, and N_E represent the total number of buses, PV buses and branches, respectively.

This way, a group of objects that represent a batch of electrical grid samples is formed. Said representation is structured so that it is ready to be used to calculate adjacency matrices and the inputs to the proposed PI-TGN based model. It should be noted that while quantities such as branch impedance do not frequently change in a single grid, to train the proposed model with grid samples with varying topology and differing branch characteristics gives it the opportunity to understand how these quantities affect the resulting PFs, thus improving the generalization across different grids.

All voltage magnitude values from PQ buses are initialized to 1 p.u. (per unit) and all voltage phase values are set to the slack bus angle reference.

The inference variables, which are the voltage phase for both PV and PQ buses, and the voltage magnitude for PQ buses are initialized as an initial 'flat' guess. The voltage magnitude of PQ buses are intialized to 1 p.u., and all voltage phase values are set to the slack bus angle reference, as shown below:

$$V_d^{(0)} = 1.0\,(\text{p.u.}), \quad d = 1, \ldots, N_D \tag{4.9}$$

$$\theta_n^{(0)} = \theta_s', \quad n = 1, \ldots, N_B \tag{4.10}$$

where N_D represents the number of PQ buses.

The voltage magnitude of PV buses is known beforehand and thus does not need to be initialized or updated at any point in the proposed model.

4.2.3 Input features

For all TGN layers except the first one, the input features are calculated from the grid state data (power loads and generation, branch features and grid topology), and the updated inferred voltage magnitude and phase states from the previous TGN layer. For the first TGN layer, the input features depend on the grid state data and the initialization of the inference variables.

The suitability of the TGN framework is highlighted when choosing the input features for the different bus nodes and branch nodes, as they must be treated differently. PV and PQ nodes are expected to produce an output that will result in voltage values such that the power equilibrium in each bus is obtained, so the inputs for these node types are set to be the voltage values and the power equilibrium error per bus. The power equilibrium error is calculated as shown in section 2.1.1 from the grid state data, and the updated voltage and power generation values. Both the active and reactive power of the slack bus, and the reactive power of PV buses are locally compensated, meaning that $\tilde{P}_{v_s}, \tilde{Q}_{v_s} = \mathbf{0}$, and $\tilde{Q}_{v_k} = \mathbf{0}$. For this reason it does not make sense to add the reactive power imbalance as an input feature to PV type nodes (they will always be zero and not provide additional information), instead the generated reactive power is used as an input feature.

A similar situation happens with the slack node where both active and reactive powers are compensated. The voltage values for the slack bus are known beforehand, the generated power at the slack bus is calculated as a byproduct after the voltage values of the other buses has been found in order to compensate for total power generation and load imbalance. The TGN layer will not produce an output for slack type nodes, as the calculation of the locally compensated generation only involves simple matrix sums; thus, the input features will only serve to communicate information about the current generation and voltage state of the slack bus to the neighboring branches and buses. Branch type nodes will also not produce an output, their input features remain unchanged throughout all TGN layers, but will serve to include information about the branch characteristics in the total model. The input features of each type of node for a TGN layer l are described in the following equations.

$$\mathbf{x}_{\nu_k}^{(l)} = \{\hat{V}_{\nu_k}^{(l-1)},\ \hat{\theta}_{\nu_k}^{(l-1)},\ \tilde{P}_{\nu_k}^{(l-1)},\ \hat{Q}g_{\nu_k}^{(l-1)}\}, \quad \forall \nu_k \in \mathcal{V}_K \tag{4.11}$$

$$\mathbf{x}_{\nu_d}^{(l)} = \{\hat{V}_{\nu_d}^{(l-1)},\ \hat{\theta}_{\nu_d}^{(l-1)},\ \tilde{P}_{\nu_d}^{(l-1)},\ \tilde{Q}_{\nu_d}^{(l-1)}\}, \quad \forall \nu_d \in \mathcal{V}_D \tag{4.12}$$

$$\mathbf{x}_{\nu_e}^{(l)} = \{\rho_{\nu_e},\ \delta_{\nu_e},\ \dot{B}_{\nu_e},\ \dot{\tau}_{\nu_e}\}, \quad \forall \nu_e \in \mathcal{V}_E \tag{4.13}$$

$$\mathbf{x}_{\nu_s}^{(l)} = \{V_{\nu_s},\ \theta_{\nu_s},\ \hat{P}g_{\nu_s}^{(l-1)},\ \hat{Q}g_{\nu_s}^{(l-1)}\} \tag{4.14}$$

where ρ and δ represent the branch admittance magnitude and phase, respectively, which are defined in subsection 3.5.

4.2.4 TGN model

In the proposed method, a predefined number L of TGN layers is used to iteratively approximate the missing voltage values at every node of the power grid. For this PI-TGN application, a fixed number of independent layers sequentially calculate an approximation of the final inference values. While the embedding, message passing, update and output functions of each TGN layer are independently parameterized, the layers are structurally the same. The initial voltage state is used for the first TGN layer, and the following TGN layers receive the voltage approximation of the previous layer; after each approximation, the power balance error ($\tilde{P}$ and $\tilde{Q}$) is calculated at each bus and is used as part of the input features for the next TGN layer.

Considering the allowed configuration of transmission grids, bus type nodes cannot be directly connected to each other but rather are always connected to at least one branch type node. Therefore, three adjacency matrices are needed to represent the topology of the TGN layers:

$$\mathbb{A}_{d,e} \in \mathbb{R}^{N_D \times N_E} \tag{4.15}$$

$$\mathbb{A}_{k,e} \in \mathbb{R}^{N_K \times N_E} \tag{4.16}$$

$$\mathbb{A}_{s,e} \in \mathbb{R}^{1 \times N_E} \tag{4.17}$$

The T message-passing and embedded update steps of each layer are represented by:

$$\nu_d^{(t+1)} = \sigma_{\nu_d}\left(\nu_d^{(t)},\ \mathbb{A}_{d,e} \cdot \mu_{e\to d}(\nu_e^{(t)})\right), \quad \forall \nu_d \in \mathcal{V}_D \tag{4.18}$$

$$\nu_k^{(t+1)} = \sigma_{\nu_k}\left(\nu_k^{(t)},\ \mathbb{A}_{k,e} \cdot \mu_{e\to k}(\nu_e^{(t)})\right), \quad \forall \nu_k \in \mathcal{V}_K \tag{4.19}$$

$$\nu_e^{(t+1)} = \sigma_{\nu_e}\left(\nu_e^{(t)},\ \mathbb{A}_{d,e}^T \cdot \mu_{d\to e}(\nu_d^{(t)}),\ \mathbb{A}_{k,e}^T \cdot \mu_{k\to e}(\nu_k^{(t)}), \mathbb{A}_{s,e}^T \cdot \mu_{s\to e}(\nu_s^{(t)})\right) \\ \forall \nu_e \in \mathcal{V}_E \tag{4.20}$$

$$\nu_s^{(t+1)} = \nu_s^{(t)} \tag{4.21}$$

where σ_d, σ_k and σ_e are the update MLPs for PQ, PV and branch nodes, respectively; $\mu_{e\to d}$ and $\mu_{e\to k}$ are the aggregation MLPs from branch type nodes to PQ and PV nodes, respectively; $\mu_{d\to e}$, $\mu_{k\to e}$ and $\mu_{s\to e}$ are the aggregation MLPs from PQ, PV and slack type nodes, respectively, to branch type nodes. As is shown in eq. (4.20), even though the branch type nodes do not produce a layer output, they do aggregate the embedded branch characteristics with information from neighboring nodes, and are thus updated in every TGN layer update step. In contrast, slack type nodes do not produce a layer output and only serve to communicate information from the previous TGN layer slack state to the neighboring nodes, for this reason they are not updated during the TGN layer update step.

As described in chapter 3, there are encoding MLPs defined for every type of node (γ_{ν_d}, γ_{ν_k}, γ_{ν_s} and γ_{ν_e}) that embed the input features into the latent space, and decoding functions for the PQ and PV type nodes (φ_{ν_d} and φ_{ν_k}) to obtain the correct number of outputs for every TGN layer. The PV type node decoding function has one output: the voltage phase change. The PQ type node decoding function has two outputs: voltage magnitude and phase changes. The outputs of the decoder functions of a TGN layer l for both of these types of nodes are shown below:

$$\varphi_{\nu_d}^{(l)} = \{\Delta V_{\nu_d}^{(l)},\ \Delta\theta_{\nu_d}^{(l)}\}, \quad \forall \nu_d \in \mathcal{V}_D \tag{4.22}$$

$$\varphi_{\nu_k}^{(l)} = \{\Delta\theta_{\nu_k}^{(l)}\}, \quad \forall \nu_k \in \mathcal{V}_K \tag{4.23}$$

where ΔV_{ν_χ} and $\Delta\theta_{\nu_\chi}$ represent the voltage magnitude and phase modifications for a χ type bus.

4.2.5 Updated and final inference values

The outputs of the decoders are used to update the missing voltage values of every bus, and generate the TGN layer output, as shown below:

$$\hat{V}_{\nu_d}^{(l)} = \hat{V}_{\nu_d}^{(l-1)} + \Delta V_{\nu_d}^{(l)} \tag{4.24}$$

$$\hat{\theta}_{\nu_d}^{(l)} = \hat{\theta}_{\nu_d}^{(l-1)} + \Delta\theta_{\nu_d}^{(l)} \\ \forall \nu_d \in \mathcal{V}_D \tag{4.25}$$

$$\hat{\theta}_{\nu_k}^{(l)} = \hat{\theta}_{\nu_k}^{(l-1)} + \Delta\theta_{\nu_k}^{(l)} \\ \forall \nu_k \in \mathcal{V}_K \tag{4.26}$$

Thus, for a layer l, eq.(4.24) represents the voltage magnitude update for PQ nodes; eqs. (4.25) and (4.26) represent the voltage phase output for PQ and PV nodes, respectively.

With the updated voltage values, eqs. (2.3) and (2.4) are used to calculate the power balance error in each bus, the reactive power compensation in generator buses, and both active and reactive power compensation in the slack bus. If the last layer has not been reached, the voltage values and power balance error values are used to determine the graph input for the next TGN layer, otherwise, if the model is training, they are used to evaluate the cost function. If the model is not being trained, the process ends with the final voltage inference values.

4.2.6 Loss function

The goal of the PF problem is to find a set of voltage magnitudes and phase angles that satisfies the power flow equations at every node. To achieve power balance, the loss function used to train the PI-TGN PF solver seeks to minimize the power imbalance at every PV and PQ type node. The loss function is given by:

$$\mathcal{L} = \frac{1}{H} \sum_{h=1}^{H} \left(\frac{1}{N_B} \sum_{n=1}^{N_B} (\tilde{P}_{n,h}^2 + \tilde{Q}_{n,h}^2) \right) \tag{4.27}$$

where N_B represents the total number of buses, and H the total number of samples (n and h being the bus and sample indices, respectively).

This way the learning process is unsupervised and the objective of the loss function is to ensure that Kirchhoff's current law is enforced. The loss function only considers the final voltage inference (not the hidden approximations) and the resulting power balance error.

4.2.7 Training

The training process is described in algorithm 3. The proposed TGN solver is trained in batches, the independent encoding, message-passing, update and decoding NNs of the TGN layers are trained simultaneously. When the model is being trained, the final voltage inference is used with the power equilibrium equations to calculate the loss function described by eq. (4.27).

The backpropagation algorithm is used to calculate the gradients of the loss function with respect to the trainable parameters of the NN functions that make up each TGN layer. These gradients are used to modify the values of the NN parameters via the Adam optimization algorithm, afterwards a new batch of data is introduced to the TGN model and the process is repeated until the cost function converges to a minimum value.

4.3 Experiments

In this section, the predictive accuracy of the proposed PI-TGN based PF solver is evaluated by comparing it to the results obtained with the Matpower PF solver based on the Newton-Raphson method.

Algorithm 3 Training algorithm for PI-TGN based PF solver

```
Require: Bus, generator and branch data
  while $i < i_{max}$ do                                          ▷ Training iterations
    $\dot{P}d$, $\dot{Q}d \leftarrow$ eqs.(4.1) & (4.2)                          ▷ Bus noise
    $\dot{V}_{\nu_k}$, $\dot{P}g_{\nu_k} \leftarrow$ eqs.(4.3) & (4.4)                ▷ Noise to PV nodes
    $\dot{R}$, $\dot{X}$, $\dot{B}$, $\dot{\tau} \leftarrow$ eqs.(4.5)-(4.8)           ▷ Transmission line noise
    $from$, $to \leftarrow$ configuration change                ▷ Random branch outage
    $\dot{G}$, $\dot{B}$, $\dot{\omega} \leftarrow$ no noise
    $\hat{V}^{(0)}_{\nu_d}$, $\hat{\theta}^{(0)} \leftarrow$ eqs.(4.9)-(4.10)                ▷ Initialized values
    $\mathbb{A}_{d,e}$, $\mathbb{A}_{k,e}$, $\mathbb{A}_{s,e} \leftarrow$ eqs.(4.15)-(4.17)        ▷ Adjacency matrices
    for $l = 1 \dots L$ do                                          ▷ $L$ TGN layers
      $\hat{Q}g^{(l)}_{\nu_k} \leftarrow$ eq.(2.5)                ▷ PV bus reactive power compensation
      $\hat{P}g^{(l)}_{\nu_s}$, $\hat{Q}g^{(l)}_{\nu_s} \leftarrow$ eqs.(2.6) & (2.7)      ▷ Slack bus power compensation
      $\tilde{P}^{(l)}$, $\tilde{Q}^{(l)} \leftarrow$ eqs.(2.3) & (2.4)                     ▷ Power balance
      $\mathbf{x}^{(l)}_{\nu_k}$, $\mathbf{x}^{(l)}_{\nu_d}$, $\mathbf{x}^{(l)}_{\nu_s}$, $\mathbf{x}^{(l)}_{\nu_e} \leftarrow$ eqs.(4.11)-(4.14)            ▷ Input features
      $\Delta V^{(l)}_{\nu_d}$, $\Delta\theta^{(l)} \leftarrow TGN\_model(\mathbf{x}^{(l)}_{\nu_g}, \mathbf{x}^{(l)}_{\nu_d}, \mathbf{x}^{(l)}_{\nu_d}, \mathbb{A}_{d,e}, \mathbb{A}_{k,e}, \mathbb{A}_{s,e})$   ▷ TGN layer
      $\hat{V}^{(l)}_{\nu_d}$, $\hat{\theta}^{(l)} \leftarrow$ eqs.(4.24)-(4.26)          ▷ Update inference variables
    end for
    $\mathcal{L}(\tilde{P}^{(L)}, \tilde{Q}^{(L)}) \leftarrow$ eq.(4.27)                          ▷ Loss function
    $gradients \leftarrow \nabla_w(\mathcal{L})$            ▷ $loss$ is differentiated against NN weights
    $weights \leftarrow w - \alpha\nabla_w(\mathcal{L})$   ▷ Adam optimizer applies gradients to NN weights
  end while
  Save trainable weights
```

4.3.1 Simulation setup

Three electrical grid sizes are employed based on case 30, case 57 and case 118 standard IEEE power grids. The characteristics of each of the grids are given in Table 4.1.

Test case	**N$_B$**	**N$_G$**	**N$_E$**	**max ΔPd (MW)**	**max ΔQd (MVAr)**	**max ΔPg (MW)**	**max ΔV (KV)**
30_ieee	30	6	40	41.7	14.8	165.1	**10.6**
57_ieee	57	7	79	**184.9**	43.1	300.8	10.1
118_ieee	118	54	185	138.5	**54.9**	**405.6**	10.4

TABLE 4.1: Test case characteristics

The noise added to each of the grids is specified in subsection 4.2.2, however, the real noise applied to each grid varies depending on the distribution and characteristics of the loads and generators on that grid. The columns of Table 4.1 indicated by **max ΔPd (MW)** and **max ΔQd (MVAr)** represent the maximum amount of active and reactive power load noise added to the test case, respectively. The last two columns, indicated by **max ΔPg (MW)** and **max ΔV (KV)** show the maximum amount of noise added to the nominal active power and nominal voltage magnitude of the *PV* nodes, respectively. In all cases, the variation from the original transmission grid is significant, and is additional to the noise introduced to the branch characteristics. The biggest differences from the original cases are shown in bold.

To further explain the inference procedure, Fig. 4.2 shows the evolution of the absolute difference between the final N-R based result and the outputs obtained at distinct TGN iterations of the proposed method: $|y - \hat{y}|$, for a variable y. This is done for a single, random sample of the case 118 grid, showing voltage phase difference in radians (left) and voltage magnitude in p.u. (right). This way, *iter* 0 corresponds to the output of the first TGN layer; *iter* l represents an intermediate layer, in this case $l = 7$; *iter* L represents the final output, with $L = 15$. The abundance of low error vertices on the magnitude side of the first layer is due to the quantity of PV nodes, for which the voltage magnitude is known from the original state of the grid. As the iterations advance, it is shown that the output of each node approaches the N-R output, even though the learning is not supervised and the N-R result is not known during training.

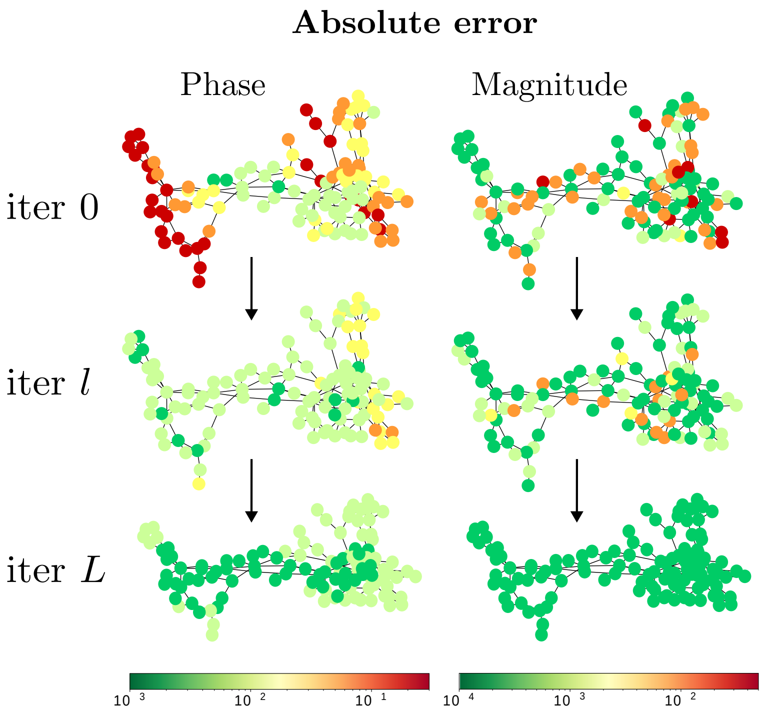


FIGURE 4.2: Evolution of absolute difference between the proposed method outputs and the final N-R based output.

4.3.2 Prediction accuracy with same size grids

For this test, three instances of the proposed solver are generated, each trained on an electrical grid of fixed size. The three instances of the PI-TGN based PF solver share the same hyperparameters:

- Number of TGN layers: $L = 5$.
- Number of message passing and update steps: $T = 2$.
- Embedded dimension of node types: $d = 16$.
- Learning rate: $\alpha = 1 \times 10^{-4}$.

number of TGN layers, number of message passing and update steps, embedded dimension of node types, and learning rate. The voltage data is normalized using the per unit system, and a batch of 20 samples is taken for testing each instance.

The first experiment consists of testing the three instances of the proposed solver on electrical grids of the same size as the ones they are trained on, only adding noise to the power grid injections and disconnecting random branches. The corresponding TGN instance is applied to infer the missing voltage values at each node. To add another point of comparison to the results, the test samples are additionally solved using the DC approximation method, which does not consider the reactive power in the electrical grid and uses linear network equations that relate real power to bus voltage angles (instead of complex bus voltages). The DC approximation is simple and robust, and for these reasons, sometimes used for contingency or real-time dispatch analyses (Van Hertem, 2006). To validate the obtained results, they are compared with the solutions calculated with the trusted and conventional Newton-Raphson (N-R) method using Matpower (Zimmerman, Murillo-Sanchez, and Thomas, 2011). Because of the way the batch samples are generated, some input samples result in non-feasible grid states that don't converge with the N-R method, for these cases the proposed solver does infer some solution, but in the presented results only those that converged with the N-R method are considered.

Table 4.2 reports the average L2 loss (squared error loss) obtained from the difference between the inferred values obtained with the PI-TGN based PF solver, the DC approximation method and the values obtained with the conventional N-R method which are taken as ground truth. Each column shows the average predicted error in percentages: $\frac{1}{N_y}\sum(y-\hat{y})^2 \times 100$, for a variable y with N_y values. The best approximations for each test case are highlighted in bold. As expected, for the voltage magnitude, the proposed PF solver always outperforms the DC approximation method. The proposed solver and the DC approximation method obtain similar results for the voltage phase.

	30 buses		**57 buses**		**118 buses**	
	PI-TGN	**DC**	**PI-TGN**	**DC**	**PI-TGN**	**DC**
V	**0.029**	0.327	**0.169**	0.465	**0.007**	0.311
$\boldsymbol{\theta}$	0.163	**0.137**	**0.146**	0.174	1.07	**0.91**

TABLE 4.2: PF Prediction errors (%)

Fig. (4.3) shows the mean (blue line) and standard deviation (green shaded area) of the L2 loss obtained with the proposed solver for the 30 bus case; the voltage magnitude (left) and voltage phase (right) for each bus. In all cases, the voltage magnitude is always zero at PV buses.

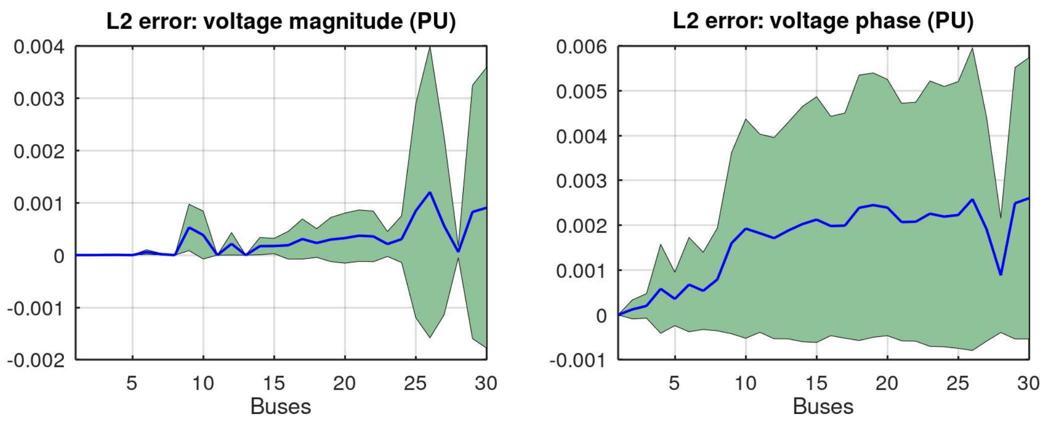


FIGURE 4.3: Mean and SD of PF L2 error (30-bus case).

Fig. (4.4) shows the mean (blue line) and standard deviation (green shaded area) of the L2 loss obtained with the proposed solver for the 57 bus case; the voltage magnitude (left) and voltage phase (right) for each bus.

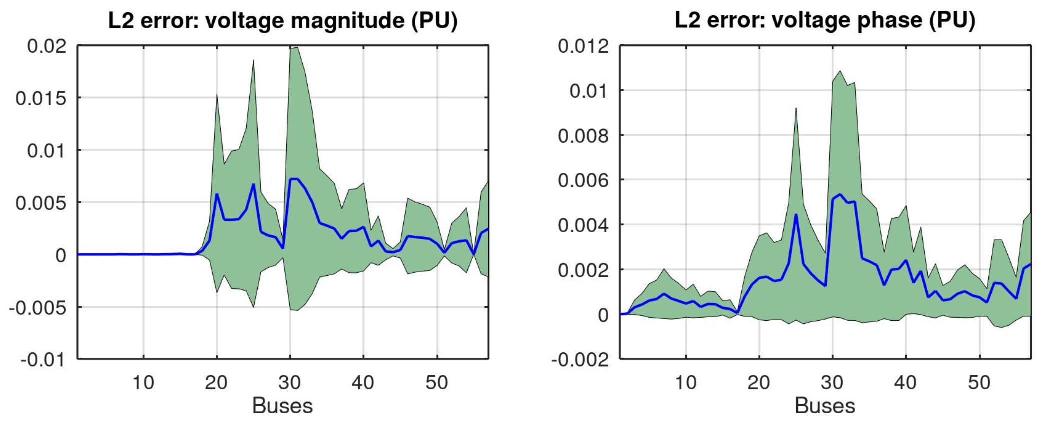


FIGURE 4.4: Mean and SD of PF L2 error (57-bus case).

Fig. (4.5) shows the mean (blue line) and standard deviation (green shaded area) of the L2 loss obtained with the proposed solver for the 118 bus case; the voltage magnitude (left) and voltage phase (right) for each bus.

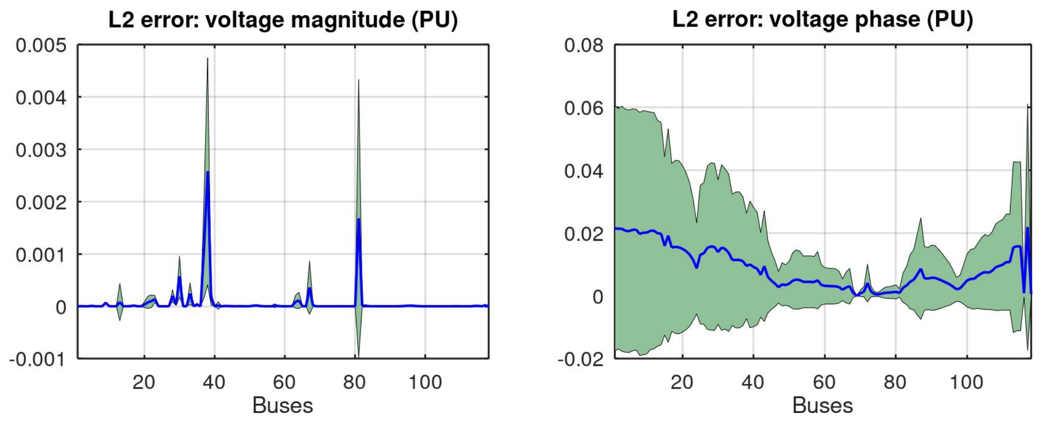


FIGURE 4.5: Mean and SD of PF L2 error (118- bus case).

4.3.3 Prediction accuracy with different size grids

One of the crucial points of the proposed model is the capability to be tested on grids of different size from the ones they are trained on, due to the graph structure representation of the system. To evaluate the generalization performance of the proposed solver when faced with different grid sizes, each of the three TGN instances is tested on batches of the other two grid sizes that do not correspond to the ones they were trained on. Each grid in the test batch is obtained as described in subsection 4.3.2, with varying injections, line characteristics and grid configuration through the elimination of a random branch.

This way three different test cases are produced for each PI-TGN instance, and each of them is additionally solved with the DC approximation method as described in subsection 4.3.2.

Table 4.3 reports the average L2 loss (squared error loss) obtained from the difference between the inferred values obtained with the PI-TGN based PF solver, the DC

	V	θ	V	θ	V	θ
	(30 buses)		(57 buses)		(118 buses)	
PI-TGN 30	**0.029**	0.163	**0.608**	6.764	**0.011**	2.574
DC	0.327	**0.137**	0.825	**2.25**	0.318	**1.89**
PI-TGN 57	**0.055**	2.862	**0.169**	**0.146**	**0.007**	5.347
DC	0.285	**0.629**	0.465	0.174	0.323	**0.927**
PI-TGN 118	**0.117**	0.93	0.604	4.996	**0.007**	1.07
DC	0.261	**0.231**	**0.518**	**0.806**	0.311	**0.91**

TABLE 4.3: PF Prediction errors (%) - Extrapolation case

approximation method and the values obtained with the conventional N-R method which are taken as ground truth. Each column represents a test case for either voltage magnitude or phase, and each row represents an instance of the PI-TGN solver or the DC approximation method. This way, the results obtained with instances trained and tested on the same size grids are the same as those obtained in subsection 4.3.2. The best approximations for each test case are highlighted in bold. For the voltage magnitude, the proposed PF solver usually outperforms the DC approximation method, even when tested on grids of different size from the ones they were trained on. Considering the voltage phase, the DC approximation method usually outperforms the proposed solver. However, the results obtained with the proposed solver are decent, and it should be taken into consideration that they are being tested on grids with significant injection difference and topology from the grids they were trained on.

4.3.4 Time considerations

As was mentioned in subsection 3.3, the time complexity of the model is linear, and thus the calculation time does not increase as sharply as the N-R method with respect to the size of the electrical grid. Additionally, the proposed method is faster and appears to have the tendency to increase the runtime slower than the DC approximation method for increasingly larger transmission systems. To illustrate this, Fig. 4.6 shows the average runtime in seconds for different sized electrical grids for the three methods tested in this work. It should be noted that running n scenarios multiplies the number of inputs by n for the proposed solver, and the scenarios are solved in parallel. With the N-R based approach, the n scenarios are processed sequentially. These considerations show an important reduction in the time needed to carry out numerous PF iterations, which in combination with the robustness to single branch outages and differences in branch characteristics, make it a beneficial tool for power grid planning and risk assessment.

Additionally, the training of all PI-TGN instances for this application was very fast, each instance took less than 1500 training iterations (around 10 minutes) to minimize the loss function.

4.4 Discussion and limitations

In this chapter the potential of the proposed physics-informed TGN model applied to the steady-state PF problem is investigated. The PF problem involves finding all the voltage magnitude, phase and active and reactive power states of all the buses in the transmission system. To find a solution to the steady-state PF problem, power

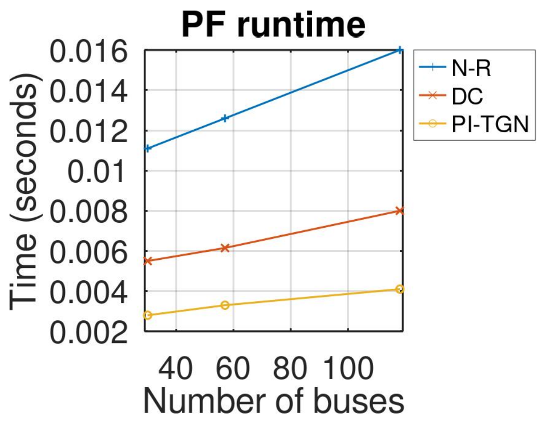


FIGURE 4.6: Comparison of PF runtimes.

equilibrium must be obtained for all buses in the system, which involves solving a set of nonlinear equations given by Kirchhoff's laws.

To make the learning independent of target cases, the loss function used for training depends only on the power imbalance for every bus. The proposed model was tested on a number of different power networks of varying sizes, with a considerable amount of noise added to different variables (including loads, generators, and branch characteristics) and with changing electrical grid configurations in order to evaluate its performance.

In contrast to the usual MLPs found in the literature, which would be completely unable to handle samples coming from different grid topologies, the proposed method does a decent job in generalizing to these cases. In this case, the difference in topology comes, not just from removing a single random branch, but from a difference in the quantity of all node types.

The proposed PF solver has been proven to obtain results very similar to those obtained with the conventional N-R method, especially when tested on grids of the same size as the ones they were trained on, but also has obtained decent results when extrapolated to different grid sizes. Additionally, the time it takes to run the PF problem is less than the time it takes for the DC approximation method.

In summary, the proposed method consists of a PI-TGN based system that abstracts the relationship between the different elements in the electrical grid to solve the steady state power flow problem for dynamical networks, i.e. considering different grid configurations, injections and branch characteristics. The proposed solver is flexible in that it can admit grid elements with different characteristics and learn their relation with the other elements of the grid. Thus, the presented work presents a valuable step towards developing a machine learning based system that is able to assist in analyzing flexible electrical grids of increasing complexity, while improving speed and reducing the computational burden of essential PF analyses.

Chapter 5

Application to Optimal Power Flow

5.1 Introduction

As mentioned in Chapter 1, the OPF problem is similar to the "simple" economic dispatch problem, but includes the additional difficulty of considering the operational limits of the power grid. The OPF problem is a notably nonlinear, non-convex optimization problem. The main objective is to find the best value of the control variables of the transmission grid which minimize the objective function, taking the physical and engineering constraints of the grid into consideration. The control variables can be: the setpoint for the active power of the generators, the setpoints for reactive power compensation devices, the voltage magnitude setpoints, or even tap settings of the transformers. These objective functions can be based on minimizing different parameters, such as: the fuel cost of the generators, the emission rate of the generators, power losses in the transmission network, or the security index of the voltage.

The use of machine learning to solve the AC optimal power flow has been gaining popularity recently to the significant run-time speedup that results from using these methods instead of traditional optimization techniques. However, most works such as the one presented in Owerko, Gama, and Ribeiro (2020) require generating large amounts of target data and do not generalize well outside similar cases to those seen in training. Other works such as those presented in Fioretto, Mak, and Hentenryck (2020) and Nellikkath and Chatzivasileiadis (2022) use semi-supervised methods, which do not require such large amounts o target values. However, these methods do not take advantage of the inherent ability of graph networks to learn structured data representations, which add accuracy and efficiency to the neural network based solvers. The use of typed graph networks presented in this work also adds interpretability to the solutions, which is an important aspect to gain confidence from operators in order to begin to applying these models.

5.2 Methodology

This section explains the layout of the proposed PI-TGN based OPF solver. It starts out explaining the elements of the loss function as one of the biggest challenges in solving the OPF problem with machine learning lies in complying with the presented constraints. This way, the chosen loss function greatly influences the structure of the model, which is described afterwards.

5.2.1 Loss function

As was mentioned in chapter 2, the optimal power flow problem can have different optimization objectives, however whatever the objective, it is constrained by physical laws governing the system and the operational constraints of the different elements that make up the electrical network. In this work, the objective to minimize is the power losses in the transmission branches.

To incorporate the different requirements of the OPF problem, the loss function used for training is multi-target and composed of three elements: one that depends on the active power line losses, a second that depends on the active and reactive power imbalance at each bus, and a final component that depends on the operation constraints of voltage magnitude on all buses, and the power generation.

The total loss of all branches in each sample is calculated using the difference between sent power and received power for each branch, which depends on the inferred voltage magnitude and phase values, and on the branch characteristics, as shown below:

$$J_{loss,h} = \sum_{e=1}^{N_E} ||Pf_{e,h}\left(\hat{\Gamma}_e, \dot{\varpi}_e\right)| - |Pt_{e,h}\left(\hat{\Gamma}_e, \dot{\varpi}_e\right)||, \quad h = 1, \ldots, H \tag{5.1}$$

where inferred values are marked with a hat ($\hat{}$), while given values are marked with a dot ($\dot{}$); Pf and Pt are defined as in eq. (2.13), they represent the power flowing from the sending bus and to the sending bus of each branch, respectively. These values are determined by the characteristics of each branch and the voltage magnitude and phase values at their extremities ("from" and "to" buses); the branch characteristics are summarized in the single variable $\varpi = (\rho_e, \delta_e, B_e, \tau_e, \omega_e)$; the voltage magnitude and phase values of the buses on each extremity of the branch are summarized in the variable $\Gamma_e = (Vf_e, \theta f_e, Vt_e, \theta t_e)$.

Thus, to infer voltage magnitude and phase values that will minimize the branch losses, the first part of the loss function is simply the mean square deviation from zero loss of each sample, as presented below:

$$\mathcal{L}_a = \lambda_a \frac{1}{H} \sum_{h=1}^{H} J_{loss,h}^2 \tag{5.2}$$

The second part of the cost function is meant to enforce the equality constraint that enforces the fulfillment of the Kirchhoff laws, it is defined by the mean square deviation from the active and reactive power equilibrium point at each bus node:

$$\mathcal{L}_b = \lambda_b \frac{1}{H} \sum_{h=1}^{H} \left(\frac{1}{N_B} \sum_{n=1}^{N_B} \left(\tilde{P}_{n,h}^2 + \tilde{Q}_{n,h}^2\right) \right) \tag{5.3}$$

where H represents the batch size and each sample is indexed with h; N_B is the number of substations in the electrical grid, as defined in chapter 2; the active and reactive power imbalances, $\tilde{P}$ and $\tilde{Q}$, are defined similarly to eqs. (2.3) and (2.4), but in this case the generated power is inferred, and the power from and to each bus depends on the inferred voltage magnitude and phase values, such that:

$$\tilde{P}_n = \hat{P}g_n - \dot{P}d_n - \dot{G}s_n\hat{V}_n^2 - \sum_{e\in\mathcal{N}(n)} Pf_e\left(\hat{V}_n, \hat{\theta}_n, \dot{\omega}_e\right) - \sum_{e\in\mathcal{N}(n)} Pt_e\left(\hat{V}_n, \hat{\theta}_n, \dot{\omega}_e\right) \tag{5.4}$$

$$\tilde{Q}_n = \hat{Q}g_n - \dot{Q}d_n + \dot{B}s_n\hat{V}_n^2 - \sum_{e\in\mathcal{N}(n)} Qf_e\left(\hat{V}_n, \hat{\theta}_n, \dot{\omega}_e\right) - \sum_{e\in\mathcal{N}(n)} Qt_e\left(\hat{V}_n, \hat{\theta}_n, \dot{\omega}_e\right) \tag{5.5}$$

$$n = 1, \ldots, N_B$$

The final term of the loss function is in itself a composition of terms designed to discourage the transgression of the inequality constraints. The violation degrees (VDs) are defined by ramp functions that depend on how much the inferred parameters deviate outside the minimum and maximum established limits, i.e. for any parameter $x \in \mathbb{R}$ with lower and upper limits x^{min}, $x^{max} \in \mathbb{R}$, the VDs for both limits are defined as:

$$\eta_{min} = \begin{cases} 0, & \text{if } x^{min} - x < 0 \\ x^{min} - x, & \text{otherwise (limit violation)} \end{cases} \tag{5.6}$$

$$\eta_{max} = \begin{cases} 0, & \text{if } x - x^{max} < 0 \\ x - x^{max}, & \text{otherwise (limit violation)} \end{cases} \tag{5.7}$$

These functions are applied to the inequality constraints that are determined by the operational constraints of voltage magnitude, and power generation. For simplicity, the ramp function is expressed as the "max" function, which outputs the maximum of two given values.

$$\eta_{Vmin} = \max(0, \dot{V}_{n,h}^{min} - \hat{V}_{n,h}) \tag{5.8}$$

$$\eta_{Vmax} = \max(0, \hat{V}_{n,h} - \dot{V}_{n,h}^{max}) \tag{5.9}$$

$$\eta_{Pmin} = \max(0, \dot{P}g_{g,h}^{min} - \hat{P}g_{g,h}) \tag{5.10}$$

$$\eta_{Pmax} = \max(0, \hat{P}g_{g,h} - \dot{P}g_{g,h}^{max}) \tag{5.11}$$

$$\eta_{Qmin} = \max(0, \dot{Q}g_{g,h}^{min} - \hat{Q}g_{g,h}) \tag{5.12}$$

$$\eta_{Qmax} = \max(0, \hat{Q}g_{g,h} - \dot{Q}g_{g,h}^{max}) \tag{5.13}$$

$$n = 1, \ldots, N_B,\ g = 1, \ldots, N_G,\ h = 1, \ldots, H$$

where eqs. (5.8) and (5.9) represent the VD of voltage magnitude from their lower and upper limits, respectively, for every bus; eqs. (5.10) and (5.11) represent the VD of the active power generation from their lower and upper limits, respectively, for every generation bus, eqs. (5.12) and (5.13) are similarly defined, but for the reactive power generation.

It should be noted that the ramp function is not differentiable in all its domain, but the gradient can be estimated through the subgradient. The two VDs of each parameter are aggregated, as shown below:

$$\eta_{V_{n,h}} = \eta_{Vmin} + \eta_{Vmax} \tag{5.14}$$

$$\eta_{P_{g,h}} = \eta_{Pmin} + \eta_{Pmax} \tag{5.15}$$

$$\eta_{Q_{g,h}} = \eta_{Qmin} + \eta_{Qmax} \tag{5.16}$$

As with the previous loss function expressions, the mean square error of each parameter is calculated (eqs. (5.17 - 5.19)), then to obtain the last piece of the loss function, all the resulting terms are added together (eq.(5.20)).

$$v_{limit} = \frac{1}{H}\sum_{h=1}^{H}\frac{1}{N_B}\sum_{n=1}^{N_B}(\eta_{V_{n,h}})^2 \tag{5.17}$$

$$p_{limit} = \frac{1}{H}\sum_{h=1}^{H}\frac{1}{N_G}\sum_{g=1}^{N_G}(\eta_{P_{g,h}})^2 \tag{5.18}$$

$$q_{limit} = \frac{1}{H}\sum_{h=1}^{H}\frac{1}{N_G}\sum_{g=1}^{N_G}(\eta_{Q_{g,h}})^2 \tag{5.19}$$

$$\mathcal{L}_c = \lambda_c(v_{limit} + p_{limit} + q_{limit}) \tag{5.20}$$

Eq. (5.21) shows the final loss function, which is simply the sum of the terms described above (eqs. (5.2), (5.3) and (5.20)). Each of the three terms has a corresponding weight, represented as λ_a, λ_b and λ_c. These weights are assigned to give more or less importance to each term in the total loss function.

$$\mathcal{L} = \mathcal{L}_a + \mathcal{L}_b + \mathcal{L}_c \tag{5.21}$$

5.2.2 TGN based OPF solver

The proposed TGN based OPF solver only considers two types of bus nodes: generator and load nodes, and branch nodes i.e. the graph for every TGN layer is defined by: $\mathcal{G} = (\mathcal{V}_D, \mathcal{V}_G, \mathcal{V}_E)$. The scheme of the TGN based OPF solver proposed in this work is shown in Fig. 5.1. The solid blue squares represent the main parts of each TGN layer, and other squares with a blue edge represent either components of the input or intermediate layer outputs (which are fed to the next layer). The orange square with the bright orange border shows the initialization of the inference variables, and is only input to the first TGN layer. The orange square with purple edge represents the output only of the last TGN layer. In the following subsections the different elements of the scheme are explained in more detail.

5.2.3 Grid state data and initial inference values

The data used to train and verify the proposed model is obtained from a benchmark library from the IEEE PES task force on benchmarks for validation of emerging power system algorithms to solve the AC optimal power flow problem, posed as a non-convex, nonlinear program; the headers used were the ones of cases 14, 30, 57 and 200 buses (Babaeinejadsarookolaee et al., 2019). All of the case files are curated in the MATPOWER data format, and thus include three main structures: bus data, generator data and transmission line data. Details on the data of the different structures is explained in more detail in section 3.5.

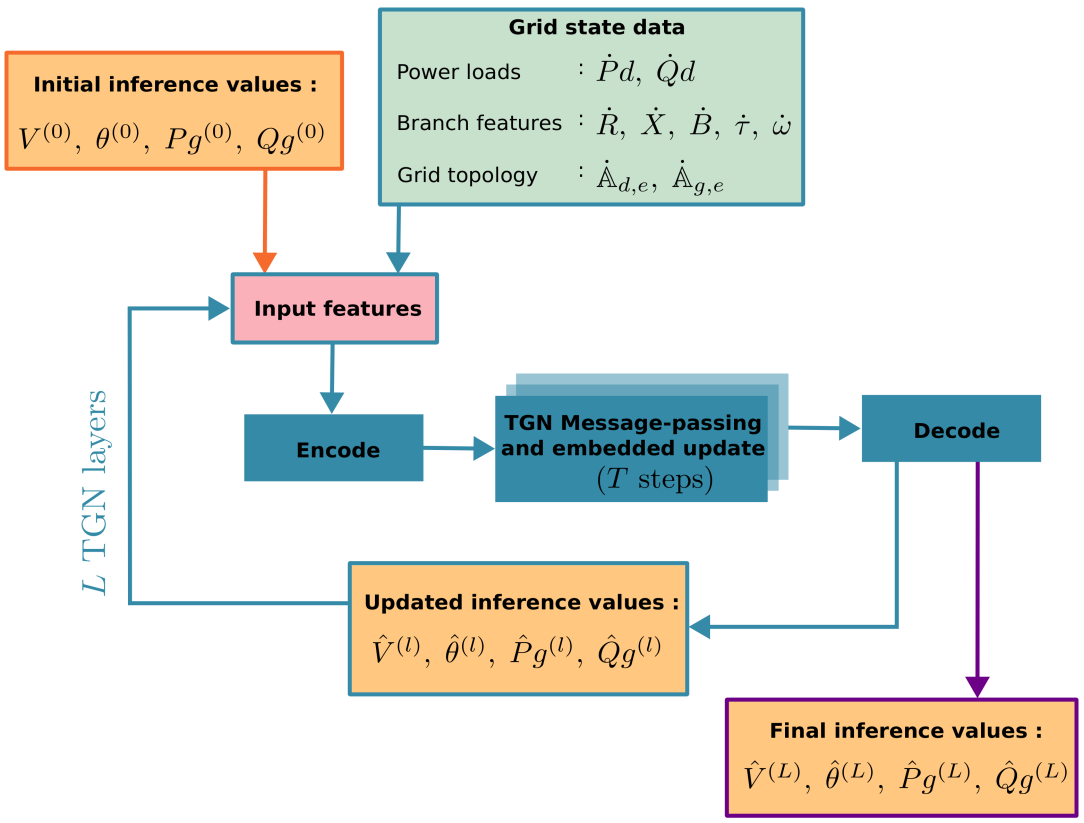

Initial inference values :
$V^{(0)},\ \theta^{(0)},\ Pg^{(0)},\ Qg^{(0)}$
Grid state data
Power loads : $\dot{P}d,\ \dot{Q}d$
Branch features : $\dot{R},\ \dot{X},\ \dot{B},\ \dot{\tau},\ \dot{\omega}$
Grid topology : $\dot{\mathbb{A}}_{d,e},\ \dot{\mathbb{A}}_{g,e}$
Input features
Encode
TGN Message-passing and embedded update (T steps)
Decode
L TGN layers
Updated inference values :
$\hat{V}^{(l)},\ \hat{\theta}^{(l)},\ \hat{P}g^{(l)},\ \hat{Q}g^{(l)}$
Final inference values :
$\hat{V}^{(L)},\ \hat{\theta}^{(L)},\ \hat{P}g^{(L)},\ \hat{Q}g^{(L)}$

FIGURE 5.1: TGN based OPF solver scheme.

The proposed model is trained by adding uniformly distributed noise to specific parts of the different data structures. A batch of data is generated for every training iteration, to do this two main steps are taken: first either random uniform noise is added to specific elements of the different benchmark cases data structures, or a predefined time series is loaded, secondly initial voltage and power generation values are defined.

The following list presents the values to which noise is added, and how it is applied. In the following, the prime symbol is used to indicate original benchmark case values.

- The active and reactive power load of every bus is uniformly distributed between 80 and 120% of the original benchmark case value:

$$\dot{P}d_n = \mathrm{U}_{PQ} \cdot Pd'_n, \quad n = 1, \ldots, N_B \tag{5.22}$$

$$\dot{Q}d_n = \mathrm{U}_{PQ} \cdot Qd'_n, \quad n = 1, \ldots, N_B \tag{5.23}$$

$$\mathrm{U}_{PQ} \sim \mathcal{U}(0.8, 1.2)$$

  It should be noted that the same noise value is applied to both active and reactive power loads (they change in a proportionate manner), and that it is only applied when time series data is not available.

- The shunt susceptance of every bus is uniformly distributed between 90 and 110% of the original value, i.e.

$$\dot{Bs}_n = \mathrm{U}_{Bs} \cdot Bs'_n, \quad n = 1, \ldots, N_B \tag{5.24}$$
$$\mathrm{U}_{Bs} \sim \mathcal{U}(0.9, 1.1)$$

- The maximum and minimum power generation limits change for every sample of the training batch, uniformly distributed noise is added:

$$\dot{Pg}_g^{max} = \mathrm{U}_{Pmax} \cdot Pg_g^{max'}, \quad g = 1, \ldots, N_G \tag{5.25}$$
$$\dot{Pg}_g^{min} = \mathrm{U}_{Pmin} \cdot Pg_g^{min'}, \quad g = 1, \ldots, N_G \tag{5.26}$$
$$\mathrm{U}_{Pmax} \sim \mathcal{U}(1, 1.1), \quad \mathrm{U}_{Pmin} \sim \mathcal{U}(0.9, 1)$$

  The active power generation limits are varied to add generalization capabilities to the proposed model, in such a way that it will better learn to deal with generators with different capacities.

- The branch resistance, reactance and susceptance noise are generated independently, but in the same manner:

$$\dot{R}_e = \mathrm{U}_R \cdot R'_e, \quad e = 1, \ldots, N_E \tag{5.27}$$
$$\dot{X}_e = \mathrm{U}_X \cdot X'_e, \quad e = 1, \ldots, N_E \tag{5.28}$$
$$\dot{B}_e = \mathrm{U}_B \cdot B'_e, \quad e = 1, \ldots, N_E \tag{5.29}$$
$$\mathrm{U}_R, \mathrm{U}_X, \mathrm{U}_B \sim \mathcal{U}(0.9, 1.1)$$

- After noise has been added to the injections and transmission line characteristics, the topology of the grid is changed by randomly deleting a transmission line for each sample.

Parameters of the power line are known to change due to change in the operational conditions, added or removed vegetation and aging. These changes are generally infrequent but still significant, which is the main reason branch noise is added.

The initial state of the parameters that will be inferred by the model are indicated by $Pg^{(0)}$, $Qg^{(0)}$, $V^{(0)}$ and $\theta^{(0)}$. The active and reactive power are independently initialized to a value between 25 and 75% of the previously established generation limits, i.e.

$$Pg_g^{(0)} = \mathrm{U}_{Pg} \cdot (\dot{Pg}_g^{max} - \dot{Pg}_g^{min}) + \dot{Pg}_g^{min}, \quad g = 1, \ldots, N_G \tag{5.30}$$
$$Qg_g^{(0)} = \mathrm{U}_{Qg} \cdot (\dot{Qg}_g^{max} - \dot{Qg}_g^{min}) + \dot{Qg}_g^{min}, \quad g = 1, \ldots, N_G \tag{5.31}$$
$$\mathrm{U}_{Pg}, \mathrm{U}_{Qg} \sim \mathcal{U}(0.25, 0.75)$$

There must be some amount of balance between power generation and load, in this work a 5% error margin is set; if the total generation and load difference is larger, the initial generation state is scaled to meet the requirement.

The voltage magnitude of every bus is initialized to a value between 30 and 70% of the established bus limit, i.e.

$$V_n^{(0)} = \mathrm{U}_V \cdot (\dot{V}_n^{max} - \dot{V}_n^{min}) + \dot{V}_n^{min}, \quad n = 1, \ldots, N_B \tag{5.32}$$
$$\mathrm{U}_V \sim \mathcal{U}(0.3, 0.7)$$

The voltage phase of each bus is initialized to the slack bus angle established in the benchmark case, i.e.

$$\theta_n^{(0)} = \theta'_{slack}, \quad n = 1, \ldots, N_B \tag{5.33}$$

For the case in which established load time series are specified (e.g. for validation), these are simply loaded and used as they are, and random noise is added to the rest of parameters not given by the time series.

This process of adding noise and changing the configuration of the benchmark electrical grid state, and initializing the voltage and power generation values is repeated every time a batch of data is needed by the proposed model, either for training, validation or testing.

5.2.4 Input features

For all TGN layers except the first one, the input features are calculated from the grid state data (power loads, branch features and grid topology), and the updated inferred voltage and power generation states from the previous TGN layer. For the first TGN layer, the input features depend on the grid state data and the initialization of the inference variables.

As was mentioned in subsection 5.2.1 which discussed the loss function, the three main aspects of the OPF problem involve the power losses in the transmission lines, the power equilibrium equations and the operational limits of the electrical grid. These components are correspondingly incorporated into the input features of the different types of nodes, such that the input feature set for every type of node is as shown below:

$$\begin{aligned}\mathbf{x}_{\nu_g}^{(l)} = \{&\hat{V}_{\nu_g}^{(l-1)},\ \hat{\theta}_{\nu_g}^{(l-1)},\ \tilde{P}_{\nu_g}^{(l-1)},\ \tilde{Q}_{\nu_g}^{(l-1)},\ \eta_{Vmin,\nu_g}^{(l-1)},\ \eta_{Vmax,\nu_g}^{(l-1)},\\ &\eta_{Pmin,\nu_g}^{(l-1)},\ \eta_{Pmax,\nu_g}^{(l-1)},\ \eta_{Qmin,\nu_g}^{(l-1)},\ \eta_{Qmax,\nu_g}^{(l-1)}\}, \quad \forall \nu_g \in \mathcal{V}_G\end{aligned} \tag{5.34}$$

$$\begin{aligned}\mathbf{x}_{\nu_d}^{(l)} = \{&\hat{V}_{\nu_d}^{(l-1)},\ \hat{\theta}_{\nu_d}^{(l-1)},\ \tilde{P}_{\nu_d}^{(l-1)},\ \tilde{Q}_{\nu_d}^{(l-1)},\ \eta_{Vmin,\nu_d}^{(l-1)},\ \eta_{Vmax,\nu_d}^{(l-1)}\},\\ &\forall \nu_d \in \mathcal{V}_D\end{aligned} \tag{5.35}$$

$$\mathbf{x}_{\nu_e}^{(l)} = \{\rho_{\nu_e},\ \delta_{\nu_e},\ \dot{B}_{\nu_e},\ \dot{\tau}_{\nu_e},\ \dot{\omega}_{\nu_e},\ J_{loss,\nu_e}^{(l-1)}\}, \quad \forall \nu_e \in \mathcal{V}_E \tag{5.36}$$

Since generator type nodes must infer all the variables that are constrained by operation limits, all the VDs of the corresponding generator nodes are used as input features. The voltage magnitude and phase, and the deviation from both active and reactive power equilibrium are considered input features to this type of node as well. Considering load type nodes only infer voltage values, from all VDs, only the voltage magnitude ones are used as input features for these type of nodes. Just like with generator nodes, voltage values and the deviation from power equilibrium are included as input features as well. For branch type nodes, branch characteristics (which do not change from TGN layer to layer) are used as input features along with the value of the branch power loss from the previous step. The branch characteristics that are considered are the branch admittance magnitude and phase, ρ and δ, respectively, the line charging susceptance, and for transformers the off nominal turns ratio and the phase shift angle.

5.2.5 TGN model

In this model the same TGN is used recursively for every TGN layer, i.e. the same encoding, message-passing, embedded updating and decoding MLPs are used for every layer. Since there are only 2 types of bus nodes which are only connected to branch nodes, only two adjacency matrices are needed to represent the topology of the TGN architecture:

$$\mathbb{A}_{d,e} \in \mathbb{R}^{N_D \times N_E} \tag{5.37}$$

$$\mathbb{A}_{g,e} \in \mathbb{R}^{N_G \times N_E} \tag{5.38}$$

The T message-passing and embedded update steps of each layer are represented by:

$$\nu_d^{(t+1)} = \sigma_{\nu_d}\left(\nu_d^{(t)},\ \mathbb{A}_{d,e} \cdot \mu_{e\to d}(\nu_e^{(t)})\right), \quad \forall \nu_d \in \mathcal{V}_D \tag{5.39}$$

$$\nu_g^{(t+1)} = \sigma_{\nu_g}\left(\nu_g^{(t)},\ \mathbb{A}_{g,e} \cdot \mu_{e\to g}(\nu_e^{(t)})\right), \quad \forall \nu_g \in \mathcal{V}_G \tag{5.40}$$

$$\begin{aligned} \nu_e^{(t+1)} &= \sigma_{\nu_e}\left(\nu_e^{(t)},\ \mathbb{A}_{d,e}^T \cdot \mu_{d\to e}(\nu_d^{(t)}),\ \mathbb{A}_{g,e}^T \cdot \mu_{g\to e}(\nu_g^{(t)})\right), \\ &\forall \nu_e \in \mathcal{V}_E \end{aligned} \tag{5.41}$$

where σ_d, σ_g and σ_e are the update MLPs for load, generation and branch nodes, respectively; $\mu_{e\to d}$ and $\mu_{e\to g}$ are the aggregation MLPs from branch type nodes to load and generator nodes, respectively; $\mu_{d\to e}$ and $\mu_{g\to e}$ are the aggregation MLPs from load and generator type nodes, respectively, to branch type nodes.

As described in chapter 3, there are encoding MLPs defined for every type of node (γ_{ν_d}, γ_{ν_g} and γ_{ν_e}) that embed the input features into the latent space, and decoding functions for the load and generation type nodes (φ_{ν_d} and φ_{ν_g}) to obtain the correct number of outputs for every TGN layer. The generator type node decoding function has four outputs: voltage magnitude and phase changes, and active and reactive power generation changes. The load type node decoding function has two outputs: voltage magnitude and phase changes. The outputs of the decoder functions of a TGN layer l for both types of nodes are shown below:

$$\varphi_{\nu_d}^{(l)} = \{\Delta V_{\nu_d}^{(l)},\ \Delta\theta_{\nu_d}^{(l)}\}, \quad \forall \nu_d \in \mathcal{V}_D \tag{5.42}$$

$$\varphi_{\nu_g}^{(l)} = \{\Delta V_{\nu_g}^{(l)},\ \Delta\theta_{\nu_g}^{(l)},\ \Delta Pg_{\nu_g}^{(l)},\ \Delta Qg_{\nu_g}^{(l)}\}, \quad \forall \nu_g \in \mathcal{V}_G \tag{5.43}$$

where ΔV_{ν_χ} and $\Delta\theta_{\nu_\chi}$ represent the voltage magnitude and phase modifications for a χ type bus; ΔPg_{ν_g} and ΔQg_{ν_g} represent the active and reactive power modifications for the generation type nodes.

5.2.6 Updated and final inference values

The outputs of the decoders are used to update the voltage and power generation values of every bus, and generate the TGN layer output, as shown below:

$$\hat{V}_{v_d}^{(l)} = \hat{V}_{v_d}^{(l-1)} + \Delta V_{v_d}^{(l)} \tag{5.44}$$

$$\hat{\theta}_{v_d}^{(l)} = \hat{\theta}_{v_d}^{(l-1)} + \Delta\theta_{v_d}^{(l)} \tag{5.45}$$

$$\forall v_d \in \mathcal{V}_D$$

$$\hat{V}_{v_g}^{(l)} = \hat{V}_{v_g}^{(l-1)} + \Delta V_{v_g}^{(l)} \tag{5.46}$$

$$\hat{\theta}_{v_g}^{(l)} = \hat{\theta}_{v_g}^{(l-1)} + \Delta\theta_{v_g}^{(l)} \tag{5.47}$$

$$\forall v_g \in \mathcal{V}_G - slack$$

$$\hat{P}g_{v_g}^{(l)} = \hat{P}g_{v_g}^{(l-1)} + \Delta Pg_{v_g}^{(l)} \tag{5.48}$$

$$\hat{Q}g_{v_g}^{(l)} = \hat{Q}g_{v_g}^{(l-1)} + \Delta Qg_{v_g}^{(l)} \tag{5.49}$$

$$\forall v_g \in \mathcal{V}_G$$

Thus, for a layer l, eqs. (5.44) and (5.46) represent the voltage magnitude output for load and generation nodes, respectively; eqs. (5.45) and (5.47) represent the voltage phase output for load and generation nodes, respectively; eqs. (5.48) and (5.49) represent the active and reactive power generation for generation nodes, respectively. Even though a slack node is not considered in this model, one of the generator buses is indeed the slack bus, as indicated by the benchmark case. For slack nodes, the output features corresponding to the voltage magnitude and phase are hard-coded to the upper limit for the magnitude, and to original value of the slack for the phase, i.e.

$$\hat{\theta}_{slack}^{(l)} = \theta_{slack}^{(0)} \tag{5.50}$$

$$\hat{V}_{slack}^{(l)} = V_{slack}^{max} \tag{5.51}$$

$$l = 1, \ldots, L$$

The TGN layer sequentially produces L outputs, the final outputs represent the final voltage and power generation inferences.

5.2.7 Training

The training process is described in algorithm 4. The proposed PI-TGN OPF solver is trained in batches, the encoding, message-passing, update and decoding NNs of the single TGN layer are trained simultaneously. The gradients of the loss function with respect to the parameters of the NNs are calculated with the backpropagation algorithm. These gradients are used to modify the values of the NN parameters via the Adam optimization algorithm, afterwards a new batch of data is introduced to the PI-TGN model and the process is repeated until the cost function converges to a minimum value.

In algorithm 4, $\eta_{min}^{(l)}$ and $\eta_{max}^{(l)}$ summarize all VD values (voltage magnitude, active and reactive power generation).

Algorithm 4 Training algorithm for PI-TGN based OPF solver

```
Require: Bus, generator and branch data (benchmark case)
  while $i < i_{max}$ do                                                          ▷ Training iterations
      $\dot{P}d,\ \dot{Q}d,\ \dot{B}s \leftarrow$ eqs.(5.22)-(5.24)                         ▷ Bus noise
      $\dot{P}g^{min},\ \dot{P}g^{max} \leftarrow$ eqs.(5.26)-(5.25)                        ▷ Active power limits noise
      $\dot{R},\ \dot{X},\ \dot{B} \leftarrow$ eqs.(5.27)-(5.29)                          ▷ Transmission line noise
      $from,\ to \leftarrow$ configuration change                                   ▷ Random branch outage
      $\dot{G},\ \dot{V}^{min},\ \dot{V}^{max},\ \dot{Q}g^{min},\ \dot{Q}g^{max},\ \dot{\tau},\ \dot{\omega} \leftarrow$ benchmark          ▷ No noise
      $\hat{V}^{(0)},\ \hat{\theta}^{(0)},\ \hat{P}g^{(0)},\ \hat{Q}g^{(0)} \leftarrow$ eqs.(5.30)-(5.33)          ▷ Initialized values
      $\mathbb{A}_{d,e},\ \mathbb{A}_{g,e} \leftarrow$ eqs.(5.37)-(5.38)                      ▷ Adjacency matrices
      for $l = 1 \dots L$ do                                                      ▷ $L$ TGN layers
          $J_{loss}^{(l)} \leftarrow$ eq.(5.1)                                          ▷ Branch power loss
          $\tilde{P}^{(l)},\ \tilde{Q}^{(l)} \leftarrow$ eqs.(5.4)-(5.5)                          ▷ Power equilibrium
          $\eta_{min}^{(l)},\ \eta_{max}^{(l)} \leftarrow$ eqs.(5.8)-(5.13)                       ▷ Violation degrees
          $\mathbf{x}_{\nu_g}^{(l)},\ \mathbf{x}_{\nu_d}^{(l)},\ \mathbf{x}_{\nu_e}^{(l)} \leftarrow$ eqs.(5.34)-(5.36)                 ▷ Input features
                                                                                  ▷ TGN layer
          $\Delta V^{(l)},\ \Delta\theta^{(l)},\ \Delta Pg^{(l)},\ \Delta Qg^{(l)} \leftarrow TGN\_model(\mathbf{x}_{\nu_g}^{(l)}, \mathbf{x}_{\nu_d}^{(l)}, \mathbf{x}_{\nu_d}^{(l)}, \mathbb{A}_{d,e}, \mathbb{A}_{g,e})$
          $\hat{V}^{(l)},\ \hat{\theta}^{(l)},\ \hat{P}g^{(l)},\ \hat{Q}g^{(l)} \leftarrow$ eqs.(5.44)-(5.51)        ▷ Update inference variables
      end for
      $\mathcal{L}(J_{loss}^{(L)}, \tilde{P}^{(L)}, \tilde{Q}^{(L)}, \eta_{min}^{(L)}, \eta_{max}^{(L)}) \leftarrow$ eq.(5.21)              ▷ Loss function
      $gradients \leftarrow \nabla_w(\mathcal{L})$                      ▷ $loss$ is differentiated against NN weights
      $weights \leftarrow w - \alpha\nabla_w(\mathcal{L})$     ▷ Adam optimizer applies gradients to NN weights
  end while
  Save trainable weights
```

5.3 Experiments

In this section, the predictive accuracy of the proposed TGN-based OPF solver is evaluated by comparing it to the resuLts obtained with the Matpower AC OPF solver. It also analyzes various design decisions in detail. All the instances of the proposed PI-TGN based OPF solver share the same hyperparameters:

- Number of TGN layers: $L = 4$.
- Number of message passing and update steps: $T = 2$.
- Embedded dimension of node types: $d = 16$.
- Learning rate: $\alpha = 1 \times 10^{-3}$.

5.3.1 Simulation setup

To test the accuracy of the proposed solver, five different test cases are analyzed, each based on a header from the IEEE PES benchmarks (Babaeinejadsarookolaee et al., 2019). The headers used are: "PGLIB OPF case14 IEEE", "PGLIB OPF case30 IEEE", "PGLIB OPF case57 IEEE", "PGLIB OPF case118 IEEE" and "PGLIB OPF case200 activ" (Birchfield et al., 2017). The characteristics of each of the grids are given in Table 5.1.

Test case	**N_B**	**N_G**	**N_E**	**max ΔPd (MW)**	**max ΔQd (MVAr)**	**max ΔPg (MW)**	**max ΔQg (MVAr)**
14_ieee	14	5	19	18.3	3.7	90.8	14.4
14_ieee (time series)	14	5	19	-	-	107.4	15.0
30_ieee	30	6	40	17.2	5.8	44.1	79.4
57_ieee	57	7	79	**74.7**	16.3	241.2	82.7
118_ieee	118	54	185	55.0	**22.4**	**301.3**	**450.3**
200_activ	200	49	244	14.4	4.11	205.3	65.3

TABLE 5.1: Test case characteristics

The noise added to each of the grids is specified in subsection 5.2.3, however, depending on the distribution and characteristics of loads and generators of each grid, the real load and generator noise applied to each of the grids differs. The last four columns of Table 5.1 show the maximum amount of load and generator alteration from the nominal case, in *MW* and *MVAr*, respectively. It can be seen, that even in the smallest grid of 14 buses, since three of the five generators are only for voltage support (they do not produce active power), all the active load has to be satisfied by only two generators, and so the noise applied to them results in large modifications from the nominal case. The opposite is true for the biggest grid of 200 buses, in which most of the generators produce active power and the load can be distributed between them, thus, the noise applied does not cause such big changes from the nominal case. As can be seen, in all cases the variation is significant, and is additional to the noise introduced in branch features, active power generation limits and shunt susceptance of the buses; the biggest differences from the nominal cases are shown in bold.

Furthermore, since in this work the branch flow limits are not considered, the branch ratings of the cases are manually modified to zero to make them unlimited when solving with the Matpower AC OPF solver, for comparison purposes. In the following tests, all Matpower results are included, even those in which some of the established limits are breached.

5.3.2 Prediction accuracy with noisy benchmark cases test

For the first test, five instances of the proposed TGN-based solver (with the same hyperparameters) are trained on the five different electrical grid benchmark cases. The data is normalized using the per unit system, and a batch of 20 samples is taken for testing each instance.

Table 5.2 reports the average L2 loss (squared error loss) obtained from the difference between the inferred values obtained with the TGN-based solver and the values obtained with the Matpower AC OPF solver using the IPOPT method. Each column shows the average predicted error in percentages: $\frac{1}{N_y}\sum(y-\hat{y})^2 \times 100$, for a variable y with N_y values.

	V	$\boldsymbol{\theta}$	**Pg**	**Qg**
14 buses	0.074	2.168	3.06	0.817
30 buses	0.056	1.554	1.98	3.03
57 buses	0.185	1.28	4.14	8.64
118 buses	0.079	1.06	13.31	14.58
200 buses	0.206	0.091	0.476	7.25

TABLE 5.2: OPF Prediction errors (%)

Fig.(5.2) shows the mean (blue line) and standard deviation (green shaded area) of the L2 loss for the 14 bus case; the voltage magnitude (top left) and voltage phase (top right) for each bus, and the active (bottom left) and reactive (bottom right) power generation for each generator bus. The voltage phase of the slack bus (bus 1 in this case) is hard-coded to the nominal value, which is why the error in this point is close to zero. The voltage magnitude of the slack bus is set to the maximum limit, which in this case also results in a similar result to the Matpower solution. Only generators 1 and 2 of the 14 bus case produce active power, which is why the active power prediction error in the other generators is trivially zero. Meanwhile, the mean reactive power error is largest for generator 3 which is the one with the widest reactive power generation limits.

Fig. (5.3) shows the mean (blue line) and standard deviation (green shaded area) of the L2 loss for the 30 bus case; the voltage magnitude (top left) and voltage phase (top right) for each bus, and the active (bottom left) and reactive (bottom right) power generation for each generator bus. Similar interpretations as with the bus 14 cases are achieved with respect of the voltage magnitude and phase of the slack bus. In this case all generators are capable of producing active power, and the generator most prone to divergence in the active and reactive power generated with respect to the Matpower case is generator 2.

Fig. (5.4) shows the mean (blue line) and standard deviation (green shaded area) of the L2 loss for the 57 bus case; the voltage magnitude (top left) and voltage phase (top right) for each bus, and the active (bottom left) and reactive (bottom right) power generation for each generator bus. Similar interpretations as with the previous cases are achieved with respect of the voltage magnitude and phase of the slack bus. In this case, similar to the case with 14 buses, generator buses 2, 4 and 6 do not generate active power, which is why the error in this nodes is so drastically

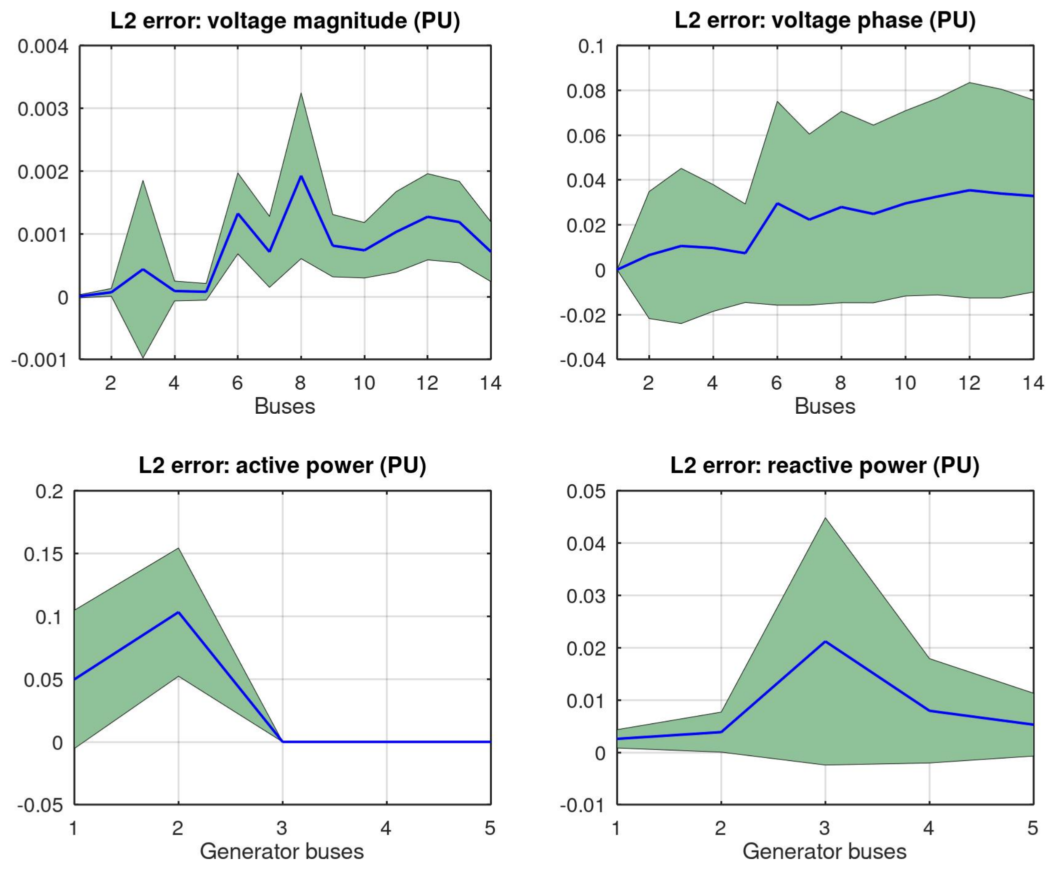


FIGURE 5.2: Mean and SD of OPF L2 error (14-bus case).

close to zero. Generators 5 and 7 are the largest, and thus most prone to divergence from the Matpower solution.

Fig. (5.5) shows the mean (blue line) and standard deviation (green shaded area) of the L2 loss for the 118 bus case; the voltage magnitude (top left) and voltage phase (top right) for each bus, and the active (bottom left) and reactive (bottom right) power generation for each generator bus. Similar interpretations as with the previous cases are achieved with respect of the voltage magnitude and phase of the slack bus, which in this case is bus 69 (generator bus 30). This case has many generators that do not generate active power, which seems to cause a negative effect on the predictive ability on the generator buses that have to compensate the load. This case is the one with the worst results in power generation inference, even though the voltage magnitude and phase errors are quite small. The lack of active power generators and the large amount of noise from the nominal case is hypothesized to be the main reasons for this.

Fig. (5.6) shows the mean (blue line) and standard deviation (green shaded area) of the L2 loss for the 200 bus case; the voltage magnitude (top left) and voltage phase (top right) for each bus, and the active (bottom left) and reactive (bottom right) power generation for each generator bus. Similar interpretations as with the previous cases are achieved with respect of the voltage magnitude and phase of the slack bus, which in this case is bus 189 (generator bus 47). In this case, most generators generate active power and are not too large; the largest generators are 29, 30 and 47, which is where the largest power generation errors are found. However, in general very good results were obtained with this case, that although large, seems to behave

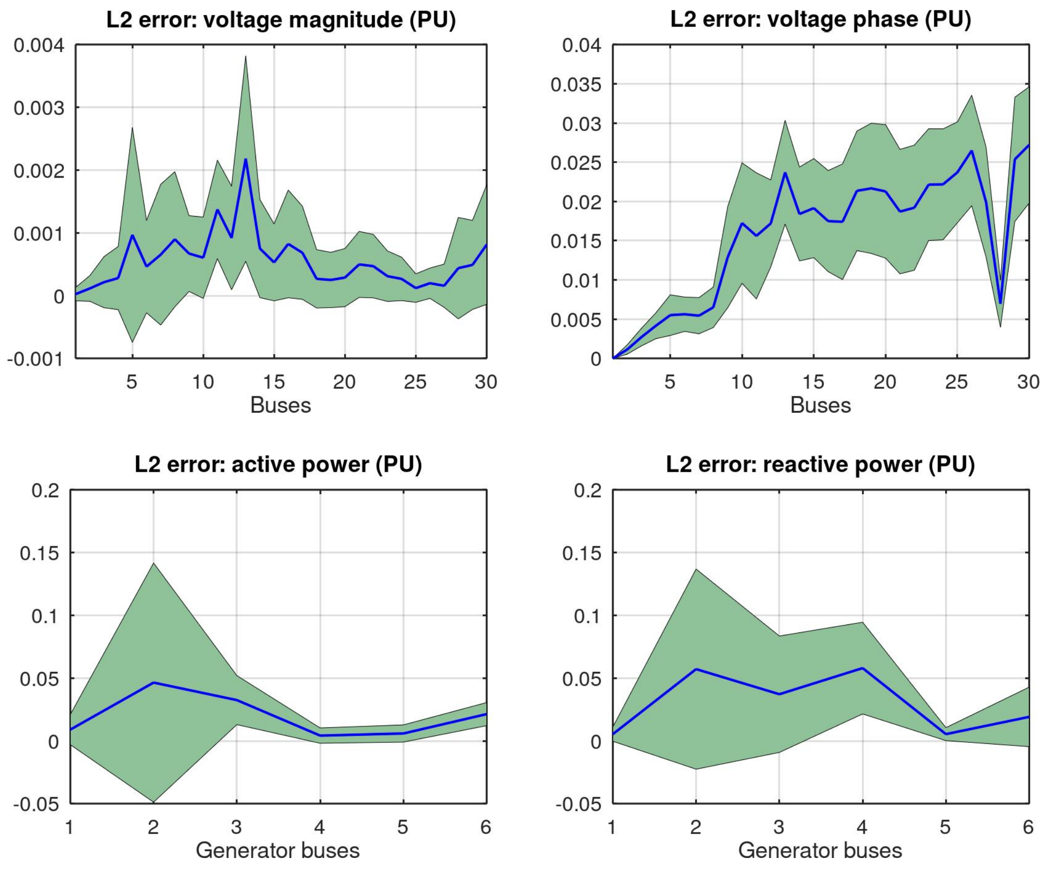


FIGURE 5.3: Mean and SD of OPF L2 error (30-bus case).

well.

In all cases the best approximation is made of the voltage magnitude, and the power generation proves to be the most difficult to infer similarly to the Matpower solution.

5.3.3 Loss function elements of test with noisy benchmark cases

Some attention must be paid to the results of adding the different constraints to the loss function of the TGN-based solver, as there are no target values, these elements greatly influence the behavior of the solver and may cause it to diverge from the Matpower results by giving more or less importance to different aspects.

The main objective of the loss function is to minimize branch power losses. In Figs. (5.7)-(5.11) the total active power loss of all branches for every sample in the test batch is shown for both the TGN-based solution and the Matpower solution. In this case, the loss is calculated as shown in eq. (5.1); as can be seen, in general the total loss is lower with the proposed solution.

The violation of operational constraints is another element of the loss function. In Figs.(5.12)-(5.18) the mean of the different VDs at each corresponding bus of the different test cases, that are not zero, are shown. For Fig. (5.12), the 14 bus test case, the only VD that has some values different from zero are from the minimum active power generation. The mean violation values are very small, because for these generators the active power should be zero but because of small numerical errors, sometimes it is inferred to be small negative active generation values. Something

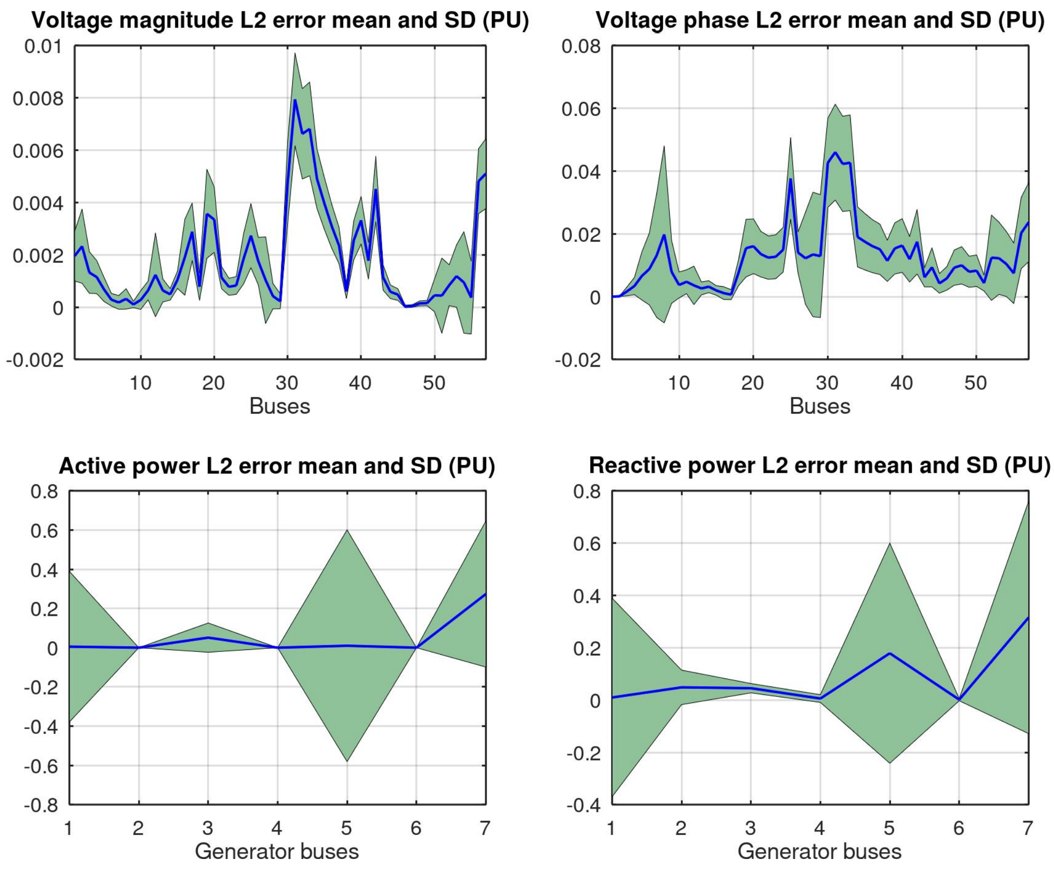


FIGURE 5.4: Mean and SD of OPF L2 error (57-bus case).

similar happens with Fig. (5.13), where generator buses 2, 4 and 6 are supposed to be zero but sometimes infer small values very close to zero. The most recurrent violation is of the maximum voltage magnitude, as shown in Figs. (5.14), (5.16) and (5.18). The test that corresponds to the 30 bus case does not violate any of the operational limits, and in general, the VDs of all cases are quite small, which suggests that the proposed method could be adequate for real-world applications.

The last element of the loss function, which is very important, is ensuring the power balance at every bus. Figs. (5.20)-(5.29) show the average active and reactive power imbalance of every bus for all the test cases for both the solutions from the proposed TGN-solver and the solutions obtained with the Matpower AC OPF solver. In most cases the imbalance of the proposed solver is smaller at every bus than the imbalance obtained with the Matpower solver, except for the case of 118 nodes, as can be seen in Figs. (5.26) and (5.27). On the other hand, for the 200 bus grid, the proposed solver successfully reduced the imbalance to virtually zero for all nodes, as observed in Figs. (5.28) and (5.29). The imbalances from the Matpower solutions could be due to the fact that all solutions were included, even those that did not converge in Matpower (no solution was found that could respect all the specified constraints).

In general it can be observed that the proposed solver does a good job at minimizing the elements of the loss function, which emphasizes the importance of the physics-informed approach of adding the physical constraints into the loss function.

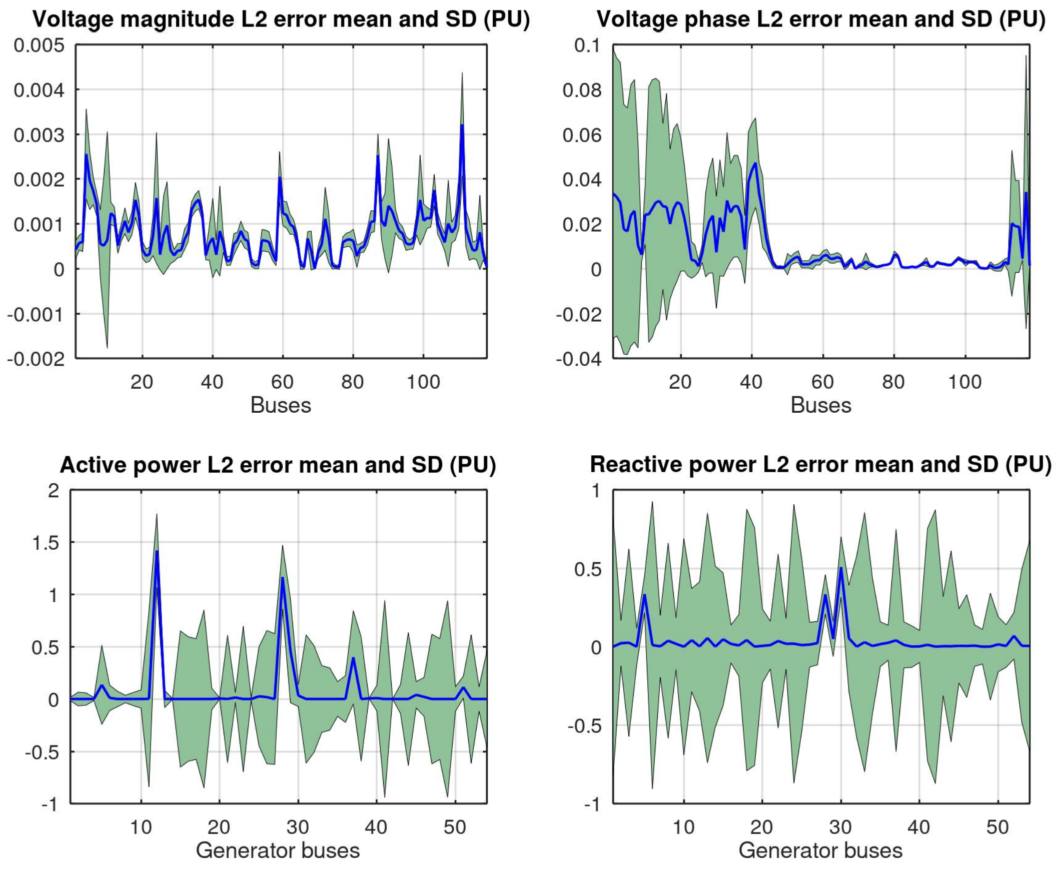


FIGURE 5.5: Mean and SD of OPF L2 error (118-bus case).

5.3.4 Prediction accuracy with time series case test

In this subsection, time series data from the 14 bus grid is used for training an instance of the proposed TGN-based OPF solver. The load profiles are obtained from the Codalab competition "Learning to Run a Power Network 2019" (Marot et al., 2020), which includes the active and reactive power load in intervals of 5 minutes. The TGN-based solver was trained and tested on batches of 300 samples (equivalent to one day and one hour of data). As with Table 5.2, in Table 5.3 the average L2 loss (squared error loss) calculated from the difference between the results obtained with the proposed solver and the results obtained with Matpower using the interior point optimizer (IPOPT) method is reported. Each column shows the average predicted error of each of the inference variables, in percentages.

Test case	**V**	θ	**Pg**	**Qg**
14_ieee (time series)	0.104	1.03	7.99	3.65

TABLE 5.3: Prediction errors in time series case (%)

As was mentioned in subsection 5.2.3, the only data considered in the time series is the active and reactive power load, the rest of the data is obtained from the benchmark case of 14 buses, adding noise just as with the noisy benchmark cases. Another important feature is that for every sample, a random branch is disconnected just as

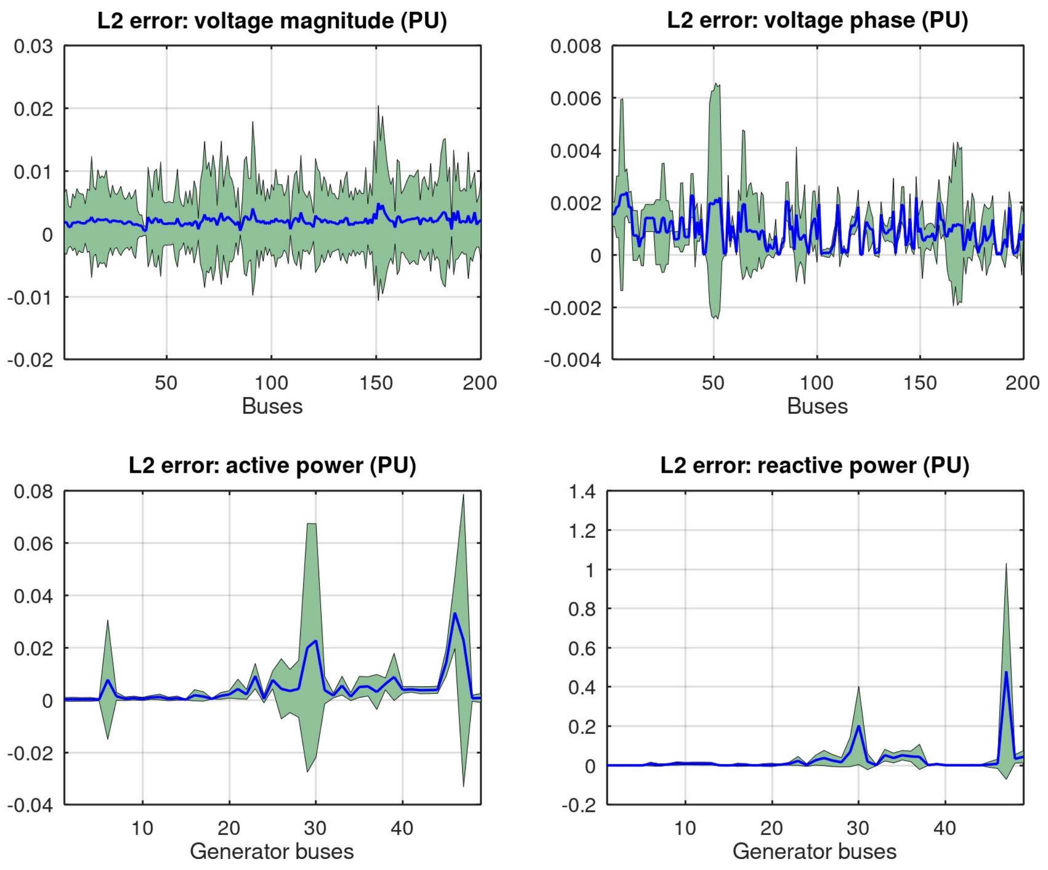


FIGURE 5.6: Mean and SD of OPF L2 error (200-bus case).

with the previous test cases, which would mean a very extreme case of a different branch outage every five minutes.

For this time series test the average and standard deviation of the L2 loss is also analyzed and shown in Fig. (5.30). It can be seen that results are similar to the noisy benchmark case, with the slack bus being the most prone to obtaining solutions that differ from the solution obtained with Matpower. The prediction of reactive power is shown to benefit slightly from the time series data, but other than that most results are only slightly better than with the noisy benchmark case.

5.3.5 Loss function elements of test with time series case

In this subsection the influence of the loss function elements on the time series case results are shown. The main objective function, which is to minimize the branch losses is represented in Fig. (5.31).

It can be seen that the proposed solver obtains results that in which the loss is at least two times smaller than the loss obtained with the Matpower AC OPF solver. Furthermore, the VDs are negligible (in the range of 1×10^{-5}) and are thus not included. The mean active and reactive power imbalance of every bus is shown in Figs. (5.32) and (5.33).

As with the previous examples, the proposed solver achieves significantly lower active and reactive power imbalance in every node. It should be noted that in the time series case, with the added noise and network topologies, most of the samples did not converge in Matpower.

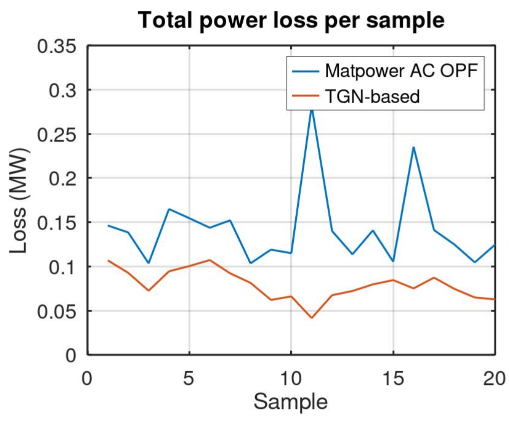


FIGURE 5.7: Power loss (14-bus case).

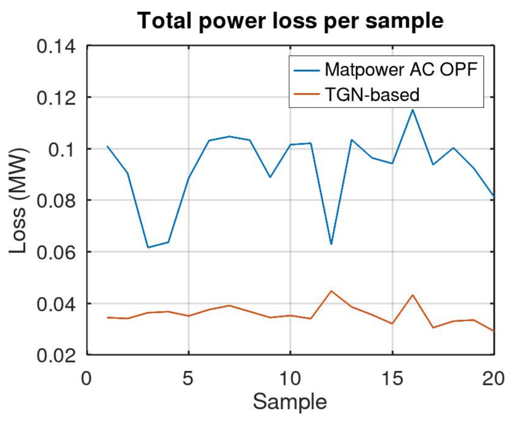


FIGURE 5.8: Power loss (30-bus case).

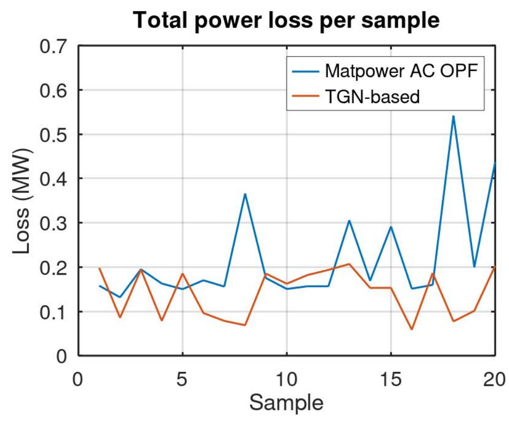


FIGURE 5.9: Power loss (57-bus case).

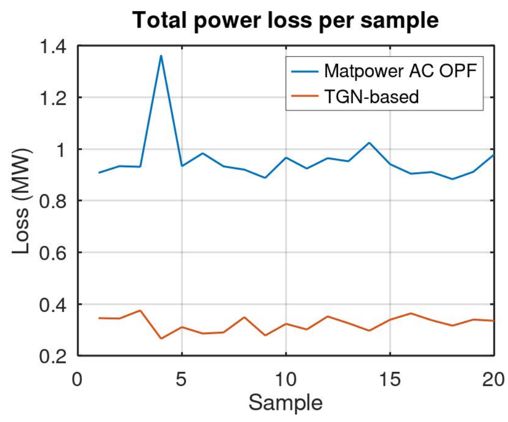


FIGURE 5.10: Power loss (118-bus case).

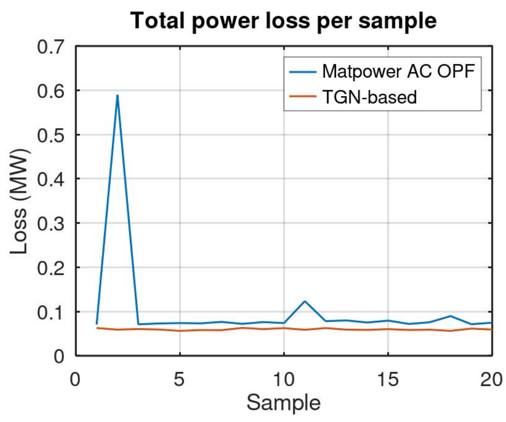


FIGURE 5.11: Power loss (200-bus case).

5.3.6 Time considerations

Table 5.4 illustrates the average time required to find an AC OPF solution with the Matpower AC OPF solver and the proposed TGN-based OPF solver. It is important to note that because of the noise and configuration changes applied to the benchmark cases, the resulting cases are more challenging to solve than their original counterparts. As shown in Table 5.4, even for the smallest grid of only 14 nodes,

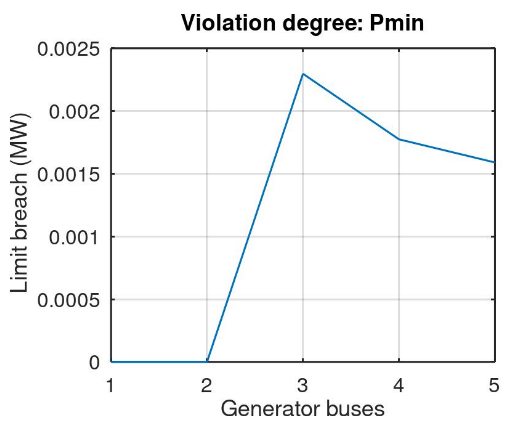


FIGURE 5.12: Power VD (14-bus case).

the proposed solver is nearly 20 times faster than the conventional AC OPF solver. However, for larger grids the scalability of the proposed solution is emphasized. For the largest analyzed case of 200 buses, the proposed solver is more than 1000 times faster than the conventional solver.

Test case	**Matpower AC OPF solver**	**TGN-based OPF solver**
14_ieee	0.0328	1.77×10^{-3}
30_ieee	0.1059	1.798×10^{-3}
57_ieee	0.3327	2.389×10^{-3}
118_ieee	0.5263	3.297×10^{-3}
200_activ	4.6744	3.646×10^{-3}

TABLE 5.4: Average runtime in seconds

It should be noted that the time considered for the proposed solver only includes the computation time of the TGN model with the different electrical grid sizes, but does not include the time it takes to add noise to the data or to generate the adjacency matrices.

Additionally, the training of all PI-TGN instances for this application was very fast, each instance took less than 4000 training iterations (less than 10 minutes) to minimize the loss function. The time is even faster than the PF application case, due to the fact that in this application computations outside the TGN to compute active and reactive power generations are avoided.

5.4 Discussion and limitations

In this chapter the potential of the proposed physics-informed TGN model applied to the AC-OPF problem is investigated. The AC-OPF problem involves finding the optimal values for variables that determine the power flow in an electric power system, while also taking into account the complex and changing interactions between voltage and power flow in the system and the physical and operational constraints that must be followed. This optimization problem is non-convex and nonlinear, meaning that it is more challenging to solve than some other types of optimization problems.

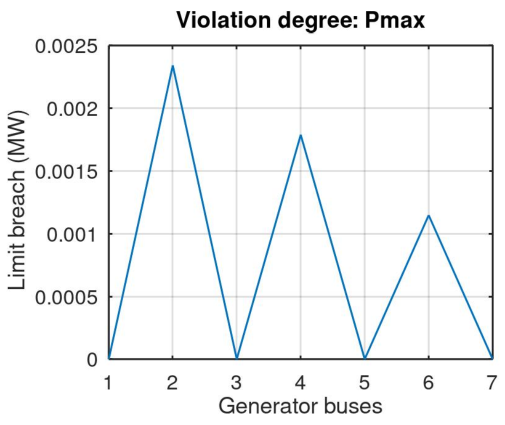


FIGURE 5.13: Power VD (57-bus case).

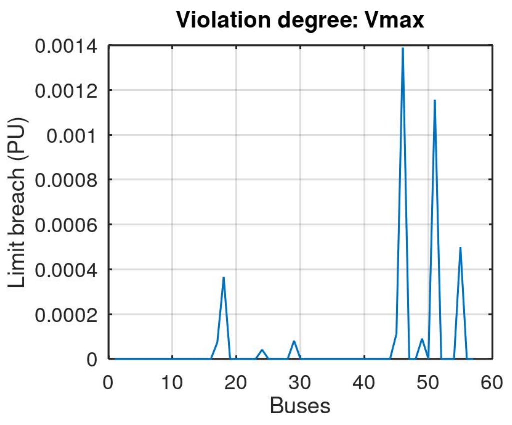


FIGURE 5.14: Voltage VD (57-bus case).

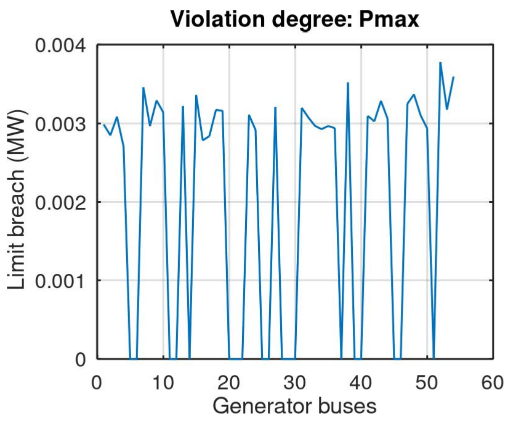


FIGURE 5.15: Power VD (118-bus case).

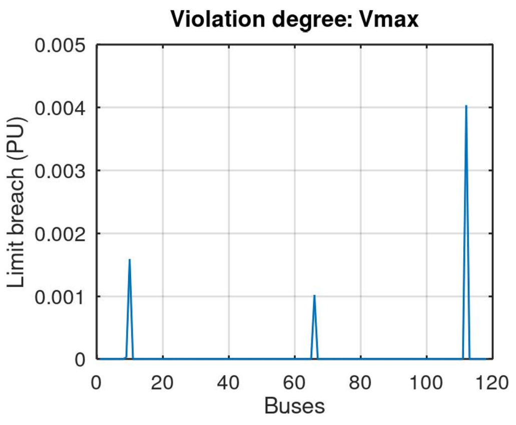


FIGURE 5.16: Voltage VD (118-bus case).

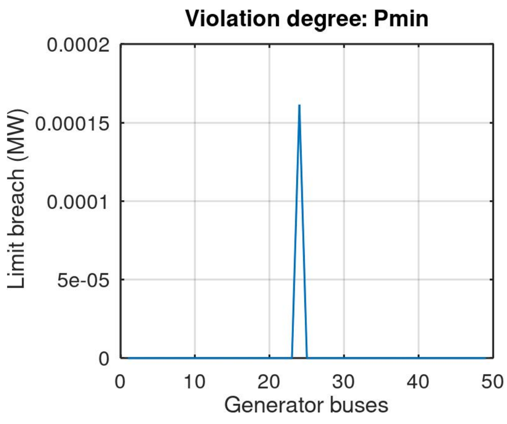


FIGURE 5.17: Power VD (200-bus case).

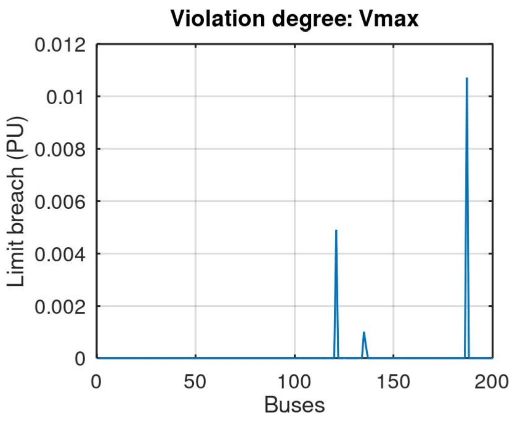


FIGURE 5.18: Voltage VD (200-bus case).

FIGURE 5.19: Different violation degrees of test case.

In this work, the goal of the established optimization problem is to minimize the amount of power lost as it is transmitted through the branches of the power system, while also taking into account the constraints related to the magnitude of the voltage and the operation of the power generation facilities. In order to account for the various constraints that must be considered when optimizing the power flow,

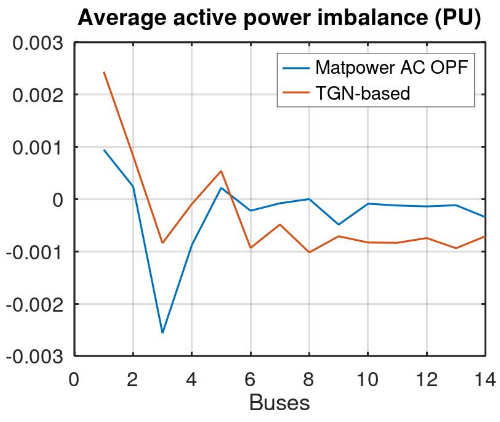


FIGURE 5.20: Active power imbalance (14-bus case).

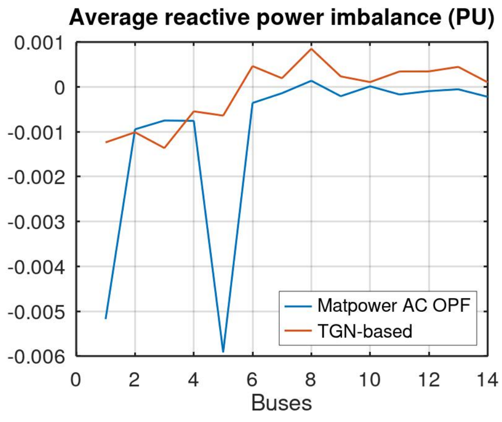


FIGURE 5.21: Reactive power imbalance (14-bus case).

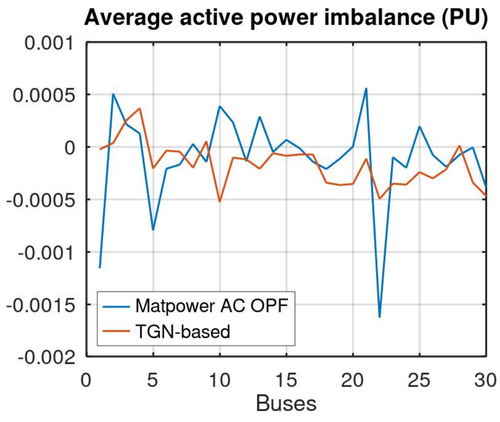


FIGURE 5.22: Active power imbalance (30-bus case).

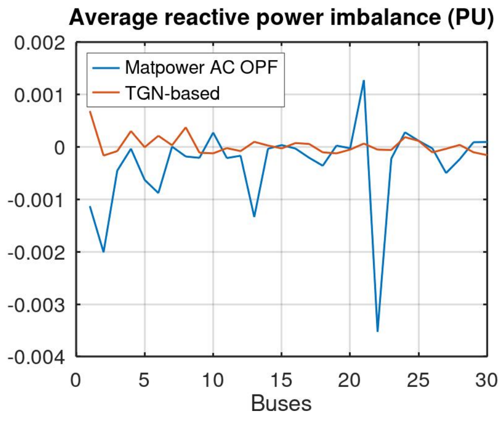


FIGURE 5.23: Reactive power imbalance (30-bus case).

the proposed PI-TGN model is trained with a loss function that incorporates several elements that not only considers the the main optimization objective, but also the established constraints.

The proposed model was tested on a number of different power networks of varying sizes, with a considerable amount of noise added to different variables (including loads, generators, and branch characteristics) and with changing electrical grid configurations in order to evaluate its performance. The tests focused on the ability of the proposed model to accurately predict the voltage magnitude and phase, and active and reactive power generation, while also considering the feasibility for use in real-world operations by analyzing the violation degrees of the operational constraints and the computational time. The model has proven to minimize the multi-target loss function, and to achieve results not very different from a conventional AC-OPF solver, while proving to be scalable and reducing considerably the computation time needed.

It is important to note that the proposed method is completely unsupervised, meaning that it does not rely on any target values in order to find a solution. While this makes the optimization process entirely self-contained, and allows it to be applied without the need of training data, the solutions obtained are not always the same as the solutions obtained with reliable, conventional AC-OPF solvers. The proposed model could profit from including into the loss function an element that depends on target data in order to obtain results closer to those from an already trusted

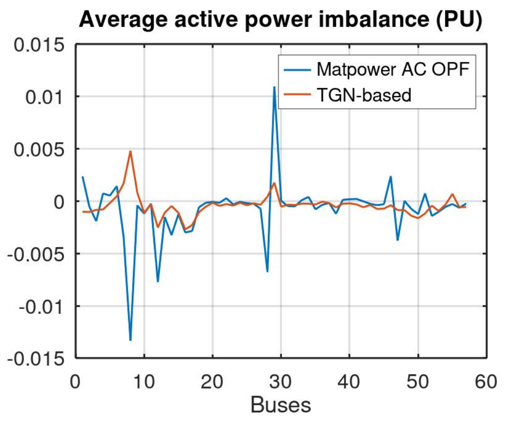


FIGURE 5.24: Active power imbalance (57-bus case).

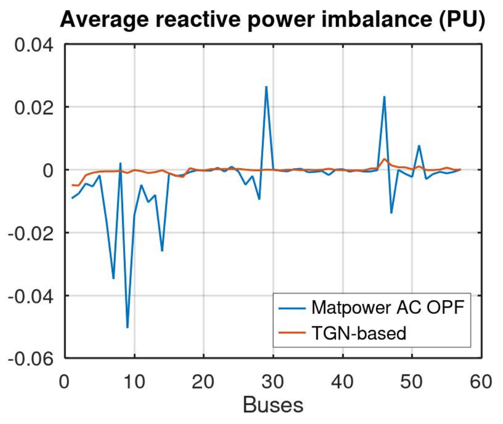


FIGURE 5.25: Reactive power imbalance (57-bus case).

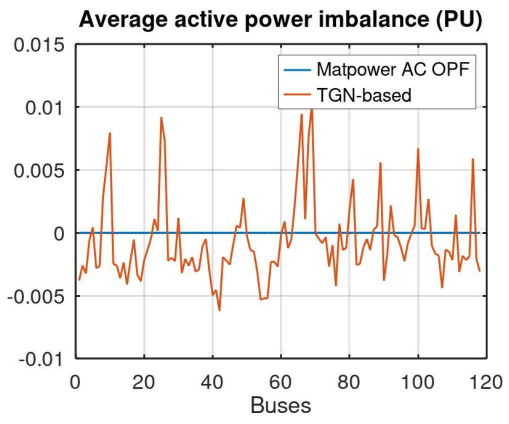


FIGURE 5.26: Active power imbalance (118-bus case).

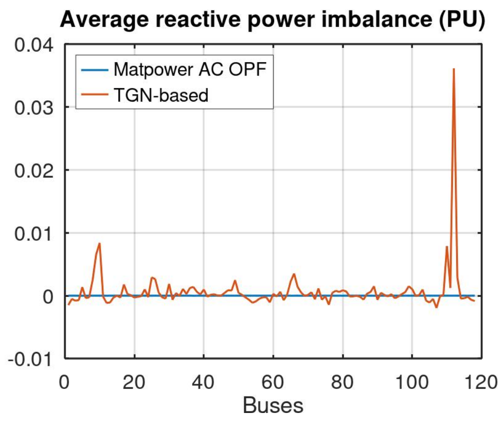


FIGURE 5.27: Reactive power imbalance (118-bus case).

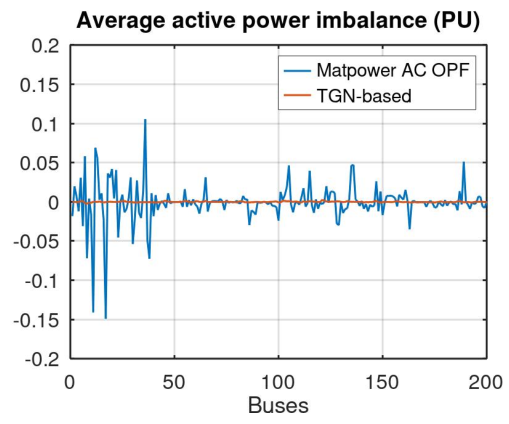


FIGURE 5.28: Active power imbalance (200-bus case).

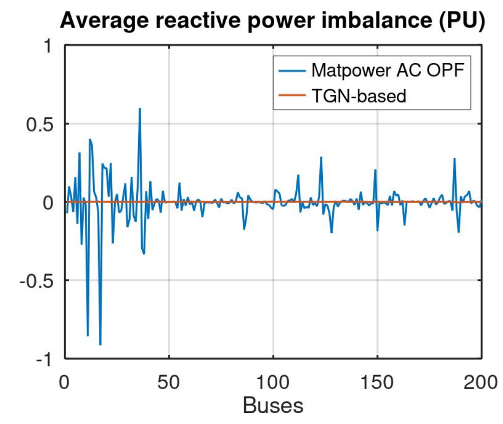


FIGURE 5.29: Reactive power imbalance (200-bus case).

numerical method. Otherwise, more information from the physical system and the constraints could be added to the model inputs and to the loss function. In this work, the branch operational constraints were not considered, but could straightforwardly be added as additional violation degrees.

In contrast to the previous chapter, in this chapter the different model instances were only tested on power networks of the same size as the ones they were trained

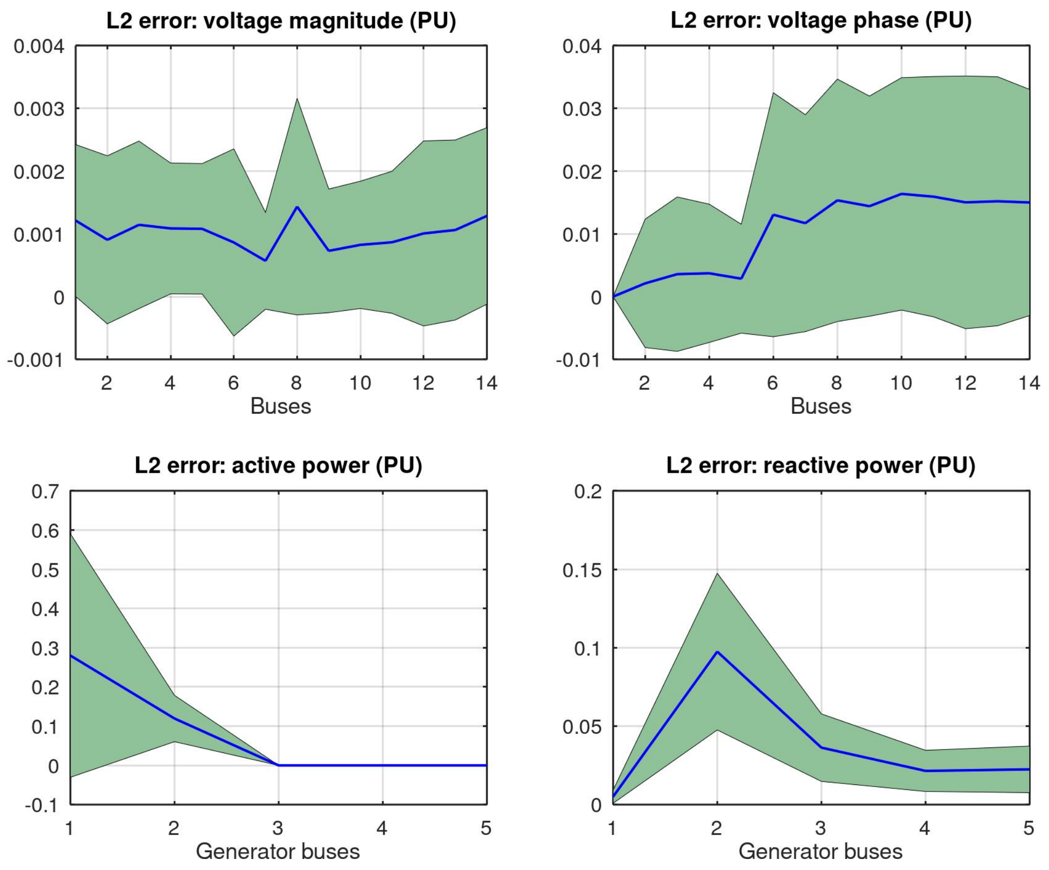


FIGURE 5.30: Mean and SD of OPF L2 error (14-bus time-series case).

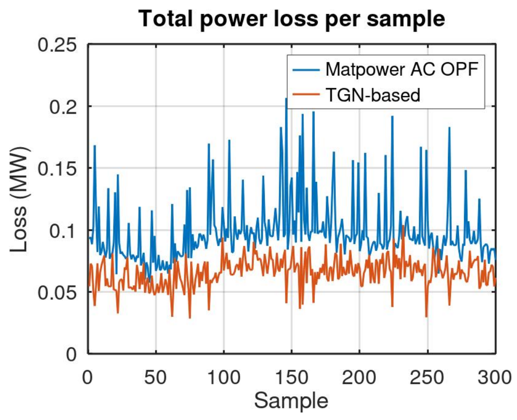


FIGURE 5.31: Power loss (14-bus time-series case).

on. The author attempted to test the model on different size networks, but found that the performance was not satisfactory and that the differences between the networks were too great for the model to be able to generalize effectively.

Additionally, improvement could be made in the implementation of the proposed method in order to test it on larger networks whose entire data sets may be too large to fit in the memory of a graphics processing unit (GPU).

The results of this study suggest that the proposed method may be a promising approach for approximating solutions to the AC-OPF problem, which is a fundamental component of many power system applications such as expansion planning and security assessments. These applications often require the use of a large number

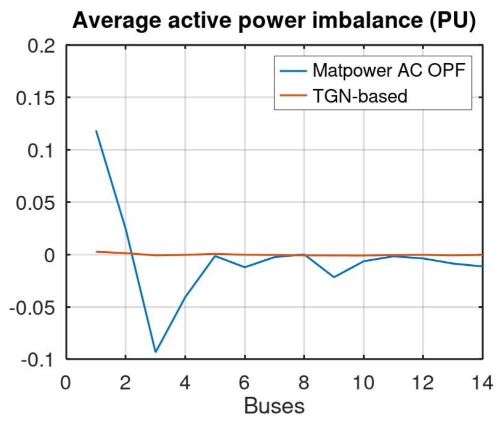


FIGURE 5.32: Active power imbalance (14-bus time-series case).

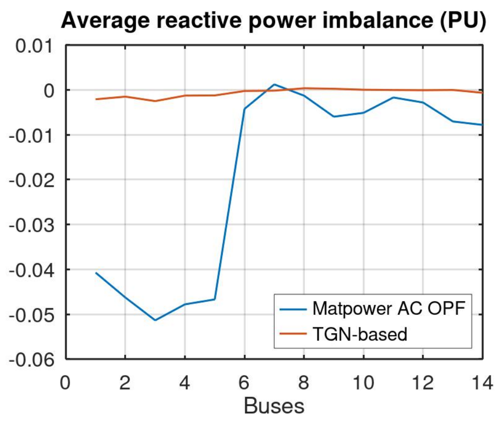


FIGURE 5.33: Reactive power imbalance (14-bus time-series case).

of simulations which can be time-consuming and resource-intensive. The solution to this problem will help to improve the efficiency of the power system operation and ensure that it is working within the established limits.

Chapter 6

Conclusions and Future Work

The main objective of this thesis is to study the applicability of physics-informed typed graph neural networks to help solve the emerging challenges involved in solving the power flow and optimal power flow analyses in increasingly complex power transmission systems.

In the first chapter, some sources of the increasing complexity of the transmission grid were highlighted. Mainly, to address climate change, power grids must become carbon-neutral, relying heavily on inverter-based resources such as solar photovoltaics, wind and other renewable energies. These sources are generally decentralized, their output to be more variable and to have less inertia, than traditional generators. Distributed generation, or the generation of electricity from multiple small, decentralized sources, has the potential to improve power availability and increase the use of renewable and alternative energy sources. However, there are challenges associated with integrating distributed generation into the utility grid in a way that maintains the quality of the generated power. The major power quality issues are voltage, frequency fluctuations, and harmonic currents.

The variable nature of renewable energy sources adds complexity and may cause problems for balancing the supply and demand of power, e.g. distributed generation may cause line overloading in certain periods of time when the sources of energy are abundant. To address the aforementioned challenges, solutions such as the use of batteries, the development of microgrids, and the implementation of smart systems can be employed. These approaches can help to control, manage, and plan operations to achieve more reliable and sustainable power systems. However, it is evident that inverter-based, renewable power sources and the electronics and technology necessary for their efficient and reliable implementation is changing the dynamics of electrical grids, and requiring fast responses and frequent monitoring of the grid. Thus, a new and deep understanding of the effect of these elements is required. Machine learning and other artificial intelligence methodologies can help in characterizing, planning and managing power transmission systems under these new and evolving paradigms.

In this thesis, the two applications for the proposed model are the power flow and optimal power flow problems. On one hand, power flow analyses are essential for transmission planning, and determining the reliability and congestion of the transmission grid. Reliability is related to the ability of the power system to respond to transmission faults without interrupting the load. Congestion occurs when the demand for electricity exceeds the capacity of the transmission lines to deliver it. When congestion occurs, power utilities may need to implement measures to reduce demand or increase higher-cost generation in order to maintain a stable electricity grid. Steady-state analysis, such as the power flow analysis, can be used to inspect if the transmission system can withstand the loss of any single major piece of equipment (such as a transmission line or a transformer) without violating voltage or other equipment loading limits. It can be challenging to perform wide-area event planning using conventional computational methods because of the complexity of the large system models and the potential for a high number of possible contingencies, which can make it difficult to achieve convergence. Additionally, simulations that involve multiple contingencies may produce results that differ significantly from the base case, and there is a risk that the simulation will not converge to a solution. In addition to reliability planning, it is becoming increasingly important to plan for congestion by assigning hourly loads to long periods of time (e.g. an entire year) and performing a load flow for each hour (accounting for generation and transmission line maintenance); each case examined in a congestion study is computationally intensive. These issues accentuate the importance of finding flexible, reliable and more efficient ways to carry out these analyses.

The other important application studied in this thesis is the optimal power flow, which is used to manage the real power output of each controlled generating unit to meet a given load and to minimize a certain criteria, which in the case of the work presented herein is the active power lost through the transmission lines. While the PF problem involves solving a system of equations, the optimal power flow is solved by minimizing a cost function subject to constraints that include the same power equilibrium equations that are the basis for the PF analysis. These computations are important because the solution of the OPF problem allows for:

- Resource allocation: OPF can be used to optimize the allocation of resources within a power system, such as deciding how much power to generate from different sources (e.g., coal, natural gas, solar, wind).
- Integration of renewable energy: OPF can be used to determine the most effective way to integrate renewable energy sources, such as solar and wind, into a power system.
- Transmission expansion planning: OPF can be used to determine the most effective way to expand the transmission grid, such as by adding new transmission lines or upgrading existing ones.
- Unit commitment: OPF can be used to determine which power generation units should be turned on or off at any given time in order to meet demand while minimizing costs or losses.
- Emergency operations: In the event of an emergency, such as a natural disaster or equipment failure, OPF can be used to quickly determine the most effective way to restore power to affected areas.

With the previous considerations, the power transmission grid can be described as a complex network with many elements. It can directly be seen as a graph, where the nodes represent entities such as producers and consumers, and the edges represent power lines and transformers. This network has physical properties that govern the interactions and dynamics between the nodes and the edges. This thesis proposes to integrate mathematical models of the physical properties through a physics-informed loss function and to take advantage of the ability of graph neural networks to reason about explicitly structured data to help identify these dynamics with a fast, scalable and reliable model. The results obtained could become an important building block for future development of a machine learning based tool to aid in multi-scenario planning, risk assessment and generation management.

As mentioned in chapter 2, graph neural networks have shown to be capable to learn structured data and to transfer learned information to cases that go beyond the training conditions (Battaglia et al., 2018). The architecture of these models is directly dependent on the structure of the system to be analyzed which guides the neural network to learn the relationships between the elements, and not just information about the elements independently. This thesis presents a model based on GNNs for analyzing transmission power systems. The use of GNNs has several advantages, including the ability to scale well to larger power grids because the computations are localized rather than global. Additionally, the model is not limited to learning from just one grid configuration, but can be trained and tested on multiple configurations. To add further flexibility to the proposed model, instead of working with conventional GNNs, the TGN framework is chosen and adapted to the problem at hand. TGNs intend to generalize the concept of graph networks, adopting the concept of *node types* instead of constraining the domain to just nodes, edges and universal elements. This is an important concept for working with transmission grids since different types of buses, such as generators, loads and slack buses behave differently and benefit from being modeled by different types of nodes.

Having covered the importance of finding new methods to solve the power flow and optimal power flow problems, and having underlined some beneficial characteristics of graph neural networks which seem to make it appropriate for modeling transmission grids, we can attempt to provide answers to the questions posed in Chapter 1 and afterwards suggest interesting directions for future work.

Research question 1: *Can we develop a computationally efficient, flexible, neural network based model of the transmission grid?*

In chapter 3, the basic and customizable architecture that was developed in this work is presented. The efficiency of the proposed model in both training and inference can be attributed to the fact that it does not require the pre-calculation of target data and does not need to calculate Jacobian matrices (and their inverse). Additionally, the use of graph networks allows for flexibility in the model due to the inherently modular nature of the framework.

In general, a fixed number of stacked TGN layers produce node-level outputs, the first layer takes data from the state of the grid and an initial guess at the values to be inferred, hidden layers take the previous layer state as input, and the last layer produces the final node-level inference. The TGN layers either have shared or independent parameters. For the case of shared parameters, the output from a single TGN layer is used recursively to obtain results. For the case of independent parameters, different instances of the TGN layer are used sequentially to produce outputs. Each TGN layer is comprised of three basic functions: (1) a message passing function that permits the exchange of information between types of nodes, (2) an aggregation function that combines the collection of received messages into a single, fixed-length representation, and (3) a nonlinear update activation function produces node-level representations given the previous node representation and the aggregated information. A decoding function is the final function of each TGN layer, so that the output of each layer can be thought of as an intermediate approximation of the final desired inference values. The TGN approach inherently allows for modularity during training and inference time, i.e. the same NN functions are used for all nodes of the same type, which permits the incorporation or disconnection of elements without the need to train a new model. Furthermore, the number of parameters to train is independent of the size of the electrical grid. The types of nodes, the embedding dimensions, the number of message-passing steps and the total number of layers used are the hyperparameters of the proposed model. Additionally, the TGN model is unsupervised, needing no previously obtained targets for training, thus amplifying the domain of applicability. The unsupervised training paradigm is obtained by either:

- Making the loss function dependant on minimizing the violation of the physical laws that govern the system, in the case of the power flow problem.
- Encoding the optimization problem, including constraints, into the loss function, in the case of the optimal power flow problem.

It can be concluded that the proposed model meets the requirements outlined in the first research question.

Research question 2: *Can the resulting neural network based model be utilized for solving the system of equations necessary for solving the power flow problem?*

Chapter 4 presents the first application of the proposed PI-TGN model: a power flow solver. The objective of the power flow problem is to find all the voltage magnitude and phase values of all buses in the electrical grid, and as a result obtain the unknown active and reactive power values, thus attaining the complete resulting state of the transmission grid. The proposed power flow solver takes advantage of the intuitive connection between the electrical grid data and graph representations to learn the relationships and dynamics between the different types of elements present in electrical grid models to analyze power flow. An important aspect of the presented work is the generalization capability to infer decent results for essentially different grids (varying injection, branch characteristics and topology). The proposed method does not imitate any other existing method, but rather is based on minimizing the active and reactive power imbalance at each node of each sample during the training of the parameters.

The proposed method exploits several benefits of GNNs, e.g. it scales linearly with the size of the grid to be analyzed and the chosen embedding size. Furthermore, since the voltage variables are not directly modified by the proposed method, the computation of Jacobian matrices and their inverse is completely avoided, which is necessary in the conventional N-R method. These two characteristics are key reasons as to why the proposed model computation time scales in a more linear way with respect to the test grid size than the N-R method. It is worth mentioning that the testing of different grid sizes is not possible for conventional multi-layer perceptron models, and that these methods are inefficient for larger grids as their size grows quickly with the grid size. The presented method grows linearly with the size of the electrical grid, due to it being based on local operations and shared modules.

With the simulation tests described in section 4.3, it is shown that the proposed PI-TGN based method obtains results very close to those obtained with a conventional N-R based method, even when the learning is unsupervised. The tests are carried out in batches, with each sample of the batch representing an independent electrical grid from the rest. This way, the proposed system is capable of analyzing many grid states by running many simulations in a parallel manner and is faster than the conventional method when presented with the larger tested electrical grids.

Thus, the results presented in chapter 4 constitute a valuable step towards developing a machine learning based system that is able to assist in analyzing flexible electrical grids of increasing complexity, while improving speed and reducing the computational burden of essential power flow analyses.

However, in future work, steps can be taken to capture sequentiality in time, i.e. instead of the samples being autonomous from each other, if each sample represents the state of the grid over a certain time duration Δt, then H samples would represent the grid state evolution over a total time of $H \cdot \Delta t$ (assuming reliable injection forecast information is available). Additionally, the framework of the proposed model can be improved to continuously learn and adapt to the environment. Finally, another potential way to improve the proposed method would be to improve the way the topology of the electrical grid is represented, so that it allows it to be tested on larger networks.

Research question 3: *Can the resulting neural network based model be utilized to minimize a cost function subject to constraints for solving the optimal power flow problem?*

Chapter 5 presents the second application of the proposed PI-TGN model: an optimal power flow solver. The objective of the studied optimal power flow problem is to find the active power generation values that minimize the loss of power loss through the transmission branches. The solution to the optimal power flow problem must adhere to the power balance equations and certain operational constraints, such as minimum and maximum voltage magnitude values and limits on active and reactive power generation.

Just as with the PF case, the proposed OPF solver uses the connection between electrical grid data and graph representations to understand how different elements in the grid interact and affect each other. The proposed method uses this knowledge to find the correct voltage and power generation values that follow Kirchhoff's current law and reduce power loss through transmission lines, while considering operation limits. The proposed method takes advantage of the benefits of GNNs, such as being able to handle large grids, and scaling well as the size of the grid increases. The proposed method does not imitate any other existing method, and the formulation of the multi-objective loss function encodes the main objective of the OPF and the given constraints.

The computational experiments carried out in section 5.3 were solved both with Matpower as a conventional and trustworthy AC OPF solver and with the proposed PI-TGN based model. The test case results obtained with Matpower were taken as the ground truth to validate the results obtained with the proposed model. While in most cases the error obtained was decently small, the proposed model results were not always the same as the ones obtained with the conventional AC OPF solver. However, it was shown that the components included in the loss function are minimized effectively, which suggests that adding more elements to the loss function (and possibly to the input features of the TGN layers) that better capture the essence of the OPF problem could lead to results that are closer to those obtained using Matpower. There was a noteworthy reduction of computational run time with the proposed model compared to Matpower, with the difference becoming more evident with larger transmission grids.

A possible future line of work could be to change the learning formulation and make the model only partially unsupervised. In other works dealing with the OPF problem using neural networks (Fioretto, Mak, and Hentenryck, 2020; Nellikkath and Chatzivasileiadis, 2022), some target values are provided, and either completely or partially make up the loss function. A hybrid approach that emphasizes the physics-informed elements but also includes some training targets could combine the benefit of reducing the training space while expanding the range of possible applications (by allowing the evaluation of points without target data using the mathematical equations of the physics-informed elements). Meanwhile, the targets could help guide the model in producing results that are more similar to those obtained using traditional methods, thus exploiting the structure of the underlying system but also taking advantage of "experience" learned from conventional methods.

On the other hand, even though the same hyperparameters were used to train the different instances of the PI-TGN model for each particular electrical grid case, and the proposed OPF solver is computationally able to obtain a result for grid cases of different sizes, it was found that satisfactory results were not obtained when testing with cases of different sizes from the ones encountered during training. This shows that the generalization capabilities of the proposed model for this application are reduced in comparison with the simpler PF problem.

An important future line of work is to improve the implementation of the proposed model to be able to test it on very large networks. We encountered memory problems when attempting to scale up computations to larger grids; developing a computational framework that enables the compression of sparse data structures could help address this issue.

In conclusion, the results presented are encouraging as they suggest that the proposed PI-TGN solver has the potential to significantly reduce the computational complexity and, more importantly, the time it takes to obtain an AC OPF solution with fidelity to the elements included in the loss function. The results obtained with this application of the proposed model constitutes a beneficial step towards developing a tool for real-time dispatch and techno-economic analyses, or as an aid to improve the different stages of power system planning, optimization, operation and control of electrical grids.

Chapter 7

Conclusiones y Trabajo Futuro

El objetivo principal de esta tesis es estudiar la aplicabilidad de las redes neuronales de grafos tipados informadas por la física para ayudar a resolver los retos emergentes implicados en la resolución de los análisis de flujo de potencia y flujo de potencia óptimo en sistemas de transmisión de energía eléctrica que son cada vez más complejos.

En el primer capítulo, se destacan algunas fuentes de la creciente complejidad en los sistemas de transmisión eléctrica. Principalmente, para hacer frente al cambio climático, las redes eléctricas deben obtener neutralidad de carbono y apoyarse en gran medida de recursos basados en inversores, como la energía solar fotovoltaica, la eólica y otras energías renovables. Estas fuentes de energía son descentralizadas, su producción es más variable y tienen menos inercia que los generadores tradicionales. La generación distribuida, o generación de electricidad a partir de múltiples fuentes pequeñas y descentralizadas, cuenta con gran potencial para mejorar la disponibilidad de energía y aumentar el uso de fuentes de energía renovables y alternativas. Sin embargo, la integración de la generación distribuida en la red de suministro plantea dificultades para mantener la calidad de la energía generada. Los principales problemas de calidad a considerar son la tensión, las fluctuaciones de frecuencia y las corrientes armónicas.

La naturaleza variable de las fuentes de energía renovables añade complejidad y puede causar problemas para equilibrar la oferta y la demanda de energía, como ejemplo: la generación distribuida puede provocar sobrecarga de líneas de transmisión en determinados periodos de tiempo cuando las fuentes de energía son abundantes. Para hacer frente a los retos mencionados, pueden emplearse soluciones como el uso de baterías, el desarrollo de microrredes y el uso de sistemas inteligentes. Dichas propuestas pueden ayudar a controlar, gestionar y planificar las operaciones necesarias para obtener sistemas eléctricos fiables y sostenibles. Sin embargo, es evidente que las fuentes de energía renovables basadas en inversores y las tecnologías necesarias para su incorporación están cambiando la dinámica de las redes eléctricas, exigen respuestas rápidas y una supervisión frecuente de la red. Así, se requiere una comprensión nueva y profunda del efecto de estos elementos. El aprendizaje automático y otras metodologías de inteligencia artificial pueden ayudar a caracterizar, planificar y gestionar los sistemas de transmisión eléctrica bajo estos cambiantes paradigmas.

En esta tesis, las dos aplicaciones para el sistema propuesto son la de flujo de potencia y flujo de potencia óptimo. Por una parte, el flujo de potencia es esencial para determinar la fiabilidad y la congestión de la red. La fiabilidad está relacionada con la capacidad del sistema eléctrico de responder a los fallos de transmisión sin interrumpir la carga. La congestión se produce cuando la demanda de electricidad supera la capacidad de las líneas de transmisión para suministrarla. Cuando se produce una congestión, es posible que las compañías eléctricas tengan que aplicar medidas para reducir la demanda o aumentar la generación de mayor coste con el fin de mantener una red eléctrica estable. El análisis de estado estacionario, como el análisis de flujo de potencia, puede utilizarse para inspeccionar si el sistema de transmisión puede soportar la pérdida de una sola pieza importante del equipo (como una línea de transmisión o un transformador) sin violar límites de tensión u otros límites de carga del equipo. La planificación de posibles eventos adversos en áreas extensas con métodos computacionales convencionales puede resultar complicada debido a la complejidad de los modelos de sistemas de gran tamaño y al elevado número de contingencias posibles, lo cual dificulta la convergencia. Además, las simulaciones que implican múltiples contingencias pueden producir resultados que difieren significativamente del caso base, y existe el riesgo de que la simulación no converja a una solución. Además de la planificación de la fiabilidad, cada vez es más importante planificar la congestión asignando perfiles de carga de largos periodos de tiempo (por ejemplo, un año entero) y realizando un flujo de carga para cada hora (teniendo en cuenta la generación y el mantenimiento de las líneas de transmisión); cada caso examinado en un estudio de congestión es intensivo desde el punto de vista computacional. Estas cuestiones acentúan la importancia de encontrar formas flexibles, fiables y más eficientes de llevar a cabo estos análisis en vista de los cambiantes paradigmas relativos a los sistemas de transmisión de energía.

La otra aplicación importante estudiada en esta tesis es el flujo de potencia óptimo, que se utiliza para gestionar la potencia real de salida de cada unidad generadora controlada para satisfacer una carga determinada y minimizar un determinado criterio, que en el caso del trabajo aquí presentado es la pérdida de potencia activa a través de las líneas de transmisión. Mientras que el problema del flujo de potencia implica resolver un sistema de ecuaciones, el flujo de potencia óptimo se resuelve minimizando una función de costo sujeta a restricciones que incluyen las mismas ecuaciones de equilibrio de potencia que son la base del análisis del flujo de potencia. Estos cálculos son importantes porque la solución del problema de flujo de potencia óptimo permite:

- Asignación de recursos: Se puede utilizar para optimizar la asignación de recursos dentro de un sistema eléctrico; como para decidir cuánta energía generar a partir de diferentes fuentes (por ejemplo, carbón, gas natural, solar, eólica).

- Integración de energías renovables: Se puede utilizar para determinar la forma más eficaz de integrar fuentes de energía renovables, como la solar y la eólica, en un sistema eléctrico.

- Planificación de la expansión de la transmisión: Se puede utilizar para determinar la forma más eficaz de ampliar la red de transporte, por ejemplo añadiendo nuevas líneas o mejorando las existentes.

- Compromiso unitario: Se puede utilizar para determinar qué unidades de generación de energía deben encenderse o apagarse en cada momento para satisfacer la demanda de energía, minimizando los costes o las pérdidas.

- Operaciones de emergencia: En caso de emergencia, como una catástrofe natural o un fallo de los equipos, el análisis de flujo de potencia óptimo puede utilizarse para determinar rápidamente la forma más eficaz de restablecer el suministro eléctrico en las zonas afectadas.

Con las consideraciones anteriores, la red de transporte de electricidad puede describirse como una red compleja con numerosos elementos. Puede verse directamente como un grafo, en el que los nodos representan entidades como generadores y cargas, y las aristas representan líneas eléctricas y transformadores. Esta red tiene propiedades físicas que rigen las interacciones y la dinámica entre los nodos y las aristas. Esta tesis propone integrar modelos matemáticos de las propiedades físicas mediante una función de pérdida informada por la física y aprovechar la capacidad de las redes neuronales de grafos para razonar sobre datos explícitamente estructurados con el fin de ayudar a identificar estas dinámicas con un modelo rápido, escalable y fiable. Los resultados obtenidos podrían convertirse en un elemento importante para el desarrollo futuro de una herramienta basada en el aprendizaje automático que ayude en la planificación multiescenario, la evaluación de riesgos y la gestión de la generación.

Como se mencionó en el capítulo 2, las redes neuronales de grafo han demostrado ser capaces de aprender datos estructurados y transferir la información aprendida a casos que van más allá de las condiciones de entrenamiento (Battaglia et al., 2018). La arquitectura de estos modelos depende directamente de la estructura del sistema a analizar, lo cual guía a la red neuronal para aprender las relaciones entre los elementos, y no sólo información sobre los elementos de forma independiente. Esta tesis presenta un modelo basado en redes neuronales de grafo para el análisis de sistemas de transmisión eléctrica. El uso de redes de grafo neuronales conlleva varias ventajas, incluyendo la capacidad de escalar fácilmente a redes eléctricas más grandes ya que los cálculos son localizados en lugar de globales. Además, el modelo no se limita a aprender una sola configuración de red, sino que puede entrenarse y probarse en múltiples configuraciones. Para añadir más flexibilidad al modelo propuesto, en lugar de trabajar con redes neuronales de grafo convencionales, se elige el marco de grafos neuronales tipados, adaptado a las aplicaciones propuestas. Las redes neuronales de grafos tipadas generalizan el concepto de redes de grafo, adoptando el concepto de *tipos de nodos* en lugar de restringir el dominio a sólo nodos, aristas y elementos universales. Esto es benéfico al lidiar con redes de transmisión, ya que los distintos tipos de buses, como generadores, cargas y buses *slack* se comportan de forma diferente.

Habiendo tratado la importancia de encontrar nuevos métodos para resolver los problemas de flujo de potencia y de flujo de potencia óptimo, y habiendo subrayado algunas características beneficiosas de las redes neuronales de grafos que parecen hacerlas apropiadas para modelar las redes de transmisión, podemos intentar dar respuesta a las preguntas planteadas en el Capítulo 1 y sugerir posibles direcciones interesantes para trabajos futuros.

Pregunta de investigación 1: *¿Podemos desarrollar un modelo de red de transporte de energía eléctrica que sea flexible, computacionalmente eficiente y basado en redes neuronales?*

En el capítulo 3, se presenta la arquitectura básica y customizable que se ha desarrollado en este trabajo. La eficiencia del modelo propuesto tanto en el entrenamiento como al momento de inferencia puede atribuirse al hecho de que no requiere el cálculo previo de datos objetivo y no necesita calcular matrices Jacobianas (ni sus inversas). Además, el uso de redes de grafos permite flexibilizar el modelo debido a su naturaleza intrínsecamente modular.

En general, un número fijo de capas apiladas de redes de grafos neuronales tipadas (TGN) producen salidas a nivel de nodo, la primera capa toma datos del estado de la red y una conjetura inicial sobre los valores que deben inferirse, las capas ocultas toman como entrada el estado de la capa anterior y la última capa produce la inferencia final a nivel de nodo. Las capas TGN pueden tener parámetros compartidos o independientes. En el caso de los parámetros compartidos, la salida de una sola capa TGN se utiliza recursivamente para obtener resultados. En el caso de parámetros independientes, se utilizan secuencialmente diferentes instancias de la capa TGN para obtener resultados. Cada capa TGN se compone de tres funciones básicas: (1) una función de paso de mensajes que permite el intercambio de información entre diferentes tipos de nodos, (2) una función de agregación que combina los mensajes recibidos en una única representación de longitud fija, y (3) una función de activación de actualización no lineal que produce representaciones a nivel de nodo dado el estado anterior del nodo y la información de los mensajes. Una función de decodificación es la función final de cada capa TGN, de modo que la salida de cada capa puede considerarse una aproximación intermedia de los valores finales de inferencia deseados. El enfoque con TGNs permite intrínsecamente la modularidad durante el entrenamiento y el tiempo de inferencia, esto es porque se utilizan las mismas funciones NN para todos los nodos del mismo tipo, lo que permite incorporar o desconectar elementos sin necesidad de entrenar un nuevo modelo. Además el número de parámetros a entrenar es independiente del tamaño de la red eléctrica analizada. Los tipos de nodos, las dimensiones ocultas, el número de pasos de envío de mensajes y el número total de capas utilizadas son los hiperparámetros del modelo propuesto. Además, el modelo TGN es no supervisado, por lo que no necesita valores objetivo obtenidos previamente para su entrenamiento, lo cual amplía el dominio de aplicabilidad. El paradigma de entrenamiento no supervisado se obtiene mediante:

- Para el caso de flujo de potencia: la función de costo depende de minimizar la violación de las leyes físicas que gobiernan el sistema.
- Para el caso de flujo de potencia óptimo: codificando el problema de optimización, incluidas las restricciones, en la función de costo.

Se puede concluir que el modelo propuesto cumple los requisitos esbozados en la primera pregunta de investigación.

Pregunta de investigación 2: *¿Puede utilizarse el modelo neuronal propuesto para resolver el sistema de ecuaciones necesario para obtener una solución al problema del flujo de potencia?*

El capítulo 4 presenta la primera aplicación del modelo de redes neuronales de grafo tipadas informadas por la física (PI-TGN) propuesto: un método para resolver el problema de flujo de potencia. El objetivo de dicho análisis es encontrar todos los valores de magnitud de tensión y fase de todas las subestaciones de la red eléctrica, y como resultado obtener los valores desconocidos de potencia activa y reactiva, obteniendo así el estado resultante completo de la red de transporte de electricidad. El método propuesto aprovecha la relación directa entre los datos de la red eléctrica y su representación gráfica para aprender las relaciones y la dinámica entre los distintos tipos de elementos presentes en los modelos de red eléctrica, y así resolver el flujo de potencia. Un aspecto importante del trabajo presentado es la capacidad de generalización para inferir buenos resultados para redes esencialmente diferentes (variando la inyección, las características de las líneas de transmisión y la topología). El método propuesto no imita ningún otro método existente, sino que se basa en minimizar el desequilibrio de potencia activa y reactiva en cada nodo de cada muestra durante el entrenamiento de los parámetros.

El método propuesto aprovecha varias ventajas de las redes de grafo neuronales, por ejemplo, se escala linealmente con el tamaño de la red eléctrica que se va a analizar y la dimensión del estado embebido. Además, dado que el método propuesto no modifica directamente las variables de tensión, se evita por completo el cálculo de matrices jacobianas y sus inversas, necesario en el método Newton-Raphson (N-R) convencional. Estas dos características son razones claves por las que el tiempo de cálculo del modelo propuesto es más lineal con respecto al tamaño de la red que el método N-R. Cabe mencionar que para los modelos neuronales convencionales, como el perceptrón multicapa, no es posible analizar redes eléctricas de diferentes tamaños, y que estos métodos son ineficientes para redes grandes, ya que su tamaño crece rápidamente con respecto al número de subestaciones y líneas de transmisión. El método presentado crece linealmente con respecto al tamaño de la red eléctrica, ya que se basa en operaciones locales y módulos compartidos.

Con las pruebas de simulación descritas en la sección 4.3, se demuestra que el método propuesto basado en PI-TGN obtiene resultados muy cercanos a los obtenidos con un método convencional basado en N-R, incluso cuando el aprendizaje es no supervisado. Las pruebas se realizan por lotes, cada muestra representa un estado de red eléctrica independiente del resto. De esta forma, el sistema propuesto es capaz de analizar muchos estados de red ejecutando muchas simulaciones de forma paralela y es más rápido que el método convencional cuando se le presentan las redes eléctricas probadas más grandes.

Así pues, los resultados presentados en el capítulo 4 constituyen un valioso paso hacia el desarrollo de un sistema basado en el aprendizaje automático capaz de ayudar en el análisis de redes eléctricas flexibles y de complejidad creciente, mejorando al mismo tiempo la velocidad y reduciendo la carga computacional de los análisis esenciales del flujo de potencia.

Sin embargo, en futuros trabajos, pueden tomarse medidas para captar la secuencialidad en el tiempo, es decir, en lugar de que las muestras sean independientes entre sí, si cada muestra representa el estado de la red durante una determinada duración de tiempo Δt, entonces H muestras representarían la evolución del estado de la red durante un tiempo total de $H \cdot \Delta t$ (suponiendo que se dispone de información fiable de previsión de carga y generación). Además, el marco del modelo propuesto puede mejorarse para que aprenda continuamente y se adapte al entorno. Por último, otra forma potencial de mejorar el método propuesto sería mejorar la forma de representar la topología de la red eléctrica, de modo que permita probarlo en redes más grandes.

Pregunta de investigación 3: *¿Puede utilizarse el modelo basado en la red neuronal resultante para minimizar una función de costo sujeta a restricciones para resolver el problema del flujo de potencia óptimo?*

El capítulo 5 presenta la segunda aplicación del modelo PI-TGN propuesto: un solucionador de flujo de potencia óptimo. El objetivo del problema de flujo de potencia óptimo estudiado es encontrar los valores de generación de potencia activa que minimicen la pérdida de potencia a través de las ramas de transmisión. La solución del problema de flujo de potencia óptimo debe respetar las ecuaciones de balance de potencia y ciertas restricciones operativas, como los valores mínimo y máximo de magnitud de tensión y los límites de generación de potencia activa y reactiva.

Al igual que en el caso de flujo de carga, el método propuesto para resolver el OPF utiliza la conexión entre los datos de la red eléctrica y su representación gráfica para deducir cómo interactúan y se influyen mutuamente los distintos elementos de la red. El método propuesto utiliza esta información para encontrar los valores correctos de tensión y generación de potencia que sigan la ley de corrientes de Kirchhoff y reduzcan las pérdidas de potencia a través de las líneas de transmisión, teniendo en cuenta al mismo tiempo los límites de funcionamiento. El método propuesto aprovecha las ventajas de las GNN, como su capacidad para gestionar grandes redes y su escalabilidad a medida que aumenta el tamaño de la red. El método propuesto no imita métodos existentes, y la formulación de la función de costo multiobjetivo codifica el objetivo principal del OPF y las restricciones dadas.

Los experimentos computacionales llevados a cabo en la sección 5.3 se resolvieron tanto con Matpower, el cual es un *solver* de OPF convencional y fiable, como con el modelo propuesto basado en PI-TGN. Los resultados de los casos de prueba obtenidos con Matpower se tomaron como los resultados reales para validar los resultados obtenidos con el modelo propuesto. Aunque en la mayoría de los casos el error obtenido fue decentemente pequeño, los resultados del modelo propuesto no siempre fueron los mismos que los obtenidos con el solver AC OPF convencional. Sin embargo, se demostró que los componentes incluidos en la función de costo se minimizan eficazmente, lo que sugiere que añadir más elementos a dicha función de costo (y posiblemente a la entrada del TGN) podría conducir a captar de mejor manera la esencia el problema OPF, y llevar a resultados más cercanos a los obtenidos con Matpower. Se ha observado una notable reducción del tiempo de ejecución computacional con el modelo propuesto en comparación con Matpower, siendo la diferencia más evidente con redes de transmisión más grandes.

Una posible línea de trabajo futura podría ser cambiar la formulación de entrenamiento de forma que el modelo sea sólo parcialmente no supervisado. En otros trabajos que abordan el problema OPF utilizando redes neuronales (Fioretto, Mak, and Hentenryck, 2020; Nellikkath and Chatzivasileiadis, 2022), se proporcionan algunos valores objetivo, los cuales componen completa o parcialmente la función de costo. Un enfoque híbrido que haga hincapié en los elementos informados por la física pero que también incluya algunos objetivos de entrenamiento podría combinar la ventaja de reducir el espacio de entrenamiento al tiempo que amplía la gama de posibles aplicaciones (al permitir la evaluación de puntos que no cuenten con valores objetivo, al utilizar las ecuaciones matemáticas de los elementos informados por la física en esos casos). Mientras tanto, los valores objetivo podrían ayudar a guiar el modelo para producir resultados más parecidos a los obtenidos con métodos tradicionales, explotando así la estructura del sistema subyacente pero aprovechando también la "experiencia" aprendida con los métodos convencionales.

Por otro lado, aunque se utilizaron los mismos hiperparámetros para entrenar las diferentes instancias del modelo PI-TGN para cada caso particular de red eléctrica, y el solucionador OPF propuesto es computacionalmente capaz de obtener un resultado para casos de red de diferentes tamaños, se encontró que no se obtienen resultados satisfactorios cuando se probaba con redes de transmisión de tamaños diferentes a los usados durante el entrenamiento. Esto demuestra que la capacidad de generalización del modelo propuesto para esta aplicación es reducida en comparación con el problema de flujo de potencia que es más simple.

Una importante línea de trabajo futura es mejorar la implementación del modelo propuesto para poder probarlo en redes muy grandes. Nos encontramos con problemas de memoria al intentar escalar los cálculos a redes más grandes; el desarrollo de un marco computacional que permita la compresión de estructuras de datos dispersas podría ayudar a resolver este problema (las matrices de adyacencia son dispersas).

En conclusión, los resultados presentados son alentadores ya que sugieren que el solucionador PI-TGN propuesto tiene el potencial de reducir significativamente la complejidad computacional y, lo que es más importante, el tiempo que se tarda en obtener una solución AC OPF con fidelidad a los elementos incluidos en la función de pérdida. Los resultados obtenidos con esta aplicación del modelo propuesto constituyen un paso beneficioso hacia el desarrollo de una herramienta para el despacho en tiempo real y los análisis tecno-económicos, o como ayuda para mejorar las distintas etapas de la planificación, optimización, operación y control de las redes eléctricas.

Appendix A

Publications List

International Journal Publications included in the Journal Citations Reports (JCR)

1. Lopez-Garcia, Tania B.; Coronado-Mendoza, Alberto; Domínguez-Navarro, José A. **Artificial neural networks in microgrids: A review.** *Engineering Applications of Artificial Intelligence,* vol. 95, 103894, 2020.

2. Lopez-Garcia, Tania B.; Domínguez-Navarro, José A. **Power flow analysis via typed graph neural networks.** *Engineering Applications of Artificial Intelligence,* vol. 117, 105567, 2023.

International Conference Publications

1. Lopez-Garcia, Tania B.; Domínguez-Navarro, José A. **Graph Neural Network Power Flow Solver for Dynamical Electrical Networks.** *2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON),* 825-830, 2022.

Bibliography

Afonso, Tiago Lopes, António Cardoso Marques, and José Alberto Fuinhas (Aug. 2019). "Accommodating renewable energy sources in a small electricity market: An analysis considering the interactions of sources within Portugal and Spain". In: *Heliyon* 5.8, e02354. ISSN: 24058440. DOI: 10.1016/j.heliyon.2019.e02354.

Al-Othman, A.K. and M.R. Irving (Aug. 2006). "Analysis of confidence bounds in power system state estimation with uncertainty in both measurements and parameters". In: *Electric Power Systems Research* 76.12, pp. 1011–1018. ISSN: 03787796. DOI: 10.1016/j.epsr.2005.12.016.

Amerongen, R.A.M. van (May 1989). "A general-purpose version of the fast decoupled load flow". In: *IEEE Transactions on Power Systems* 4.2, pp. 760–770. ISSN: 08858950. DOI: 10.1109/59.193851.

Babaeinejadsarookolaee, Sogol et al. (Aug. 2019). "The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms". In.

Babatunde, O.M., J.L. Munda, and Y. Hamam (Feb. 2020). "Power system flexibility: A review". In: *Energy Reports* 6, pp. 101–106. ISSN: 23524847. DOI: 10.1016/j.egyr.2019.11.048.

Baker, Kyri (June 2021). "Solutions of DC OPF are Never AC Feasible". In: *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. New York, NY, USA: ACM, pp. 264–268. ISBN: 9781450383332. DOI: 10.1145/3447555.3464875.

Baran, M.E. and F.F. Wu (1989). "Optimal capacitor placement on radial distribution systems". In: *IEEE Transactions on Power Delivery* 4.1, pp. 725–734. ISSN: 08858977. DOI: 10.1109/61.19265.

Battaglia, Peter W. et al. (2016). "Interaction Networks for Learning about Objects, Relations and Physics". In: *arXiv*.

Battaglia, Peter W. et al. (June 2018). "Relational inductive biases, deep learning, and graph networks". In: *arXiv*. URL: https://arxiv.org/pdf/1806.01261.pdf.

Bienstock, Daniel and Abhinav Verma (Nov. 2019). "Strong NP-hardness of AC power flows feasibility". In: *Operations Research Letters* 47.6, pp. 494–501. ISSN: 01676377. DOI: 10.1016/j.orl.2019.08.009.

Birchfield, Adam B. et al. (July 2017). "Grid Structural Characteristics as Validation Criteria for Synthetic Networks". In: *IEEE Transactions on Power Systems* 32.4, pp. 3258–3265. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2016.2616385.

Borges, Carmen Lucia Tancredo and Vinícius Ferreira Martins (Mar. 2012). "Multistage expansion planning for active distribution networks under demand and Distributed Generation uncertainties". In: *International Journal of Electrical Power & Energy Systems* 36.1, pp. 107–116. ISSN: 01420615. DOI: 10.1016/j.ijepes.2011.10.031.

Bregere, Margaux and Ricardo J. Bessa (2020). "Simulating Tariff Impact in Electrical Energy Consumption Profiles With Conditional Variational Autoencoders". In: *IEEE Access* 8, pp. 131949–131966. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3009060.

Chowdhury, B.H. and S. Rahman (1990). “A review of recent advances in economic dispatch”. In: *IEEE Transactions on Power Systems* 5.4, pp. 1248–1259. ISSN: 08858950. DOI: 10.1109/59.99376.

Dimitrovski, Aleksandar, Kevin Tomsovic, and Alfredo Vaccaro (2011). “Reliable Algorithms for Power Flow Analysis in the Presence of Data Uncertainties”. In: *Innovations in Power Systems Reliability*. Ed. by George Anders and Alfredo Vaccaro. Springer, pp. 329–357.

Donon, Balthazar et al. (Dec. 2020). “Neural networks for power flow: Graph neural solver”. In: *Electric Power Systems Research* 189. ISSN: 03787796. DOI: 10.1016/j.epsr.2020.106547.

Fikri, Meriem et al. (Oct. 2018). “Power flow analysis by numerical techniques and artificial neural networks”. In: *3rd Renewable Energies, Power Systems and Green Inclusive Economy, REPS and GIE 2018*. Institute of Electrical and Electronics Engineers Inc. ISBN: 9781538685150. DOI: 10.1109/REPSGIE.2018.8488870.

Fioretto, Ferdinando, Terrence W K Mak, and Pascal Van Hentenryck (2020). “Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods”. In: *AAAI Conference on Artificial Intelligence*. DOI: https://doi.org/10.1609/aaai.v34i01.5403. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5403.

Fisher, E.B., R.P. O’Neill, and M.C. Ferris (Aug. 2008). “Optimal Transmission Switching”. In: *IEEE Transactions on Power Systems* 23.3, pp. 1346–1355. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2008.922256.

Giannakis, Georgios B. et al. (2013). “Monitoring and optimization for power grids: A signal processing perspective”. In: *IEEE Signal Processing Magazine* 30.5, pp. 107–128. ISSN: 10535888. DOI: 10.1109/MSP.2013.2245726.

Gilmer, Justin et al. (Apr. 2017). “Neural Message Passing for Quantum Chemistry”. In: *International Conference on Machine Learning*. Ed. by JMLR.org. Sydney, pp. 1263–1272. DOI: 10.5555/3305381.3305512.

Glover, J. Duncan, Mulukutla S. Sarma, and Thomas J. Overbye (2012). *Power System Analysis and Design*. 5th. Cengage Learning. ISBN: 978-1-111-42579-1.

Gomez-Exposito, Antonio, Antonio J. Conejo, and Claudio Canizares (2009). *Electric Energy Systems Analysis and Operation*. 2nd. Taylor & Francis Group.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. MIT Press. URL: http://www.deeplearningbook.org.

Gori, M., G. Monfardini, and F. Scarselli (2005). “A new model for learning in graph domains”. In: *IEEE International Joint Conference on Neural Networks*. IEEE, pp. 729–734. ISBN: 0-7803-9048-2. DOI: 10.1109/IJCNN.2005.1555942.

Hansen, Jonas Berg, Stian Normann Anfinsen, and Filippo Maria Bianchi (2022). “Power Flow Balancing With Decentralized Graph Neural Networks”. In: *IEEE Transactions on Power Systems*. ISSN: 15580679. DOI: 10.1109/TPWRS.2022.3195301.

Heymann, Fabian et al. (Apr. 2021). “Forecasting Energy Technology Diffusion in Space and Time: Model Design, Parameter Choice and Calibration”. In: *IEEE Transactions on Sustainable Energy* 12.2, pp. 802–809. ISSN: 1949-3029. DOI: 10.1109/TSTE.2020.3020426.

Heymann, Fabian et al. (Jan. 2023). “Digitalization in decarbonizing electricity systems – Phenomena, regional aspects, stakeholders, use cases, challenges and policy options”. In: *Energy* 262, p. 125521. ISSN: 03605442. DOI: 10.1016/j.energy.2022.125521.

Hong, Tao and Shu Fan (July 2016). “Probabilistic electric load forecasting: A tutorial review”. In: *International Journal of Forecasting* 32.3, pp. 914–938. ISSN: 01692070. DOI: 10.1016/j.ijforecast.2015.11.011.

Hu, Xinyue et al. (May 2021). “Physics-Guided Deep Neural Networks for Power Flow Analysis”. In: *IEEE Transactions on Power Systems* 36.3, pp. 2082–2092. ISSN: 15580679. DOI: 10.1109/TPWRS.2020.3029557.

Jianhui Wang, M. Shahidehpour, and Zuyi Li (Aug. 2008). “Security-Constrained Unit Commitment With Volatile Wind Power Generation”. In: *IEEE Transactions on Power Systems* 23.3, pp. 1319–1327. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2008.926719.

Karniadakis, George Em et al. (June 2021). *Physics-informed machine learning*. DOI: 10.1038/s42254-021-00314-5.

Kipf, Thomas et al. (Feb. 2018). “Neural Relational Inference for Interacting Systems”. In: *International Conference on Machine Learning*. URL: http://arxiv.org/abs/1802.04687.

Köhnen, Clara Sophie et al. (Mar. 2022). “The potential of deep learning to reduce complexity in energy system modeling”. In: *International Journal of Energy Research* 46.4, pp. 4550–4571. ISSN: 0363-907X. DOI: 10.1002/er.7448.

Lopez-Garcia, Tania B., Alberto Coronado-Mendoza, and José A. Domínguez-Navarro (Oct. 2020). “Artificial neural networks in microgrids: A review”. In: *Engineering Applications of Artificial Intelligence* 95. ISSN: 09521976. DOI: 10.1016/j.engappai.2020.103894.

Lopez-Garcia, Tania B. and José A. Domínguez-Navarro (Jan. 2023). “Power flow analysis via typed graph neural networks”. In: *Engineering Applications of Artificial Intelligence* 117, p. 105567. ISSN: 09521976. DOI: 10.1016/j.engappai.2022.105567.

Marot, Antoine et al. (Dec. 2020). “Learning to run a power network challenge for training topology controllers”. In: *Electric Power Systems Research* 189, p. 106635. ISSN: 03787796. DOI: 10.1016/j.epsr.2020.106635.

Nellikkath, Rahul and Spyros Chatzivasileiadis (2022). “Physics-Informed Neural Networks for AC Optimal Power Flow”. In: *22nd Power Systems Computation Conference PSCC*. DOI: 10.1016/j.epsr.2022.108412.

Niharika, Sumit Verma, and Vivekananda Mukherjee (Apr. 2016). “Transmission expansion planning: A review”. In: *2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS)*. IEEE, pp. 350–355. ISBN: 978-1-5090-1534-4. DOI: 10.1109/ICEETS.2016.7583779.

Owerko, Damian, Fernando Gama, and Alejandro Ribeiro (Oct. 2020). “Optimal Power Flow Using Graph Neural Networks”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5930–5934. DOI: 10.1109/ICASSP40776.2020.9053140.

Prates, Marcelo O. R. et al. (Jan. 2019). “Typed Graph Networks”. In: *arXiv*. DOI: https://doi.org/10.48550/arXiv.1901.07984. URL: https://doi.org/10.48550/arXiv.1901.07984.

Sakaguchi, T. and K. Matsumoto (Feb. 1983). “Development of a Knowledge Based System for Power System Restoration”. In: *IEEE Transactions on Power Apparatus and Systems* PAS-102.2, pp. 320–329. ISSN: 0018-9510. DOI: 10.1109/TPAS.1983.317770.

Santoro, Adam et al. (2017). “A simple neural network module for relational reasoning”. In: *arXiv*.

Sauer, Peter W., M.A. Pai, and Joe H. Chow (2017). *Power System Dynamics and Stability: With Synchropasor Measurement and Power System Toolbox*. 2nd. John Wiley & Sons.

Scarselli, F. et al. (Jan. 2009). “Computational Capabilities of Graph Neural Networks”. In: *IEEE Transactions on Neural Networks* 20.1, pp. 81–102. ISSN: 1045-9227. DOI: 10.1109/TNN.2008.2005141.

Schäfer, Benjamin et al. (Sept. 2022). “Understanding Braess’ Paradox in power grids”. In: *Nature Communications* 13.1, p. 5396. ISSN: 2041-1723. DOI: 10.1038/s41467-022-32917-6.

Schweppe, Fred and J. Wildes (Jan. 1970). “Power System Static-State Estimation, Part I: Exact Model”. In: *IEEE Transactions on Power Apparatus and Systems* PAS-89.1, pp. 120–125. ISSN: 0018-9510. DOI: 10.1109/TPAS.1970.292678.

Simonovsky, Martin and Nikos Komodakis (Apr. 2017). “Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs”. In: URL: http://arxiv.org/abs/1704.02901.

Smith, Oliver et al. (Mar. 2022). “The effect of renewable energy incorporation on power grid stability and resilience”. In: *Science Advances* 8.9. ISSN: 2375-2548. DOI: 10.1126/sciadv.abj6734.

Stott, Brian (1974). “Review of Load-Flow Calculation Methods”. In: *Proceedings of the IEEE* 62.7, pp. 916–929. ISSN: 15582256. DOI: 10.1109/PROC.1974.9544.

Sweeney, Conor et al. (Mar. 2020). “The future of forecasting for renewable energy”. In: *WIREs Energy and Environment* 9.2. ISSN: 2041-8396. DOI: 10.1002/wene.365.

Topping, Jake et al. (2022). “Understanding Over-squashing and Bottlenecks on Graphs via Curvature”. In: *International Conference on Learning Representations*.

Tovar-Facio, Javier, Mariano Martín, and José María Ponce-Ortega (Mar. 2021). “Sustainable energy transition: modeling and optimization”. In: *Current Opinion in Chemical Engineering* 31, p. 100661. ISSN: 22113398. DOI: 10.1016/j.coche.2020.100661.

Vaccaro, Alfredo and Domenico Villacci (Feb. 2009). “Radial Power Flow Tolerance Analysis by Interval Constraint Propagation”. In: *IEEE Transactions on Power Systems* 24.1, pp. 28–39. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2008.2009383.

Van Hertem, D. (2006). “Usefulness of DC power flow for active power flow analysis with flow controlling devices”. In: *8th IEE International Conference on AC and DC Power Transmission (ACDC 2006)*. IEE, pp. 58–62. ISBN: 0 86341 613 6. DOI: 10.1049/cp:20060013.

Vankayala, Vidya Sagar S. and Nutakki D. Rao (1993). “Artificial neural networks and their applications to power systems—a bibliographical survey”. In: *Electric Power Systems Research* 28.1, pp. 67–79.

Vinuesa, Ricardo et al. (Jan. 2020). “The role of artificial intelligence in achieving the Sustainable Development Goals”. In: *Nature Communications* 11.1, p. 233. ISSN: 2041-1723. DOI: 10.1038/s41467-019-14108-y.

Wan, Yih-huei and B.K. Parsons (Aug. 1993). *Factors relevant to utility integration of intermittent renewable technologies*. Tech. rep. Golden, CO: National Renewable Energy Laboratory (NREL). DOI: 10.2172/10186066.

Wang, Gang et al. (Jan. 2019). “Distribution system state estimation: an overview of recent developments”. In: *Frontiers of Information Technology & Electronic Engineering* 20.1, pp. 4–17. ISSN: 2095-9184. DOI: 10.1631/FITEE.1800590.

Wildberger, M. (Oct. 1994). “Stability and nonlinear dynamics in power systems”. In: *IEEE Power Engineering Review* 14.10, pp. 16–18. ISSN: 0272-1724. DOI: 10.1109/39.318698.

Xie, Le et al. (Aug. 2021). “Toward carbon-neutral electricity and mobility: Is the grid infrastructure ready?” In: *Joule* 5.8, pp. 1908–1913. ISSN: 2542-4351. DOI: 10.1016/J.JOULE.2021.06.011.

Zhang, Z.Z., G. S. Hope, and O. P. Malik (1989). "Expert systems in electric power systems - a bibliographical survey". In: *IEEE Transactions on Power Systems* 4.4, pp. 1355–1362. ISSN: 15580679. DOI: 10.1109/59.41685.

Zimmerman, Ray Daniel, Carlos Edmundo Murillo-Sanchez, and Robert John Thomas (Feb. 2011). "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education". In: *IEEE Transactions on Power Systems* 26.1, pp. 12–19. ISSN: 0885-8950. DOI: 10.1109/TPWRS.2010.2051168.

Zuiderwijk, Anneke, Yu-Che Chen, and Fadi Salem (July 2021). "Implications of the use of artificial intelligence in public governance: A systematic literature review and a research agenda". In: *Government Information Quarterly* 38.3, p. 101577. ISSN: 0740624X. DOI: 10.1016/j.giq.2021.101577.