Universidade Federal do Espírito Santo

Technological Center

Graduate Program in Electrical Engineering

Rogerio José Menezes Alves

Graph Machine Learning Models Applied on the Identification of Critical Transmission Lines in Electrical Power Systems

Vitória, ES

2022

Rogerio José Menezes Alves

Graph Machine Learning Models Applied on the Identification of Critical Transmission Lines in Electrical Power Systems

Dissertation presented to the Graduate Program in Electrical Engineering of the Universidade Federal do Espírito Santo as a partial requirement for the degree of Master in Electrical Engineering.

Universidade Federal do Espírito Santo

Technological Center

Graduate Program in Electrical Engineering

Advisor: Marcia Helena Moreira Paiva

Vitória, ES

2022

Ficha catalográfica disponibilizada pelo Sistema Integrado de
Bibliotecas - SIBI/UFES e elaborada pelo autor

A474g

Alves, Rogerio José Menezes, 1995-
Graph machine learning models applied on the identification of critical transmission lines in electrical power systems / Rogerio José Menezes Alves. - 2022.
103 f. : il.

Orientadora: Marcia Helena Moreira Paiva.
Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Aprendizado do computador. 2. Sistemas de energia elétrica. I. Paiva, Marcia Helena Moreira. II. Universidade Federal do Espírito Santo. Centro Tecnológico. III. Título.

CDU: 621.3

Rogerio José Menezes Alves

Graph Machine Learning Models Applied on the Identification of Critical Transmission Lines in Electrical Power Systems

Dissertation presented to the Graduate Program in Electrical Engineering of the Universidade Federal do Espírito Santo as a partial requirement for the degree of Master in Electrical Engineering.

Vitória, ES, April 4th, 2022:

M. Paiva

Profa. Dra. **Marcia Helena Moreira Paiva**
Universidade Federal do Espírito Santo
Advisor

M. Paiva

p/ **Profa. Dra. Elizete Maria Lourenço**
Universidade Federal do Paraná
External Examiner

[illegible]

Prof. Dr. Jorge Leonid Aching Samatelo
Universidade Federal do Espírito Santo
Internal Examiner

Vitória, ES
2022

Acknowledgements

First and more importantly, I thank my family and dear friends for all the support through all the years of my academic studies that resulted in this dissertation. Without their support and encouragement, this text would not be written. A special acknowledgement to my dear Joyce Meireles, that stood with me everyday and never gave up supporting me on this journey.

I thank my advisor Marcia Helena Moreira Paiva for all her patience in guiding me during these years, since 2014. Her example of dedicated researcher and friend was decisive for me to not give up on the task until the end. A special thanks to Prof. Gilles Caporossi for the time spent at GERAD, since the research environment that I found greatly contributed to my understanding of scientific development.

I also thank the professors and staff from the Department of Electrical Engineering at UFES that contributed for my knowledge and moral construction. Their effort allowed me to now act as an electrical engineer and contribute to the society that we share.

In addition, I thank the professors and the staff at LabTel for all the material support, and more recently at CPID. Without the provided computational infrastructure, this work would not be accomplished.

For those that I did not mention but also contributed for my journey, thank you!

Rogerio José Menezes Alves

"Instrua-se, porque necessitaremos de toda a sua inteligência. Atue, porque necessitaremos de todo o seu entusiasmo. Organize-se, porque necessitaremos de toda a sua força."

(Antonio Gramsci)

Abstract

When the operation of electric power systems is concerned, one may associate security with predicting a future event, and then be always ready to what may happen in the future. The number of possible events increases even further when not only single contingencies are evaluated, but also multiple contingencies. Therefore, one must seek for the balance between desired security level and practical cost-efectiveness. A critical event that is not mapped will not be analyzed and might represent an issue in the system's event response mechanisms. On the other hand, the screening is vital because the number of possible contingency scenarios turns impractical to make an exhaustive detailed simulation approach. One alternative approach to make analyses on power systems is to consider topological aspects instead of, or even together with electrical information from the systems. In this case, graph models of the power systems can be constructed and evaluated. In this dissertation, Graph Machine Learning techniques are evaluated on graph models constructed from the system data, aiming to classify critical and non-critical transmission lines. First, a literature review is made, highlighting the main models and methods, which are then adapted and applied. Then, a set of test systems that are commonly used in benchmarks is selected for the evaluation step. Three approaches for learning architectures are proposed, given that the formulated learning problem is not directly treated in the known literature. The learning approaches are trained for reproducing a known criticality index for transmission lines in a graph model, and the obtained results are analyzed. Finnaly, conclusion about the obtained results are made and possible future research themes are proposed.

Keywords: power systems. security. graph theory. complex networks. graph machine learning.

Resumo

No que diz respeito à operação de sistemas elétricos de potência, pode-se associar segurança à previsão de um evento futuro e, assim, estar sempre pronto para possíveis contingências. O número de eventos possíveis aumenta ainda mais quando não apenas contingências únicas são avaliadas, mas também múltiplas contingências. Portanto, deve-se buscar o equilíbrio entre o nível de segurança desejada e o custo associado. Um evento crítico que não seja mapeado não será analisado e pode representar um problema nos mecanismos de resposta a eventos do sistema. Por outro lado, a triagem é vital porque o número de cenários de contingência possíveis torna impraticável fazer uma abordagem exaustiva de simulação detalhada. Uma abordagem alternativa para fazer análises em sistemas de potência é considerar aspectos topológicos ao invés de, ou mesmo em conjunto com informações elétricas dos sistemas. Neste caso, modelos de grafos dos sistemas de potência podem ser construídos e avaliados. Nesta dissertação, técnicas de Aprendizado de Máquina em Grafos são avaliadas em modelos de grafos construídos a partir dos dados do sistema, visando realizar a classificação de linhas de transmissão críticas e não críticas, com relação a contingências simples e múltiplas. Primeiramente é feita uma revisão da literatura, levantando os modelos e métodos que serão adaptados e avaliados. Em seguida, um conjunto de sistemas de teste que são comumente usados em *benchmarks* é selecionado para aplicação dos métodos de aprendizagem. Três abordagens para arquiteturas de aprendizado são propostas, tendo em vista que o problema de aprendizagem que foi construído, na forma de classificação de arestas, não é diretamente tratado na literatura visitada. As arquiteturas foram treinadas para reproduzir um índice de criticidade conhecido para linhas de transmissão em um modelo de grafo e, ao final, os resultados obtidos são analisados. Por fim, conclui-se sobre os resultados que foram obtidos nas estratégias adotada e são propostas possíveis pesquisas futuras.

Palavras-chave: sistemas de potência. segurança. teoria de grafos. redes complexas. aprendizado de máquina em grafos.

List of Figures

List of Tables

List of abbreviations and acronyms

EPS	Electric Power System
SIN	Brazilian Interconnected System
RB	Basic Network
DC	Direct Current
CFB	Current Flow Betweenness Centrality
ER	Erdos-Renyi
BA	Barabasi-Albert
LE	Laplacian Eigenmaps
GML	Graph Machine Learning
HOPE	High-Order Proximity Preserved Embedding
GNN	Graph Neural Network
MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
GCN	Graph Convolutional Network
GIN	Graph Isomorphism Network
RGCN	Relational Graph Convolutional Network
GAT	Graph Attention Network
CI	Contingency Impact
GI	Global Impact
MSE	Mean Squared Error
MAE	Mean Absolute Error

Contents

1 Introduction

Since mid 19th century, when they were initially build to provide lighting in urban areas, the Electric Power Systems (EPS) have grown, not only in size, but also in complexity around the world. The first EPS to operate in scale aggregated generation, distribution and loads in a spatially limited urban area, not being able to power a whole district (ANDRADE; LEAO, 2012). Currently, some EPS cover continental areas, connecting consumers through several thousands kilometers of transmission lines, being operated to ensure power stability and quality.

In order to enable such scale of EPS, many devices and technologies have been developed over the decades. Nowadays, every EPS contains not only generating units, distribution lines and loads, but a huge set of transformers, reactors, capacitor banks and protection devices such as circuit breakers, fuses and protection relays (PAPAILIOU, 2021). This increase in the number of devices, together with the growth in scale, makes the reliable operation of EPS more challenging every day.

1.1 The Need for Electric Power

Electric power is an essential resource for daily life in the modern world. Since late 1990s electric power is already considered an necessity for most activities in a daily routine (GUY; MARVIN, 1998). More than an ease in people's life, it has been a key resource for powering transport systems, communications, food production, hospitals and many processes that happen to enable life in modern society (ZOHURI; MCDANIEL, 2019).

When there is a threat to the electric power supply, many structures in modern society are affected. For instance, prices can begin to oscilate due to inflation (VIECELI, 2021), factories may have to interrupt their activities due to preventive power rationing decisions and social chaos can happen when a lasting power outage occurs, as happened recently in the northen brazilian state of Amapá, where a transformer short-circuit isolated the entire state from the interconnected transmission system for 22 days in late 2020s. During this period, the population suffered with problems in the water distribution system, food transport and storage and property security issues (G1, 2020; ONS, 2020).

Therefore, as essential as the electric power is for the modern society, also it is vital to keep the systems that enable the daily use of electric power in normal operation. However, due to the size and complexity of those systems, meeting this necessity becomes a challenge.

1.2 Challenges in EPS Operations

In order to meet the demand for broad electrical power supply, EPS have grown in size and scale. In order to enable the modern EPS operation, the number of installed field devices, sensors and control applications increase each day. For the Brazilian Interconnected System's (SIN) Basic Network (RB), keeping the system with all devices in operation is nearly impossible. This system has over 3000 installed field devices among transformers, buses, capacitor banks, reactors and syncronous and static compensators to support more than 1000 transmission lines, with more than 100,000 km of extension (ONS, 2021).

With a large scale EPS, device failure will occur almost every day. However, it is not expected that the system suffer a major breakdown every time a device fails. It is expected that some devices are more critical than others for the entire system operation. This may happen in the case of a large generating unit, transmission line, or a transformer that lowers the voltage for an important load.

In the case of failure in such devices, it is expected that the system reacts as fast as possible to restore the normal operation. However, this cannot be always done by repairing the damaged device. Some events require the system to recover only through electrical maneuvers, changing the operation state of other devices to re-enable the supply for the affected region. This can only happen if a specific study regarding that contingency scenario has been previously developed, and all the actions that must be taken in that case are already mapped.

Given the number of devices and transmission lines in modern EPS, it is not practical to evaluate all possible scenarios and make detailed studies about every contingency that might occur. This kind of study usually is made through complex simulations, that consider a great number of parameters and models for the system, and the system expansion alone makes this approach impractical due to changes in the existing devices and the model parameters. Therefore, a certain number of scenarios, namely the most critical scenarios must be evaluated and studied in detail.

1.3 Security in Electric Power Systems

In the context of Electric Power Systems, one can define security for many specific cases. For instance, when analyzing devices, the security can be associated with robustness, meaning that the device will not break or enter an defectuous state if the propper maintanence is made. If analyzing locations, such as substations, security can be associated with not allowing unauthorized entrance or, if it happens, not allowing the intruder to make any harmful maneuvers.

When the operation of the EPS is concerned, one may associate security with predicting a future event, and then be always ready to what may happen in the future. For instance, in the morning the amount of power generated by photovoltaic plants may increase by degrees of magnitude in the time interval of minutes, so a secure operation would be to already prepare the devices and transmission lines around these plants to be able to transmit all this power when it reaches the system.

Considering all the aforementioned cases, when dealing with the EPS as a whole, one can define security as the ability of the power system to withstand contingencies (BALU et al., 1992). This definition is broad in the sense it admits that contingencies will occur. Otherwise, it could be rewritten as "the ability of the power system to avoid contingencies".

When associating security with withstanding contingencies, obtaining a more secure EPS becomes a task deeply entangled with having robust and solid contingency plans. In this context, the power system operators are required to continuosly evaluate their contingency strategies to ensure the fastest response to a contingency scenario that might not have happened before. The number of possible events increases even further when not only single contingencies are evaluated, but also multiple contingencies.

In order to have the most secure EPS as possible, one might suggest to develop the system to withstand all possible multiple contigencies, but that may be economically unfeasible. Therefore, one must seek for the balance between desired security level and practical cost-efectiveness. For normal operation, an EPS with N components is considered to have moderate security if the outage of any single component does not affect the normal operating state. This is known as the $N-1$ criterion, which is the most used criterion in contingency analysis (WOOD; WOLLENBERG; SHEBLÉ, 2013).

Some improvements to the $N-1$ criterion have been proposed in the literature, but besides generalizing the analyzed contingencies to $N-k$ with $k>1$, other modifications in the way of evaluating the criticality of each contingency have been made, such as the optimal transmission switching (KHANABADI; GHASEMI; DOOSTIZADEH, 2013) and the transmission capacity expansion planning (MAJIDI-QADIKOLAI; BALDICK, 2016b; MAJIDI-QADIKOLAI; BALDICK, 2016a).

1.4 Contingency Analysis

A number of common approaches in evaluating scenarios to be considered in EPS contingency analysis are based on performance indices (DORAISWAMI; CARVALHO, 1979; EJEBE; WOLLENBERG, 1979), which are non-monotonic functions and can have unpredictable behavior for multiple contingencies (BULAT; FRANKOVIć; VLAHINIć, 2021). Even though the EPS configuration changes in time, the operators have pratical knowledge about the system's behavior in many contingency scenarios, and can predict which cases may be worth of studying based on which cases were studied earlier.

Although some other criteria may be considered in evaluating the contigency scenarios, such as the devices that have the greatest rated power capacities, or the ones nearest the largest power sources, some methods can be found in the literature to rank the most critical scenarios for a given EPS, such as the sensitivity factors (WOOD; WOLLENBERG; SHEBLÉ, 2013), which approximate changes in line flows through a DC load flow aiming to detect possible overloads in generators and transmission lines. This initial step, of evaluating the scenarios that will be studied and simulated in detail, is called the screening step (EJEBE et al., 1996).

The screening step is critical to the contingency analysis and security assessment of power systems, mainly because it constrains the future study scenarios on a limited set of contingencies. A critical event that is not mapped in the screening step will not be analyzed in further simulations, and might represent an issue in the system's event response mechanisms. On the other hand, the screening is vital because the number of possible contingency scenarios turns impractical to make an exhaustive detailed simulation approach.

As the security assessment of EPS mostly uses the $N-1$ criterion, contingency screening is also generally made in terms of single contingency scenarios, where the system operates with a single component in failure state. This criterion by itself already provides a broad search space for the contingency screening step, where the number of possible contingencies equals the number of components in the EPS.

However, the number of scenarios becomes even larger when the system operates with $N-k$ components, with $k > 1$. These are called multiple contingency (multi-contingency) scenarios. In order to make a viable evaluation of a multi-contingency scenario, Yang, Guan e Zhai (2017) developed a method for DC grid security assessment using Optimal Power Flow by constructing a reduced set of representative constraints based on the network parameters. This approach enabled the analysis of multi-contingency scenarios in DC networks, but did not provide a way of ranking the most severe multi-contingencies or the most critical elements in the system with respect to single and multi-contigencies.

Even though the probability of a multi-contingency $(N-k)$ is much smaller than a single contingency $(N-1)$, decreasing with the value of k, one may need to evaluate such scenarios and, therefore, may have efficient methods to study these cases and rank a subset of contingencies.

One approach of evaluating and ranking the most severe contingencies and most critical devices in the EPS was developed using centrality measures from graph theory on graph models of the EPS. Analyses that are made in graph models are called *topological* since the results are obtained by considering the topology of the connections between the system's elements instead of power flow simulations.

In Gorton et al. (2009), Jin et al. (2010), new methods and algorithms for calculating a previously existing centrality measure, the *betweenness centrality*, and through this measure obtaining the most relevant devices in the system, are proposed. These methods provided speed-ups in parallel computing environments and for specific graph models of the EPS. In Cetinay, Kuipers e Mieghem (2016), spectral graph theory is used for estimating sensitivities of electrical variables by line and bus removal. In Wang, Scaglione e Thomas (2010), modifications on classical graph centrality measures are proposed for EPS vulnerability analysis.

More than one centrality measure is evaluated in Coelho et al. (2019), where the *betweenness*, *closeness* and *current-flow betweenness* centralities are applied in the same critical node and bus exhaustive identification method and the results are compared with Bompard, Wu e Xue (2011) and Yan, He e Sun (2014), indicating that the method had a high rate of agreement.

In order to extend the topological contingency analysis to greater networks and to reduce computation time, an alternative implementation of the proposed method using metaheuristics was proposed in Coelho et al. (2022), showing a reasonable agreement with the exhaustive method for a given set of standard test power systems.

In this dissertation, a Graph Machine Learning (GML) approach is evaluated for solving the contingency screening problem in EPS graph models, for the single and multiple contingency cases. A known Criticality Index, proposed in Coelho (2019) is considered and machine learning methods are applied instead of using metaheuristics, as in Coelho et al. (2022). The considered approaches are based on the formulation of an edge classification problem, which is not a classical problem in the Graph Machine Learning literature, since the extensively studied problems are either the node or graph classification (HAMILTON, 2020). The methodology is applied to a set of benchmark test systems and the learning results are compared to the exhaustive approach from Coelho (2019).

1.5 Objectives

The main objective of this dissertation is to extend the topological multi-contigency analysis and evaluation method proposed in Coelho (2019) by using Graph Machine Learning techniques on a formulated edge classification problem. Instead of using metaheuristics, alternative approaches are proposed using graph embedding and Graph Neural Networks.

The specific objectives can be listed as follows:

- Overview and consider the usage of complex network generation methods for improving the learning methods.

- Overview and analyze current graph, node and edge embedding techniques for further steps of the contingency screening methodology.
- Define graph for evaluating the methodology and compare the obtained results with the exhaustive topological contingency screening approach from (COELHO, 2019).
- Evaluate different approaches for the contingency screening problem through edge classification with Graph Machine Learning and other alternative formulations.

1.6 Text Organization

This dissertation is organized as follows:

- **Introduction**: makes an overview and contextualizes the contingency analysis problem and the chosen approach to solve it.
- **Graph Theory Overview and Graph Models of Power Systems**: summarizes the relevant aspects of graph theory useful for the comprehension of the remaining text, such as graph transformations and centrality measures.
- **Graph Machine Learning Fundamentals**: makes an introduction to the graph machine learning resources that are evaluated and used in the proposed methodology for the multi-contingency analysis.
- **Methodology**: explains the chosen approach and the evaluated alternatives to solve the problem, together with the chosen test data.
- **Results**: exposes the behavior of the criticality index in the chosen test data and the obtained results for the learning approaches.
- **Conclusion and Future Developments**: in the final chapter the conclusions are presented and possible future works are enumerated.

2 Graph Theory Overview and Graph Models of Power Systems

In this chapter the basics of Graph Theory are reunited in order to substantiate the key concepts and results that will be used in further sections of this dissertation. Graph Theory is a widely researched area in pure mathematics and there are many applications that are entirely based on graph data models, which are also illustrated in this chapter. The concepts and results more closely related to Graph Machine Learning are presented in Chapter 3 and the reader which is more familiar with Graph Theory fundamental concepts and mainstream applications may skip this chapter. The definitions regarding graph theory were mainly extracted from Balakrishnan e Ranganathan (2012) and Benjamin, Chartrand e Zhang (2017), being explicitly cited if otherwise.

2.1 Introduction

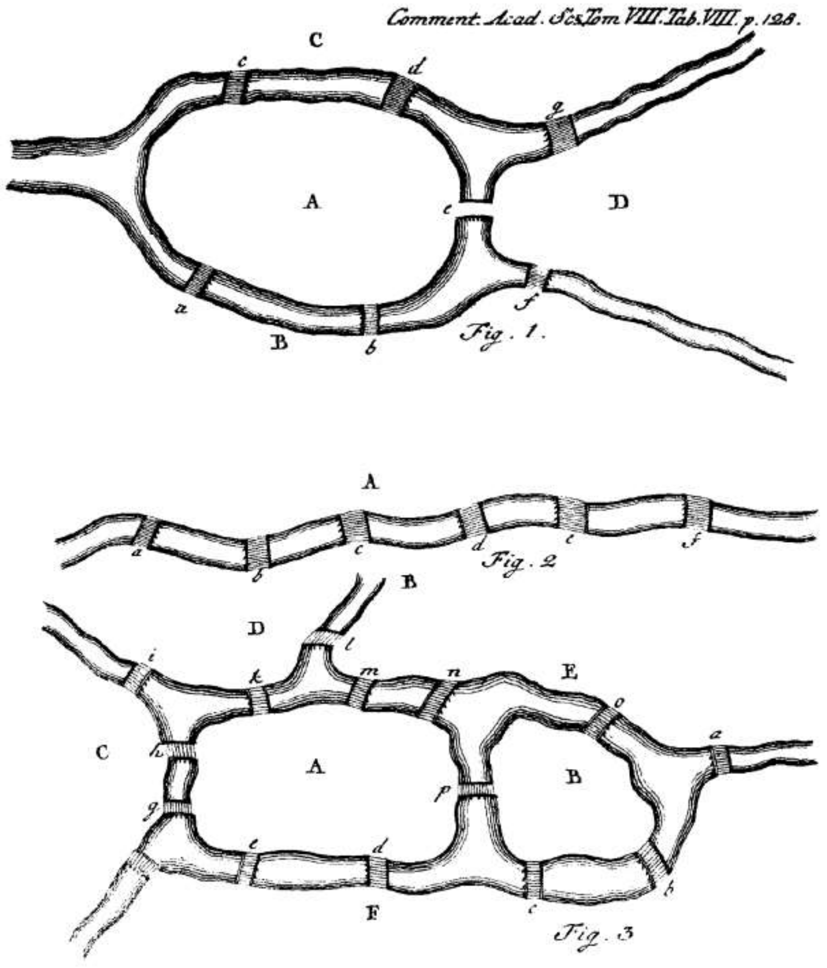
The origin of graph theory as a field in mathematics is considered to be in early 18th century, when the Swiss born mathematician Lehonard Euler published the solution for the contemporary puzzle of The Bridges of Könisberg (EULER, 1736), which now refers to the city of Kaliningrad, Russia. The problem was to design a route that crossed every bridge that was built across the Pregel river exactly once, without crossing any bridge twice. At that time, there was a total of 7 bridges that connect 4 pieces of land, as shown in Figure 1. Euler approached a real problem by extracting an abstract graph model with the pieces of land as nodes and bridges as edges, connecting them and giving an rigorous proof that it was not possible to be done in that case.

This approach of abstracting the unnecessary elements from the real world that will not have any influence in the outcome, such as houses, streets, if there were any squares in the pieces of land, which one was larger, and so on, was essential for Euler to demonstrate such simple and comprehensive result. In the end, only the pieces of land and bridges were considered.

For this dissertation, more relevant than Euler's result, which now characterizes a family of graphs, called *Eulerian Graphs*, is the adopted methodology of simplifying a real world scenario in existing elements, which are called vertices or nodes, and connections that represent some kind of similarity or bond between them, which are called edges or links. This approach leads to the definition of a graph as considered nowadays.

Definition 1 *A graph $G = G(V, E)$ is composed of the sets V and E. Elements $v \in V$ are* *called the vertices of G. Elements of E are unordered pairs of elements of V, $e = \{u, v\} \in E$, $u, v \in V$, and are called the edges of G.*

Figure 1 – Annotations made by Euler in order to abstract the problem only with the relevant information in the model



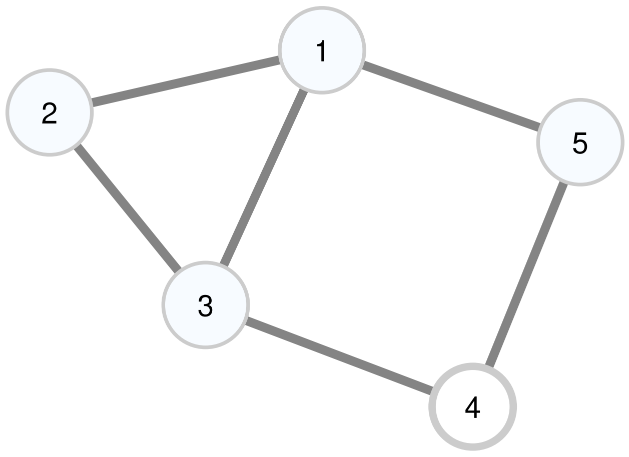
Source: extracted from (EULER, 1736)

This dissertation uses the words *vertex* and *node* as synonyms, and well as *edge* and *link*. By definition a graph is a very generic structure. In a great set of problems that can be solved with graphs, it is common to model the entities with a certain family of graphs called *simple graphs*, which are formally defined in Definition 2 and one example is shown in Figure 2.

Definition 2 (Simple Graph) *A graph $G = G(V, E)$ is called a simple graph when the vertex set V is non-empty and finite, and the elements of the edge set $e \in E$ are made of distinct vertices $e = \{u, v\} \in E$, $u \neq v \in V$.*

For the simple graph shown in Figure 2 the vertex set is $V = \{1, 2, 3, 4, 5\}$, whereas the edge set is $E = \{\{1, 2\}, \{1, 3\}, \{1, 5\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\}$. This graph will be kept for illustrating other definitions in this section, therefore it will be valuable to have its vertex and edge sets fully described as above.

Figure 2 – A simple graph following the Definition 2

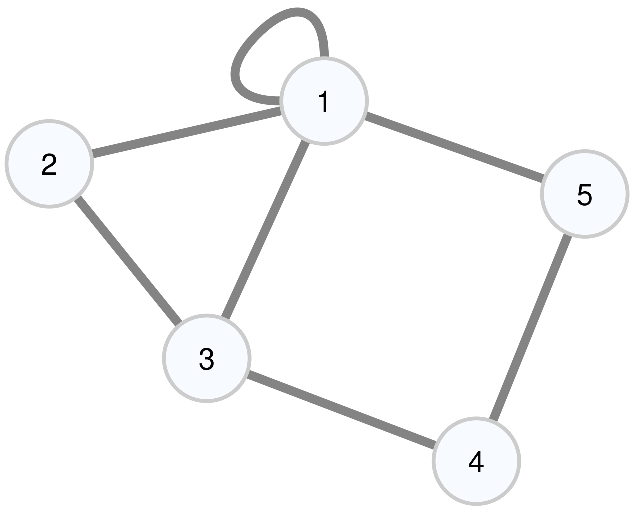


Source: the author

When the constraint regarding the edges of a simple graph is removed, one is allowed to have edges that relate only to a single vertex. This kind of edge is said to be a *self-loop*, or simply a *loop*. A graph which has a loop is shown in Figure 3. If a graph has a self-loop, then it cannot be considered a simple graph anymore.

Another way to add edges to a graph that violates the simple graph edges constraint is adding parallel edges to the same vertex pair. A graph in which a vertex pair has more than one edge is called a multigraph, which is illustrated in Figure 4.

Figure 3 – A graph with a self-loop in vertex 1

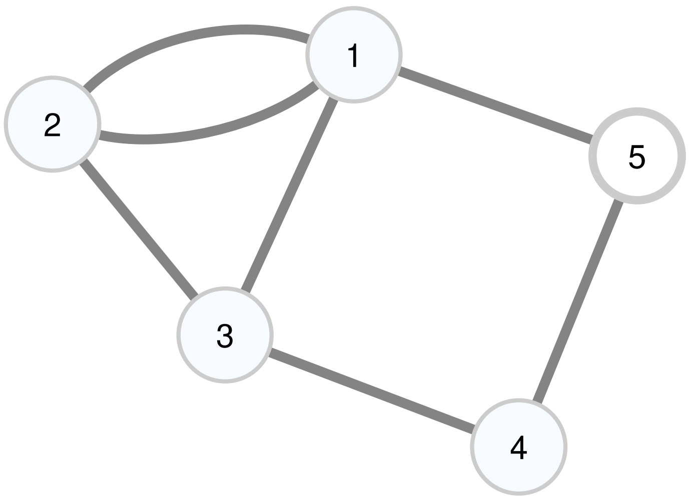


Source: the author

Definition 3 (Multigraph) *A graph $G = G(V, E)$ is called a multigraph when the edge set E has more than one edge that connects the same vertex pair. Given $e_1, e_2 \in E$, one may write $e_1 = \{u, v\}$ and $e_2 = \{u, v\}$, $u, v \in V$.*

A graph can also contain both self-loops and repeated edges, in which case it would also be a multigraph. If any of the aforementioned structures are present among the graph's edge set, then it cannot be considered a simple graph, and a lot of results from graph theory cannot be applied in the approached problem. Actually, if it is possible to construct a consistent model only with simple graphs, it is the preferred solution.

Figure 4 – A multigraph with duplicated edges connecting vertices 1 and 2



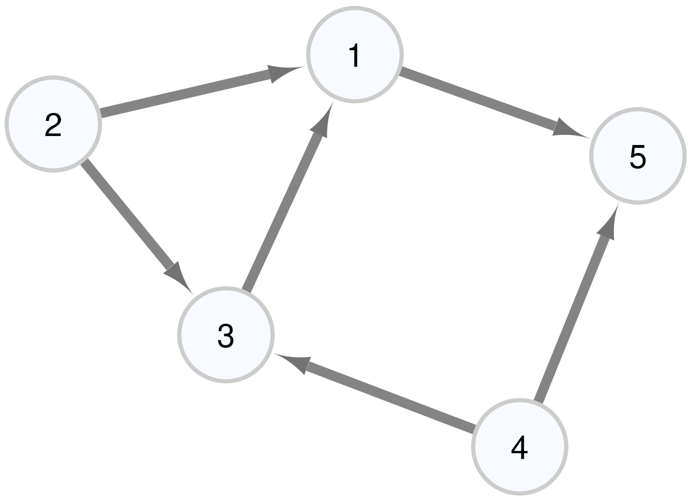
Source: the author

In all the above defined and illustrated graphs, the edges connect a pair of vertics in a symmetric way. If one says that, for a graph $G = G(V, E)$, $e = \{u, v\} \in E$, then it can be said that u is connected to v as much as v is connected to u. However, some applications require modeling of unidirectional relationships, or asymmetrical connections. In order to cover these models, a certain class of graphs, called the *digraphs* (short for directed graphs).

Definition 4 (Digraph) *A graph $G = G(V, E)$ is called a digraph when the edge set E has elements that are defined through ordered pairs or vertices $(u, v) \in E$, instead of unordered ones.*

For the digraph illustrated in Figure 5, which has the same vertex set as the previously mentioned graphs but directed edges, the edge set could be written as $E = \{(1, 5), (2, 1), (2, 3), (3, 1), (4, 3), (4, 5)\}$. It would be incorrect to say that $(5, 1) \in E$ because, as the edges are directed and written as ordered pairs of vertices, $(5, 1) \neq (1, 5)$.

Figure 5 – A digraph



Source: the author

With the aforementioned graph definitions one can comprehend most graph models for power systems. As will be described in Section 2.5, there are different approaches for extracting a graph from the electric diagram. For instance, multiple circuits can be denoted as repeated edges, resulting in a multigraph, or the edges can be oriented according to the power flow simulation result.

2.2 Paths and Connectivity

In order to use graphs to model the topology of an Electrical Power System (EPS), one must introduce topological definitions from graph theory. In a contingency scenario, it is expected that a contingency in a given device will have greater impact on the device's neighborhood than in some distant load. Therefore, in order to introduce the concept of distance, one must define a *walk*.

Definition 5 (Walk) *Given a graph $G = G(V, E)$, a walk W is a sequence of vertices $W = (v_1, v_2, \dots, v_k)$ such that, for any $i = 1, 2, \dots, k-1$, the edge $e_i = \{v_i, v_{i+1}\}$ exists in G.*

Intuitively, if a person is placed in a given vertex $v_1 \in V$ in a graph G, it can walk to any vertex that has a connection with v_1, namely v_2. In a second moment, being in v_2, the person can walk again, going to any other vertex that has a connection with v_2, even repeating v_1.

If a constraint on repeating vertices is put, than the walk becomes a *path*, as formally stated in Definition 6. The first and last vertices of the sequence are called *source* and *destination*, respectively. The only exception for repeating a vertex in a path is when the path is called *closed*, i.e., when $v_1 = v_k$. This special kind of path is called *cycle*.

Definition 6 (Path) *Given a graph $G = G(V, E)$, a path P is a sequence of distinct vertices $P = (v_1, v_2, \ldots, v_k)$ such that, for any $i = 1, 2, \ldots, k-1$, the edge $e_i = \{v_i, v_{i+1}\}$ exists in G.*

In certain applications some level of redundancy is required, and for modeling this kind of situation one may define distinct paths. Actually, if two paths P_1 and P_2 share the same source and destination, but have a single different vertex in the middle, then these paths are already distinct ones. However this situation may not reflect the desired redundancy status. A stronger requirement is the situation on which P_1 and P_2 are disjoint. In graph theory, paths can be *vertex-disjoint*, when they do not share any common vertex, or *edge-disjoint*, when they don't share any common edge (implicitly, a path can be defined in terms of the edges instead of the vertices). If a path is vertex-disjoint, than it is edge-disjoint, but the reciprocal is not always true.

If dealing with a digraph or a graph with repeated edges, both walk and path definitions hold. Given a graph $G = G(V, E)$ and a pair of vertices $u, v \in V$ such there is a path with u as origin and v as destination, we can say that the vertices u and v are *connected*. More generally, this leads to the definition of a connected graph.

Definition 7 (Connected Graph) *A graph $G = G(V, E)$ is called connected if, for each pair of vertices $u, v \in V$, there is a path P_{uv} connecting u and v.*

If a graph is not connected, it is said to be *disconnected*. The graph model of an EPS should be always connected. If transmission lines are modeled as edges and their contingencies are modeled as edge removals, contingencies that turn the graph model into an disconnected graph are expected to be treated as severe ones.

For a given path $P \subset V$, the length of P is the number of vertices in P minus one, i.e., the number of edges that were considered in order to define the path. Using paths and their lengths one can define the distance between a pair of vertices.

Definition 8 (Distance) *Given a pair of vertices $u, v \in V$ that are connected in a graph $G = G(V, E)$, the distance $d(u, v)$ between u and v is the length of the shortest path that has u as source and v as destination.*

The distance between vertices in a graph is a widely used metric for topological analysis and characterization. For instance, a simple characterization is done by evaluating all the distances between pairs of vertices in a graph and taking the average, which is called *mean distance* and the maximum, which is called the *diameter*.

Another way of characterize a graph is by analyzing the distribution of the vertices connections in the graph. If there are few vertices with many connected edges and many vertices with few edges, or if the edges are uniformly distributed among the vertices is a way of obtaining insights about the graph being dealt with. Formally, the definition behind this is the *degree* of a vertex.

Definition 9 (Degree) *The degree* $\deg(v)$ *of a vertex* $v \in V$ *in a graph* $G = G(V, E)$ *is the number of edges that are connected to* v*.*

The degrees of the vertices in a graph G can be considered themselves a centrality measure, called the *degree centrality*. This centrality is based on measuring the relevance of a vertex by the number of other vertices connected to it. More details about this centrality and others are given in Section 2.4.

Another important definition in graph theory is the *neighborhood* of a vertex. There are two main definition for a neighborhood, which can be a *open neighborhood* and a *closed neighborhood*, as stated in Definitions 10 and 11.

Definition 10 (Open Neighborhood) *Let* $G = G(V, E)$ *be a graph and* $v \in V$ *a vertex in* G*. The open neighborhood of* v *is the set of vertices* $N(v) \subset V$ *that are connected to* v*.*

Definition 11 (Closed Neighborhood) *Let* $G = G(V, E)$ *be a graph and* $v \in V$ *a vertex in* G*. The closed neighborhood of* v *is* $\bar{N}(v) = \{v\} \cup N(v)$*, the union of the open neighborhood of* v *with the vertex* v *itself.*

There are some transformations for extracting properties from graphs. In particular, the *line graph* is useful since it switches the vertices and edges in the graph, as defined in Definition 12.

Definition 12 (Line Graph) *Let* $G = G(V, E)$ *be a graph. The line graph of* G *is denoted* $G' = G'(V', E')$ *and its vertex and edge sets are constructed as follows:*

- *Each vertex of* G' *is associated with an edge of* G
- *Two vertices of* G' *are connected if and only if their corresponding edges incide in a common vertex in* G*.*

Another relevant definition is associated with graph editing. The process of removing edges from a graph has the possibility of turning a connected graph into a disconnected one. In this context, an *cut set* is defined in Definition 13.

Definition 13 (Cut Set) *Let $G = G(V, E)$ be a connected graph. A set of edges $e^k = \{e_1, \ldots, e_k\} \subset E$ is called a cut set if their removal from graph G turns G into a disconnected graph. If the set is unitary, $e^k = \{e\}$, e is called a disconnecting edge.*

2.3 Graph Matrix Representations

While being formally defined in Definition 2, graphs have many other ways of representation. In the area of spectral graph theory, many matrix representation for graphs were developed, and one of the most direct and comprehensive ones is the *adjacency matrix.* Two vertices are called *adjacent* if they are connected by an edge.

Definition 14 (Adjacency Matrix) *The adjacency matrix $A = A(G)$ of a graph G with vertices $\{v_1, v_2, \ldots, v_n\}$ is a matrix of order n and entries:*

$$a_{i,j} = \begin{cases} 1, & \text{if vertices i and j are adjacent} \\ 0, & \text{otherwise} \end{cases}$$

For the simple graph shown in Figure 2, considering the rows and columns are ordered following the labels given to the vertices, the adjacency matrix would be as in Equation 2.1.

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.1}$$

A common approach while modeling some problem using graphs is adding weights to connections. If an edge $e = \{v_i, v_j\}$ is given a weight $w_e \in \mathbb{R}$, then one can define an *weighted adjacency matrix*, in which the value of $a_{ij} = a_{ji} = 1$ is replaced by w_e.

Another common matrix representation in spectral graph theory is the *laplacian matrix*, which considers both adjacencies and degrees of the vertices.

Definition 15 (Laplacian Matrix) *The laplacian matrix $L = L(G)$ of a graph G with vertices $\{v_1, v_2, \ldots, v_n\}$ is a matrix of order n given by:*

$$L(G) = D(G) - A(G)$$

where $D = D(G)$ is a diagonal matrix of order n in which the i^{th} entry is the degree of vertex v_i, $d_{i,i} = \deg(v)$, and $A(G)$ is the adjacency matrix of G.

For the simple graph shown in Figure 2, the laplacian matrix is in Equation 2.2

$$L = \begin{bmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix} \tag{2.2}$$

Another matrix representation of graphs, which together with both adjacency and laplacian matrices is widely used in defining centrality measures is the *incidence matrix*.

Definition 16 (Incidence Matrix) *The incidence matrix $B = B(G)$ of a graph G with vertices $\{v_1, v_2, \ldots, v_n\}$ and edges $\{e_1, e_2, \ldots, e_m\}$ is a matrix of size m by n and given by:*

$$b_{i,j} = \begin{cases} 1, & \text{if edge } e_i \text{ incides in vertex } v_j \\ -1, & \text{if edge } e_i \text{ originates from vertex } v_j \\ 0, & \text{otherwise} \end{cases}$$

In simple graphs, since edges are not directed, a random orientation can be made. For the simple graph shown in Figure 2, if we make the convention of labeling the edges following $e_1 = (1, 2)$, $e_2 = (1, 3)$, $e_3 = (1, 5)$, $e_4 = (2, 3)$, $e_5 = (4, 3)$ and $e_6 = (4, 5)$, then the incidence matrix would be as in Equation 2.3.

$$B = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{2.3}$$

All the aforementioned matrix representations of graphs are used in the formulation of many of the centrality measures that are used in this dissertation, as shown in the following section. More results about the adjacency, laplacian and incidence matrices, including the description of many eigenvalues of their spectra and their meaning are known and can be found in Balakrishnan e Ranganathan (2012).

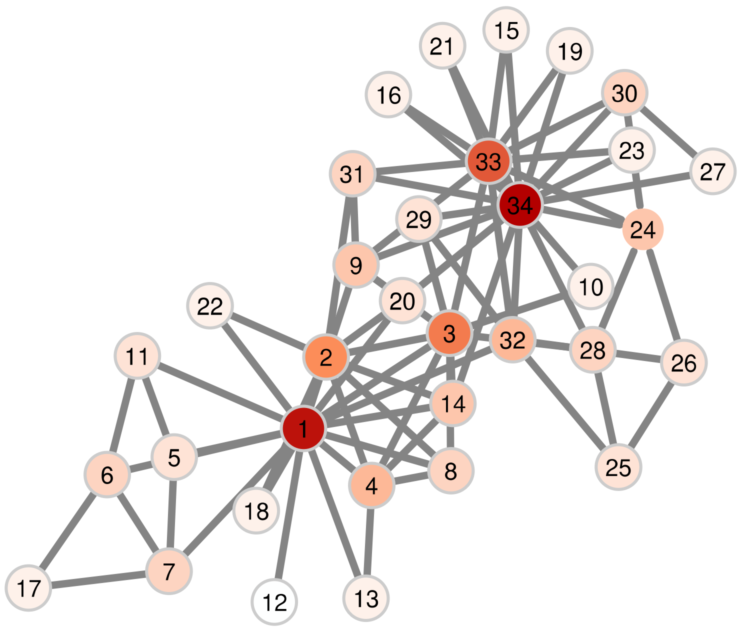
2.4 Centrality Measures

The center, originated from the ancient greek *kéntron* was used to give meaning to a little wooden stick, which greeks used to draw a circle by tying up a small woolen yarn. The stick was at the center of the circle, and became known as *kéntron*. As it was not possible to draw the circle without the stick, the meaning of importance was associated with the center of something, and from this one can understand the purpose of centrality measures.

The most classical way of measuring the centrality, or the importance in a graph was defined for vertices, in terms of the number of connections a vertex has, as mentioned in Section 2.2. The *degree centrality* of a vertex is simply the number of edges that connect to it, sometimes being normalized by $n - 1$, where n is the number of vertices in the graph.

In Figure 6 one can see an illustration of the degree centrality for a widely used reference graph in graph theory, the Zachary's Karate Club, which was originally used in Zachary (1977) for describing community structures in graphs. This graph models a social network of members from a karate club studied by Wayne Zachary from 1970 to 1972, where nodes represents students and two students are linked if they developed some kind of relationship outside the club. There was a conflict during the study period which led the club to be split in two, and Zachary predicted the students that would be in each side of the splits, except for one. This centrality measure clearly points to the vertices 34 and 1 as the most central ones, as the have degrees of 17 and 16, respectively. These nodes represent the two leaders of each split of the club. The third most central vertex is 33, with a degree of 12, a large drop when compared to the second most central vertex. Vertices 34 and 1, which have degrees significantly greater than the other vertices in the graph, and are called *hubs*.

Figure 6 – Zachary's Karate Club with vertices colored by their degree centrality. Darker colors means higher values.



Source: the author

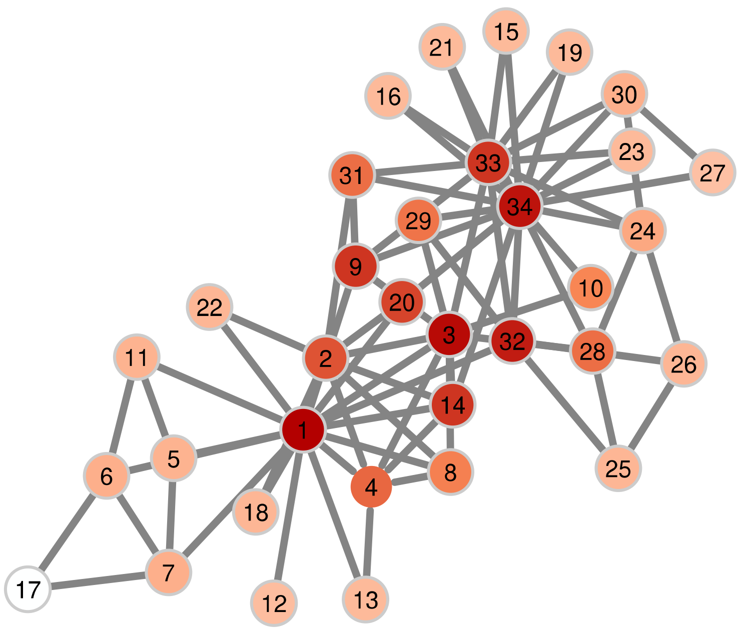
Another widely used centrality measure, which uses the concept of distance instead of degree is the *closeness centrality*, initially proposed in Freeman (1978).

Definition 17 (Closeness Centrality) *Given a graph $G = G(V, E)$ with n vertices, the closeness centrality of a vertex u is given, following Definition 8, by*

$$C(u) = \frac{n-1}{\sum_{v \in V \setminus \{u\}} d(u, v)}$$

As for the degree centrality, higher values of closeness indicates higer centrality. As it is entirely based on distance, the most central vertex in a graph will be the vertex that is closer to all the others. Figure 7 shows the evaluation of the closeness centrality for the Zachary's Karate Club. By evaluating centrality through distances, the vertices closer to the hubs have greater centrality since they can reach most vertices in the graph through their connection to the hub, despite the hubs being defined in terms of degrees and not the closeness itself.

Figure 7 – Zachary's Karate Club with vertices colored by their closeness centrality. Darker colors means higher values.



Source: the author

Instead of using the explicit distance from a vertex u to all the others in a graph G, one could consider the vertices that appear more frequently in shortest paths. This is pictured in the *betweenness centrality*, first published in Freeman (1977), but receiving modifications for better suiting certain applications, such as in Caporossi et al. (2011). For the betweenness centrality, there is also an alternative formulation for measuring the centrality of the edges, called *edge betweenness centrality*, with an equivalent formulation (BRANDES, 2008).

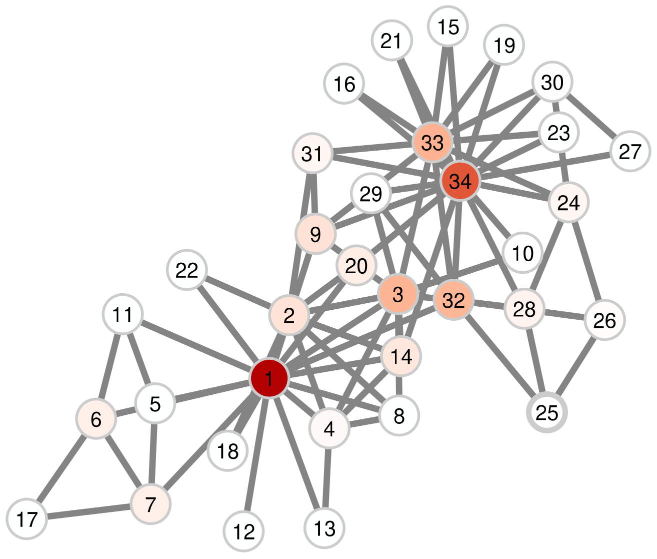
Definition 18 (Betweenness Centrality) *Given a graph $G = G(V, E)$ and a vertex $u \in V$, the betweenness centrality of u is given by*

$$c_B(u) = \sum_{s,t \in V} \frac{\sigma(s,t\,|\,u)}{\sigma(s,t)}$$

where $\sigma(s,t)$ is the number of shortest paths with s as source and t as destination and $\sigma(s,t\,|\,u)$ is the number of shortest paths with s as source and t as destination that passes through vertex u.

In Figure 8, the same aforementioned graph is represented with its vertices colored with respect to their betweenness centrality. When compared to the closeness centrality, at a first glance it is possible to notice how different the relative centrality is, for example, in vertices 3 and 32, which were as central as 1 and 34 for the closeness, but are considerably less for the betweenness.

Figure 8 – Zachary's Karate Club with vertices colored by their betweenness centrality. Darker colors means higher values.



Source: the author

The last centrality measure presented in this section, which is also the one whose definition embbeds more EPS concepts is the *current flow betweenness centrality*, first proposed by Newman (2005), soon having its evaluation algorithm improved in Brandes e Fleischer (2005), when it was proposed its edge equivalent centrality measure, the *edge current flow betweenness centrality*.

In order to define the equation for the current flow betweenness centrality, one must go through its formulation. A bus v, from a circuit modeled by a graph G, with lines weighted by $R = 1\Omega$, has the current flow betweenness measured by $C_{CFB}(v)$. The current enters the circuit from $s \in V$, that is a single source and out to the circuit by a sinking bus $t \in V \setminus s$. A vector $b : V \to \mathbb{R}$ called *supply* is the representation of it. Only unitary currents are considered and b is defined as:

$$b_{st}(v) = \begin{cases} 1, & v = s \\ -1, & v = t \\ 0, & otherwise \end{cases} \tag{2.4}$$

Considering the direction of current flow, and an arbitrary orientation of these edges, these oriented edges represents arcs. A set of arcs which is given by $\vec{E}$ is formed by the oriented pairs (v, w), where $v, w \in V$, or simply by $\vec{e}$.

A vector of *electrical currents*, defined by $x : \vec{E} \to \mathbb{R}$, satisfies both:

$$\sum_{(v,w)\in\vec{E}} x(v, w) - \sum_{(u,v)\in\vec{E}} x(u, v) = b(v), \tag{2.5}$$

which holds for every $v \in V$ and is known as Kirchhoff's Current Law (KCL), and

$$\sum_{i=1}^{k} x(\vec{e_i}) = 0, \tag{2.6}$$

which holds for every non-oriented cycle $e_1, \ldots, e_k$ in G and is known as Kirchhoff's Voltage Law (KVL).

By convention, when the current flows in the direction given to the edge e a positive value is given to $x(\vec{e})$, otherwise $x(\vec{e})$ has a negative value.

The vector of voltages can be defined as $\hat{p} : \vec{E} \to \mathbb{R}$ with $\hat{p} = x$, with only unitary edge weights. The vector $p : V \to \mathbb{R}$, with $\hat{p}(v, w) = p(v) - p(w)$ represents the absolute potentials, chosen by assigning the potential of the vertex v_1 as p_1. The vectors were taken with the st-supply and it is indicated by $x_{st}, \hat{p}_{st}$ and p_{st}.

The *Laplacian* matrix $L(G)$ is used to compute the absolute potentials, which is the application of Kirchhoff's Current Law to each vertex in the graph.

Taking the above explanation, a supply b and $L(G)$, it is possible compute the b vector:

$$Lp = b. \tag{2.7}$$

One can verify that the Laplacian matrix $L(G)$ is singular, since all its rows and columns sum 0 and 0 must be an eigenvalue. Therefore, one forces the potential in a vertex, namely $p(v_1)$, to be a known value, for instance 0. By removing the entries relative to v_1

in the vectors p and b, together with the first line and column of L, it is obtained a new matrix, denoted $\tilde{L}$. In order to solve the system given in Equation 2.7, one might invert $\tilde{L}$, obtaining $\tilde{L}^{-1}$, and then construct the matrix M as given in Equation 2.8.

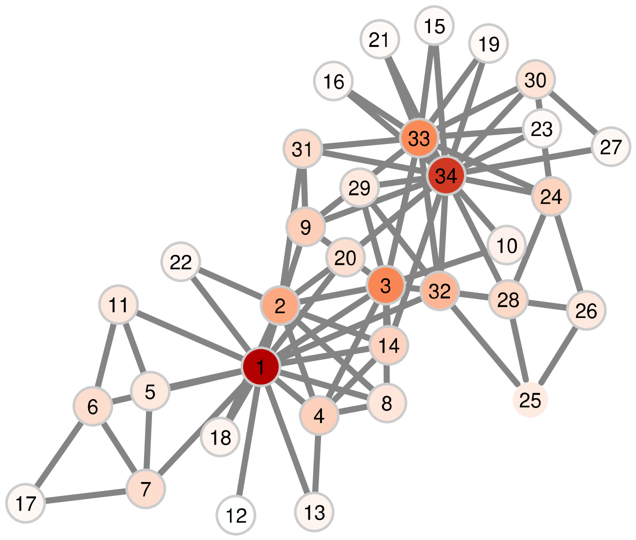
$$M = \begin{pmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \tilde{L}^{-1} \end{pmatrix} \tag{2.8}$$

The absolute potentials can now be obtained since $p = Mb$. With the obtained absolute potentials p and the supply vector b, the current x is computed. Then, with these results it is possible to define the *throughput* of an vertex v, based on Equation 2.5, as:

$$\tau(v) = \frac{1}{2}\left(-|b(v)| + \sum_{e:v\in e} |x(\vec{e})|\right) \tag{2.9}$$

where $e : v \in e$ means that the sum is made for every edge incident on vertex v.

Figure 9 – Zachary's Karate Club with vertices colored by their current flow betweenness centrality. Darker colors means higher values.



Source: the author

Definition 19 (Current Flow betweenness Centrality) *Taking into account Equation 2.9, which defines the throughput $\tau(v)$ of a vertex v, the current flow betweenness centrality $C_{CFB} : V \to \mathbb{R}$ is defined as:*

$$C_{CFB}(v) = \frac{1}{(n-1)(n-2)} \sum_{s,t\in V} \tau_{st}(v). \tag{2.10}$$

The current flow betweenness centrality is deduced by making an analogy between a graph and an electric circuit. By not considering only the shortest paths, one can give another degree of relevance to vertices that are near the shortest paths. This happens since the centrality is evaluated as the average node current throughput on unitary electrical circuit solutions. In Figure 9 the same aforementioned graph is shown, with its vertices colored with respect to the current flow betweenness centrality. It is immediate to notice the correlation between the betweenness centrality shown in Figure 8 and the current flow betweenness centrality, but the difference is also noticed for the vertices with lower values of centrality.

Table 1 – Comparison of the vertices with greatest values of centrality in the Zachary's Karate Club.

Ranking	$\deg(v)$	$C(v)$	$c_B(v)$	$C_{CFB}(v)$
1	34	1	1	1
2	1	34	3	34
3	33	33	34	3
4	3	3	32	33
5	2	32	33	2
6	32	9	9	32
7	4	2	14	9
8	9	14	20	4
9	14	20	2	14

Source: the author

In Table 1 it is shown a comparison among the vertices with the greatest values of centrality evaluated by each of the centrality measures in this section. From this result one may conclude, at least for the Zachary's Karate Club network, that the centrality measures have a similar behavior with respect to the most central vertices. However, as shown in Coelho et al. (2019), the chosen centrality measure can have significant influence in the contigency analysis result.

2.5 Complex Networks and Network Models of Power Systems

2.5.1 Complex Networks and Network Science

Even when one is able to represent real world systems in graph models, another challenge arises. For a specific given graph, it is possible to evaluate many centrality measures, but modeling a single network does not give one a characterization of how other networks that model similar systems might be.

One of the challenges of the area called network science is to build models that reproduce some topological properties of real networks (BARABÁSI, 2016). This task begun even before the research field nowadays known as network science had this name, with the research of Pál Erdős and Alfréd Rényi on random graphs (ERDÖS; RÉNYI, 1959).

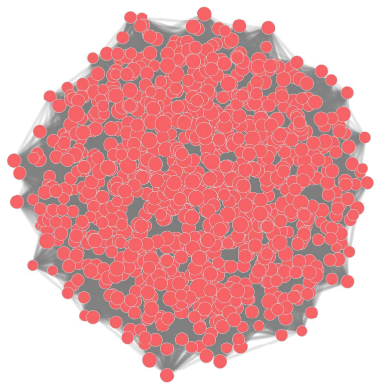
The model known as Erdős-Rényi (ER) or $G(n, m)$ builds random networks using the most naive approach: given a network with n nodes, m links will be randomly distributed among the $n(n-1)/2$ possible pairs of nodes. A second model, proposed in Gilbert (1959) and called $G(n, p)$ became more popular. In this model, given a network with n nodes, to each of the node pairs, an edge was added with probability p. This became known as the classical *random network* model. Figure 10 shows one random graph generated with the $G(n, p)$ graph model, together with its degree distribution, with is known to follow the binomial distribution, given in Equation 2.11.

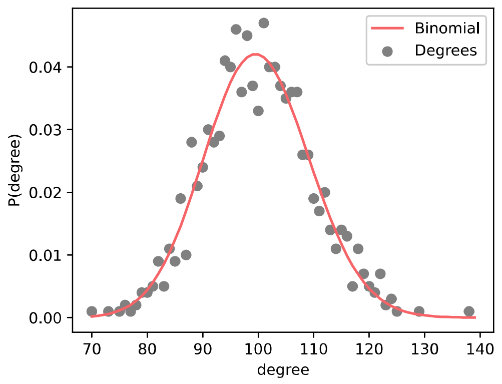
$$p(k) = \begin{pmatrix} n-1 \\ k \end{pmatrix} p^k (1-p)^{n-1-k} \tag{2.11}$$

When the network is sparse, i.e. $\bar{k} << n$, where $\bar{k} = \frac{1}{n}\sum_{i=1}^{n} k_n$ is the average degree of the network, it is common to approximate it's degree distribution by the Poisson, given in Equation 2.12.

$$p(k) = e^{-\bar{k}} \frac{\bar{k}^k}{k!} \tag{2.12}$$

Figure 10 – A generated graph for the $G(n, p)$ random graph model, together with its degree distribution.


(a) Graph generated with the $G(n, p)$ graph model for $n = 1000$ and $p = 0.1$. Vertex size is proportional to degree.



(b) Degree distribution for the generated $G(n, p)$ graph.

Source: the author

Despite its theoretical richness, the random graph model does not represent most real world networks, mainly because the interactions are not random (BARABÁSI, 2016). Instead of expect the nodes's degrees to have a binomial distribution, where most are around the average degree, with very small probabilities of deviating from this value, by observing real world network models, it was found that there are many nodes with small degrees, and few nodes with degrees much greater than the average. This property was considered an lack of scale, and this was called the *scale-free* property (BARABÁSI; ALBERT, 1999).

The Barabási-Albert Model (BA) is able to generate scale-free networks which aims to approximate a power-law degree distribution, described in Equation 2.13.

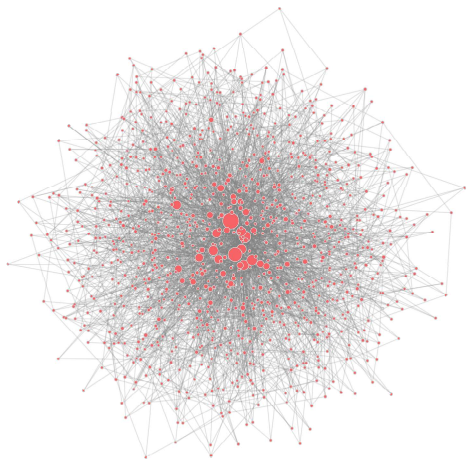
$$p(k) = k^{-\gamma} \tag{2.13}$$

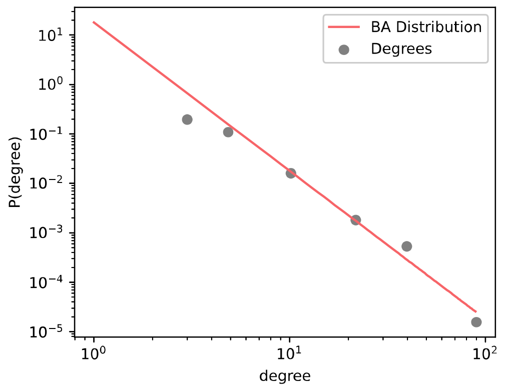
The key concept for generating scale-free networks is the prefferential attachment. In order to generate a scale-free network with n nodes, the BA model starts with an initial graph G, iteratively adding nodes with w edges, where the probability of attachment on a given vertex is proportional to the vertex degree. The approximate degree distribution for a BA network is given in Equation 2.14.

$$p(k) \approx 2w^2 k^{-\gamma} \tag{2.14}$$

Figure 11 illustrates a BA network generated with $n = 1000$ nodes and prefferential attachment with $w = 3$ edges added in each iteration, together with its degree distribution.

Figure 11 – A graph generated for the BA graph model, together with its degree distribution.


(a) Graph generated with the BA graph model for $n = 1000$ and $w = 3$. Vertex size is proportional to degree.



(b) Degree distribution for the generated BA graph.

Source: the author

The BA model generates graphs that model many real world networks, such as the WWW (BARABÁSI; ALBERT, 1999) and citation networks (LESKOVEC; KLEINBERG; FALOUTSOS, 2007). When dealing specificaly with power systems, it has been found that the preferential attachment hypotesis is not valid when modeling the growth, mainly because the large hubs are not found in this kind of system. Some other degrees distributions such as the *power law with exponential cutoff* and the *stretched exponential*, or even the actual *exponential* distribution can also fit these networks (BARABÁSI, 2016).

2.5.2 Network Models of Power Systems

When modeling power systems using graphs, there is a number of considerations to be made on which elements are relevant to the model and which are not. For instance, if the vertices are generators or buses and edges are the transmission or distribution lines, the direction of the power flow in the lines in normal operation can be used to embbed additional information in the edges, resulting in a digraph, such as made in Bompard, Pons e Wu (2012), which also weighted the edges by the magnitude of their equivalent impedance, and called this an "electrical distance". In the same work, an extension for the betweenness cetrality, called "electrical betweenness centrality" is proposed.

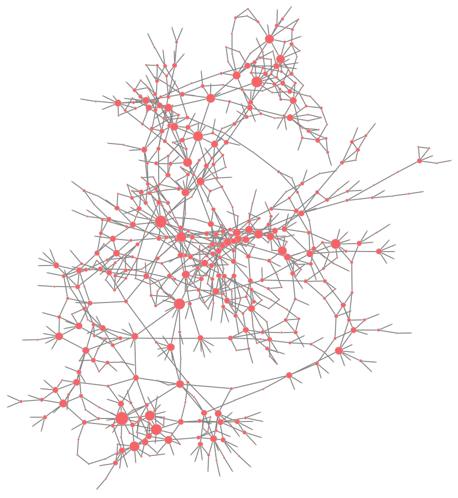
In Pang e Kezunovic (2011), the EPS is also modeled as an directed graph, using the direction given in the power flow result, but the lines are not weighted and the classical betweenness centrality is used. The directed model is enforced as being more coherent when analyzing steady state systems in Dwivedi e Yu (2013), which uses an alternative maximum flow approach to evaluate vulnerabilities.

However, not all graph models of power systems are based in digraphs. In Nasiruzzaman e Pota (2014), an undirected graph model is used for studying extensions of traditional centrality measures for electrical applications. In Pagani e Aiello (2013), a survey is made, containing directed and undirected models with weighted and unweighted elements. More recently in Chu e Iu (2017), another survey is made, again enforcing the use of undirected models when applying centrality measures.

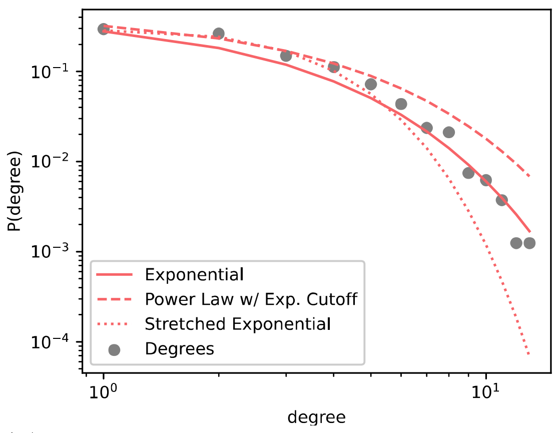
One of the results presented in the aforementioned surveys is the characterization of power systems as graphs through the usage of complex networks models. For this task, the node degrees are evaluated and a statistical distribution is fitted, in order to picture how the connections are made in the system. In Pagani e Aiello (2013), most power system models were fitted to either power law degree distributions or exponential distributions. As stated in Barabási (2016), the exponential degree distribution is related to the lack of preferential attachment, whereas the power law degree distribution reflects its presence.

For the Brazilian Interconnected System (SIN), a simple graph model was extracted considering vertices as substations (buses) and transmission lines as edges. All voltage levels were considered. The data used for constructing the graph is available in ONS (2022). The resulting graph model and its degree distribution are represented in Figure 12. As one can verify in the degree distribution, there is a faster decay in the probability for hubs when compared to power law networks, meaning the BA graph model would not

Figure 12 – The constructed graph model for the Brazilian Interconnected System (SIN), together with its degree distribution and the fitted exponential curve.



(a) Graph model for the Brazilian Interconnected System (SIN)



(b) Degree distribution for the SIN graph model with fitted distributions.

Source: the author

be suited for modeling the SIN graph. However, some variations of the BA graph model which consider vertex removal might be suited for modeling the SIN graph, since their degree distribution can range from power law to exponential (BAUKE et al., 2011).

2.6 Summary

The graph theory has plenty of resources for modeling a wide range of problems in present day. In the case of power systems analysis, there are many forms of obtaining graph models for the underlying system, such as considering the power flow direction, the different kinds of connection between electric devices, and so on.

For the contingency analysis problem, the developed graph models have recurrently considered centrality measures to formulate the decision process, with the *betweenness* centrality and its generalizations having a distinct role among the other measures.

In this dissertation, in order to apply learning techniques for the graph models of power systems, given the reduced amount of available real-world data and that for training most deep learning techniques a large amount of data is nedded, graph generation models must be considered.

For what can be seen from the liteature and for the built model for the SIN graph, the popular BA model does not give a good fit in terms of degree distribution for most power system graphs. However, some variations of the BA model which regulates the preferential attachment can be applied, since the given degree distributions approximate the exponential.

In the next chapter, an introduction to the graph machine learning techniques is made. Some classical methods for dealing with graph data are shown, being compared to the most recent literature results.

3 Graph Machine Learning Fundamentals

When dealing with conventional serial or tabular data, machine learning techniques have notorious success, being widely publicized and applied both in research and real world applications. These conventional techniques usually take as input well-structured data and apply pattern recognition strategies. However, when one has to deal with relational data, the number of available techniques decreases dramatically, most being recently developed (HAMILTON, 2020). In this chapter, the fundamental problems of dealing with graph data with conventional machine learning are presented, together with some approaches and techniques from the literature, which are used in the proposed methodology in Chapter 4.

3.1 Machine Learning on Graphs

Machine learning problems are often categorized based on the type of task they seek to solve. When the goal is to predict a target output in unknown test data, based on given training data, it is called a *supervised* task. On the other hand, when the task is to infer patterns in the data, such as clusters, the task is said to be *unsupervised*. On graphs, these usual categories are mostly not informative, and the learning tasks are said to be *self-supervised*, since that the graph topological structure is used for training, even if not all the vertex labels are known (NICKEL et al., 2016). Three of the fundamental tasks of machine learning in graphs are introduced in the following: node classification, link prediction and community detection.

3.1.1 Node Classification

Let $G = G(V, E)$ be a graph where each vertex $u \in V$ is assigned to a label y_u, which can be a class, an attribute or any other property. If only a subset of vertices have their labels known, called the *training set* $V_{train} \subset V$, the *node classification* task is to determine the labels of the remaining vertices in the graph.

It is usual that only a small subset of vertices have their labels known in a graph, as for classifying the topic of documents referenced by hyperlinks in web graphs or papers in citation graphs (KIPF; WELLING, 2016) and for the role of proteins in an interactome (HAMILTON; YING; LESKOVEC, 2017a). This task appears to be, but is not a straightforward variation of the classical supervised classification. First, the samples, or the vertices, are not independent from each other, breaking a series of assumptions taken in most methods. Second, it is not possible to ensure that the labels are identically distributed, frequently leading to unbalanced problems. One key concept that is behind many of the successful methods for node classification is to exploit *homophily*, the tendency for vertices to share attributes and influence their neighbors (MCPHERSON; SMITH-LOVIN; COOK, 2001). In the particular object of study for this dissertation, applying this concept to graphs that model power systems would mean that buses that are immediately connected to critical buses tend to be critical.

3.1.2 Link Prediction

The second classical problem when dealing with graph data is the *link prediction*. As with node classification, the graph G may have missing data, such as the labels of the vertices that had to be obtained, sometimes the graph that represents the underlying network may have some links missing. Link prediction has been applied in many areas, such as content recommendation in social networks (YING et al., 2018) or inferring facts in relational databases (BORDES et al., 2013).

Given a graph G with an incomplete edge set $E_{train} \subset E$, the goal of link prediction is to use this incomplete information to infer the missing edges. For graphs which model only one kind of relation, such as social network graphs, there are simple heuristics that can reach good performance (LU; ZHOU, 2011). The link prediction task can also vary among the prediction object, which can be the missing edges of a single graph or across multiple disjoint graphs (TERU; HAMILTON, 2019). For this dissertation, the power systems databases are closely analyzed and studied, so the probability of existing missing link information in the data is low. However, instead of predicting the existence or not of some link, one might be interested in predicting one link's attributes or classifying the link in some given classes, such as critical and non-critical links.

3.1.3 Community Detection

For as much as both node classification and link prediction tasks can be seen as analogs of classical machine learning supervised tasks, *community detection* can be seen as the analog of unsupervised clustering for graph data.

Networks inherently have clusters, being them the areas of knowledge in a knowledge graph, a common circle of friends from the university on a social network, or even diseases with similar symphtoms in a disease interactome. The task is to identify the communities of vertices only using the graph topology. More formally, the task is, given a graph $G = G(V, E)$, determining the partition of V that best suits in the graph data. It means that each vertex u must be put on a subset $V_u \subset V$ of the vertices that belong to the same community if u shares a number of characteristics with the remaining vertices in V_u. Some real world applications of community detection are to identify fraudulent groups of users in transaction networks (PANDIT et al., 2007) and extract functional modules in genetic networks (AGRAWAL; ZITNIK; LESKOVEC, 2018).

3.2 Node Embeddings

A fundamental process of learning with graph data is to define an *embedding* for the vertices of a graph, or even for a graph itself. An embedding of a node u in a graph G is a projection of u in a low-dimensional latent space $\mathbb{R}^d$ that summarizes its position inside G, where nodes that are similar to u are close to its projection in the latent space (HOFF; RAFTERY; HANDCOCK, 2002).

The notation used to define the embedding techniques that are used in this text consider a graph $G = G(V, E)$ with adjacency matrix A and an optional matrix of node attributes $X \in \mathbb{R}^{d \times |V|}$, where $|V|$ is the size of the vertex set, i.e., the number of vertices and $d \in \mathbb{N}$ is the number of node attributes supplied for the embedding process. The goal is to use the existing information from both A and X to map each node, or subgraph, to a vector $z \in \mathbb{R}^d$, where $d << |V|$. Most methods do this task in an unsupervisied method, without making use of the downstream machine learning task that the embeddings will be used for. However, some methods optimize the embeddings to give the best results on a certain task, such as node classification or community detection, which are said to be supervised learning embeddings.

3.2.1 Encoders and Decoders

In the first years of research in node embeddings there were developed a diversity of notations and models, which lead to issues in understanding and comparing models. In order to make an overview of the existing approaches, Hamilton, Ying e Leskovec (2017b) developed a general framework to organize the embedding methods, called the *encoder-decoder* framework.

The encoder and the decoder are two key mapping functions. The encoder's role is to map each node to a low-dimensional vector that holds structural information of the node and it's neighborhood, i.e., the nodes connected to it and how they are connected. The decoder is able to extract the relevant information for the specific machine learning task for each node from its embedding vector. More formally, the encoder is a function that maps nodes v_i to low-dimensional vectors $z_i \in \mathbb{R}^d$, given in Equation 3.1.

$$\text{ENC} : V \to \mathbb{R}^d \tag{3.1}$$

The decoder is another function, which accepts sets of embeddings and decodes specified graph statistics. For instance, a simple decoder can predict the existence of edges between nodes, given their embeddings, or might predict the community that a node belongs. Most works make use of the basic *pairwise decoder*, which is defined in Equation 3.2. This decoder maps pairs of node embeddings to a real node similarity measure.

$$\text{DEC} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+ \tag{3.2}$$

The goal is to optimize the encoder-decoder process so that the *reconstruction* error is minimized, which means that the entire or most part of the structural information contained in the graph is recovered after the embedding process, as denoted in Equation 3.3, where $s_G(v_i, v_j)$ is a similarity measure in graph G, applied to nodes v_i and v_j.

$$\text{DEC}(\text{ENC}(v_i), \text{ENC}(v_j)) = \text{DEC}(z_i, z_j) \approx s_G(v_i, v_j) \tag{3.3}$$

The canonical example is when the similarity measure is set to the adjacency relation, where $s_G(v_i, v_j) = 1$ if v_i and v_j are adjacent, and 0 otherwise. In any case, the embeddings are learned by minimizing a loss function $\mathcal{L}$ over a subset of vertex pairs T, as in Equation 3.4, where $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function defined by the user, which measures the discrepancy between the estimated and true values for the similarity.

$$\mathcal{L} = \sum_{\{v_i, v_j\} \in T} l(\text{DEC}(z_i, z_j), s_G(v_i, v_j)) \tag{3.4}$$

Most embedding approaches are said to be *shallow embeddings*. This kind of embedding allows the encoder function to be written as a simple "embedding lookup", as in Equation 3.5, where $Z \in \mathbb{R}^{d \times |V|}$ is a matrix with all the embedding vectors and $\mathbf{v}_i$ is a one-hot indicator vector (values 1 on the i-th coordinate and 0 elsewhere) for selecting the column associated with v_i. These approaches are inspired by matrix factorization techniques, and more recent developments, called *Graph Neural Networks*, are not included in this group.

$$\text{ENC}(v_i) = Z\mathbf{v}_i \tag{3.5}$$

3.2.2 Factorization-Based Embeddings

The embeddings described in this section are called factorization-based embeddings because, averaging over all vertices, their loss functions are optimized, approximately, in the form shown in Equation 3.6, where S is the matrix containing the pairwise similarity measures: $s_{i,j} = s_G(v_i, v_j)$.

$$\mathcal{L} \approx ||Z^T Z - S||_2^2 \tag{3.6}$$

One of the earliest embedding techniques for graphs was developed in the context of computer vision. Despite the complete algorithm included a step for constructing a graph from a set of data points extracted from images, the Laplacian eigenmaps (LE) technique can still be used if the graph is already known (BELKIN; NIYOGI, 2002). The embedding process can be summarized as a solution for the generalized eigenvector problem.

Given a graph G where its edges are weighted and denoting by w_{ij} the weight of the edge that connects vertices v_i and v_j, the diagonal weighted matrix D_W, which is a generalization of the diagonal degree matrix used in Definition 15, is defined in Definition 20. The weighted adjacency matrix, which is the adjacency matrix for weighted graphs, is defined in Definition 21.

Definition 20 *Let $G = G(V, E)$ be a simple graph with vertices $v_i \in V$, $i \in \{1, \ldots, n\}$ and weighted edges $e_w \in E$. Let $w_e \in \mathbb{R}$ be the weight of edge $e_w \in E$. The diagonal weighted matrix $D_W \in \mathbb{R}^{n\times n}$ is such that its entry $d_w(v_i)$, associated with vertex v_i, is valued by sum of weights of edges that incide on v_i.*

Definition 21 *Let G be a simple graph with vertices $u, v \in V$ and weighted edges $e_w \in E$. Let $w \in \mathbb{R}$ be the weight of edge $e_w \in E$. The weighted adjacency matrix $A_W \in \mathbb{R}^{n\times n}$ is such that the value of entry $a_w(u, v)$ is the weight of the edge that connects vertices u and v, being 0 if the vertices are not connected by any edge.*

The generalized laplacian matrix for weighted graphs is given by $L_W = D_W - A_W$ and the embedding is obtained by solving Equation 3.7. If $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{k-1}$ are the solutions of Equation 3.7, the embedding for vertex v_i in the lower-dimensional space $\mathbb{R}^d$ is given by $z_i = (\mathbf{y}_1(i), \ldots, \mathbf{y}_d(i))$.

$$L_W y = \lambda D_W y \tag{3.7}$$

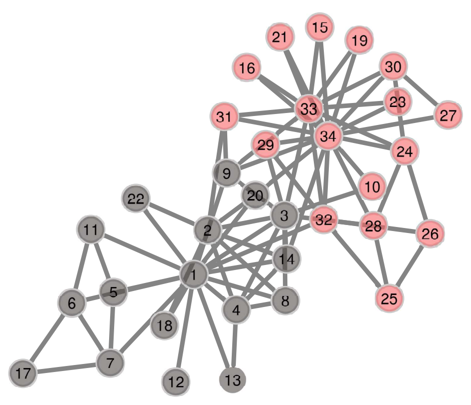
The decoder for the LE technique can be written as in Equation 3.8 and the loss function in Equation 3.9. These results are obtained in the formulation of the LE methodology as proposed in (BELKIN; NIYOGI, 2002).

$$\text{DEC}(z_i, z_j) = ||z_i - z_j||_2^2 \tag{3.8}$$

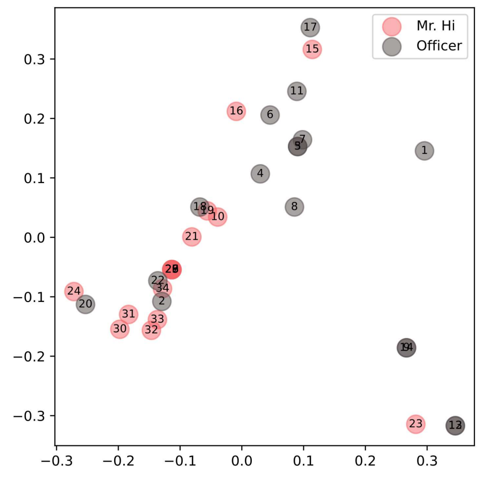
$$\mathcal{L} = \sum_{\{v_i, v_j\} \in T} \text{DEC}(z_i, z_j) \cdot s_G(v_i, v_j) \tag{3.9}$$

It is relevant to observe that the pairs of vertices can receive any real weight, which means that the true value of the similarity measure, expressed by the weighted adjacency matrix W, can be any real attribute specific for the application. Figure 13 shows the result of the embedding using LE for the Zachary's Karate Club graph.

Figure 13 – Zachary's Karate Club graph with node embeddings generated using the Laplacian Eigenmaps method.



(a) Zachary's Karate Club with vertices colored according to the existing communities.



(b) Node embedding with LE with $d = 2$ for the Zachary's Karate Club in $\mathbb{R}^2$.

Source: the author

Another technique for node embedding that is frequently cited in the literature is the *high-order proximity preserved embedding* (HOPE), which belongs to a larger group of embeddings, called *inner-product methods* (OU et al., 2016). Among this group, the decoders can be written in the form of an inner-product, as in Equation 3.10.

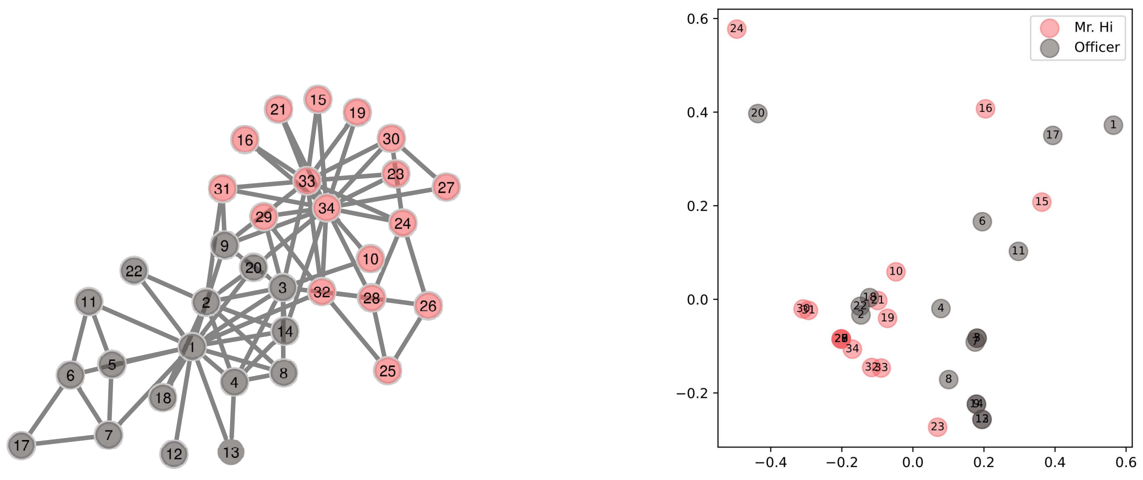
$$\text{DEC}(z_i, z_j) = z_i^T z_j \tag{3.10}$$

The HOPE approach aims to preserve a property called asymmetric transitivity, present in digraphs and which pictures the differences in path lengths for a given pair of vertices when the source and destination are commuted. The equation for the HOPE embedding writes the learning problem in a very standard way for factorization methods, as shown in Equation 3.11. S is the matrix for the similarity measures, which is called proximities in the original proposition of the method. $Z^s, Z^t \in \mathbb{R}^{n \times d}$ are the embedding matrices for a graph with n nodes and embeddings with dimension d.

$$\min ||S - Z^s \cdot Z^{t^T}||_F^2 \tag{3.11}$$

In (OU et al., 2016), closed forms for writing the S matrix are proposed for a given set of classical similarity measures, such as the number of common neighbors, the Katz Index (KATZ, 1953) and the Adamic-Adar Index (ADAMIC; ADAR, 2003). Figure 14 shows the embedding using the HOPE approach for the Zachary's Karate Club graph.

Figure 14 – Zachary's Karate Club graph with node embeddings generated using the HOPE method.



(a) Zachary's Karate Club with vertices colored according to the existing communities.

(b) Node embedding with HOPE ($d = 10$, $\beta = 0.1$, reduced with PCA) for the Zachary's Karate Club in $\mathbb{R}^2$.

Source: the author

3.2.3 Random Walk Embeddings

Despite the progress in defining methods for embedding that produce a low reconstruction error and are able to encode the structural information of a node in a graph, all factorization-based embeddings employ deterministic measures of similarity. On recent years, a lot of methods that use stochastic measures were developed by using random walks in graphs, as these approaches promised to be more scalable. For the methods that are cited in this section, it is necessary to define a random walk, which is done in Definition 22.

Definition 22 (Random Walk) *Given a graph $G = G(V, E)$ and a vertex $v_i \in V$, a random walk rooted at v_i is denoted $\mathcal{W}_{v_i}$. It is a stochastic process with random variables $\mathcal{W}_{v_i}^1, \mathcal{W}_{v_i}^2, \ldots, \mathcal{W}_{v_i}^k, \ldots, \mathcal{W}_{v_i}^K$ such that $\mathcal{W}_{v_i}^{k+1}$ is a vertex chosen randomly from the neighborhood of $\mathcal{W}_{v_i}^k$. We can denote $\mathcal{W}_{v_i}^0 = v_i$. K is the length of the random walk.*

The standard random walk is also known as *unbiased random walk*, which is a random walk following the stated in Definition 22. However, the process of choosing a vertex from the neighborhood of v_k does not have to be done with uniformly distributed probabilities. Some neighbord might have a grater chance of selection than others, what it said to be a bias in the selection process. The distribution of probabilities for vertex selection in a random walk is called a *random walk strategy*. Random walks that are made following a strategy different from the random selection are called *biased random walks*.

One of the first successful methods to deliver good performance and scalability in the embedding was the deepwalk approach (PEROZZI; AL-RFOU; SKIENA, 2014), which was developed in a context of embedding social network graphs. The embedding method was inspired in language modeling techniques, which tried to predict the probability of a given word showing in a context, given the words that are already present. In this approach, the embedding look-up matrix $Z \in \mathbb{R}^{|V| \times d}$ is initialized randomly, with the entries being updated each iteration by using stochastic gradient descent. The loss function for computing the update is a variant of the softmax function, which approximates the probability $p_G(v_j|v_i)$ of visiting a given node v_j in a random walk rooted in v_i, called *hierarchical softmax*. The equations for the deepwalk method in the encoder-decoder framework are shown in Equations 3.12 and 3.13 for the decoder and the loss function, respectively. In these equations, T is a training set generated by sampling random walks starting from each node.

$$\mathrm{DEC}(z_i, z_j) = \frac{e^{z_i^T z_j}}{\sum_{v_k \in V} e^{z_i^T z_k}} \approx p_G(v_j|v_i) \tag{3.12}$$

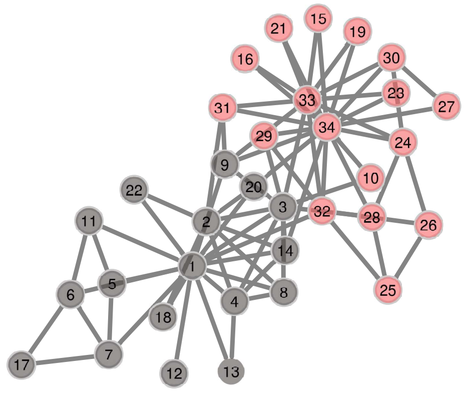
$$L = \sum_{\{v_i, v_j\} \in T} -\log(\mathrm{DEC}(z_i, z_j)) \tag{3.13}$$

Instead of evaluating the sum in the denominator of Equation 3.12, deepwalk uses an approximation called *hierarchical softmax*, which consists in assigning the vertices visited in the random walk to the leaves of a binary tree and computing the probabilities and products from paths in the tree (MNIH; HINTON, 2009). This approximation reduces the complexity of evaluating Equation 3.12 from $O(|V|)$ to $O(\log |V|)$. Figure 15 shows the node embedding for the Zachary's Karate Club using DeepWalk.

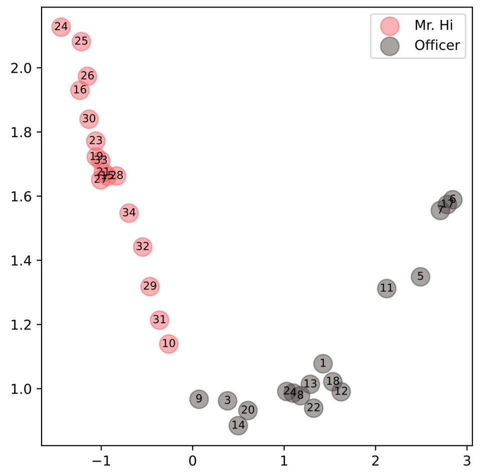
Another embedding method, which is also based on random walks is the node2vec (GROVER; LESKOVEC, 2016). Coming from the same language modeling background, there are many similarities between node2vec and deepwalk, except for two main differences. First, the walks in node2vec are biased by input parameters that control how far from the root node the walks tend to go, called the *in-out* parameter p, or how much the neighborhood of the root node is revisited, called the *return* parameter q.

Also, instead of using hierarchical softmax to approximate the softmax function, node2vec uses another strategy from language processing called *negative sampling* (MIKOLOV et al., 2013). Equations 3.12 and 3.13 also describe well the node2vec embedding method in the encoder-decoder framework. On its publication, node2vec outperformed most shallow embedding methods from the state of the art, including ones based in factorization methods and the deepwalk. Figure 16 shows the Zachary's Karate Club graph with embeddings evaluated using node2vec.

Figure 15 – Zachary's Karate Club graph with node embeddings generated using the DeepWalk method.



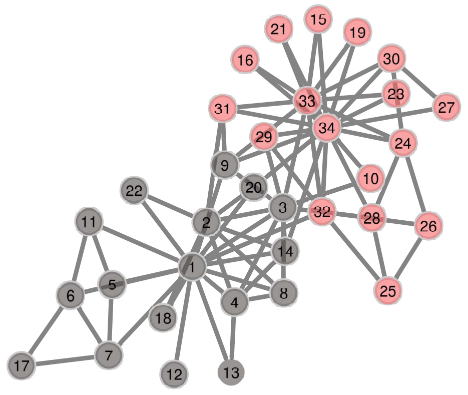
(a) Zachary's Karate Club with vertices colored according to the existing communities.



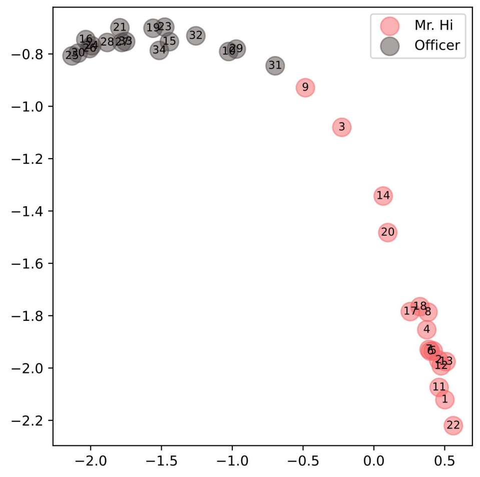
(b) Node embedding with DeepWalk ($t = 5$, $\gamma = 20$, $d = 2$, $w = 5$) for the Zachary's Karate Club in $\mathbb{R}^2$.

Source: the author

Figure 16 – Zachary's Karate Club graph with node embeddings generated using the node2vec method.



(a) Zachary's Karate Club with vertices colored according to the existing communities.



(b) Node embedding with node2vec ($d = 2$, $r = 20$, $l = 5$, $k = 5$, $p = 0.25$, $q = 4$) for the Zachary's Karate Club in $\mathbb{R}^2$.

Source: the author

All of the aforementioned embedding methods were developed to evaluate the optimal embeddings only looking at the graph's structure. This means that the embedding methods are agnostic to which machine learning task will be evaluated with the embeddings, which might not capture the desired information in some learning tasks. Also, the embedding methods do not make use of an embedding that was evaluated for a given vertex in the evaluation process of the embedding of another vertex, which turns to be computationally inefficient. However, shallow embeddings are still very useful since a whole set of classical machine learning techniques can be applied to the node embedding data once the embedding is done.

The following section introduces a different framework for accomplishing the embedding task by optimizing the embedding together with the machine learning task, the *Graph Neural Networks*.

3.3 Graph Neural Networks

The aforementioned embedding techniques are approaches to learn low-dimensional *shallow* embeddings of nodes in a graph. However, these embeddings are not learned while of the classification or regression task that has to be accomplished, or considering any external feature information that might help to solve the problem.

Graph Neural Networks (GNNs) are a general framework for defining neural networks in graph (or relational) data. The goal is to learn representation of nodes that are not a simple look-up embedding and consider both structural information from the graph and additional feature information.

A first obstacle for defining neural networks in graph data is that graphs cannot be represented in a canonical way with structured data (or the problem of embedding would not be a problem). If one tries to use common neural networks such as convolutional neural networks (CNNs) - by inputting the adjacency matrix of the graph - and recurrent neural networks (RNNs) - by inputting a serialization of the edge list - the output of the networks would change by a simple relabeling of the nodes, what should not happen.

Although the fundamental GNN models can be motivated in a number of ways, such as a generalization of convolutions (BRUNA et al., 2014), a common framework for understanding most GNN models will be introduced, where the nodes communicate a sort of *vector messages*, updating their internal state using neural networks (GILMER et al., 2017).

3.3.1 Neural Message Passing Framework

For the description of the framework and the different kinds of RNN, the notation will be as follows: given a graph $G = G(V, E)$ with a set of node features $X \in \mathbb{R}^{d \times |V|}$, the goal is to learn node embeddings $z_u, u \in V$. The formulation of the framework follows the notation from (HAMILTON, 2020).

To each node a *hidden embedding* $h_u^{(k)}$ is defined, where $u \in V$ and k is the iteration number of the learning process. The embedding information is updated with information from node u's open neighborhood $N(u)$. The iterative process is shown in Equation 3.14.

$$h_u^{(k+1)} = \text{UPDATE}^{(k)}\left(h_u^{(k)}, \text{AGGREGATE}^{(k)}(\{h_v^{(k)}, v \in N(u)\})\right) \tag{3.14}$$

The UPDATE and AGGREGATE are arbitrary functions, with the constraint to be differentiable (for example, neural networks). It is common to denote $m_{N(u)}^{(k)} = \text{AGGREGATE}^{(k)}(\{h_v^{(k)}, v \in N(u)\})$, which is called the *message* aggregated from u's neighborhood. The role of the AGGREGATE function is to summarize the information from all the nodes in u's neighborhood. The summarized information is then combined with the current hidden embedding $h_u^{(k)}$ of u through the UPDATE function. The initial embeddings are set to the input features of each node $h_u^{(0)} = x_u$, $u \in V$. After K iterations of message passing, one can use the hidden embeddings to define the actual node embeddings, as in Equation 3.15.

$$z_u = h_u^{(K)}, u \in V \tag{3.15}$$

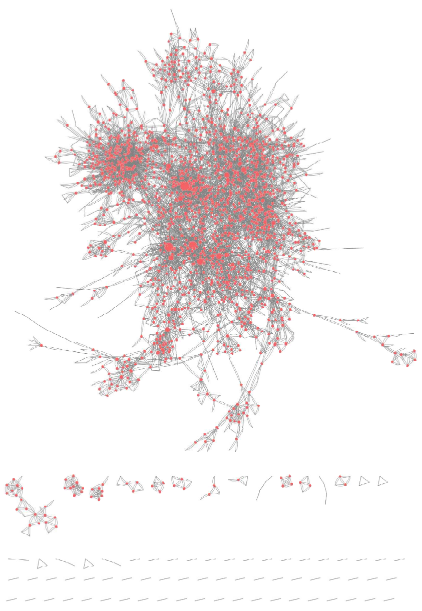
One of the most popular tasks which GNNs are used for is node classification. For this task, a subset of nodes $V_{train} \subset V$ of a graph $G = G(V, E)$ is selected for training in a supervised manner. A one-hot encoded vector $y_v \in \mathbb{Z}^c$ is defined for each vertex $v \in V$, where c is the number of classes. The classification is done by using a softmax function and a negative log-likelihood loss as in Equation 3.16, where z_v is the embedding of node v and w_i, $w_j \in \mathbb{R}^d$ are trainable parameters. The loss minimization is usually done using stochastic gradient descent (RUMELHART; HINTON; WILLIAMS, 1986).

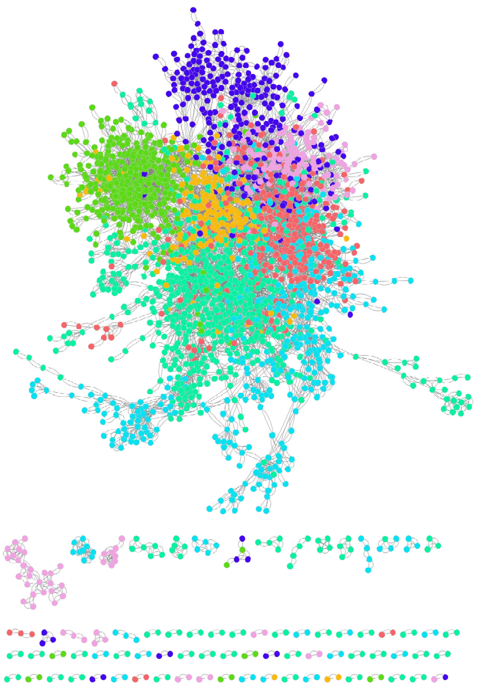
$$\mathcal{L} = \sum_{v \in V_{train}} -\log\left(\text{softmax}(z_v, y_v)\right) = \sum_{v \in V_{train}} -\log\left[\sum_{i=1}^{c} y_v(i) \frac{e^{z_v^T w_i}}{\sum_{j=1}^{c} e^{z_v^T w_j}}\right] \tag{3.16}$$

When applying GNNs for link prediction, the usual approach is to model edge features as a combination of pairwise node feature, such as addition, average or dot product. For the loss function, since link prediction can be seen as an special case of binary classification, the most common option is using binary cross entropy, which is a special case of Equation 3.16 with 2 classes (SCHLICHTKRULL et al., 2018). For graph classification tasks, the same loss function is used, but applied to graph-level embeddings, denoted z_G.

For node classification tasks, which are the case of this dissertation, the graph neural networks are commonly compared in citation datasets with nodes being papers, labeled based on the paper topic (BOJCHEVSKI; GÜNNEMANN, 2018). One of the most used datasets is *Cora*, which was built from an online portal of computer science research papers. It contains 19793 nodes with 8710 features each, 65311 edges and the nodes are labeled according to the paper topic (MCCALLUM et al., 2000). Edges represent citations. The goal is to consider only a subset of paper topics to predict the topics of the remaining ones. Usually a subset of the entire graph is used for evaluating and benchmarking GNNs, such as in (YANG; COHEN; SALAKHUDINOV, 2016), with 2708 nodes with 1433 features each and 10556 edges. There are 7 paper topics in the dataset, which results in a multi-class classification task. Figure 17 shows the Cora network subgraph, together with the nodes colored by class.

Figure 17 – Cora citation network subgraph commonly used for benchmarking in node classification.


(a) Cora citation network subgraph with node sizes proportional to degrees.


(b) Cora citation network subgraph with node labels colored according to their classes.

Source: the author

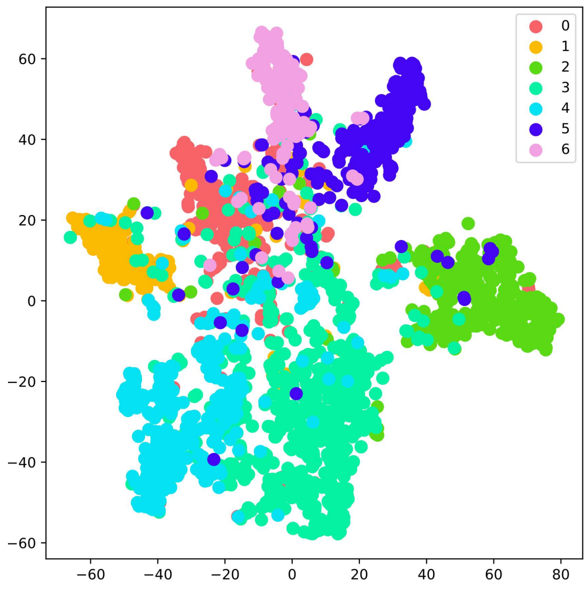
3.3.2 Graph Convolutional Networks

One widely applied GNN model is the Graph Convolutional Network (GCN), which can be viewed in the message passing framework in Equation 3.17. In this equation, both AGGREGATE and UPDATE functions were summarized in a single one by considering the closed neighborhood $\bar{N}(v)$. Moreover, the aggregation of neighborhood information is done using a trainable weight matrix W and a normalization with respect to the geometric mean of the size of the neighborhoods $\bar{N}(v)$ and $\bar{N}(u)$. The hidden embedding update uses a sigmoid function σ.

$$h_v^{(k)} = \sigma\left(W^{(k)} \sum_{u \in N(v) \cup \{v\}} \frac{h_u}{\sqrt{|N(v)||N(u)|}}\right) \tag{3.17}$$

Given the requirement of differentiability, the GCN uses very simple operations in both AGGREGATE and UPDATE functions. Despite this fact, the GCN can obtain decent results in a number of benchmarks (KIPF; WELLING, 2016). Figure 18 shows the node embeddings extracted from the GCN trained for node classification in the Cora dataset, which achieved 81.5% of accuracy. The dimensionality reduction from the chosen embedding dimension $d = 128$ to 2, for visualization, was done using the t-SNE method (MAATEN; HINTON, 2008).

Figure 18 – Visualization of the node embeddings produced using a GCN for node classification in the Cora dataset, with nodes colored according to their classes.



Source: the author

3.3.3 Graph Isomorphism Networks

In the opposite direction of the AGGREGATE and UPDATE simplicity from the GCN, Xu et al. (2018) develops a theoretical analysis for determining the most representative class of GNNs. The analysis results in a class of "maximally powerful" GNNs that learn the most representative functions by defining a Multi-Layer Perceptron (MLP) neural network in the UPDATE step. The authors still propose an architecture called the Graph Isomorphism Network (GIN), which can be defined in the message passing framework as in Equation 3.18, where ϵ is a trainable parameter, commonly set to 0.

$$h_v^{(k)} = \mathrm{MLP}^{(k)} \left(\left(1 + \epsilon^{(k)}\right) \cdot h_v^{(k-1)} + \sum_{u \in N(v)} h_u^{(k-1)} \right) \tag{3.18}$$

3.3.4 Relational Graph Neural Networks

The two aforementioned GNN structures, despite being able to consider directed graphs, cannot make use of any information of multi-relational graphs or heterogenous graphs. One of the first approaches to consider information from the graph edges, such as edge features, is known as the Relational Graph Neural Networks (RGCN). In this approach, Schlichtkrull et al. (2018) define an aggregation function that contains one trainable matrix per relation type, such as in Equation 3.19, where R is the number of relation types in the graph, $N_r(v)$ is the neighborhood of node v considering the relation type r and $c_{v,r}$ is a normalization constant that can be defined in a number of ways, mainly considering some operation over the node degrees.

$$h_v^{(k)} = \sigma \left(\sum_{r \in R} \sum_{u \in N_r(v)} \frac{1}{c_{v,r}} W_r^{(k-1)} h_u^{(k-1)} + W_0^{(k-1)} h_v^{(k)} \right) \tag{3.19}$$

The main downside of the RGCN is that the number of trainable parameters grows significantly when compared to the other GNN models, as the learning structure size is proportional to the number of relation types in the graph. Together with the proposition of the RGCN, the authors already propose a parameter-sharing scheme where the learnable matrices $W_r^{(k-1)}$ are written in terms of a set of basis matrices, reducing the number of total parameters in the model.

3.3.5 Graph Attention Networks

To add another layer of complexity to the aggregation process and inspired on the application of attention mechanisms in sequence processing, Veličković et al. (2017) introduced the Graph Attention Networks (GAT). The aggregation step in GATs uses attention weights $\alpha_{v,u}$ to each neighbor u of node v, defining a weighted average to aggregate information, instead of a simple average, as in Equation 3.20. The attention weights are also trained in the learning process, computed as in Equation 3.21, where $a \in \mathbb{R}^{2d}$ is a trainable attention vector, W is a trainable matrix and $\oplus$ denotes the concatenation operation.

$$h_v^{(k)} = \sigma \left(\sum_{u \in N(v)} \alpha_{v,u} W h_u^{(k)} \right) \tag{3.20}$$

$$\alpha_{v,u} = \frac{e^{a^T[Wh_v \oplus Wh_u]}}{\sum_{u' \in N(u)} e^{a^T[Wh_v \oplus Wh_{u'}]}} \tag{3.21}$$

3.4 Summary

The application of classical machine learning techniques on graphs can be done in a variety of approaches. The first step is to represent the graph elements in a structured way using an embedding technique. After having succesfully obtained an embedding for the graph, one might apply classical machine learning algorithms for classification or regression of the graph elements or the graph itself.

However, it is possible to go even further and, instead of obtaining the embeddings based on the graph structure or a given feature matrix, one might be able to train a Graph Neural Network (GNN) for obtaining the best embedding for doing the desired learning task, being that classification or regression.

In this dissertation, as part of the contingency analysis process consists on evaluating the possibly most critical contigencies in a graph model for an EPS, which can be framed as an binary (or possibly multi-class) classification problem, the Graph Machine Learning techniques can contribute in a relevant way to deliver consistent results.

In the next chapter, the proposed methodology for approaching the contingency analysis problem is described. The reference model from (COELHO, 2019) is considered, while making some modifications to the criticality measure. An exhaustive method is considered as the ground-truth for the learning methods to approximate.

4 Methodology

In this chapter the methodology for the application and evaluation of Graph Machine Learning (GML) in the screening of possibly critical contigencies in power systems is introduced. More than one formulation of the screening problem is made, through classical topological embedding such as node2vec and GNN methods such as Graph Convolutional Networks (GCN). The learning is made upon a reference criticality index, which is computationaly expensive and not feasible for evaluation in large systems. The role of this index is to express the relevance of a line or a bus in a single or multi contingency scenario. For this dissertation, the considered criticality measure is the sum of differences in the Current Flow Betweenness (CFB) Centrality for the simple graph model of the power system as proposed in Coelho (2019). However, for learning purposes, other indices could be used.

4.1 Criticality Index and Contingency Screening

For evaluating the criticality of transmission lines in possible single and multi-contingencies, Coelho (2019) propose the sum of the absolute differences in the CFB for the power system graph and the graph with the element of interest removed, or the edited graph.

In the following, the methodology is introduced for the cases of single and multi-contingencies of transmission lines, which is the scope of this dissertation. The case of contingencies in buses is not treated in this text. More formally, given a connected graph $G = G(V, E)$, let $C_G(v)$ be the CFB for vertex $v \in V$ in the power system graph G. If one denotes by $G \setminus e^k$, $e^k \subset E$ the graph obtained by editing G with the removal of edge set e^k, where $|e^k| = k \in \mathbb{N}$, then the CFB for vertex v in the edited graph is given by $C_{G\setminus e^k}(v)$. In this context, $G \setminus e^k$ is considered to be connected. If it is not, then the edge set e^k is immediately critical and should be treated separately. The absolute variation of the considered centrality measure, the CFB, is given in Equation 4.1.

$$\Delta C[G \setminus e^k](v) = |C_{G\setminus e^k}(v) - C_G(v)| \tag{4.1}$$

In order to obtain the most critical single contingencies, one should make $k = 1$ and then edge sets with single elements would be evaluated. For obtaining one value that defines the contingency for the entire graph, it is worth to define the Contingency Impact

CI of an edge set e^k as the sum of the ΔC for every vertex $v \in V$, as in Equation 4.2.

$$CI(e^k) = \sum_{v \in V} \Delta C[G \setminus e^k](v) \tag{4.2}$$

When analyzing single contingencies, each edge is associated with one possible contingency. However, for values of $k > 1$, the same edge may be involved with multiple events. Therefore, it is not possible to evaluate the impact of a single edge in multi-contingencies through CI. For addressing this issue, Equation 4.3 defines the Global Impact $GI(e)$ for an edge $e \in E$ as the sum of all Contingency Impacts $CI(e^k)$ for which the edge sets e^k contain edge e. As previously stated, if any of the edge sets removal disconnects the graph, then it is not considered in the calculation and should be treated separately. Therefore, only the edge sets e^k whose removal does not disconnect graph G are used in the computation of GI.

$$GI(e) = \sum_{e \subset E,\, e \in e^k} CI(e^k) \tag{4.3}$$

For $k = 1$, i.e., when analyzing single contingencies, $GI(e) = CI(e)$. However, for $k > 1$, the Global Impact $GI(e)$ is an index of the criticality of a given edge e with respect to k-contingencies. With the above definitions one may already rank the most and the least critical lines with respect to multi-contingencies. However, the last step in the methodology proposed by (COELHO, 2019) is normalizing the values of GI to the range $[0, 1]$, which are denoted gi, by dividing by the greatest value among all the edges, as in Equation 4.4. Then, thresholds are stablished for defining critical, non-critical and regular edges. The chosen thresolds, that presented good results when compared to previous methodologies in a set of test systems are $gi(e) \geq 70\%$ for considering an edge e *critical* and $gi(e) \leq 20\%$ for considering an edge *non-critical*. The remaining edges, for which $20\% \leq gi(e) \leq 70\%$ are said to be *regular*.

$$gi(e) = \frac{GI(e) - \min_{e \in E}\{GI(e)\}}{\max_{e \in E}\{GI(e)\} - \min_{e \in E}\{GI(e)\}} \tag{4.4}$$

As originally formulated, the methodology gives three classes for the edges of a graph, either critical, non-critical or regular. For this dissertation, the methodology will be adjusted so that the classification problem that will be formulated in the following becomes a binary classification. Thus, only the upper threshold of 70% is considered and the existing classes are *critical* and *regular* edges. This is done because the objective of the learning methodology is to classify possibly crtitical edges, with the non-critical ones not being much relevant in this context. The process of obtaining $gi(e)$ for an edge e of a graph G is summarized in Algorithm 1.

Algorithm 1: Obtaining the normalized criticality index $gi(e)$

```
Data: Simple graph $G$, contingency order $k$
Result: Normalized criticality index $gi(e)$ for every edge $e$ of $G$
Compute every subset of edges with $k$ elements $e^k \subset E(G)$;
Compute the current flow betweeness centrality $C_G(v)$ for every vertex $v$ of $G$;
foreach $e^k$ do
    if $G \setminus e^k$ is connected then
        foreach $v \in V(G)$ do
            $C_{G\setminus e^k}(v)$;
            $\Delta C[G \setminus e^k](v) = |C_{G\setminus e^k}(v) - C_G(v)|$ ;
        $CI(e^k) = \sum_{v\in V} \Delta C[G \setminus e^k](v)$;
    else
        Continue;
Compute the global impact;
$GI(e) = \sum_{e\subset E,\, e\in e^k} CI(e^k)$;
$gi(e) = \frac{GI(e)-\min_{e\in E}\{GI(e)\}}{\max_{e\in E}\{GI(e)\}-\min_{e\in E}\{GI(e)\}}$
```

4.2 Formulation of the Screening Problem with Graph Machine Learning

The Contingency Screening problem for lines consists in evaluating the lines that could lead to adverse operational conditions on their absence, which are called *critical* lines. Therefore, since every edge is labeled as critical or non-critical (regular), the contingency screening can be seen as en edge classification problem in the graph model of the power system.

However, as mentioned in Chapter 3, the GML approaches were developed to learn on graph based on node embeddings. More specifically, the GNN architectures are formulated for solving either the node classification or the link prediction problems, and no direct approach for the edge classification problem was found in the literature. Mainly, graph models are supposed to be made with the nodes as the elements of interest and the links are the connections between these elements. Despite some GNN architectures such as the RGCN being able to deal with multigraphs, the link types are previously defined and used as input for a classification task on the nodes. Meanwhile, the link prediction task requires the assumption of an incomplete graph model, and the outputs are possibly existing edges, and therefore are not made for classifying already existing edges.

For dealing with the absence of GML models for edge classification, a couple of alternatives are developed, as detailed in Section 4.3. Mainly, either the edge embeddings are obtained through a commutative operation on the node embeddings or the classification is made in the line graph of the power system graph model.

The edge classification task in the contingency screening problem has an unbalanced nature. From a large set of edges, the goal is to identify the most critical ones, that could lead to severe outages in the power system. Therefore, it is expected that the number of critical edges is much less than the total number of edges in the graph model, leading to an unbalanced binary classification task.

For the unbalanced classification problem, the adopted strategy is to balance the training set. The balancing strategy is the *random under-sampling*. For the node classification task, given a graph $G = G(V, E)$ with $|V| = n$, if the nodes are divided into classes V_1 and V_2 such that $|V_1| = n_1$, $|V_2| = n_2$ and $n_1 + n_2 = n$, one defines $n_{min} = \min\{n_1, n_2\}$. If the desired training split is $p\%$, then the same ratio is sampled from the class with least nodes, obtaining a number of training nodes for the rarest class $\lfloor n_{min} \cdot p\% \rfloor$. The other class have the same amount of nodes sampled, in order to balance the training node set with the same number of elements on both classes.

4.3 Proposed Approaches

4.3.1 Classification on Generated Embeddings (CGE)

The first evaluated approach for the edge classification on the power systems graph models was using the node embeddings generated without any GNN, but with shallow embedding techniques. The chosen technique was the node2vec, since it is known as the state of the art for shallow node embeddings. The node embeddings were converted into edge embeddings by using the element-wise product operation, i.e., given the embeddings $z_1, z_2 \in \mathbb{R}^d$ for nodes $v_1, v_2 \in V$, respectively, the embedding for the edge $e = \{v_1, v_2\}$ is given by $z_e = z_1 * z_2$, where $*$ denotes the element-wise product operation. The process of obtaining the training, validation and testing datasets is shown in Algorithm 2.

Algorithm 2: Obtaining the dataset for the *CGE* approach

```
Data: Simple graph $G$, criticality label for each edge $y_e$
Result: Pairs in the form of $(z_e, y_e)$, where $z_e$ is the edge embedding and $y_e$ the
        criticality label for edge $e \in E(G)$;
Compute the node embeddings $z_v$ with node2vec for every vertex $v \in V(G)$;
foreach $v \in V(G)$ do
    $z_v = \text{node2vec}(v)$
Compute the edge embeddings $z_e$ with the inner product of its vertices embeddings;
foreach $e \in E(G)$ do
    $z_e = z_1 * z_2$, where $e = \{v_1, v_2\}$
Associate each edge embedding $z_e$ with its label $y_e$;
```

Since the node2vec method only generates embeddings, the classification task relies on conventional machine learning classifiers, which are applied in a set of samples in

$\mathbb{R}^d$ and trained against a set of labels, obtained via the exhaustive method described previously in Section 4.1. The selected classification method is the Random Forests from Breiman (2001), since there is no known result about the distribution of the criticality indices and the embeddings can be high-dimensional. This approach will be reffered to as CGE throughout the text.

4.3.2 GNN Node Classification in the Line Graph (CLG)

The second evaluated approach is to use a conventional node classification GNN architecture applied to the line graph of the power system graph model. Since each node in the line graph is associated with an edge in the power system graph model, classifying the nodes in the line graph can be seen as classifying the edges in the model graph. However, the edges in the line graph represent the existence of nodes that are shared by the edges, which is usually greater then number of edges in the model graph. Additionaly, the greater number of nodes and edges in the line graph tends to slow down the training of the GNN methods in this approach. Algorithm 3 demonstrates the process of obtaining the dataset for the CLG approach.

Algorithm 3: Obtaining the dataset for the CLG approach

```
Data: Simple graph $G$, criticality label for each edge $y_e$
Result: Pairs in the form of $(v', y_{v'})$, where $v'$ is the vertex in the line graph
        associated with edge $e$ and $y_{v'}$ the criticality label for vertex $v'$;
Compute the line graph $G'$ of $G$;
Denote by $v'$ the vertex in $V(G')$ associated with edge $e \in E(G)$ ;
Define the classification label for each vertex in the line graph;
foreach $v' \in V(G')$ do
    $y_{v'} = y_e$
All the vertices $(v')$ now have labels $(y_{v'})$ for training a $GNN$ classifier;
```

Since GNNs rely on learning the node embedding together with the classification step, some initial embedding must be given to the graph model. For defining the initial embeddings, one might consider a number of conventional graph measures that intuitively tend to facilitate the classification process. However, the influence of the initial embeddings tend to be small in the final result, so the common practice is to randomly initialize the embeddings for all nodes and let the GNN learn the best embeddings. This approach will be reffered to as CLG throughout the text.

4.3.3 GNN Regression on the Line Graph (RLG)

The third and last approach for the transmission line classification problem is an indirect approach for obtaining the critical and regular edges by estimating the criticality index. A GNN architecture is trained for a regression task on the line graph of the power system model graph, where the target values are the criticality indices on each node. This approach requires evaluating both the regression and the conclusion of the classification done with the estimated values for each criticality index. This approach will be reffered to as *RLG* throughout the text, and the process for obtaining the dataset for this approach is shown in Algorithm 4.

Algorithm 4: Obtaining the dataset for the *RLG* approach

```
Data: Simple graph $G$, criticality index for each edge $gi(e)$
Result: Pairs in the form of $(v', y_{v'})$, where $v'$ is the vertex in the line graph
        associated with edge $e$ and $y_{v'}$ the regression target for vertex $v'$;
Compute the line graph $G'$ of $G$;
Denote by $v'$ the vertex in $V(G')$ associated with edge $e \in E(G)$ ;
Define the regression target for each vertex in the line graph;
foreach $v' \in V(G')$ do
    $y_{v'} = gi(e)$
All the vertices $(v')$ now have targets $(y_{v'})$ for training a $GNN$ for regression;
```

4.4 Evaluating the GML Application

The proposed approaches are applied to the test systems described in Section 5.1. For each system, each approach is evaluated for single contingencies ($k = 1$) and multi-contingencies up to $k = 4$. A number of hyperparameters is tested for each approach, ensuring that the obtained results are actually reflecting the learning capability of the proposed architecture.

For each combination of hyperparameters, multiple training tasks with different random embedding initializations are made, in order to evaluate the relevance of each parameter in the architecture instead of mistaking the obtained values due to lucky embedding initializations.

For scoring the binary classification task, classical classification metrics are used. For the following definitions, a *positive* sample will be associated with a critical line, whereas a *negative* sample will be associated with a regular line. In the next equations TP represent the number of *true positives*, i.e., the critical lines that are predicted as critical. TN if the number of *true negatives*, i.e., the regular lines that are predicted as regular. Similarly, FP and FN are *false positives* and *false negatives*, respectively, which represent the lines that are predicted in a wrong manner.

The accuracy is probably the most required metric of a classifier, but is not enough to ensure that the learning was properly done. Specially in the task of classifying lines as critical or regular, which is an imbalanced classification task, high values of accuracy can be obtained by simply classifying all samples as belonging to the most frequent class.

For preventing this kind of conclusion, another set of classical metrics is used. The first is the *F1-score*, which is defined using two other metrics, the *precision* and the *recall*, which are defined in Equations 4.5 and 4.6. The *F1-score* is defined in Equation 4.7.

$$Precision = \frac{TP}{TP + FP} \tag{4.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.6}$$

$$F1 = 2\,\frac{Precision \cdot Recall}{Precision + Recall} \tag{4.7}$$

All the above defined metrics take values in the interval $[0, 1]$, where greater values means better performance. The precision requires that the classifier has a low rate of false negatives, whereas the recall requires that the classifier has a low rate of false negatives. The *F1-score* is the harmonic mean of *Precision* and *Recall*, only assuming values close to 1 if the classifier has low rates of both false positives and false negatives.

For the regression approach, the evaluation is done in two steps. First, the capacity of predicting the correct values of the criticality index for each line is evaluated through regression metrics, which are defined in the following. Then, the predicted values are used, together with the 70% threshold to determine the critical and regular lines, which follows an evaluation using classification metrics. For the regression step, the considered evaluation metric is the R^2, which is defined in Equation 4.8. In the definitions, the test set is denoted by $\{y_1, \ldots, y_N\}$ and the predicted samples by $\{\hat{y}_1, \ldots, \hat{y}_N\}$. The average of the test set is denoted by $\bar{y}$, which correspond to the baseline regressor that always outputs the mean.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{4.8}$$

4.5 Summary

In this chapter, the adopted methodologies for the contingency screening and the application of GML to the screening task are explained. Finnaly, the metrics that are used for scoring the learning architectures are defined. In the next chapter, the results of the application of the above described approaches are shown, together with the proposal of improvements for the learning architectures and the applied criticality index.

5 Results

In this chapter, the results of the described methodology applied to the test systems are shown. In the beginning of the chapter, the test systems are described. Then, the behavior of the criticality index on each test system is analyzed and the distribution of the indices $gi(e)$ are shown. Following, each of the approaches are shown separately, highlighting the architectures and the evaluated hyperparameters. Each of the test systems is evaluated for single and multi-contingencies up to $k = 4$. After, the approaches are compared for each test system considering the best set of hyperparameters. In the end, some analysis and considerations are made about the adopted criticality measure and the results obtained by each of the learning approaches.

The developed code for evaluating both the global impacts $gi(e)$ and the GML analysis is available in <https://github.com/rjmalves/k-contingency-screening> and <https://github.com/rjmalves/gnn-contingency-analysis>, respectively.

5.1 Test Systems

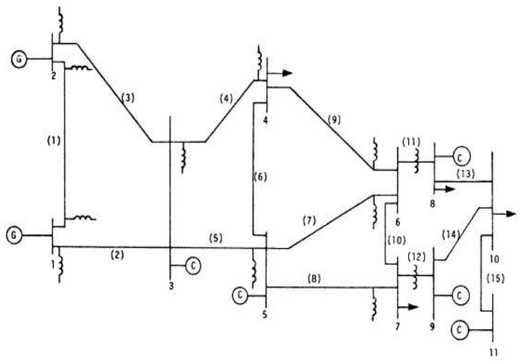
For evaluating the learning of criticality index though GML, a set of test systems is used, which are commonly considered when proposing methodologies for power system analysis. For each test system, different training and testing data splits are evaluated.

The first considered system is an 11-bus model of a designed system for transfering generated power from Itaipu generating facilities in Parana River from Southern Brazil along 800 kV long transmission lines, extracted from (EJEBE; WOLLENBERG, 1979). Previous reliability analysis in this system indicated that voltage problems might occur after the loss of critical lines, which was assessed through power transfer reduction. For this reason, the system was filled with synchronous condensers and reactors in order to balance reactive power, and was considered a good option for the power transfer issue in the region.

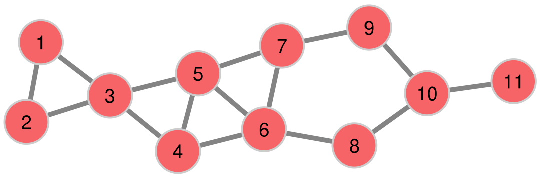
Other test systems that are used in the evaluation of the methodology are the IEEE Bus Test Systems, which are widely used for evaluation of new methodologies in sensitivity and power flow analysis. IEEE 39 Bus is known as New England power system, which contains 10 generators and 46 transmission lines and is shown in Figure 20 together with its graph model.

IEEE 57 Bus and 118 Bus are approximations of the American Electric Power system in early 1960s. The IEEE 300 Bus is a synthetic power system which was developed by IEEE for test purposes. All the test systems, together with their graph models are

Figure 19 – ITAIPU 11 bus single line diagram and graph model.



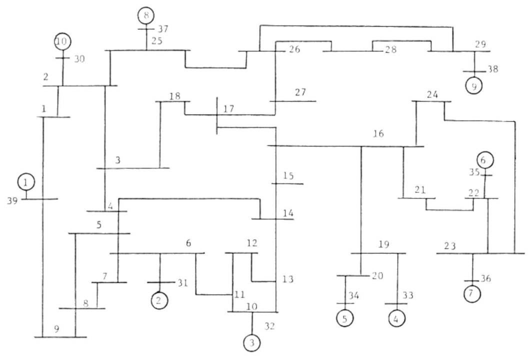
(a) ITAIPU 11 bus single line diagram.



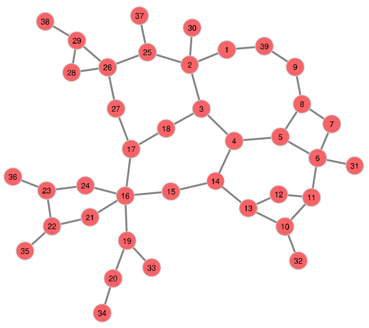
(b) ITAIPU 11 bus graph model.

Source: (EJEBE; WOLLENBERG, 1979) (diagram) and the author (graph model).

Figure 20 – IEEE 39 bus single line diagram and graph model.



(a) IEEE 39 bus single line diagram.



(b) IEEE 39 bus graph model.

Source: (ATHAY; PODMORE; VIRMANI, 1979) (diagram) and the author (graph model).

shown in Figures 21, 22 and 23.

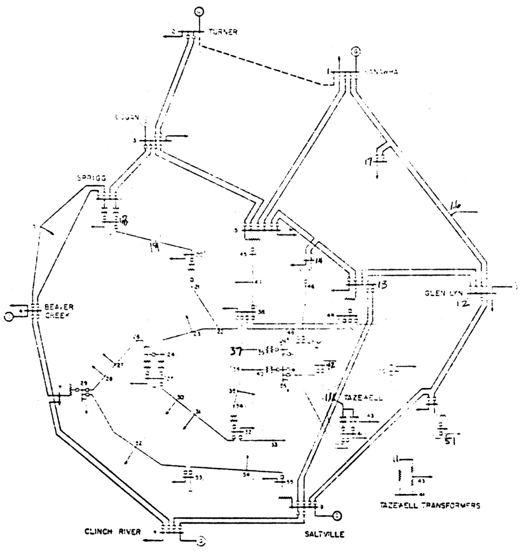
5.2 The Criticality Index on the Test Systems

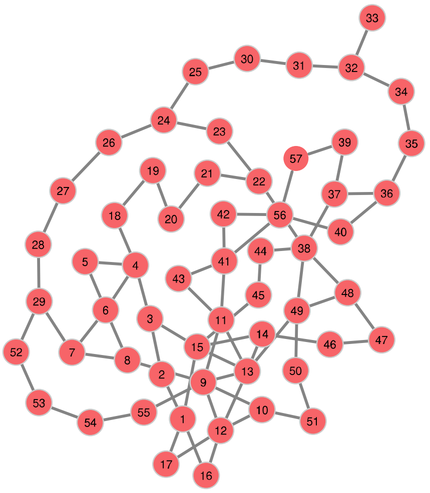
5.2.1 ITAIPU 11

The ITAIPU 11 test system is an 11-bus system with 15 lines. It is a small test system with respect to the number of buses and lines, since most real world EPS have hundreds or thousands of buses and lines.

Figure 24 shows the ITAIPU 11 test system with critical edges colored as orange and disconnecting edges colored green and dashed. Edge widths are determined by their criticality. The edges considered as critical differ significantly when comparing single and multiple contingencies. For $k = 1$, edges $\{6, 8\}$, $\{7, 9\}$ and $\{9, 10\}$ are considered critical

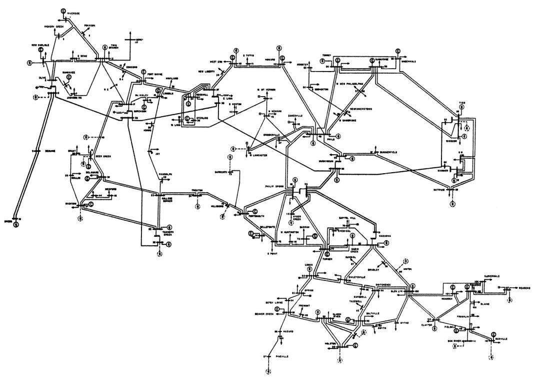
Figure 21 – IEEE 57 bus single line diagram and graph model.

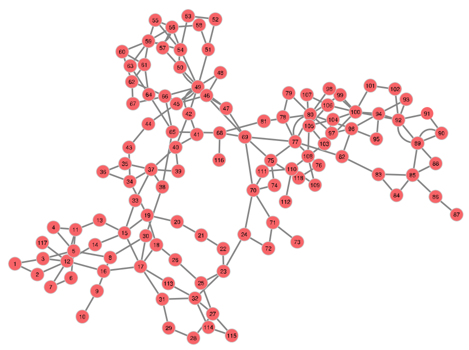

(a) IEEE 57 bus single line diagram.


(b) IEEE 57 bus graph model.

Source: (CHRISTIE, 1993c) (diagram) and the author (graph model).

Figure 22 – IEEE 118 bus single line diagram and graph model.


(a) IEEE 118 bus single line diagram.

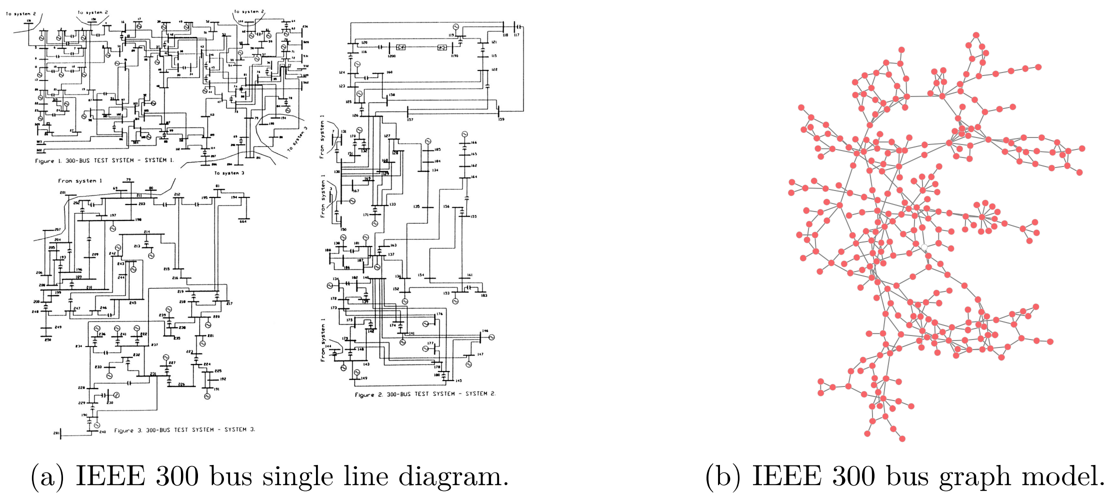

(b) IEEE 118 bus graph model.

Source: (CHRISTIE, 1993a) (diagram) and the author (graph model).

following the methodology from (COELHO, 2019). For $k = 4$, completely different edges are considered critical, mostly in the highly connected part of the graph, such as edges $\{3, 5\}$ and $\{6, 7\}$.

This is a consequence of the criticality index formulation. For computing the global criticality $gi(e)$ of and edge e through the exhaustive method, only the non-disconnecting sets of removals are considered. Therefore, since the edges considered critical for $k = 1$ mostly disconnect the graph for higher values of k, when they are removed simultaneously, their global criticality for multiple contingencies receive a lower score.

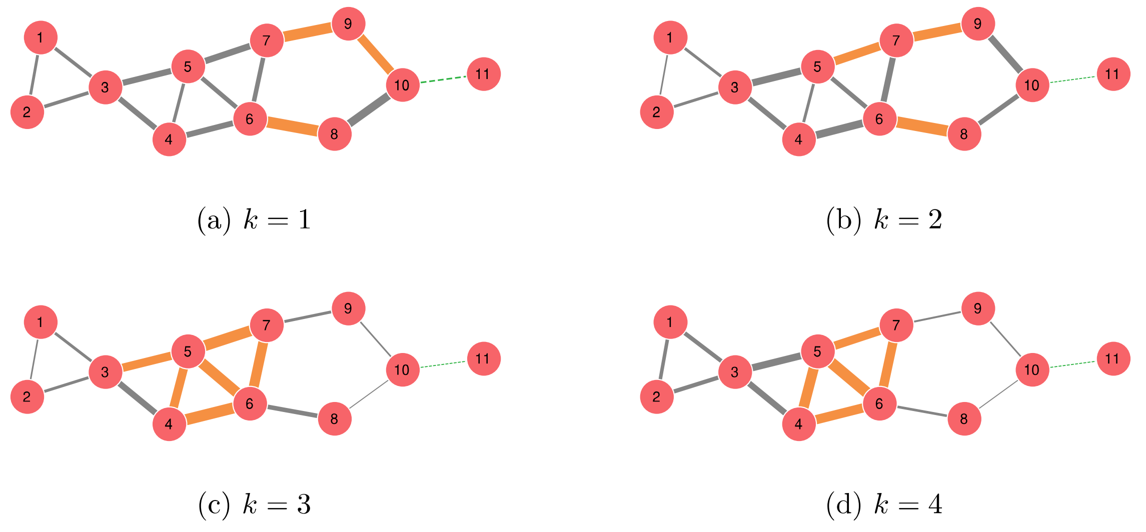
Figure 23 – IEEE 300 bus single line diagram and graph model.


(a) IEEE 300 bus single line diagram. (b) IEEE 300 bus graph model.

Source: (CHRISTIE, 1993b) (diagram) and the author (graph model).

Another consequence of the formulation of the Criticality Index is that edges which connect to degree 1 vertices are never considered for evaluation, since their removal always disconnect the graph. For the ITAIPU 11 test system, this is the case of edge $\{10, 11\}$. Therefore, only 14 of the 15 edges are considered in the classification.

Figure 24 – Criticality Index and critical lines on the ITAIPU 11 graph model.



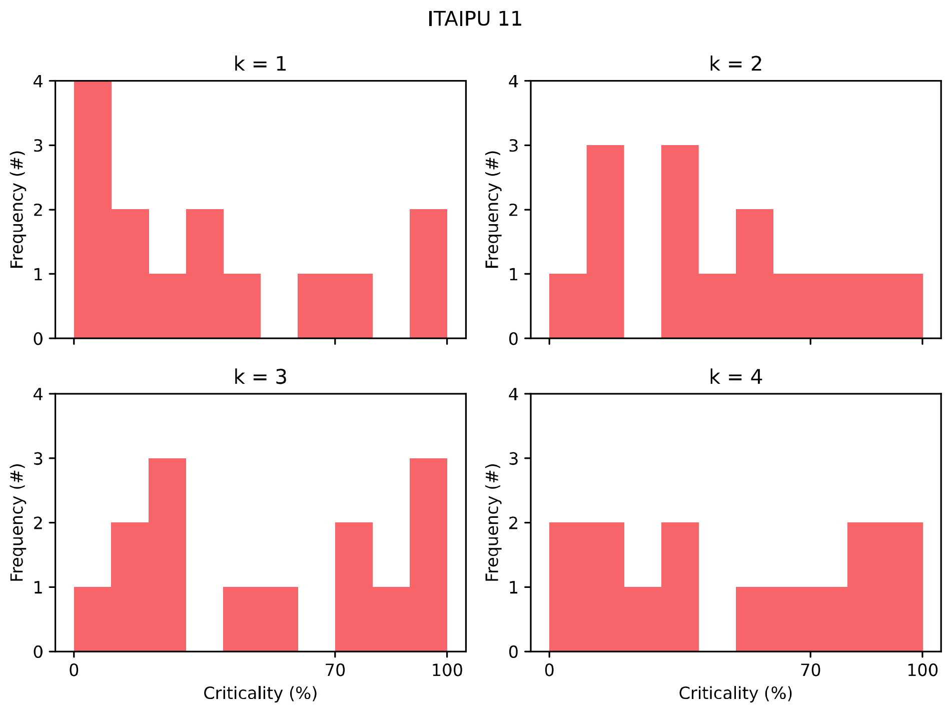
(a) $k = 1$ (b) $k = 2$

(c) $k = 3$ (d) $k = 4$

Source: the author.

Given that the formulated classification problems in approaches *CGE* and *CLG* are imbalanced, as highlighted in Section 4, it is important to analyze the degree of imbalancement of the lines among the critical and regular classes. This task is done through the distributions of the criticality indices, shown in Figure 25. As one can see, for the ITAIPU 11 test system, with $k = 1$ and $k = 2$ there are 3 critical edges. For $k = 3$ and $k = 4$, the number of critical edges increase to 6 and 5, respectively.

Figure 25 – Distribution of the Criticality Index $gi(e)$ on ITAIPU 11.



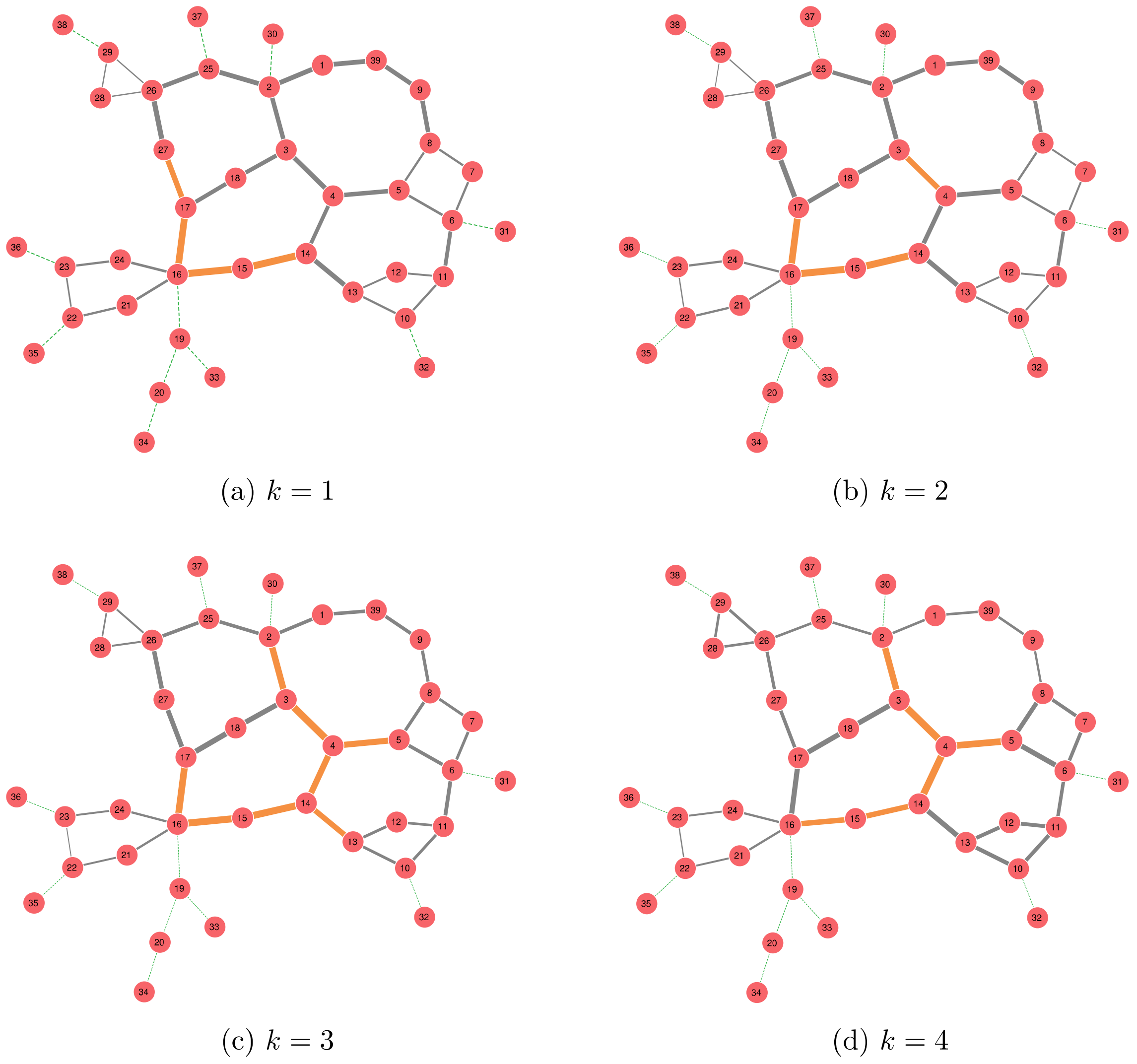
Source: the author

5.2.2 IEEE 39

The IEEE 39 test system is significantly larger than the ITAIPU 11, with 39 buses and 46 lines. In Figure 26, critical edges are colored as orange, whereas disconnecting edges are colored green and dashed. Edge widths are determined by their criticality. One can see that the edges $\{14, 15\}$ and $\{15, 16\}$ are considered critical for all k. Edge $\{16, 17\}$ is critical for $k = 2$ and $k = 3$, whereas $\{17, 27\}$ is critical only for $k = 1$. There are edges that are critical in multi-contingency scenarios but are not critical in single-contingencies, such as $\{2, 3\}$ and $\{4, 5\}$.

The distribution of the Criticality Index for IEEE 39 is shown in Figure 27. As in the ITAIPU 11 test system, the number of critical edges following the 70% threshold from the methodology and the distribution of the criticality index itself change with the value of k. For $k = 1$ and $k = 2$, 4 edges are considered critical. For $k = 3$ and $k = 4$, the number of critical edges increases to 8 and 6, respectively.

Figure 26 – Criticality Index and critical lines on the IEEE 39 graph model.


(a) $k = 1$ (b) $k = 2$

(c) $k = 3$ (d) $k = 4$

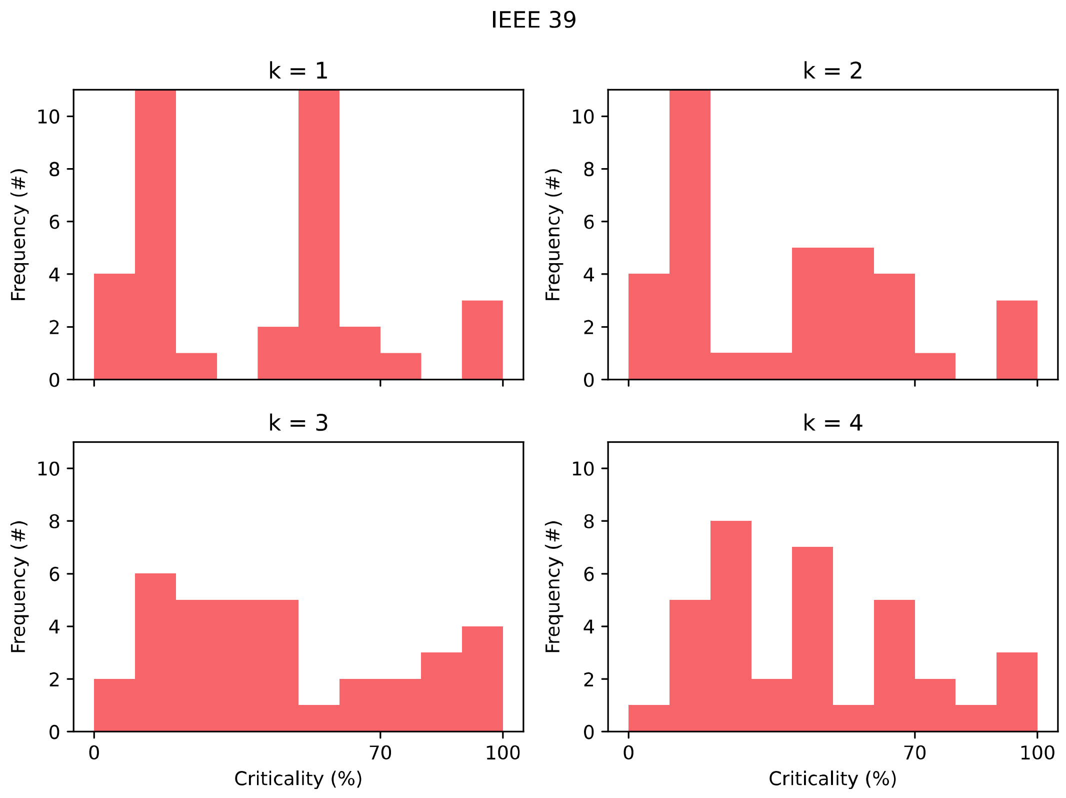
Source: the author.

5.2.3 IEEE 57

For the IEEE 57, with 57 buses and 78 lines, there is a significant difference between the edges considered critical for $k = 1$ and $k = 2$ when compared to $k = 3$ and $k = 4$, as can be seen in Figure 28. Critical edges are colored as orange, while disconnecting edges are colored green and dashed. Edge widths are determined by their criticality. For single contingencies and $k = 2$, a large number of critical edges is located in the less dense region of the graph, namely the edges $\{24, 25\}$, $\{24, 26\}$, $\{26, 27\}$ and $\{28, 29\}$. Some edges that are critical for $k = 3$ are not considered critical for and $k = 4$, namely edges $\{22, 23\}$, $\{22, 38\}$ and $\{37, 38\}$. Actually, the number of critical edges decreases with k. For k from 1 to 4, the number of critical edges is, respectively, 10, 10, 5 and 3.

The distribution of the Criticality Index for the IEEE 57, shown in Figure 29 has

Figure 27 – Distribution of the Criticality Index $gi(e)$ on IEEE 39.



Source: the author

a distinct characteristic when compared to both ITAIPU 11 and IEEE 39 test systems. Despite the variation in the number of critical edges, the distributions have a similar shape for all k. A large number of edges have criticality close to 0 and the frequency decreases as the criticality increases.

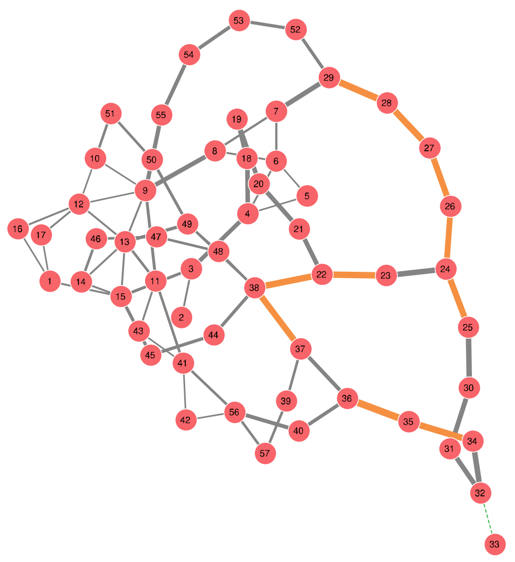
The IEEE 57 is the first evaluated graph in this dissertation that shows a similar distribution of the Criticality Index for single and multiple contingencies. Even with such similarities in the distribution, the constant 70% threshold results in a considerable different number of critical labels when approaching the classification problem since, despite some edges are critical for all k, the number of critical edges varies as k increases.

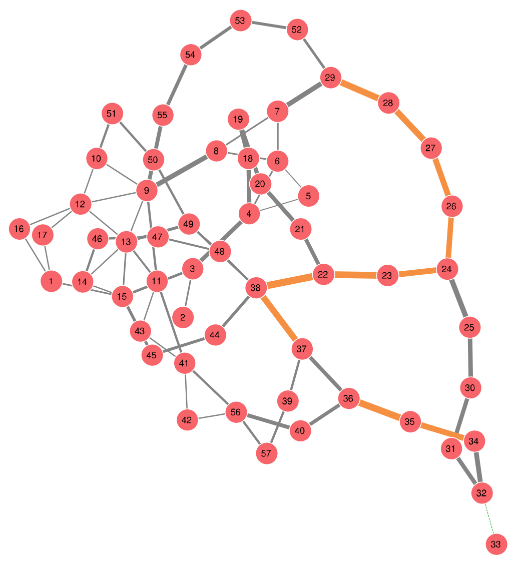
5.2.4 IEEE 118

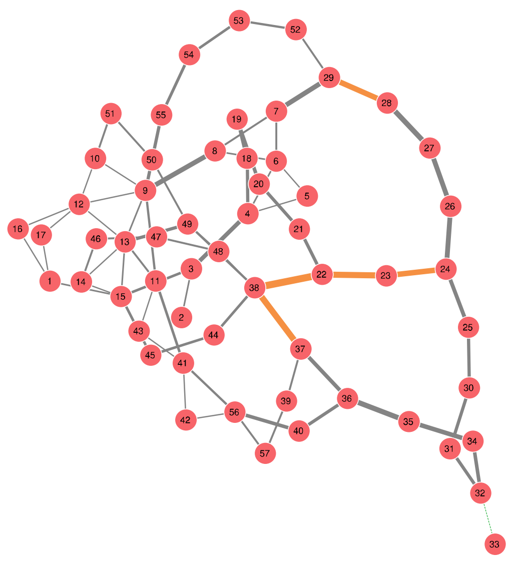
For the graph model of the IEEE 118 test system, as can be seen in Figure 30, the same edges are considered critical for all values of k. Critical edges are colored as orange, while disconnecting edges are colored green and dashed. Edge widths are determined by their criticality. Edges $\{23, 24\}$ and $\{38, 65\}$ are the only critical ones in the entire graph, which contains 118 buses and 179 edges.

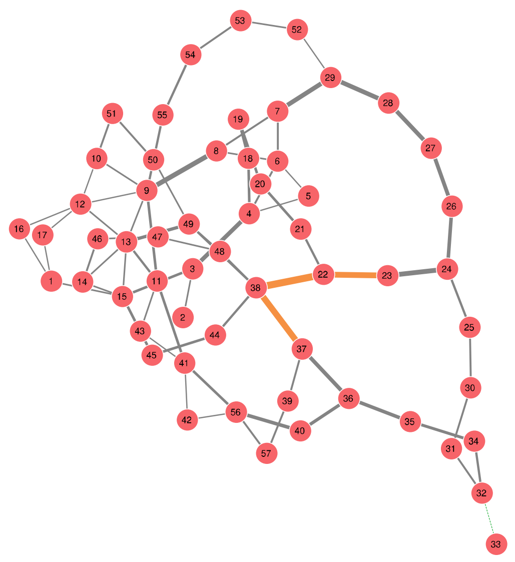
One can notice an emerging pattern regarding the number of critical edges in the

Figure 28 – Criticality Index and critical edges on the IEEE 57 graph model.


(a) $k = 1$


(b) $k = 2$


(c) $k = 3$

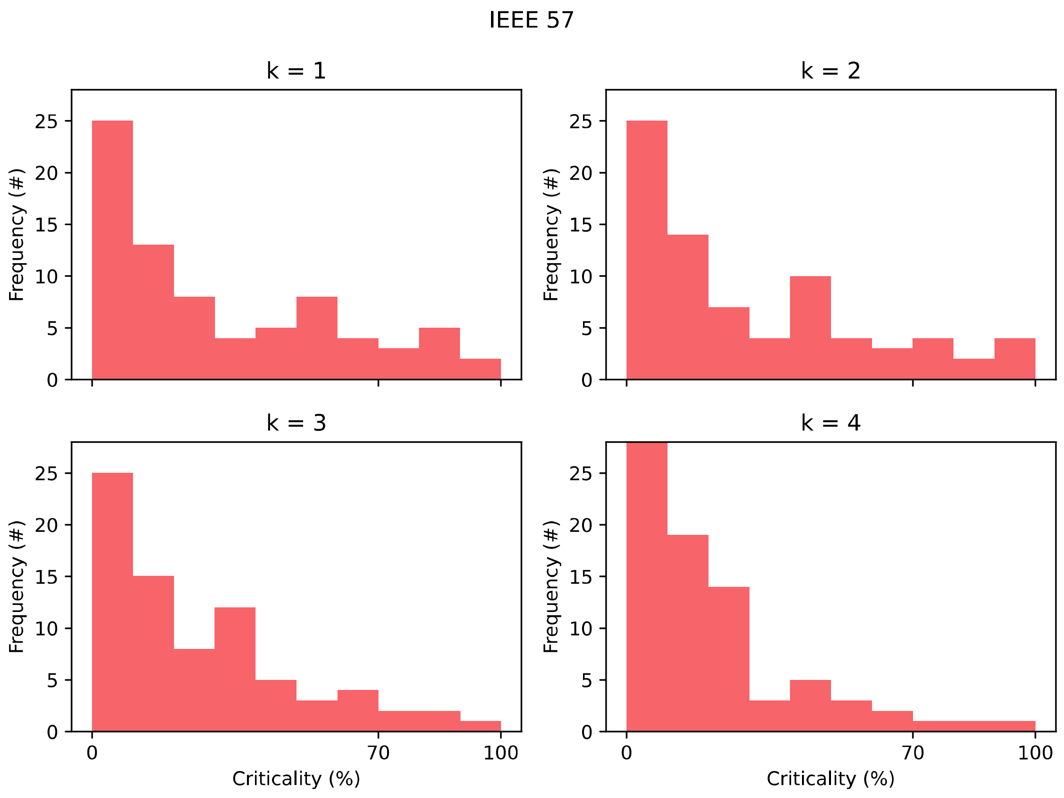

(d) $k = 4$

Source: the author.

test systems. For larger graphs, a smaller percentage of the edges is considered critical. While for the ITAIPU 11 test system 6 of the 15 edges are critical for $k = 3$, in the IEEE 118 only 2 out of 179 edges are considered critical. This is an important factor, since the imbalancement in the classification problem leads to difficulties in training the classifier, which will not learn to properly differentiate the existing classes.

Figure 31, which illustrates the distribution of the Criticality Index for the IEEE 118 graph, shows another emerging pattern in the used test systems. The shape of the distribution changes even less with the values of k when compared to the IEEE 57 graph. Also, a larger portion of the edges have lower criticality values. For instance, more than

Figure 29 – Distribution of the Criticality Index $gi(e)$ on IEEE 57.



Source: the author

100 of the 179 edges have criticality below 10%.

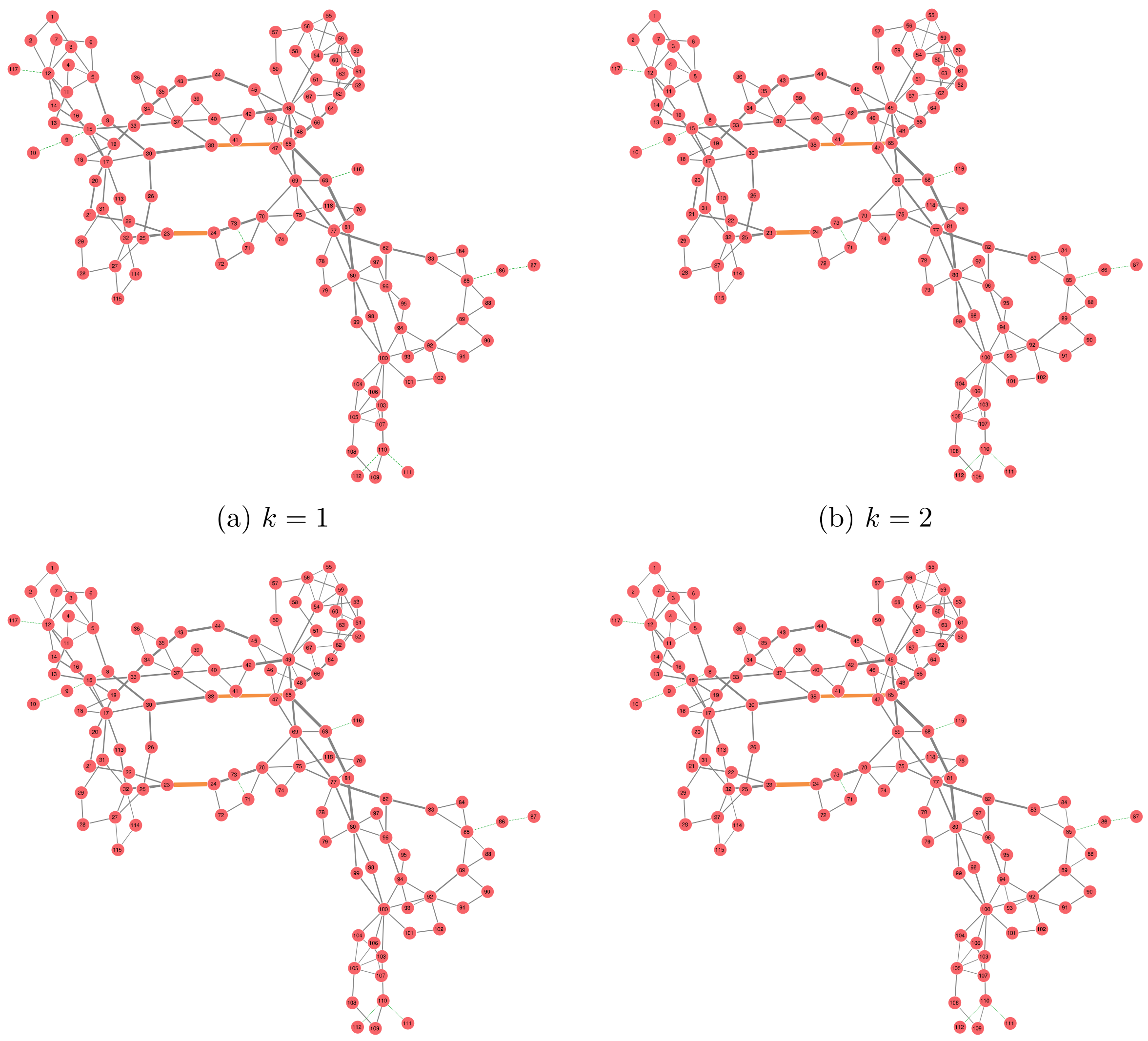
The imbalance in the frequency of the lower criticalities when compared to the greater criticalities tends to be an obstacle to the application of learning techniques in this context. As it will be shown in the following sections, for the test systems with only 2 critical edges, as random under-sampling is used for treating the imbalance during the training phase, the learning techniques are trained with only 2 edges and applied to the remaining ones.

In this context, it is reasonable to expect difficulties for adjusting the architecture and the hyperparameters of the learning techniques to ensure the maximum information extraction from a small number of samples.

5.2.5 IEEE 300

For the IEEE 300, some patterns that emerged while analyzing the IEEE 118 graph are still present. As can be seen in Figure 32, only 3 edges are considered critical, namely the $\{54, 123\}$, $\{122, 123\}$ and $\{116, 119\}$. As the IEEE 300 graph has 300 buses and 409 lines, the imbalance in the edge labels is even more noticeable than in the IEEE 118 graph. The classification task for the IEEE 300 tends to be challenging, since the training will

Figure 30 – Criticality Index and critical edges on the IEEE 118 graph model.



(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

(d) $k = 4$

Source: the author.

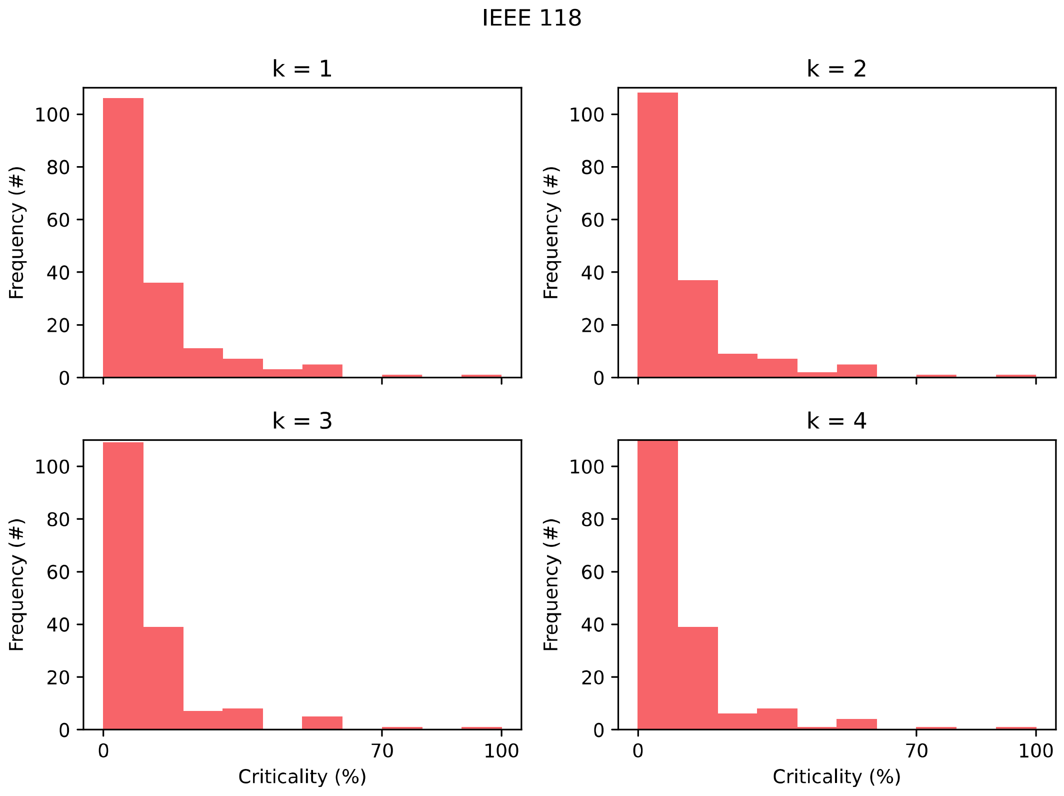
only be done with 1 or 2 edges of each class, resulting in a total of 2 or 4 edges.

The distribution of the criticalities for the IEEE 300 graph is shown in Figure 33. As for the IEEE 118 graph, the shape of the distributions have little change with the values of k.

For all the evaluated cases, around 250 edges have criticality below 10% and approximately 50 have criticality between 10% and 20%. Given the total number of edges in the graph is 409, more than 75% of the edges have criticality below 20%.

Given the observed emerging patterns, a couple of conjectures can be made. First, for large graphs, the multi-contingency screening following the methodology described in Chapter 4 tends to reach the same results of the single-contingency screening, i.e., the value of k does not matter for determining critical edges. In this scenario, the problem

Figure 31 – Distribution of the Criticality Index $gi(e)$ on IEEE 118.



Source: the author

could be solved by inspecting only single contingencies.

A second conjecture is associated with the imbalanced nature of the classification task. For large graphs, the distribution of the Criticality Index is such that the frequency is monotonic decreasing with the criticality. In this context, large scale EPS graphs tend to result in more imbalanced classification problems, beign an obstacle for the application of GML for contingency screening using the Criticality Index from (COELHO, 2019).

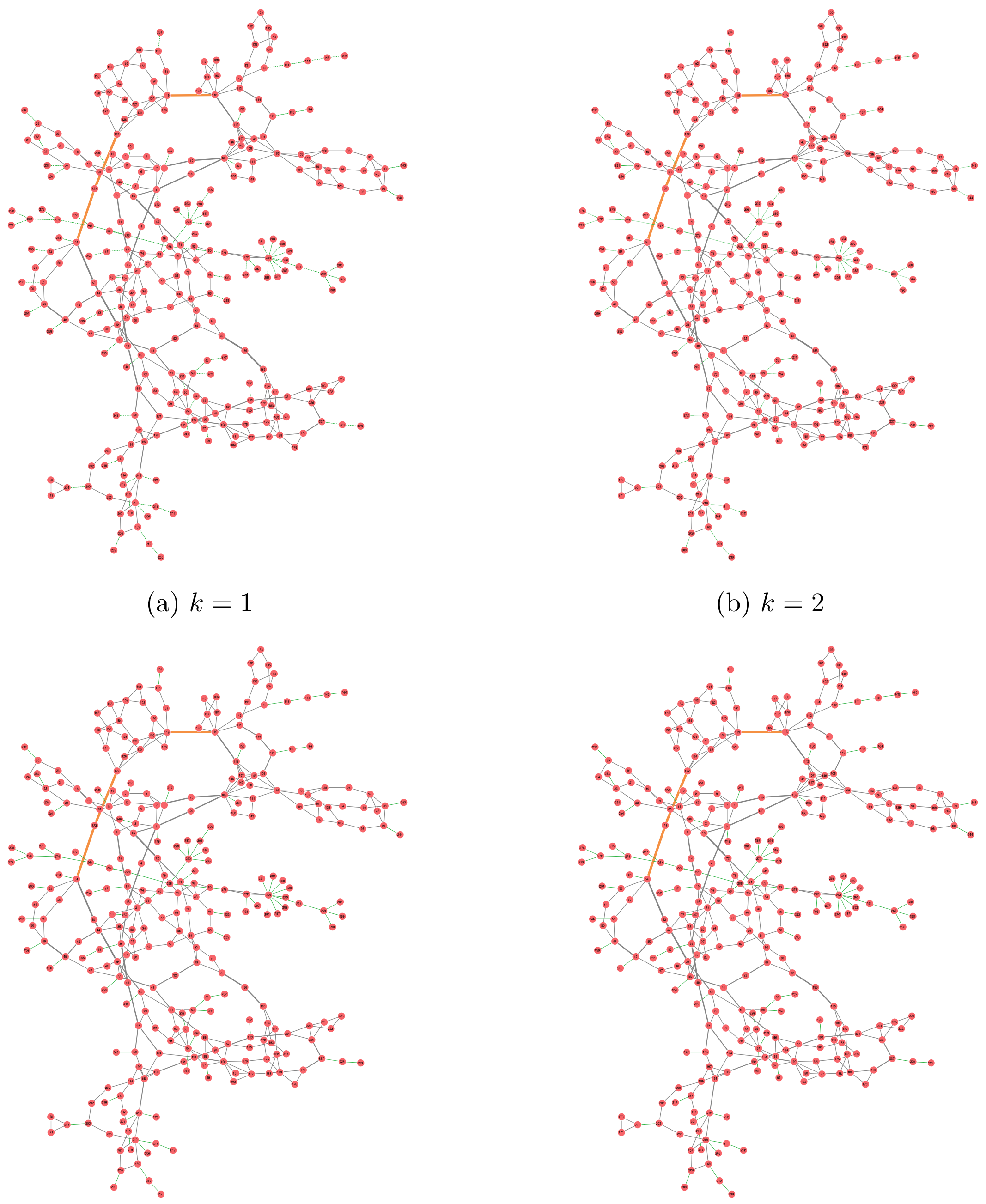
5.3 Learning Approaches Applied to the Test Systems

5.3.1 Architectures and Evaluated Hyperparameters

In this section, the results of the application of the learning approaches proposed in Chapter 4 are shown. A preliminary analysis of hyperparameters was made, and the chosen architectures and hyperparameters are shown in the following.

Every set of hyperparameters had 50 fitted models for each of the approaches, and the average values of the respective classification and regression metrics were considered. For all the tables in this chapter, the results are presented in the form of $AVG \pm STD$, where AVG is the average value among the 50 fitted models and STD is the standard

Figure 32 – Criticality Index and critical edges on the IEEE 300 graph model.



(a) $k = 1$

(b) $k = 2$

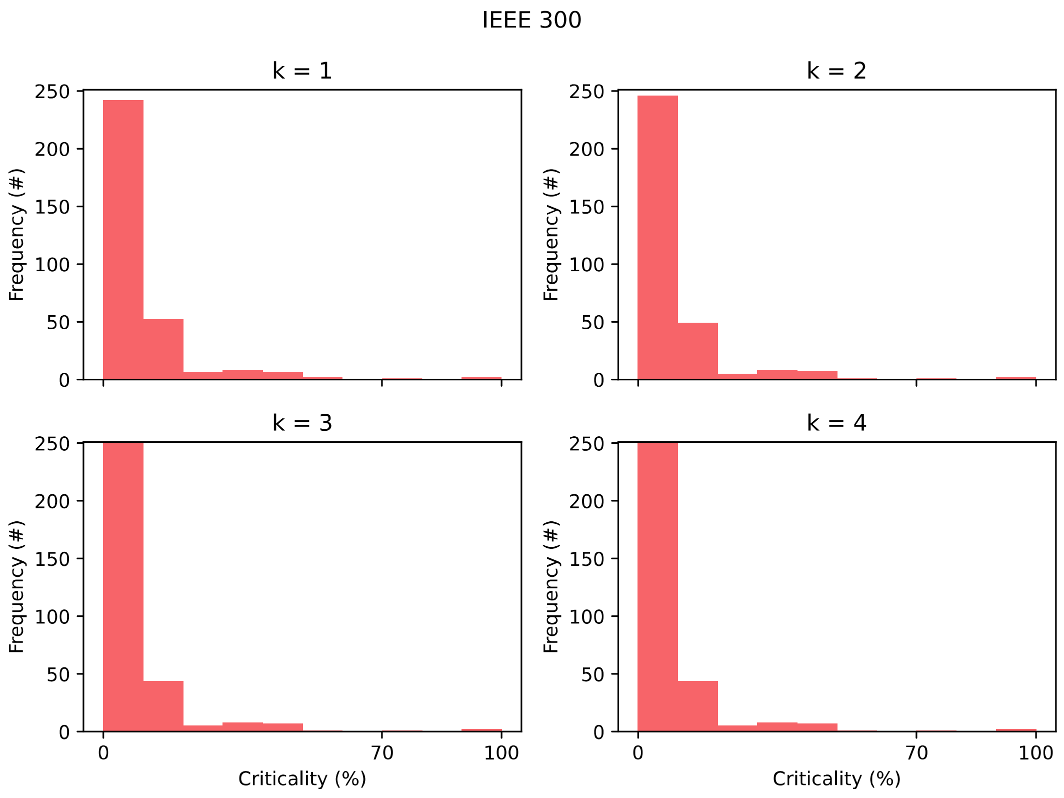
(c) $k = 3$

(d) $k = 4$

Source: the author.

deviation of the respective metric for the 50 samples. The architectures described in the following text are the ones whose hyperparameters provide the best results. Every approach was trained for different training splits, i.e. the ratio of nodes that were used for training. Since the classification tasks are heavily imbalanced, instead of showing the fraction of nodes used in the training, the explicit number of edges belonging to the critical class that were considered for the training set are presented.

Figure 33 – Distribution of the Criticality Index $gi(e)$ on IEEE 300.



Source: the author

As mentioned in Chapter 4, given that *random under sampling* is used to attenuate the effect of the imbalancement, each training set contained the same number of samples of each class. For example, if 2 critical edges are used in the training step, 2 regular edges are randomly chosen to compose the training set. For the approach RLG, since it is a regression task, the fraction of used nodes and the results are shown in terms of the actual training split among all the nodes in the test graphs.

For approach CGE, the considered hyperparameters are shown in Table 2.

For approach CLG, a 3-layer GCN architecture with dropout was chosen. After each GCN convolutional layer, the ReLU activation function was applied, followed by a dropout, except on the last convolution, which gave the resulting embedding. The considered hyperparameters are shown in Table 3.

For approach RLG, the regression architectures consists of a 4-layer GCN with 2 GCN convolutional layers followed by a 2 linear layers. After each convolutional and linear layers the ReLU activation function is applied. The considered hyperparameters are shown in Table 4.

Table 2 – Hyperparameters for the *CGE* approach

Name	Value
node2vec	
Embedding size d	128
Walk length l	10
Context window size k	10
Walks per node r	10
Negative samples	1
Walk bias parameters p and q	1
Learning rate	10^{-2}
Epochs	200
Random Forest	
Number of trees	100
Split quality metric	*gini*
Tree depth limit	∞

Source: the author

Table 3 – Hyperparameters for the *CLG* approach

Name	Value
Num. of GCN layers	3
Activation function	ReLU
Dropout rate	50%
Embedding size d	128
GCN hidden unis	64
Learning rate	10^{-3}
Epochs	200

Source: the author

Table 4 – Hyperparameters for the *RLG* approach

Name	Value
Num. of GCN layers	2
Num. of linear layers	2
Activation function	ReLU
Embedding size d	32
GCN hidden unis	32
Learning rate	10^{-3}
Epochs	100

Source: the author

5.3.2 ITAIPU 11

The ITAIPU 11 test system is previously known to be a small network when compared to most real EPS graph models. Given that GNN applications are mostly done in large graphs as the Cora citation network shown in Chapter 3, it is expected that the learning approaches might have some issues in extracting the relevant information due to the lack of samples.

Table 5 – Classification results for the ITAIPU 11 test system with the *CGE* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	37.9 ± 11.8	$\mathbf{30.8 \pm 8.7}$
2	$\mathbf{46.5 \pm 15.3}$	16.7 ± 22.7
$k = 2$		
1	31.6 ± 14.2	$\mathbf{29.7 \pm 9.8}$
2	$\mathbf{42.1 \pm 15.4}$	15.5 ± 20.5
$k = 3$		
1	46.3 ± 15.1	59.7 ± 10.0
2	$\mathbf{61.5 \pm 18.2}$	55.5 ± 24.7
3	59.7 ± 18.5	$\mathbf{61.2 \pm 16.7}$
$k = 4$		
1	43.8 ± 17.7	52.7 ± 11.3
2	$\mathbf{65.1 \pm 16.0}$	$\mathbf{56.0 \pm 22.0}$

Source: the author

Table 5 shows the results from the application of *CGE* approach to the ITAIPU 11 graph model. As can be seen, the *CGE* approach was able to reasonably extract information for classifying the nodes of the line graph of ITAIPU 11. For $k = 3$ and $k = 4$, where the total number of critical edges raised from 3 to 6 and 5, respectively, the $F1$-score for the critical class was able to hit 61.2 and 56.0 in average, which were considered good results.

The *CGE* approach does not use any topological information for the classification process, since the embedding step is done separately. The *CLG* approach does the embedding and classification in a single step using a GCN architecture. Table 6 shows the results for the critical line classification in ITAIPU 11 test system using the *CLG* approach. It is possible to note that the best average values for every k are significantly greater than respective values in the *CGE* approach.

The *RLG* approach, which consists in a regression task for the criticality index, did not provide good results. Table 7 shows the results for the ITAIPU 11 system only for the 50% training split, which was the greatest evaluated split. The negative R^2 values shows that the GCN regression could not reproduce the trend in the data.

In general, *CGE* and *CLG* approaches were successful in extracting information for the ITAIPU 11 test system. However, one relevant aspect of these results are that the classification task for the ITAIPU 11 is not highly imbalanced, since the graph contained 15 edges, from which only 14 are valid for removal, since one turns the graph into a

Table 6 – Classification results for the ITAIPU 11 test system with the *CLG* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	$\mathbf{63.1 \pm 20.5}$	$\mathbf{47.8 \pm 25.2}$
2	60.3 ± 19.8	45.9 ± 23.4
$k = 2$		
1	$\mathbf{44.5 \pm 15.2}$	$\mathbf{33.2 \pm 13.3}$
2	42.5 ± 15.8	31.7 ± 16.7
$k = 3$		
1	54.2 ± 9.1	$\mathbf{64.0 \pm 7.9}$
2	55.7 ± 8.4	63.5 ± 10.3
3	$\mathbf{56.7 \pm 8.6}$	63.6 ± 10.4
$k = 4$		
1	$\mathbf{55.1 \pm 10.4}$	$\mathbf{58.7 \pm 8.5}$
2	53.5 ± 11.5	57.7 ± 11.3

Source: the author

Table 7 – Regression results for the ITAIPU 11 test system with the *RLG* approach

Train Split	R^2
$k = 1$	
50%	-0.5 ± 1.2
$k = 2$	
50%	-0.8 ± 1.6
$k = 3$	
50%	-1.0 ± 1.6
$k = 4$	
50%	-0.4 ± 0.8

Source: the author

disconnected one. The case with $k = 3$, which reaches the highest $F1$-score for the critical class, has 6 critical edges, which represent around 40% of the total.

5.3.3 IEEE 39

For the IEEE 39 test system, Tables 8 and 9 show the results for the application of *CGE* and *CLG* approaches, respectively.

Both *CGE* and *CLG* approaches continue to produce good results as for the ITAIPU 11. The *CLG* approach obtains better results for the $F1$-score of the critical class, except for $k = 4$. For this graph, it is important to highlight that the classification task is significantly more imbalanced than for the ITAIPU 11 graph. For $k = 1$ and $k = 2$, only 4 edges are considered critical by the Criticality Index, which represents only 9% of the 46 edges of the graph.

Table 8 – Classification results for the IEEE 39 test system with the *CGE* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	24.4 ± 10.4	16.9 ± 4.4
2	$\mathbf{42.8 \pm 11.0}$	$\mathbf{17.7 \pm 9.0}$
$k = 2$		
1	25.4 ± 9.8	$\mathbf{17.2 \pm 4.3}$
2	$\mathbf{43.0 \pm 10.1}$	15.7 ± 9.7
$k = 3$		
1	34.1 ± 10.1	$\mathbf{38.6 \pm 6.6}$
2	49.0 ± 9.0	33.1 ± 12.7
3	$\mathbf{53.7 \pm 10.0}$	33.3 ± 13.8
4	52.8 ± 10.9	35.0 ± 13.6
$k = 4$		
1	28.3 ± 10.0	$\mathbf{26.8 \pm 5.1}$
2	$\mathbf{45.7 \pm 12.3}$	25.5 ± 11.3
3	39.8 ± 11.6	23.6 ± 7.8

Source: the author

Table 9 – Classification results for the IEEE 39 test system with the *CLG* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	40.7 ± 13.4	$\mathbf{24.5 \pm 7.8}$
2	$\mathbf{43.5 \pm 10.7}$	20.8 ± 6.8
$k = 2$		
1	33.6 ± 11.2	$\mathbf{19.3 \pm 6.8}$
2	$\mathbf{37.3 \pm 13.1}$	16.4 ± 8.4
$k = 3$		
1	44.8 ± 9.4	40.5 ± 8.1
2	56.5 ± 11.7	46.3 ± 10.8
3	59.7 ± 8.7	47.1 ± 9.5
4	$\mathbf{60.3 \pm 11.9}$	$\mathbf{48.9 \pm 11.3}$
$k = 4$		
1	33.6 ± 11.2	$\mathbf{19.3 \pm 6.8}$
2	32.2 ± 10.5	18.5 ± 5.9
3	$\mathbf{37.4 \pm 13.6}$	15.7 ± 8.9

Source: the author

The *RLG* approach again fails to produce significant results, as shown in Table 10. Even with the training split of 50%, which means to use 23 edges for training the regression task, the R^2 remains close to 0 with negative average values, meaning that the architecture is not able to learn the trend of the data.

Table 10 – Regression results for the IEEE 39 test system with the *RLG* approach

Train Split	R^2
$k = 1$	
50%	-0.5 ± 0.5
$k = 2$	
50%	-0.6 ± 0.4
$k = 3$	
50%	-0.5 ± 0.4
$k = 4$	
50%	-0.4 ± 0.4

Source: the author

5.3.4 IEEE 57

For the IEEE 57 test system, Tables 8 and 9 show the results for the CGE and CLG approaches, respectively. Both approaches follow each other in terms of the evaluated scores. The CLG approach is able to achieve better results in most cases, such as the best results for $k = 1$ and $k = 2$. The critical $F1$-scores are greater for the CLG approach. Also, the critical $F1$-score achieves reasonably better results for $k = 1$ and $k = 2$, since the number of critical edges for this system, which are 10 for both k values mentioned earlier, drop to 5 and 3 for $k = 3$ and $k = 4$, respectively. For this graph, the imbalancement takes a fundamental role in preventing the achievement of good results in the application of these learning approaches.

The RLG approach still fails to produce significant results, as shown in Table 13. The R^2 also tends to be closer to 0, and the prediction now tends to have no correlation with the true criticality values. These results can be explained by the distribution of the criticality indices in the graph. Since the IEEE 57 is the first evaluated test system on which most values tends to be closer to 0, as shown in Figure 29, the predicted values tend to smaller values, minimizing the absolute and square errors. However, the R^2 allows one to ensure that this does not mean the learning is being done succesfully, and the predicted values are close to random.

5.3.5 IEEE 118

The IEEE 118 test system, as shown previously in this chapter, continues to display the patterns that emerged on the IEEE 57 system for the criticality indices. For the results obtained by the learning approaches, the system also displays the same trend announced in the IEEE 57 system, but the classification approaches CGE and CLG reach their worst values among all the test systems. Since the approaches use *random under sampling* for balancing the training set and there are only 2 critical edges for all values of k, all the models for this test system were trained with only 2 edges in the training set, being 1

Table 11 – Classification results for the IEEE 57 test system with the *CGE* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
	$k = 1$	
1	26.4 ± 9.2	$\mathbf{20.6 \pm 4.3}$
2	43.1 ± 7.8	18.2 ± 8.5
3	38.7 ± 8.8	20.1 ± 5.2
4	$\mathbf{45.6 \pm 7.3}$	16.2 ± 8.0
5	40.4 ± 9.4	16.3 ± 6.7
	$k = 2$	
1	26.2 ± 10.4	$\mathbf{21.7 \pm 3.8}$
2	44.7 ± 9.1	20.9 ± 8.0
3	34.2 ± 9.7	18.9 ± 4.7
4	$\mathbf{46.9 \pm 6.6}$	17.2 ± 8.2
5	42.8 ± 7.3	16.5 ± 6.5
	$k = 3$	
1	20.9 ± 10.5	$\mathbf{10.3 \pm 2.9}$
2	$\mathbf{40.5 \pm 9.8}$	9.8 ± 6.3
3	39.4 ± 8.1	8.3 ± 5.9
	$k = 4$	
1	19.3 ± 9.7	$\mathbf{6.1 \pm 1.5}$
2	$\mathbf{39.1 \pm 8.7}$	5.2 ± 4.8

Source: the author

regular edge and 1 critical edge. As can be seen in Tables 14 and 15, the $F1$-scores display the lack of learning for the models with such reduced training sets.

For the *RLG* approach, the errors also follow the same trend from the results for the IEEE 57 test system. Since the Criticality Index distribution from Figure 31 is even more unbalanced than the IEEE 57 case, the results also tend to smaller errors but with no correlation with the truth values.

5.3.6 IEEE 300

The results for the IEEE 300 test system break the observed trend from the previous IEEE test systems when taken into account the distribution of the Criticality Index, as in Figure 33. Since this system has 3 critical edges, the training is made with 1 or 2 critical edges in training. The $F1$-scores continue to display values close to 0. Actually, when the training is done with 1 critical edge and the 2 other critical edges remain for the testing, the $F1$ values are greater than in the other case, when only 1 critical edge is present for training.

This fact highlights another issue with the learning done with the decribed approaches for the given Criticality Index. The heavy imbalancement in the test set may also distort the obtained results when there are very low numbers of test samples for the

Table 12 – Classification results for the IEEE 57 test system with the *CLG* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	51.6 ± 20.2	$\mathbf{32.5 \pm 15.1}$
2	$\mathbf{51.7 \pm 19.4}$	31.8 ± 14.3
3	51.3 ± 15.7	28.4 ± 12.5
4	51.0 ± 18.2	30.9 ± 13.8
5	50.8 ± 19.5	32.1 ± 14.8
$k = 2$		
1	$\mathbf{51.0 \pm 18.7}$	32.1 ± 13.0
2	50.7 ± 20.2	32.3 ± 14.4
3	50.9 ± 19.9	$\mathbf{33.2 \pm 13.9}$
4	50.6 ± 19.4	32.4 ± 12.6
5	49.8 ± 20.9	30.4 ± 14.2
$k = 3$		
1	37.4 ± 17.1	13.2 ± 9.5
2	38.1 ± 17.7	$\mathbf{14.2 \pm 8.8}$
3	$\mathbf{39.5 \pm 15.3}$	13.1 ± 8.9
$k = 4$		
1	$\mathbf{33.0 \pm 17.0}$	$\mathbf{10.6 \pm 5.7}$
2	28.2 ± 15.2	8.8 ± 4.8

Source: the author

Table 13 – Regression results for the IEEE 57 test system with the *RLG* approach

Train Split	R^2
$k = 1$	
50%	-0.2 ± 0.2
$k = 2$	
50%	-0.2 ± 0.2
$k = 3$	
50%	-0.2 ± 0.3
$k = 4$	
50%	-0.2 ± 0.2

Source: the author

critical class, as is the case for the IEEE 118 and IEEE 300 test systems.

The *RLG* approach in the IEEE 300 system also demonstrates low performance, confirming that the regression task on the line graph is not a good approach for learning the critical and regular edges in the evaluated test systems.

Table 14 – Classification results for the IEEE 118 test system with the *CGE* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	14.3 ± 10.8	1.2 ± 0.6
$k = 2$		
1	14.1 ± 10.3	1.1 ± 0.6
$k = 3$		
1	15.6 ± 9.6	1.2 ± 0.8
$k = 4$		
1	13.9 ± 10.5	1.1 ± 0.6

Source: the author

Table 15 – Classification results for the IEEE 118 test system with the *CLG* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	26.3 ± 13.8	1.0 ± 0.9
$k = 2$		
1	28.6 ± 13.9	0.7 ± 0.9
$k = 3$		
1	25.4 ± 14.9	0.8 ± 0.9
$k = 4$		
1	28.6 ± 14.4	0.8 ± 1.0

Source: the author

Table 16 – Regression results for the IEEE 118 test system with the *RLG* approach

Train Split	R^2
$k = 1$	
50%	-0.2 ± 0.3
$k = 2$	
50%	-0.1 ± 0.1
$k = 3$	
50%	-0.1 ± 0.1
$k = 4$	
50%	-0.1 ± 0.1

Source: the author

5.4 Summary

In this chapter, the results for the Criticality Index applied to the test systems and the performance of the learning approaches were presented.

First, the test systems used for evaluating the methodology and analyzing the results are also presented, highlighting the graph models that are obtained from their line

Table 17 – Classification results for the IEEE 300 test system with the *CGE* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	18.2 ± 12.3	$\mathbf{1.6 \pm 0.7}$
2	$\mathbf{36.1 \pm 10.1}$	0.6 ± 0.8
$k = 2$		
1	18.8 ± 13.4	$\mathbf{1.4 \pm 0.6}$
2	$\mathbf{36.7 \pm 9.5}$	1.0 ± 1.1
$k = 3$		
1	16.2 ± 13.0	$\mathbf{1.4 \pm 0.6}$
2	$\mathbf{36.1 \pm 11.5}$	0.9 ± 1.2
$k = 4$		
1	16.2 ± 13.0	$\mathbf{1.4 \pm 0.6}$
2	$\mathbf{36.1 \pm 11.5}$	0.9 ± 1.2

Source: the author

Table 18 – Classification results for the IEEE 300 test system with the *CLG* approach

# Critical Edges in Train	Avg. F1-score (%)	Critical F1-score (%)
$k = 1$		
1	29.4 ± 18.6	$\mathbf{3.9 \pm 5.5}$
2	$\mathbf{31.9 \pm 15.6}$	1.0 ± 1.1
$k = 2$		
1	26.6 ± 18.4	$\mathbf{3.2 \pm 3.9}$
2	$\mathbf{28.1 \pm 15.5}$	0.8 ± 1.1
$k = 3$		
1	28.3 ± 17.3	$\mathbf{2.9 \pm 3.4}$
2	$\mathbf{31.2 \pm 13.9}$	0.9 ± 0.8
$k = 4$		
1	28.3 ± 17.3	$\mathbf{2.9 \pm 3.4}$
2	$\mathbf{31.2 \pm 13.9}$	0.9 ± 0.8

Source: the author

diagrams.

For the ITAIPU 11 test system, the results of evaluating the Criticality Index on it are significantly different than the results obtained on the other test systems, given the small size with respect to the number of buses and lines. In the other way, the classification task obtained better results, since the critical and regular classes are not imbalanced as in the other test systems.

In the first part, the Criticality Index has shown to be more imbalanced as the graph size grows. In other words, larger graphs tend to have less critical edges. This has proven to be a challenge for the GML approaches, since the classification tasks have more difficulties for learning with imbalanced classes. The regression approach *RLG* was shown

Table 19 – Regression results for the IEEE 300 test system with the *RLG* approach

Train Split	R^2
$k = 1$	
50%	-0.0 ± 0.1
$k = 2$	
50%	-0.0 ± 0.1
$k = 3$	
50%	-0.0 ± 0.1
$k = 4$	
50%	-0.0 ± 0.1

Source: the author

to be inneficient for learning any relevant information for predicting the Criticality Index in the evaluated test systems. In the next chapter the conclusions and proposed further developments are exposed.

6 Conclusion and Future Developments

6.1 Considerations

Ensuring security and reliability is challenging for the electrical power systems operations, and the contingency screening methods play an important role for accomplishing this task. With the constant increase in the complexity and scale of most power systems, alternative approaches for evaluating the most critical contingency scenarios among all the possible events have been considered, such as topological analysis with graph models.

Even when the system model is simplified by the use of simple graph models, the number of possible event combinations turns to be infeasible for an exhaustive analysis in multiple contingencies. Therefore, alternative approaches for approximating these contingency indices must be developed. In this dissertation, this approximationg through Graph Machine Learning was evaluated.

The application of GML for solving the contingency screening problem in EPS through graph models has demonstrated to be challenging. Considering the chosen Criticality Index, the imbalancement between the critical and regular lines has proven to be a challenge for the learning techniques, and the results for large graphs with few critical edges were shown to be poor.

The regression learning approach *RLG* - GNN Regression on the Line Graph - was shown to be specialy inneficient for this task, and there was no success in obtaining a good learning methodology for this approach on the given test systems, with the chosen Criticality Index.

However, the classification approaches *CGE* - Classification on Generated Embeddings - and *CLG* - GNN Node Classification on the Line Graph - have shown good results for the evaluated scores in smaller systems, where the classes are not as imbalanced as in larger graphs. The obtained results are considered acceptable for a first work in the edge classification problem in the context of EPS graph models, specially since there are not specific architectures for dealing with edges as the main goal of the learning task.

The *CLG* approach demonstrated that, even when the processed graph is the line graph of the original model, the capabilities of GNNs to use topological information together with the learning algorithms has proven to frequently provide better results than the *CGE* approach, where the embedding is done separately.

Finnaly, the application of GML techniques for the contingency screening can be further improved by addressing the mentioned issues such as the class imbalancement.

6.2 Future Works

In future works, one might address some issues that were observed in the results for the evaluated approaches. First, the high variance of each score might be due to the random initialization of the node embeddings. There might be some reasonable methodology for initializing the node embeddings before applying the GCN architecture, in order to make the results more predictable. Also, the learning approaches can be applied to different criticality indices that the one used for this dissertation. Different initial labelings might lead to better classification results, which can be even more evident if the classes become more balanced.

As the evaluated approach through GML is based on learning patters from data, one might consider learning from a set of multiple graphs instead of training and testing in a single graph. Through this, the number of edges in each class can become more relevant, overcoming some issues that were observed with 2 or 3 edges belonging to the critical class.

Another possible approach is developing an data augmentation approach for using in the learning tasks. This could contribute for balancing the classes, possibly boosting the classification scores.

One specially interesting kind of methodology to be researched and evaluated is the complex networks models for generating EPS-kind of graphs. If such models were known, the training data for the learning approaches could be synthetic, generated with desired degree distributions that matches the power system graph models.

At last, more modern GNN architectures can be considered for the task, such as the GAT or the GIN. However, these layers turn the training process slower when compared to the GCN, as they contain more parameters and the GIN has an entire MLP network in the encoding step. The hyperparameter and data split analysis made in this work might be difficult to be reproduced with slower training GNN architectures. The learning can be even improved if a network generation model for EPS graphs is known, such as the BA model for social networks, since the training could be done in multiple graphs with the same characteristics.

Bibliography

ADAMIC, L. A.; ADAR, E. Friends and neighbors on the web. *Social Networks*, v. 25, n. 3, p. 211–230, 2003. ISSN 0378-8733. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378873303000091>. Cited on page 52.

AGRAWAL, M.; ZITNIK, M.; LESKOVEC, J. Large-scale analysis of disease pathways in the human interactome. In: . [S.l.: s.n.], 2018. p. 111–122. Cited on page 49.

ANDRADE, L. de; LEAO, T. P. de. A brief history of direct current in electrical power systems. *2012 Third IEEE HISTory of ELectro-technology CONference (HISTELCON)*, p. 1–6, 2012. Cited on page 21.

ATHAY, T.; PODMORE, R.; VIRMANI, S. A practical method for the direct analysis of transient stability. *IEEE Transactions on Power Apparatus and Systems*, PAS-98, n. 2, p. 573–584, 1979. Cited on page 72.

BALAKRISHNAN, R.; RANGANATHAN, K. *A Textbook of Graph Theory*. [S.l.]: Springer New York, 2012. (Universitext). ISBN 9781461445296. Cited 2 times on pages 27 and 35.

BALU, N. et al. On-line power system security analysis. *Proceedings of the IEEE*, v. 80, n. 2, p. 262–282, 1992. Cited on page 23.

BARABÁSI, A. *Network Science*. Cambridge University Press, 2016. ISBN 9781107076266. Disponível em: <https://books.google.com.br/books?id=ZVHesgEACAAJ>. Cited 3 times on pages 41, 42, and 44.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *Science*, v. 286, n. 5439, p. 509–512, 1999. Disponível em: <https://www.science.org/doi/abs/10.1126/science.286.5439.509>. Cited on page 43.

BAUKE, H. et al. Topological phase transition in a network model with preferential attachment and node removal. *The European Physical Journal B: Condensed Matter and Complex Systems*, v. 83, n. 4, p. 519–524, October 2011. Disponível em: <https://ideas.repec.org/a/spr/eurphb/v83y2011i4p519-524.html>. Cited on page 45.

BELKIN, M.; NIYOGI, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: DIETTERICH, T.; BECKER, S.; GHAHRAMANI, Z. (Ed.). *Advances in Neural Information Processing Systems*. MIT Press, 2002. v. 14. Disponível em: <https://proceedings.neurips.cc/paper/2001/file/f106b7f99d2cb30c3db1c3cc0fde9ccb-Paper.pdf>. Cited on page 51.

BENJAMIN, A.; CHARTRAND, G.; ZHANG, P. *The Fascinating World of Graph Theory*. Princeton University Press, 2017. ISBN 9780691175638. Disponível em: <https://books.google.com.br/books?id=Y3GYDwAAQBAJ>. Cited on page 27.

BOJCHEVSKI, A.; GÜNNEMANN, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In: *International Conference on Learning Representations*. [s.n.], 2018. Disponível em: <https://openreview.net/forum?id=r1ZdKJ-0W>. Cited on page 57.

BOMPARD, E.; PONS, E.; WU, D. Extended topological metrics for the analysis of power grid vulnerability. *IEEE Systems Journal*, v. 6, n. 3, p. 481–487, 2012. Cited on page 44.

BOMPARD, E.; WU, D.; XUE, F. Structural vulnerability of power systems: A topological approach. *Electric Power Systems Research*, v. 81, n. 7, p. 1334–1340, 2011. ISSN 0378-7796. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378779611000332>. Cited on page 25.

BORDES, A. et al. Translating embeddings for modeling multi-relational data. In: BURGES, C. J. C. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2013. v. 26. Disponível em: <https://proceedings.neurips.cc/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf>. Cited on page 48.

BRANDES, U. On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, v. 30, n. 2, p. 136–145, 2008. ISSN 0378-8733. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378873307000731>. Cited on page 38.

BRANDES, U.; FLEISCHER, D. Centrality measures based on current flow. In: DIEKERT, V.; DURAND, B. (Ed.). *STACS 2005*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 533–544. ISBN 978-3-540-31856-9. Cited on page 38.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Cited on page 67.

BRUNA, J. et al. Spectral networks and locally connected networks on graphs. In: *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*. [S.l.: s.n.], 2014. Cited on page 56.

BULAT, H.; FRANKOVIć, D.; VLAHINIć, S. Enhanced contingency analysis—a power system operator tool. *Energies*, v. 14, n. 4, 2021. ISSN 1996-1073. Disponível em: <https://www.mdpi.com/1996-1073/14/4/923>. Cited on page 23.

CAPOROSSI, G. et al. Centrality and betweenness: vertex and edge decomposition of the wiener index. *Les Cahiers du GERAD ISSN*, v. 711, p. 2440, 2011. Cited on page 37.

CETINAY, H.; KUIPERS, F. A.; MIEGHEM, P. V. A topological investigation of power flow. *IEEE Systems Journal*, v. 12, n. 3, p. 2524–2532, 2016. Cited on page 25.

CHRISTIE, R. *Power Systems Test Case Archive - 118 Bus Power Flow Test Case*. 1993. Disponível em: <http://labs.ece.uw.edu/pstca/pf118/pg_tca118bus.htm>. Cited on page 73.

CHRISTIE, R. *Power Systems Test Case Archive - 300 Bus Power Flow Test Case*. 1993. Disponível em: <http://labs.ece.uw.edu/pstca/pf300/pg_tca300bus.htm>. Cited on page 74.

CHRISTIE, R. *Power Systems Test Case Archive - 57 Bus Power Flow Test Case*. 1993. Disponível em: <http://labs.ece.uw.edu/pstca/pf57/pg_tca57bus.htm>. Cited on page 73.

CHU, C.-C.; IU, H. H.-C. Complex networks theory for modern smart grid applications: A survey. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, v. 7, n. 2, p. 177–191, 2017. Cited on page 44.

COELHO, E. P. et al. Topological multi-contingency screening based on current flow betweenness. *Electric Power Systems Research*, v. 203, p. 107609, 2022. ISSN 0378-7796. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378779621005903>. Cited on page 25.

COELHO, E. P. R. *Análise de Vulnerabilidade em Smart Grid Utilizando Métricas de Centralidade em Grafos*. Tese (Doutorado) — Federal University of Espirito Santo, Brazil, 2 2019. Cited 7 times on pages 25, 26, 61, 63, 64, 73, and 81.

COELHO, E. P. R. et al. A new approach for contingency analysis based on centrality measures. *IEEE Systems Journal*, v. 13, n. 2, p. 1915–1923, 2019. Cited 2 times on pages 25 and 41.

DORAISWAMI, R.; CARVALHO, M. Reliability indices for a power network considering static, transient, and dynamic performance. *IEEE Transactions on Reliability*, R-28, n. 2, p. 120–123, 1979. Cited on page 23.

DWIVEDI, A.; YU, X. A maximum-flow-based complex network approach for power system vulnerability analysis. *IEEE Transactions on Industrial Informatics*, v. 9, n. 1, p. 81–88, 2013. Cited on page 44.

EJEBE, G. et al. Methods for contingency screening and ranking for voltage stability analysis of power systems. *IEEE Transactions on Power Systems*, v. 11, n. 1, p. 350–356, 1996. Cited on page 24.

EJEBE, G. C.; WOLLENBERG, B. F. Automatic contingency selection. *IEEE Transactions on Power Apparatus and Systems*, PAS-98, n. 1, p. 97–109, 1979. Cited 3 times on pages 23, 71, and 72.

ERDÖS, P.; RÉNYI, A. On random graphs i. *Publicationes Mathematicae Debrecen*, v. 6, p. 290, 1959. Cited on page 42.

EULER, L. Solutio problematis ad geometriam situs pertinentis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, v. 8, p. 128–140, 1736. Cited 2 times on pages 27 and 28.

FREEMAN, L. C. A set of measures of centrality based on betweenness. *Sociometry*, [American Sociological Association, Sage Publications, Inc.], v. 40, n. 1, p. 35–41, 1977. ISSN 00380431. Disponível em: <http://www.jstor.org/stable/3033543>. Cited on page 37.

FREEMAN, L. C. Centrality in social networks conceptual clarification. *Social Networks*, v. 1, n. 3, p. 215–239, 1978. ISSN 0378-8733. Disponível em: <https://www.sciencedirect.com/science/article/pii/0378873378900217>. Cited on page 37.

G1. *Apagao no Amapa: veja a cronologia da crise de energia elétrica.* 2020. Disponível em: <https://g1.globo.com/ap/amapa/noticia/2020/11/18/apagao-no-amapa-veja-a-cronologia-da-crise-de-energia-eletrica.ghtml>. Cited on page 21.

GILBERT, E. N. Random Graphs. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 30, n. 4, p. 1141 – 1144, 1959. Disponível em: <https://doi.org/10.1214/aoms/1177706098>. Cited on page 42.

GILMER, J. et al. Neural message passing for quantum chemistry. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. [S.l.]: JMLR.org, 2017. (ICML'17), p. 1263–1272. Cited on page 56.

GORTON, I. et al. A high-performance hybrid computing approach to massive contingency analysis in the power grid. In: *2009 Fifth IEEE International Conference on e-Science*. [S.l.: s.n.], 2009. p. 277–283. Cited on page 25.

GROVER, A.; LESKOVEC, J. Node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 855–864. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939754>. Cited on page 54.

GUY, S.; MARVIN, S. Electricity in the marketplace: Reconfiguring the consumption of essential resources. *Local Environment*, Routledge, v. 3, n. 3, p. 313–331, 1998. Cited on page 21.

HAMILTON, W. *Graph Representation Learning*. Morgan & Claypool Publishers, 2020. (Synthesis Lectures on Artificial Intelligence and Machine Learning). ISBN 9781681739649. Disponível em: <https://books.google.com.br/books?id=Csj-DwAAQBAJ>. Cited 3 times on pages 25, 47, and 56.

HAMILTON, W. L.; YING, R.; LESKOVEC, J. Inductive representation learning on large graphs. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 1025–1035. ISBN 9781510860964. Cited on page 47.

HAMILTON, W. L.; YING, R.; LESKOVEC, J. Representation learning on graphs: Methods and applications. *CoRR*, abs/1709.05584, 2017. Disponível em: <http://arxiv.org/abs/1709.05584>. Cited on page 49.

HOFF, P. D.; RAFTERY, A. E.; HANDCOCK, M. S. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, Taylor & Francis, v. 97, n. 460, p. 1090–1098, 2002. Disponível em: <https://doi.org/10.1198/016214502388618906>. Cited on page 49.

JIN, S. et al. A novel application of parallel betweenness centrality to power grid contingency analysis. In: *2010 IEEE International Symposium on Parallel Distributed Processing (IPDPS)*. [S.l.: s.n.], 2010. p. 1–7. Cited on page 25.

KATZ, L. A new status index derived from sociometric analysis. *Psychometrika*, v. 18, n. 1, p. 39–43, mar. 1953. ISSN 1860-0980. Disponível em: <https://doi.org/10.1007/BF02289026>. Cited on page 52.

KHANABADI, M.; GHASEMI, H.; DOOSTIZADEH, M. Optimal transmission switching considering voltage security and n-1 contingency analysis. *IEEE Transactions on Power Systems*, v. 28, n. 1, p. 542–550, 2013. Cited on page 23.

KIPF, T. N.; WELLING, M. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv e-prints*, p. arXiv:1609.02907, set. 2016. Cited 2 times on pages 47 and 59.

LESKOVEC, J.; KLEINBERG, J.; FALOUTSOS, C. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, Association for Computing Machinery, New York, NY, USA, v. 1, n. 1, p. 2–es, mar 2007. ISSN 1556-4681. Disponível em: <https://doi.org/10.1145/1217299.1217301>. Cited on page 43.

LU, L.; ZHOU, T. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, v. 390, n. 6, p. 1150–1170, 2011. ISSN 0378-4371. Disponível em: <https://www.sciencedirect.com/science/article/pii/S037843711000991X>. Cited on page 48.

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. *Journal of machine learning research*, v. 9, n. 11, 2008. Cited on page 59.

MAJIDI-QADIKOLAI, M.; BALDICK, R. Integration of n - 1 contingency analysis with systematic transmission capacity expansion planning: Ercot case study. *IEEE Transactions on Power Systems*, v. 31, n. 3, p. 2234–2245, 2016. Cited on page 23.

MAJIDI-QADIKOLAI, M.; BALDICK, R. Stochastic transmission capacity expansion planning with special scenario selection for integrating n - 1 contingency analysis. *IEEE Transactions on Power Systems*, v. 31, n. 6, p. 4901–4912, 2016. Cited on page 23.

MCCALLUM, A. K. et al. Automating the construction of internet portals with machine learning. *Information Retrieval*, Springer, v. 3, n. 2, p. 127–163, 2000. Cited on page 58.

MCPHERSON, M.; SMITH-LOVIN, L.; COOK, J. M. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, v. 27, n. 1, p. 415–444, 2001. Disponível em: <https://doi.org/10.1146/annurev.soc.27.1.415>. Cited on page 48.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Red Hook, NY, USA: Curran Associates Inc., 2013. (NIPS'13), p. 3111–3119. Cited on page 54.

MNIH, A.; HINTON, G. E. A scalable hierarchical distributed language model. In: KOLLER, D. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2009. v. 21. Disponível em: <https://proceedings.neurips.cc/paper/2008/file/1e056d2b0ebd5c878c550da6ac5d3724-Paper.pdf>. Cited on page 54.

NASIRUZZAMAN, A.; POTA, H. Bus dependency matrix of electrical power systems. *International Journal of Electrical Power & Energy Systems*, v. 56, p. 33–41, 2014. ISSN 0142-0615. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0142061513004481>. Cited on page 44.

NEWMAN, M. J. A measure of betweenness centrality based on random walks. *Social Networks*, v. 27, n. 1, p. 39–54, 2005. ISSN 0378-8733. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378873304000681>. Cited on page 38.

NICKEL, M. et al. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, Institute of Electrical and Electronics Engineers (IEEE), v. 104, n. 1, p. 11–33, Jan 2016. ISSN 1558-2256. Disponível em: <http://dx.doi.org/10.1109/JPROC.2015.2483592>. Cited on page 47.

ONS. *Análise da perturbação do dia 03/11/2020 às 20h48min com início nos transformadores de 230/69/13,8 kV da SE Macapá, com desligamento da UHE Coaracy Nunes e do Sistema Amapá.* [S.l.], 2020. Disponível em: <http://www.ons.org.br/AcervoDigitalDocumentosEPublicacoes/DGL-REL-0016_2020%20-%20RAP%2003.11.2020_20h48min_Amap%C3%A1_VF.pdf>. Cited on page 21.

ONS. *Relatório de Análise Estatística de Desligamentos Forçados Referente ao Ano de 2020.* [S.l.], 2021. Cited on page 22.

ONS. *Portal de Dados Abertos.* 2022. Disponível em: <https://dados.ons.org.br/dataset/linha-transmissao>. Cited on page 44.

OU, M. et al. Asymmetric transitivity preserving graph embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 1105–1114. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939751>. Cited on page 52.

PAGANI, G. A.; AIELLO, M. The power grid as a complex network: A survey. *Physica A: Statistical Mechanics and its Applications*, v. 392, n. 11, p. 2688–2700, 2013. ISSN 0378-4371. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378437113000575>. Cited on page 44.

PANDIT, S. et al. Netprobe: A fast and scalable system for fraud detection in online auction networks. In: *Proceedings of the 16th International Conference on World Wide Web.* New York, NY, USA: Association for Computing Machinery, 2007. (WWW '07), p. 201–210. ISBN 9781595936547. Disponível em: <https://doi.org/10.1145/1242572.1242600>. Cited on page 48.

PANG, C.; KEZUNOVIC, M. Static security analysis based on weighted vulnerability index. In: *2011 IEEE Power and Energy Society General Meeting.* [S.l.: s.n.], 2011. p. 1–6. Cited on page 44.

PAPAILIOU, K. *Springer Handbook of Power Systems.* [S.l.]: Springer Singapore, 2021. (Springer Handbooks). ISBN 9789813299375. Cited on page 21.

PEROZZI, B.; AL-RFOU, R.; SKIENA, S. Deepwalk: Online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA: Association for Computing Machinery, 2014. (KDD '14), p. 701–710. ISBN 9781450329569. Disponível em: <https://doi.org/10.1145/2623330.2623732>. Cited on page 54.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group, v. 323, n. 6088, p. 533–536, 1986. Cited on page 57.

SCHLICHTKRULL, M. et al. Modeling relational data with graph convolutional networks. In: SPRINGER. *European semantic web conference.* [S.l.], 2018. p. 593–607. Cited 2 times on pages 57 and 60.

TERU, K. K.; HAMILTON, W. L. Inductive relation prediction on knowledge graphs.

CoRR, abs/1911.06962, 2019. Disponível em: <http://arxiv.org/abs/1911.06962>. Cited on page 48.

VELIČKOVIĆ, P. et al. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. Cited on page 60.

VIECELI, L. *Piora da crise hídrica impacta planos de empresas e ameaça economia até 2022*. 2021. Disponível em: <https://www1.folha.uol.com.br/mercado/2021/09/piora-da-crise-hidrica-impacta-planos-de-empresas-e-ameaca-economia-ate-2022.shtml>. Cited on page 21.

WANG, Z.; SCAGLIONE, A.; THOMAS, R. J. Electrical centrality measures for electric power grid vulnerability analysis. In: *49th IEEE Conference on Decision and Control (CDC)*. [S.l.: s.n.], 2010. p. 5792–5797. Cited on page 25.

WOOD, A.; WOLLENBERG, B.; SHEBLÉ, G. *Power Generation, Operation, and Control*. Wiley, 2013. ISBN 9781118733912. Disponível em: <https://books.google.com.br/books?id=JDVmAgAAQBAJ>. Cited 2 times on pages 23 and 24.

XU, K. et al. How powerful are graph neural networks? In: *International Conference on Learning Representations*. [s.n.], 2018. Disponível em: <https://openreview.net/forum?id=ryGs6iA5Km>. Cited on page 59.

YAN, J.; HE, H.; SUN, Y. Integrated security analysis on cascading failure in complex networks. *IEEE Transactions on Information Forensics and Security*, v. 9, n. 3, p. 451–463, 2014. Cited on page 25.

YANG, Y.; GUAN, X.; ZHAI, Q. Fast grid security assessment with n - k contingencies. *IEEE Transactions on Power Systems*, v. 32, n. 3, p. 2193–2203, 2017. Cited on page 24.

YANG, Z.; COHEN, W.; SALAKHUDINOV, R. Revisiting semi-supervised learning with graph embeddings. In: PMLR. *International conference on machine learning*. [S.l.], 2016. p. 40–48. Cited on page 58.

YING, R. et al. Graph convolutional neural networks for web-scale recommender systems. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: Association for Computing Machinery, 2018. (KDD '18), p. 974–983. ISBN 9781450355520. Disponível em: <https://doi.org/10.1145/3219819.3219890>. Cited on page 48.

ZACHARY, W. W. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, v. 33, n. 4, p. 452–473, 1977. Disponível em: <https://doi.org/10.1086/jar.33.4.3629752>. Cited on page 36.

ZOHURI, B.; MCDANIEL, P. The electricity: An essential necessity in our life. In: *Advanced Smaller Modular Reactors: An Innovative Approach to Nuclear Power*. [S.l.]: Springer International Publishing, 2019. p. 1–21. ISBN 978-3-030-23682-3. Cited on page 21.