

Graph Machine Learning Edge Classification Approaches in Unbalanced Datasets: Transmission Line Contingency Screening in Electrical Power Systems

Rogério Jose Menezes Alves

Graduate Program on Electrical Engineering, Universidade Federal do Espírito Santo, ES
Av. Fernando Ferrari, 514 - Goiabeiras, Vitória, ES
rogerio.alves@edu.ufes.br

Elizandra Pereira Roque Coelho, Marcia Helena Moreira Paiva

Electrical Engineering Department, Universidade Federal do Espírito Santo, ES
Av. Fernando Ferrari, 514 - Goiabeiras, Vitória, ES
{elizandra.roque, marcia.paiva}@ufes.br

RESUMO

A fim de lidar com dados relacionais, as técnicas clássicas de *machine learning* estão sendo melhoradas para considerar estruturas de dados de grafos, o que compreende o campo de *graph machine learning*. Muitas redes podem ser modeladas utilizando estruturas de grafos, tais como redes sociais, redes de comunicação e redes de sistemas de energia. Quando se trata da avaliação de segurança de redes de sistemas de energia, a etapa de identificação dos cenários de contingência mais severos é chamada de *contingency screening*, que pode ser enquadrada como um problema de edição de grafos no caso de falhas de equipamentos. Neste trabalho, é avaliada a utilização de abordagens de *graph machine learning* para resolver o problema de *contingency screening* de linhas de transmissão. O problema é formulado como uma tarefa de classificação binária nas arestas de um grafo e são propostas e comparadas duas abordagens de aprendizagem.

PALAVRAS CHAVE. Sistemas Elétricos de Potência. Graph Machine Learning. Graph Neural Networks.

EN&PG, IC

ABSTRACT

In order to deal with relational data, classical machine learning techniques are being improved to consider graph data structures, which comprises the graph machine learning field. Many networks can be modeled using graph structures, such as social networks, communication networks and power system networks. When dealing with security assessment of power system networks, the identification of the most severe contingency scenarios is called the contingency screening step, which can be frame as a graph editing problem in the case of equipment failures. In this work, the use of graph machine learning approaches for solving the transmission line contingency problem is evaluated. The problem is formulated as a binary classification task on the edges of a graph and two learning approaches are proposed and compared.

KEYWORDS. Electrical Power Systems. Graph Machine Learning. Graph Neural Networks.

EN&PG, IC

1. Introduction

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].

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].

The most common problem in graph machine learning is the node classification. As in a general classification task, each element in a set is given a label and using only the labels of a subset of elements, called the *training set*, one must train a learning algorithm to predict the labels of the other elements. In the node classification task, the elements are nodes (or vertices) of a 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 e Welling, 2016] and for the role of proteins in an interactome [Hamilton et al., 2017]. 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. In this work, the node classification task is applied in the contingency screening step for the security assessment of Electric Power Systems (EPS).

In the context of EPS, 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 proper maintenance 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. In order to have the most secure EPS as possible, one might suggest to develop the system to withstand all possible multiple contingencies, but that may be economically unfeasible. Therefore, one must seek for the balance between desired security level and practical cost-effectiveness. 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 et al., 2013]. The initial step in the contingency analysis process is to extract a set of contingencies that can be more severe than the others. These contingencies are said to have greater criticality and this task is called the *contingency screening* [Ejebe et al., 1996]. Once the most critical contingencies are selected, appropriate contingency plans are designed for each case.

In this work, the viability of formulating the contingency screening problem through a classification task is analyzed. First, a graph model is obtained for the EPS being evaluated. Then, the criticality index from Coelho et al. [2019] is evaluated for each of the transmission lines, which are modeled as the edges of the graph. The output of the criticality index $GI(e)$ is used for determining the label of edge e in the classification task. Then, two approaches for edge classification are proposed.

The first, named *CGE* (Classification on Generated Embeddings), uses a combination of a random walk node embedding method called *node2vec* [Grover e Leskovec, 2016] and a SVM (Support Vector Machine) classifier for classifying the critical and non-critical transmission lines (edges) in the graph model. The second, named *CLG* (GNN Node Classification in the Line Graph), uses a graph neural network architecture that does the node embedding task together with the node classification, called *Graph Convolutional Network* [Kipf e Welling, 2016], but applied to the line graph of the graph model, which maps each node to a transmission line in the electric power system. The proposed approaches are evaluated in four IEEE electric power system graph models.

The remainder of the paper is organized as follows. Section 2 contains a literature review for the contingency screening problem and machine learning techniques applied to relational data. Section 3 describes detailed aspects of the proposed methodology. Section 4 contains the results obtained in the evaluations of the proposed methodology. Finally, the conclusions and an enumeration of future works are shown in Section 5.

2. Literature Review

2.1. Contingency Screening

The first proposed approaches for screening and ranking the contingencies in terms of their criticality were done through a series of power flow simulations, with every contingency scenario being considered [Ejebe e Wollenberg, 1979]. This approach may be suitable for the $N - 1$ criterion and for average-sized systems, but the computation time and demanded resources tend to grow in an combinatoric rate for multiple contingencies.

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 et al., 2013] and the transmission capacity expansion planning [Majidi-Qadikolai e Baldick, 2016b,a].

A number of common approaches in evaluating scenarios to be considered in EPS contingency analysis are based on performance indices [Doraiswami e Carvalho, 1979; Ejebe e Wollenberg, 1979], which are non-monotonic functions and can have unpredictable behavior for multiple contingencies [Bulat et al., 2021]. Even though the EPS configuration changes in time, the operators have practical 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 contingency 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 et al., 2013], which approximate changes in line flows through a DC load flow aiming to detect possible overloads in generators and transmission lines.

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. In Gorton et al. [2009] and Jin et al. [2010], new methods and algorithms for calculating a previously existing centrality measure, the *betweenness centrality*, are proposed, providing speed-ups in parallel computing environments and for specific graph models of the EPS. In Cetinay et al. [2016], spectral graph theory is used for estimating sensitivities of electrical variables by line and bus removal. In Wang et al. [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 et al. [2011] and Yan et al. [2014], indicating that the method had a high rate of agreement. In order to extend the topological analysis to multiple contingencies, metaheuristics are applied in Coelho et al. [2022], avoiding an exhaustive method that rapidly grows in terms of computational cost.

2.2. Machine Learning on Graph Data

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 that summarizes its position inside G , where nodes that are similar to u are close to its projection in the latent space [Hoff et al., 2002].

One of the first successful methods to deliver good performance and scalability in the embedding was the deepwalk approach [Perozzi et al., 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.

Another embedding method, which is also based on random walks is the node2vec [Grover e 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 or how much the neighborhood of the root node is revisited. 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]. On its publication, node2vec outperformed most shallow embedding methods from the state of the art, including ones based in factorization methods and the deepwalk.

Differently from the aforementioned embedding techniques, which learn low-dimensional embeddings of nodes in a graph without knowing the downstream learning task for which the embedding will be applied, Graph Neural Networks (GNNs) use the information of the associated learning task for the embedding process. One of the first GNNs was the Graph Convolutional Network (GCN) from Kipf e Welling [2016], which achieved decent accuracies in many graph classification datasets. Later, Xu et al. [2018] developed a theoretical analysis for determining the most representative class of GNNs through the Graph Isomorphism Network (GIN). At the same time, Schlichtkrull et al. [2018] and Veličković et al. [2017] developed GNN architectures for considering properties from the edges in node and graph learning tasks, the Relational Graph Neural Networks (RGCN) and Graph Attention Networks (GAT), respectively.

3. Methodology

3.1. Transmission Line Criticality Index

The criticality index from Coelho et al. [2019] is evaluated by comparing the current flow betweenness centrality from Brandes e Fleischer [2005] for the original graph model of the EPS and the model of each of the possible k -contingencies, where k is the number of simultaneous line outages. The current flow betweenness centrality for a node v of a graph model G is denoted $C_G(v)$.

Let e^k denote a set of k edges of the graph G . The graph obtained by its removal from G is denoted by $G \setminus e^k$. The absolute variation on the current flow betweenness centrality for a node v and criticality index for an edge $e \in E$, where E is the edge set of graph G , are defined in Equations 1 and 2, respectively.

$$\Delta C[G \setminus e^k](v) = |C_{G \setminus e^k}(v) - C_G(v)| \quad (1)$$

$$GI(e) = \sum_{e \in E} \sum_{e \in e^k} \sum_{v \in V} \Delta C[G \setminus e^k](v) \quad (2)$$

After evaluating $GI(e)$ for every edge e of a graph model G , the approach proposed by Coelho et al. [2019] stated that edges for which $GI(e)$ is greater than a given threshold are considered critical, whereas the remaining ones are considered regular (or non-critical). For this work, a different criteria is used in this second step, in order to measure the impact of the class imbalance in the classification tasks more directly. Instead of considering a fixed threshold, edges with criticality indices above a given percentile p , taken upon the values of $GI(e)$ for all e in G are considered critical. In this approach, instead of defining the edges as critical following an universal metric, the critical edges are the $(1 - p)\%$ most critical edges in the graph.

Although k can take any value, with e^k being any tuple of edges, only results for $k = 1$ will be analyzed, considering the binary classification task.

3.2. Machine Learning Models

For dealing with the absence of Graph Machine Learning (GML) models for edge classification, a couple of alternatives were developed. 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 $t\%$, then the same ratio is sampled from the class with least nodes, obtaining a number of training nodes for the rarest class $\lfloor n_{min} \cdot t\% \rfloor$. 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.

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 1. 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. The selected classification method is the Support Vector Machine from Platt [1999]. This approach is referred to as *CGE*.

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

Algorithm 1: 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 element-wise 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 ;

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 than number of edges in the model graph. Additionally, 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 2 demonstrates the process of obtaining the dataset for the CLG approach.

Algorithm 2: 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;

Both approaches are evaluated and the best configurations results are compared for a number of combinations of the criticality percentile p , which determines the imbalance of the classes for the classification problem and the training split t .

For scoring the unbalanced binary classification task, the *F1-score* from Van Rijsbergen [1979] is used, together with the *AUC score* from McClish [1989]. The ROC curves of the trained classifiers are also compared, together with a visualization of the edge embeddings of the graph model for the evaluated power systems, made with t-SNE [Van der Maaten e Hinton, 2008].

4. Results

The proposed learning approaches are evaluated in graph models of four IEEE test systems, which are widely used for evaluation of new methodologies in sensitivity and power flow analysis. The graph model for these systems are shown in Figure 1.

Both proposed approaches aim to correctly classify each edge of the graph models by using embedding information. The CGE approach aims to accomplish such task with embeddings generated via a classical embedding method, whereas the CLG approach aims to use a GCN to do

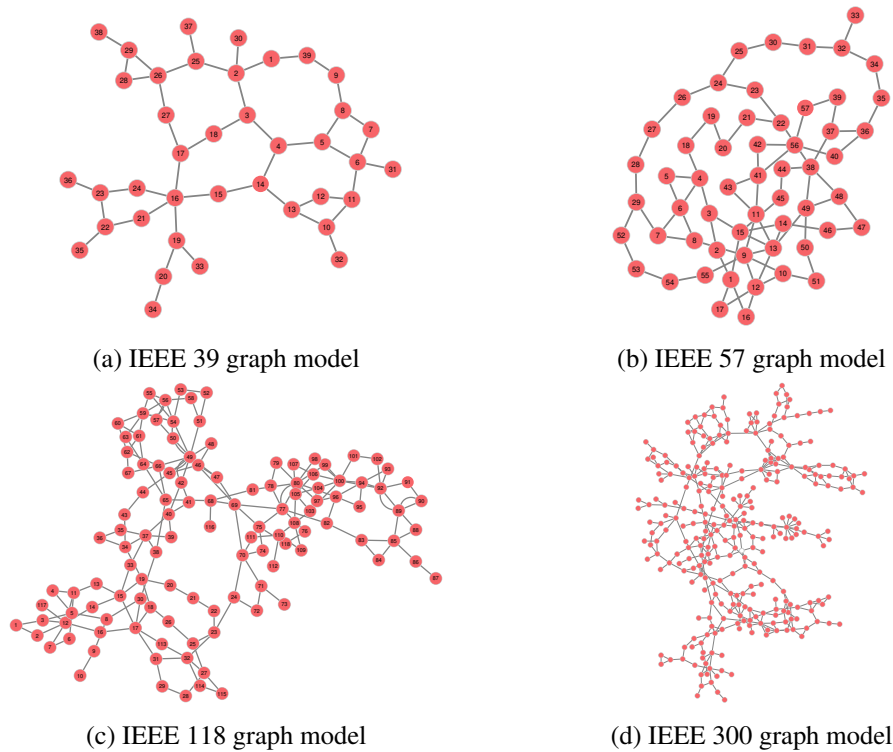


Figure 1: Graph models for the IEEE test systems used for evaluating the screening learning methodology.

both embedding and classification steps. For both approaches, the initial embeddings are generated randomly. In the end of the training step, the embeddings display some sort of spatial distinction among the classes, as can be seen in Figure 2.

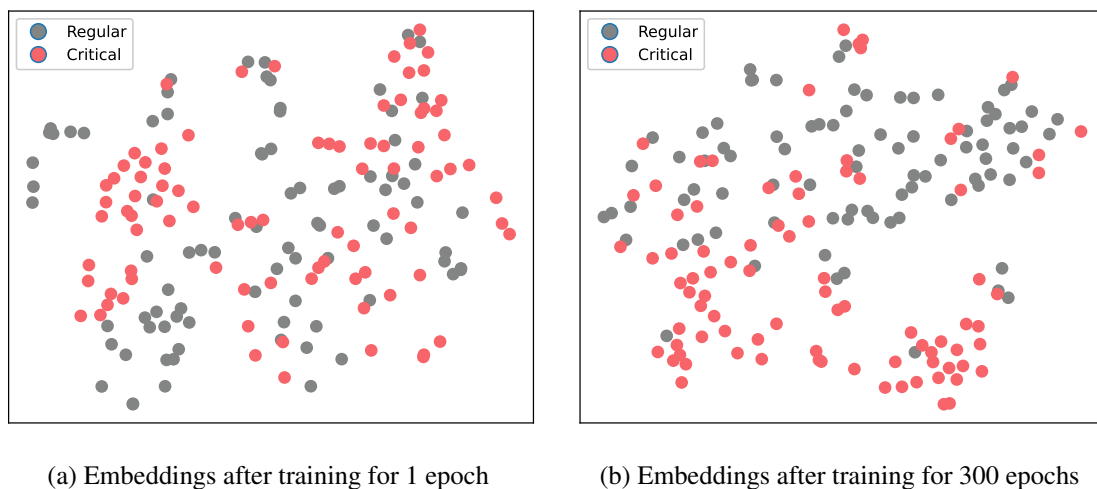


Figure 2: t-SNE dimensional reduction for the evolution of the obtained embeddings by the CLG approach for the IEEE 118 graph in one of the training realizations.

For each graph model, the training split t is chosen to be 10%, 30% and 50%. Considering that random under-sampling is applied, these values are taken independently for each class, and the class with least elements, which tends to be the critical class, which is considered the positive class in this study, determines the amount of elements in the training set. The percentile p for determining the critical and non-critical elements is taken to be 90, 70 and 50. Figure 3 illustrates a comparison among the ROC curves obtained for the IEEE 118 and IEEE 300 graph models.

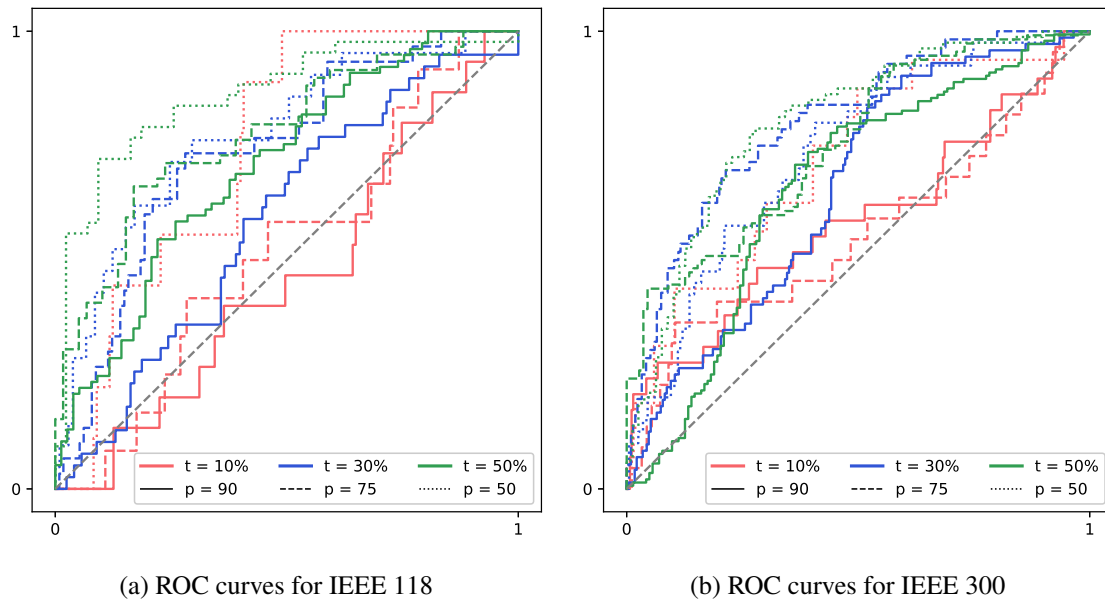


Figure 3: Comparison of the ROC curves for variations on the training split t and percentile p . The line color is determined by t , whereas the line style varies with p . The curves were obtained using the CLG approach.

The most balanced case is when $p = 50$ and $t = 50\%$, which consists in a scenario where half of the edges are considered critical and a large amount of data is provided for the training step. The least balanced case is when $p = 90$ and $t = 10\%$, which consists in the scenario where only 10% of the edges are considered critical, and only 10% of the edges are supplied for the training step, considering the class with less elements. These can be also seen in Figure 3, where the solid red lines, which represent the least balanced case, are consistently below the dashed and dotted lines and the green dotted lines, which represent the most balanced case, is consistently above the others.

The main results for all test systems, training splits t and percentiles p for determining the criticality criteria are found in Tables 1 and 2, which show the average $F1$ -score of the positive (critical) class and AUC score, respectively, taken from 10 independent realizations of each approach, for each combination of t , p and test system. The hyperparameters that were used in each approach, for each graph, can be found in Tables 3 and 4.

As can be seen in both Tables 1 and 2, the approaches do not reach consistently good scores on small graphs, which is an expected result, since machine learning techniques in general have better performance on large datasets. When a classifier reaches values of $F1$ and AUC scores above 0.50, it can be considered that it has the capacity of extracting some useful information for obtaining the correct label from the given input data. Another expectation, which is confirmed by the obtained result, is that both approaches deliver worse results for more imbalanced tasks.

$t(\%)$	p	CGE				CLG			
		IEEE 39	IEEE 57	IEEE 118	IEEE 300	IEEE 39	IEEE 57	IEEE 118	IEEE 300
10	90	0.06	0.09	0.11	0.14	0.17	0.14	0.12	0.23
	75	0.31	0.29	0.31	0.31	0.19	0.27	0.34	0.39
	50	0.37	0.44	0.55	0.45	0.45	0.65	0.71	0.63
30	90	0.12	0.08	0.09	0.12	0.18	0.11	0.13	0.14
	75	0.21	0.23	0.28	0.42	0.37	0.55	0.45	0.54
	50	0.20	0.51	0.48	0.64	0.66	0.66	0.68	0.63
50	90	0.03	0.07	0.09	0.11	0.15	0.32	0.15	0.18
	75	0.28	0.21	0.24	0.39	0.33	0.48	0.36	0.41
	50	0.49	0.52	0.57	0.51	0.51	0.48	0.75	0.70

Table 1: Average $F1$ -score of the positive class taken from 10 evaluations for both proposed approaches. Values above 0.50 are highlighted.

$t(\%)$	p	CGE				CLG			
		IEEE 39	IEEE 57	IEEE 118	IEEE 300	IEEE 39	IEEE 57	IEEE 118	IEEE 300
10	90	0.31	0.40	0.48	0.45	0.31	0.41	0.50	0.63
	75	0.55	0.41	0.42	0.48	0.53	0.59	0.67	0.62
	50	0.37	0.43	0.55	0.49	0.39	0.68	0.74	0.64
30	90	0.51	0.35	0.43	0.51	0.68	0.45	0.46	0.51
	75	0.38	0.34	0.46	0.68	0.67	0.87	0.72	0.82
	50	0.17	0.47	0.51	0.68	0.72	0.76	0.78	0.72
50	90	0.19	0.28	0.45	0.56	0.62	0.89	0.70	0.70
	75	0.51	0.46	0.50	0.69	0.60	0.86	0.75	0.74
	50	0.27	0.67	0.65	0.62	0.75	0.60	0.85	0.79

Table 2: Average AUC score taken from 10 evaluations for both proposed approaches. Values above 0.50 are highlighted.

When comparing both approaches, the CLG had consistently better results than the CGE, frequently obtaining AUC values above 0.70 for both IEEE 118 and IEEE 300, despite being trained in the line graph of the graph model made for the test power systems. These results suggest the validity of the line graph as a mean for obtaining edge embeddings of the original graph.

Name	IEEE 39	IEEE 57	IEEE 118	IEEE 300
Embedding size	8	16	16	16
Walk length	4	4	5	10
Context window size	3	3	3	6
Walks per node	8	8	10	15
Negative samples	1	1	1	1
Walk bias parameters	1	1	1	1
Learning rate	10^{-2}	10^{-2}	10^{-2}	10^{-2}
Epochs	200	200	200	200

Table 3: Hyperparameters for the CGE approach.

Name	IEEE 39	IEEE 57	IEEE 118	IEEE 300
Num. of GCN layers	2	2	2	2
Num. of linear layers	1	1	1	1
Dropout rate	0.5	0.5	0.5	0.5
Embedding size	32	64	128	256
GCN hidden units	16	32	32	64
Learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Epochs	500	200	300	200

Table 4: Hyperparameters for the CLG approach.

Regarding the chosen hyperparameters for approaches CGE and CLG, it can be seen that larger graphs demand more dimensions for both the embedding and classification steps. Tables 3 and 4 show that, for the *node2vec* embedding, the dimension did not increase, but the length of the random walks, the context window size and the number of walks per node were greater for the larger graphs. In the CLG approach, the number of layers and the dropout rate did not change, but the embedding size increased significantly with the number of nodes in the graph.

5. Conclusion and Future Works

This work proposes and evaluates two approaches for solving the edge binary classification problem using graph machine learning methods. The first approach combines the *node2vec* embedding with a SVM classifier, called CGE. The second approach is entirely based on graph neural networks, using an GCN for both the embedding and classification tasks, called CLG. Both approaches are evaluated for four test systems and for different levels of training splits t and imbalance among the existing classes, given by the criticality percentile p . The obtained results have shown that the CLG approach consistently obtains better results than the CGE one, considering the average $F1$ and AUC scores taken from 10 realizations of each combination of parameters.

For the smaller graphs, IEEE 39 and IEEE 57, it can be seen that the results either were values below the minimal threshold for acceptance of the classifiers, or did not follow a reasonable pattern. For these cases, it can be considered that the random initialization of the embeddings and the random splitting of edges among training and test sets had a large influence on the test result. For the larger graphs IEEE 118 and IEEE 300, since the number of training and test edges is greater, together with the chosen embedding dimensions, the impact of this randomness is minimized.

In future works, one might develop a methodology for deterministic initialization of the embeddings for a given graph, probably considering electrical measures. This could help in making the results more predictable. Another possible approach is developing a data augmentation approach for using in the learning tasks. This could contribute for balancing the classes, possibly boosting the classification scores, since this classification task is naturally imbalanced. At last, more modern GNN architectures can be considered for the task. However, these architectures turn the training process slower when compared to the GCN, as they contain more parameters in the encoding step.

References

- Bompard, E., Wu, D., e Xue, F. (2011). Structural vulnerability of power systems: A topological approach. *Electric Power Systems Research*, 81(7):1334–1340. ISSN 0378-7796.
- Brandes, U. e Fleischer, D. (2005). Centrality measures based on current flow. In Diekert, V. e Durand, B., editors, *STACS 2005*, p. 533–544, Berlin, Heidelberg. Springer Berlin Heidelberg.

- Bulat, H., Franković, D., e Vlahinić, S. (2021). Enhanced contingency analysis—a power system operator tool. *Energies*, 14(4).
- Cetinay, H., Kuipers, F. A., e Van Mieghem, P. (2016). A topological investigation of power flow. *IEEE Systems Journal*, 12(3):2524–2532.
- Coelho, E. P. R., Paiva, M. H. M., Segatto, M. E. V., e Caporossi, G. (2019). A new approach for contingency analysis based on centrality measures. *IEEE Systems Journal*, 13(2):1915–1923.
- Coelho, E. P., Alves, R. J., Paiva, M. H., Rocha, H. R., Caporossi, G., e Segatto, M. E. (2022). Topological multi-contingency screening based on current flow betweenness. *Electric Power Systems Research*, 203:107609. ISSN 0378-7796.
- Doraiswami, R. e Carvalho, M. (1979). Reliability indices for a power network considering static, transient, and dynamic performance. *IEEE Transactions on Reliability*, R-28(2):120–123.
- Ejebe, G. C. e Wollenberg, B. F. (1979). Automatic contingency selection. *IEEE Transactions on Power Apparatus and Systems*, PAS-98(1):97–109.
- Ejebe, G., Irisarri, G., Mokhtari, S., Obadina, O., Ristanovic, P., e Tong, J. (1996). Methods for contingency screening and ranking for voltage stability analysis of power systems. *IEEE Transactions on Power Systems*, 11(1):350–356.
- Gorton, I., Huang, Z., Chen, Y., Kalahar, B., Jin, S., Chavarría-Miranda, D., Baxter, D., e Feo, J. (2009). A high-performance hybrid computing approach to massive contingency analysis in the power grid. In *2009 Fifth IEEE International Conference on e-Science*, p. 277–283.
- Grover, A. e Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, p. 855–864, New York, NY, USA. Association for Computing Machinery.
- Hamilton, W. L., Ying, R., e Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, p. 1025–1035, Red Hook, NY, USA. Curran Associates Inc.
- Hamilton, W. (2020). *Graph Representation Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.
- Hoff, P. D., Raftery, A. E., e Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.
- Jin, S., Huang, Z., Chen, Y., Chavarría-Miranda, D., Feo, J., e Wong, P. C. (2010). A novel application of parallel betweenness centrality to power grid contingency analysis. In *2010 IEEE International Symposium on Parallel Distributed Processing (IPDPS)*, p. 1–7.
- Khanabadi, M., Ghasemi, H., e Doostizadeh, M. (2013). Optimal transmission switching considering voltage security and n-1 contingency analysis. *IEEE Transactions on Power Systems*, 28(1): 542–550.
- Kipf, T. N. e Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *arXiv e-prints*, p. arXiv:1609.02907.

- Majidi-Qadikolai, M. e Baldick, R. (2016a). Integration of $n - 1$ contingency analysis with systematic transmission capacity expansion planning: Ercot case study. *IEEE Transactions on Power Systems*, 31(3):2234–2245.
- Majidi-Qadikolai, M. e Baldick, R. (2016b). Stochastic transmission capacity expansion planning with special scenario selection for integrating $n - 1$ contingency analysis. *IEEE Transactions on Power Systems*, 31(6):4901–4912.
- McClish, D. K. (1989). Analyzing a portion of the roc curve. *Med Decis Making*, 9(3):190–5.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G., e Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13*, p. 3111–3119, Red Hook, NY, USA. Curran Associates Inc.
- Nickel, M., Murphy, K., Tresp, V., e Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.
- Perozzi, B., Al-Rfou, R., e Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, p. 701–710, New York, NY, USA. Association for Computing Machinery.
- Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, p. 61–74. MIT Press.
- Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., e Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference*, p. 593–607. Springer.
- Van der Maaten, L. e Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).
- Van Rijsbergen, C. (1979). *Information Retrieval*. Butterworths.
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., e Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- Wang, Z., Scaglione, A., e Thomas, R. J. (2010). Electrical centrality measures for electric power grid vulnerability analysis. In *49th IEEE Conference on Decision and Control (CDC)*, p. 5792–5797.
- Wood, A., Wollenberg, B., e Sheblé, G. (2013). *Power Generation, Operation, and Control*. Wiley.
- Xu, K., Hu, W., Leskovec, J., e Jegelka, S. (2018). How powerful are graph neural networks? In *International Conference on Learning Representations*.
- Yan, J., He, H., e Sun, Y. (2014). Integrated security analysis on cascading failure in complex networks. *IEEE Transactions on Information Forensics and Security*, 9(3):451–463.