UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

Jorge Luiz Franco

Graph machine learning for flight delay prediction due to holding manouver

São Carlos
2024

Jorge Luiz Franco

Graph machine learning for flight delay prediction due to holding manouver

Monograph presented to the Undergraduate Program in Computer Science at the Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, as part of the requirements for obtaining a bachelor's degree.

Advisor: Prof. Filipe Alves Neto Verri, Ph.D.

São Carlos

2024

Franco, Jorge Luiz
S856m Graph machine learning for flight delay prediction due to holding manouver / Jorge Luiz Franco ; orientador Filipe Alves Neto Verri. – São Carlos, 2024.
23 p. : il. (algumas color.) ; 30 cm.

1. Machine Learning. 2. Flight Delay. 3. Graphs I. Filipe Alves Neto Verri. II. Universidade de São Paulo. III. Instituto de Ciências Matemáticas e de Computação. IV. Graph ML for Flight Delay Prediction due to Holding Manouver

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my family for helping me through this journey, specially my father, João Luiz Franco, who always has encouraged my studies. Second, I express my gratitude to my advisor, Filipe Alves Neto Verri, for guiding me this semester and introducing me to this problem. Third, I would like to thank professor Vitor Venceslau Curtis, since this work would not be possible without his help. Last but not least, I express my gratitude to my fiance, Isis, for all the patience and caring during this time, my friends for the support, and God who never abandoned me.

“Without education, we are in a horrible and deadly danger of taking educated people seriously.”

G.K. Chesterton

ABSTRACT

FRANCO, J. L. **Graph ML for Flight Delay Prediction due to Holding Manouver**. 2024. 23p. Monografia (Trabalho de Conclusão de Curso) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

The growing demand for efficient last-mile delivery solutions, driven by the rise of e-commerce and urbanization, necessitates innovative approaches to manage the complexities of urban logistics. This study investigates the use of drones for last-mile delivery through a graph-based approach, focusing on the Multi-Agent Pathfinding (MAPF) problem, which involves routing multiple drones to deliver packages efficiently. Most algorithms proposed for the Last Mile Delivery Drones (LMDD) problem tend to overlook the inherent complexities of the MAPF problem. This research adopts a graph-based representation of the delivery area, transforming the problem into a network flow optimization. The proposed graph-based heuristic method is evaluated against a purely Mixed Integer Linear Programming (MILP)-based solution. Experimental results demonstrate that the heuristic approach not only enhances computational efficiency but also maintains high solution quality. Specifically, the heuristic outperforms MILP in terms of runtime while achieving comparable accuracy in path optimization. Additionally, a hybrid implementation combining heuristic and MILP is proposed. The exact MILP model has exponential complexity time as the number of drones is increased, which is expected since the MAPF paradigm is NP-Complete. The complexity of the heuristic is bounded by $\mathcal{O}(N^3 K \max(\log N, \log K))$, where K is the number of drones and N is the dimension, if the grid is square. The study highlights the potential of centralized control systems for managing a fleet of delivery drones. The findings from extensive simulations indicate that the graph-based heuristic effectively balances computational efficiency and operational reliability, making it a viable solution for real-world last-mile delivery applications. This research contributes to the broader field of drone logistics by offering a scalable and robust method for optimizing drone delivery paths, thereby supporting the integration of drones into commercial delivery systems.

Keywords: Optimization. Graphs. Last Mile Delivery Drones.

RESUMO

FRANCO, J. L. **Graph ML for Flight Delay Prediction due to Holding Manouver**. 2024. 23p. Monografia (Trabalho de Conclusão de Curso) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

A crescente demanda por soluções eficientes de entrega de última milha, impulsionada pelo crescimento do comércio eletrônico e urbanização, exige abordagens inovadoras para gerenciar as complexidades da logística urbana. Este estudo investiga o uso de drones para entrega de última milha por meio de uma abordagem baseada em grafos, focando no problema de Multi-Agent Pathfinding (MAPF), que envolve o roteamento de múltiplos drones para entregar pacotes de forma eficiente. A maioria dos algoritmos propostos para o problema de Drones de Entrega de Última Milha (LMDD, em inglês) tendem a ignorar as complexidades inerentes do problema MAPF. Esta pesquisa adota uma representação baseada em grafos da área de entrega, transformando o problema em uma otimização de fluxo de rede. O método heurístico baseado em grafos proposto é avaliado em comparação com uma solução puramente baseada em Programação Linear Inteira Mista (MILP, em inglês). Os resultados experimentais demonstram que a abordagem heurística não apenas melhora a eficiência computacional, mas também mantém alta qualidade nas soluções. Especificamente, a heurística supera a MILP em termos de tempo de execução, enquanto alcança uma precisão comparável na otimização de caminhos. Além disso, é proposta uma implementação híbrida combinando heurística e MILP. O modelo MILP exato tem complexidade exponencial conforme o número de drones aumenta, o que é esperado, uma vez que o paradigma MAPF é NP-Completo. A complexidade da heurística é limitada por $\mathcal{O}(N^3 K \max(\log N, \log K))$, onde K é o número de drones e N é a dimensão, se a grade for quadrada. O estudo destaca o potencial de sistemas de controle centralizados para gerenciar uma frota de drones de entrega. Os resultados de simulações extensivas indicam que a heurística baseada em grafos equilibra a eficiência computacional e a confiabilidade operacional de forma eficaz, tornando-a uma solução viável para aplicações de entrega de última milha do mundo real. Esta pesquisa contribui para o campo mais amplo da logística de drones oferecendo um método escalável e robusto para otimizar os caminhos de entrega de drones, apoiando assim a integração de drones em sistemas de entrega comerciais. Este trabalho também sugere avanços com agentes inteligentes usando Aprendizado por Reforço (RL, em inglês) e Redes Neurais de Grafos (GNNs, em inglês) para melhorar o paradigma descentralizado, o qual nós provamos ser inferior à nossa metodologia centralizada, uma vez que utiliza uma otimização local em contraste com nossa otimização global.

Palavras-chave: Otimização. Grafos. Drones de Entrega de Última Milha.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

LMDD	Last Mile Delivery Drones
MAPF	Multi-Agent Pathfinding
BFS	Breadth First Search
MILP	Mixed Integer Linear Programming
MOEA	Multiobjective Evolutionary Algorithm
UAV	Unmanned aerial vehicle
NSGA-I	Non-dominated Sorting Genetic Algorithm
PLS	Pareto Local Search
MIP	Mixed Integer Programming
TSP	Traveling Salesman Problem
TS	Tabu Seach
GVNS	General Variable Neighborhood Search
ALNS	Adaptive Large Neighborhood Search
SA	Simulated Annealing
UTM	Unmanned Aircraft System Traffic Management
FAA	Federal Aviation Administration
BVLOS	Beyond Visual Line of Sight
NASA	National Aeronautics and Space Administration
RL	Reinforcement Learning
GNN	Graph Neural Network

CONTENTS

1 INTRODUCTION

2 THEORETICAL FRAMEWORK AND RELATED WORKS

Graph machine learning can be tracked backwards to the problem of 'learning' on data that is inherently a graph (SILVA; ZHAO, 2016; CHAMI *et al.*, 2022) or can be modeled as a graph (VERRI; ZHAO, 2013; YOU *et al.*, 2020). This field encompasses a variety of tasks, including node/edge classification, network construction, link prediction, graph classification, graph cut/partitioning, network embeddings, graph coarsening/reduction, which rely on learning representations from graph-structured data. Over the last decades, researchers have developed numerous approaches to tackle these challenges, initially these techniques were most developed by complex networks researchers. However, in the last decade with the advancements in deep learning, the field has seen a significant shift towards the merging of three main communities: graph signal processing, deep learning and complex nets.

As described, defining the field of graph machine learning is not straightforward, as it encompasses a broad range of methods and applications. The tasks mentioned above are just a few examples of the many challenges that can be addressed through graph-based learning techniques. For clarity, these tasks can be categorized into three main learning paradigms: supervised, unsupervised, and semi-supervised learning. In this study, we are interested on the (semi-)supervised learning paradigm, which encompasses a variety of techniques designed to leverage learning to (partially-)labeled data (VERRI; URIO; ZHAO, 2018; AMANCIO *et al.*, 2014). But we can refine even more, in fact, this work will focus in the subset of graph elements prediction(classification/regression) methods.

In this chapter, we provide an overview of the theoretical framework of graph machine learning for node/edge prediction. Here we consider the division of the field into `classical` graph learning and `deep` graph learning, where here 'classical' refers to the machine learning techniques applied to graphs before the advent of graph neural networks, where standard ML algorithms were applied to graph data and the topological information measures were encoded as features together with the tabular data (COSTA, 2007; SILVA; ZHAO, 2016). This bipartition is what will pave the way of our explanation, since the last decade has seen a complex interplay between these two approaches. The field's evolution can be traced back to when Bruna *et al.* (2013) introduced one of the first GNN architectures leaned on the theory of graph signal processing. Concurrently, researchers were developing node embedding techniques like DeepWalk (PEROZZI; AL-RFOU; SKIENA, 2014) and node2vec (GROVER; LESKOVEC, 2016), which bridged classical and deep approaches while remaining using complex networks concepts. The subsequent years saw a surge in GNN architectures, including Graph Convolutional Networks(GCNs) (KIPF; WELLING, 2016) and GraphSAGE (HAMILTON; YING; LESKOVEC, 2017), marking a shift towards more sophisticated deep learning approaches for graphs and the unification of the field.

In the following sections, we explain each subset, their theory and applications, and how they have evolved over time. We also discuss the challenges and limitations of these methods.

2.1 Classical graph learning

These early efforts focused on shallow learning techniques such as feature engineering, graph traversal algorithms, and spectral methods, which laid the foundation for understanding graph structure and dynamics. Methods like community detection, centrality measures, and link prediction (SILVA; ZHAO, 2016) became key tools for analyzing large-scale networks in areas such as social science, biology, and infrastructure systems (NEWMAN, 2018; BOCCALETTI *et al.*, 2006). A significant focus of these techniques was to develop graph-based features that could be integrated into traditional machine learning models, effectively transforming graph data into a format compatible with standard algorithms like logistic regression, decision trees, and support vector machines. By encoding graph topology through hand-crafted features, such as connectivity and centrality, researchers could leverage these features for tasks like classification, regression, and clustering in tabular machine learning frameworks.

Among these features, centrality measures became particularly important due to their ability to capture the relative importance or influence of nodes in a graph, not just nodes (BONACICH, 1987), but other graph elements such as edges (LU; ZHANG, 2013; BRÖHL; LEHNERTZ, 2019) and hyperedges (TUDISCO; HIGHAM, 2021). Centrality measures, such as degree, betweenness, and closeness, served as input features in machine learning pipelines, helping to predict outcomes based on the structural role of nodes within the network.

Spectral centrality, particularly eigenvector centrality (BONACICH, 1987), has proven valuable in machine learning applications due to its ability to identify globally influential nodes. Eigenvector centrality assigns a score to each node by considering not only its direct connections but also the centrality of its neighbors, which results in a recursive definition. Mathematically, the eigenvector centrality x of a node in a graph can be defined as the solution to the equation $Ax = \lambda x$, where A is the adjacency matrix of the graph, and λ is the largest eigenvalue, thus x is the eigenvector associated with the largest eigenvalue. This relationship arises from the fact that the centrality of a node is proportional to the sum of the centralities of its neighbors, if we normalize the adjacency we get an stochastic matrix and then $\lambda = 1$ is the largest eigenvalue, named the `Perron vector`. The eigenvector centrality captures both local and global structure in a network, making it a powerful feature for tasks such as node classification, ranking, and recommendation systems. A related and widely used spectral measure is PageRank

(BRIN, 1998), which extends the idea of eigenvector centrality by introducing a damping factor to model random surfing behavior,

$$PR(v) = \frac{1-d}{N} + d \sum_{u \in \mathcal{N}(v)} \frac{PR(u)}{\deg(u)},$$

where $PR(v)$ is the PageRank score of node v, d is the damping factor, and $\mathcal{N}(v)$ represents the neighbors of node v. This iterative computation converges to a stationary distribution of scores, which can be interpreted as the probability of landing on a given node after a long random walk, in this sense the `Perron vector` signifies the convergence of the process in the infinite. PageRank has been widely used in ranking tasks, such as identifying important websites in search engines or recommending influential users in social networks.

However, these spectral-based centralities come with limitations. Eigenvector centrality requires the computation of the principal eigenvector of the adjacency matrix, which involves finding the largest eigenpair problem. This has a time complexity of $\mathcal{O}(n^2 d)$ for exact methods, where n is the number of nodes in the graph and d is the ratio of convergence for the power method. Furthermore, spectral methods can suffer from limitations rooted in the Perron-Frobenius theorem, which guarantees the existence of a unique largest eigenvalue only for irreducible, non-negative matrices. For graphs that are disconnected or have negative weights, these conditions are violated, and the eigenvector centrality may not be well-defined or interpretable. That is, the adjacency matrix should be non-negative and irreducible, where we could use the Perron test $\sum A^k > 0$ to see if the graph is strongly connected. These centralities also tend to be node-centric, lacking a direct extension to edge importance. For edge centrality, betweenness remains crucial, particularly in directed graphs, where the structural role of links (edges) must be considered to capture flow dynamics. Additionally, spectral centralities can be sensitive to noise and small perturbations in the graph structure, leading to instability in the centrality scores. Despite these challenges, spectral centrality remains a powerful tool for machine learning tasks that benefit from capturing global graph structure, provided that the computational and stability issues can be managed.

2.2 Deep graph learning

The rise of deep learning has revolutionized the field of graph machine learning, enabling the development of more powerful and scalable models for graph data. Graph neural networks can be divide in two main categories: spectral-based and spatial-based. Here is a trick thing, the GCN architecture (KIPF; WELLING, 2016) is commonly divulgated as a spatial-based method, since it is more intuitive talking about the convolution operation in the spatial domain, where we simply aggregate information from the immediate neighbors. However, the GCN is a spectral-based method, in fact, it can be thought as a simplification of the first spectral GNN (BRUNA *et al.*, 2013) proposed and that builds the math behind

GCNs. That said, first we introduce the spectral-based GNNs and then the spatial-based ones.

2.2.1 Spectral-based GNNs

Spectral methods are rooted in graph signal processing. The core idea is that a signal on a graph can be represented as node features, where each feature vector at a node corresponds to a 'signal' defined over the graph. In this context, the graph Laplacian $\mathcal{L} = D - A$, where D is the degree matrix and A is the adjacency matrix, plays a crucial role. It captures the structure of the graph and can be used to perform operations analogous to Fourier transforms in classical signal processing. Spectral methods can be categorized into two types: eigenvalue-based, where the focus is on creating a graph filter in the Fourier domain, and eigenvector-based, where the goal is to use a spectral basis to decompose the signal (BO *et al.*, 2023).

Bruna *et al.* (2013) introduced the first spectral Graph Neural Network (GNN), termed the Spectral CNN (SCNN), which aimed to translate ideas from standard Convolutional Neural Networks for images to graphs. The SCNN leverages the spectral decomposition of the graph Laplacian $\mathcal{L} = U\Lambda U^T$ to define a filter convolution operation in the Fourier domain. In this framework, the graph Fourier transform of a signal f is represented as $\hat{f} = U^T f$, and the convolution operation ($\star$) is defined as $g_\theta \star f = U g_\theta U^T f$, where g_θ is a learnable filter parameterized by θ. While powerful, the SCNN faces significant challenges: it requires $\mathcal{O}(n^3)$ computational complexity to calculate the entire graph spectrum, which is prohibitively expensive for large graphs. Moreover, the non-localized nature of eigenvectors means global information can overshadow local structural details, leading suboptimal balance between local and global information aligned with a huge parameter complexity (CHEN, 2020).

To address these limitations, ChebNetDefferrard, Bresson and Vandergheynst (2016) introduces Chebyshev polynomials to approximate spectral filters, effectively reducing computational complexity while preserving the ability to capture localized patterns in the graph structure. The main ideia is to redefine our previous filtering operation to $g_\theta(\mathcal{L})f = \sum_{k=0}^{K-1} \theta_k T_k(\widetilde{\mathcal{L}})f$, where $T_k(\widetilde{\mathcal{L}}) =$ is the Chebyshev polinomial of order k evaluated at the scaled Laplacian $\widetilde{\mathcal{L}} = 2\frac{\mathcal{L}}{\lambda_{\max}} - I_n$. This innovation not only makes spectral GNNs more scalable to larger graphs, since we just need to calculate the first eigenpair ($\mathcal{O}(n^2)$ through the power method) for the approximations, but also enhances their ability to balance local and global information processing. In fact, the filters are K-localized for polinomials of order K, that is intuitive by remembering that $\mathcal{L}^K$ represents the paths with length less or equal to K. The ChebNet laid the foundation for GCNs (KIPF; WELLING, 2016). Although GCNs are commonly referred to as spatial methods, their underlying principle is rooted in the truncation of the Chebyshev expansion to $K = 1$, which limits the filter to first-order neighbors. This simplification reduces computational complexity significantly while preserving effectiveness. Instead of requiring the full spectral decomposition of the Laplacian matrix, GCNs use a localized approximation of the graph convolution, expressed as: $g_\theta \star f \approx \theta(I_n + \widetilde{A})f$, where $\widetilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized adjacency matrix, where A is the adjacency matrix, and D is the degree matrix. This approximation results in an efficient propagation rule that aggregates information from a node's immediate neighbors while updating the node's features. This propagation mechanism is often confused as a spatial method because it effectively propagates information from adjacent nodes—akin to a spatial neighborhood aggregation. Although its already a simple model, results have shown that GCNs can achieve state-of-the-art performance on a variety of tasks with even more simplifications (WU *et al.*, 2019).

However, as we can note, all these spectral methods works just in undirected graphs, since it needs the spectral decomposition

2.2.2 Spatial-based GNNs

Spatial-based GNNs differ from spectral-based approaches by directly leveraging the graph structure to perform convolutions in the spatial domain, rather than relying on the spectral decomposition of graph operators like the Laplacian. In spatial-based methods, the convolution operation is interpreted as an aggregation of node features from a node's local neighborhood, akin to how standard convolutional neural networks aggregate pixel information from nearby regions in image data. These methods operate by iteratively updating node representations by propagating information between neighboring nodes, making them intuitive and highly scalable for large-scale graphs.

The general framework for message passing in spatial-based GNNs can be described as follows. For each node i, at layer t, we aggregate the features of its neighbors $\mathcal{N}(i)$ to produce an updated node embedding: $\mathbf{m}_i^{(t+1)} = \text{AGGREGATE}^{(t)}\left(\left\{\mathbf{h}_j^{(t)} : j \in \mathcal{N}(i)\right\}\right)$, where $\mathbf{h}_j^{(t)}$ is the feature of neighboring node j at layer t. Then, we update the node i's representation: $\mathbf{h}_i^{(t+1)} = \text{UPDATE}^{(t)}\left(\mathbf{h}_i^{(t)}, \mathbf{m}_i^{(t+1)}\right)$, where $\text{AGGREGATE}^{(t)}$ is a neighborhood aggregation function, and $\text{UPDATE}^{(t)}$ is the node update function.

The general idea behind spatial-based GNNs is that, for each node, we aggregate the features of its neighbors to produce an updated node embedding. A key example of this is the GraphSAGE architecture (HAMILTON; YING; LESKOVEC, 2017), which computes node representations by sampling and aggregating features from the node's neighbors. The GraphSAGE model employs several types of aggregation functions, including mean, LSTM-based, and pooling aggregators, which allow for flexible and inductive learning on large graphs. In particular, GraphSAGE enables the generation of embeddings for unseen nodes, making it suitable for inductive learning tasks, where the model needs to generalize to new nodes that were not present during training. Unlike spectral-based methods, which are constrained to a fixed graph size and structure due to their reliance on the graph Laplacian, spatial-based GNNs are inherently more flexible and can be applied to dynamic and evolving graphs. These models perform neighborhood aggregation locally, and therefore do not require the global knowledge of the graph structure that spectral methods need. This flexibility makes them particularly useful for large-scale graphs and for graphs where the structure may change over time, such as social networks or knowledge graphs.

Another prominent spatial-based GNN is the Graph Attention Network (GAT) (VELICKOVIC *et al.*,), which introduced attention mechanisms into graph learning. GAT models learn to assign different weights to the neighbors of a node, allowing the model to focus more on the most relevant neighbors during the feature aggregation process. This is achieved using a self-attention mechanism, where the importance of neighboring nodes is learned through a shared attention coefficient, $e_{ij} = \text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i||\mathbf{W}\mathbf{h}_j])$, where e_{ij} represents the attention coefficient between nodes i and j, $\mathbf{W}$ is a learnable weight matrix, $\mathbf{h}_i$ and $\mathbf{h}_j$ are the feature vectors of nodes i and j, and $||$ denotes concatenation. The attention coefficients are then normalized across all of a node's neighbors using the softmax function, $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik})}$, this normalization ensures that the attention coefficients sum to 1, allowing the model to perform a weighted aggregation of the neighbors' features, $\mathbf{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}\mathbf{W}\mathbf{h}_j\right)$, here $\mathbf{h}'_i$ is the updated representation of node i, and σ is a non-linear activation function. By learning attention coefficients, GATs can capture both the importance and the structure of the graph, making them particularly effective in tasks where the relationships between nodes are not equally important, such as in citation networks or social media graphs.

3 PROBLEM DESCRIPTION

4 MATERIALS AND METHODS

Our data is a lot imbalanced, but there is little work on imbalanced data in the literature for graph machine learning and most of the work is based on oversampling

5 RESULTS

6 CONCLUSION

REFERENCES

AMANCIO, D. R. *et al.* A systematic comparison of supervised classifiers. **PLOS ONE**, Public Library of Science, v. 9, n. 4, p. 1–14, 04 2014. Available at: https://doi.org/10.1371/journal.pone.0094137.

BO, D. *et al.* **A Survey on Spectral Graph Neural Networks**. 2023. Available at: https://arxiv.org/abs/2302.05631.

BOCCALETTI, S. *et al.* Complex networks: Structure and dynamics. **Physics Reports**, v. 4, n. 424, p. 175–308, 2006.

BONACICH, P. Power and centrality: A family of measures. **American journal of sociology**, University of Chicago Press, v. 92, n. 5, p. 1170–1182, 1987.

BRIN, S. The pagerank citation ranking: bringing order to the web. **Proceedings of ASIS, 1998**, v. 98, p. 161–172, 1998.

BRÖHL, T.; LEHNERTZ, K. Centrality-based identification of important edges in complex networks. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, AIP Publishing, v. 29, n. 3, 2019.

BRUNA, J. *et al.* Spectral networks and locally connected networks on graphs. **arXiv preprint arXiv:1312.6203**, 2013.

CHAMI, I. *et al.* Machine learning on graphs: A model and comprehensive taxonomy. **Journal of Machine Learning Research**, v. 23, n. 89, p. 1–64, 2022. Available at: http://jmlr.org/papers/v23/20-852.html.

CHEN, X. Understanding spectral graph neural network. 12 2020. Available at: https://arxiv.org/abs/2012.06660.

COSTA, L. d. F. Characterization of complex networks: A survey of measurements. **Advances in physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007.

DEFFERRARD, M.; BRESSON, X.; VANDERGHEYNST, P. Convolutional neural networks on graphs with fast localized spectral filtering. **Advances in neural information processing systems**, v. 29, 2016.

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**. [*S.l.: s.n.*], 2016. p. 855–864.

HAMILTON, W.; YING, Z.; LESKOVEC, J. Inductive representation learning on large graphs. **Advances in neural information processing systems**, v. 30, 2017.

KIPF, T. N.; WELLING, M. Semi-supervised classification with graph convolutional networks. **arXiv preprint arXiv:1609.02907**, 2016.

LU, L.; ZHANG, M. Edge betweenness centrality. *In*: _____. **Encyclopedia of Systems Biology**. New York, NY: Springer New York, 2013. p. 647–648. ISBN 978-1-4419-9863-7. Available at: https://doi.org/10.1007/978-1-4419-9863-7_874.

NEWMAN, M. **Networks**. [*S.l.: s.n.*]: Oxford university press, 2018.

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**. [*S.l.: s.n.*], 2014. p. 701–710.

SILVA, T.; ZHAO, L. **Machine Learning in Complex Networks**. Springer International Publishing, 2016. ISBN 9783319172897. Available at: https://books.google.com.br/books?id=WdDurQEACAAJ.

TUDISCO, F.; HIGHAM, D. J. Node and edge nonlinear eigenvector centrality for hypergraphs. **Communications Physics**, Nature Publishing Group UK London, v. 4, n. 1, p. 201, 2021.

VELICKOVIC, P. *et al.* Graph attention networks. **stat**, v. 1050, n. 20, p. 10–48550.

VERRI, F. A. N.; URIO, P. R.; ZHAO, L. Advantages of edge-centric collective dynamics in machine learning tasks. **Journal of Applied Nonlinear Dynamics**, L&H Scientific Publishing, v. 7, n. 3, p. 269–285, 2018.

VERRI, F. A. N.; ZHAO, L. High level data classification based on network entropy. *In*: **The 2013 International Joint Conference on Neural Networks (IJCNN)**. [*S.l.: s.n.*], 2013. p. 1–5.

WU, F. *et al.* Simplifying graph convolutional networks. *In*: PMLR. **International conference on machine learning**. [*S.l.: s.n.*], 2019. p. 6861–6871.

YOU, J. *et al.* Handling missing data with graph representation learning. **Advances in Neural Information Processing Systems**, v. 33, p. 19075–19087, 2020.