UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Classificação de Alto Nível Baseada em Redes Complexas

Josimar Edinson Chire Saire

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

ICMC USP
SÃO CARLOS

Josimar Edinson Chire Saire

Classificação de Alto Nível Baseada em Redes Complexas

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Zhao Liang

USP – São Carlos
Janeiro 2024

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

```
        Chire Saire, Josimar Edinson
C541cc     Classificação de Alto Nivel Baseada em Redes
        Complexas  / Josimar Edinson Chire Saire;
        orientador Zhao Liang. -- São Carlos, 2023.
           76 p.


           Tese (Doutorado - Programa de Pós-Graduação em
        Ciências de Computação e Matemática Computacional) --
         Instituto de Ciências Matemáticas e de Computação,
        Universidade de São Paulo, 2023.


           1. INTELIGÊNCIA ARTIFICIAL. 2. RECONHECIMENTO DE
        PADRÕES. I. Liang, Zhao, orient. II. Título.
```

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

Josimar Edinson Chire Saire

Complex Network-Based High-Level Classification

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Zhao Liang

USP – São Carlos
January 2024

This work is a significant milestone in my journey as a researcher. I am profoundly grateful to God for His daily strength, to my family (Celso, Matilde, Yadira, Joel, Lucio) for their unwavering support from my undergraduate studies to my Ph.D., and to my friends from Brazil, Peru, Bangladesh. And to Research4Tech, a project with let me meet people from Latin America. Lastly, as I approach the final stage, I wish to express my heartfelt appreciation to LFS, whose unwavering support and boundless energy have been a true blessing.

AGRADECIMENTOS

I am grateful for the opportunity to study at one of the top universities in Brazil and Latin America. I extend my thanks to the team that reviewed my application and believed in my capabilities. And, I extend my sincere thanks to Prof. Zhao for accepting to be my supervisor and for invaluable assistance in refining my proposal.

I am also appreciative of Research4Tech, a project that allowed me to connect with researchers/professionals from Latin America and provided me with a valuable opportunity for professional growth.

“This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001”.

“Whatever you do, work at it with all your heart, as working for the Lord, not for human masters
since you know that you will receive an inheritance from the Lord as a reward.
It is the Lord Christ you are serving”
Colossians 3:23-24

RESUMO

CHIRE SAIRE, J. E. **Classificação de Alto Nível Baseada em Redes Complexas**. EN. 76 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, EN.

A classificação de dados desempenha um papel crucial no campo do aprendizado de máquina. Enquanto as técnicas tradicionais se concentram nas características físicas do conjunto de dados, denominadas *classificação de baixo nível*, propomos uma abordagem inovadora. Em contraste, o cérebro humano destaca-se na capacidade de discernir objetos com base na organização semântica, inspirando a *classificação de alto nível*. Apesar do progresso em diversas técnicas, desafios persistem. Neste contexto, apresentamos soluções para enfrentar esses desafios: 1) Em técnicas tradicionais, os testes de confirmação de padrões mostram pouca variabilidade, especialmente em redes de treinamento em grande escala, considerando medidas convencionais como assortatividade e coeficiente de clustering. Introduzimos medidas inovadoras baseadas em ciclos curtos e *Minimum Spanning Tree* (MST) da rede, ampliando a sensibilidade da confirmação de padrões. 2) Nas técnicas tradicionais, as redes correspondentes a cada classe são construídas com regras locais, como o método dos *k* Vizinhos Mais Próximos (*k*NN). Propomos otimizar as redes de treinamento para um desempenho aprimorado. 3) Aplicamos nossa técnica inovadora ao diagnóstico de COVID-19, classificando imagens de raio-X do tórax. Avançamos na precisão e eficácia da classificação de dados de alto nível, especialmente no diagnóstico de COVID-19 por meio de imagens de raio-X do tórax. Esses esforços representam avanços significativos na melhoria das técnicas de classificação, abrindo caminho para aplicações específicas, como o diagnóstico de COVID-19 com imagens de raio-X do tórax.

Palavras-chave: Redes Complexas, Classificação de Alto Nível, Classificação de Dados, Algoritmo de Colônia de Formigas, Algorítmo Genético, COVID-19.

ABSTRACT

CHIRE SAIRE, J. E. **Complex Network-Based High-Level Classification**. EN. 76 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, EN.

Data classification plays a crucial role in the field of machine learning. While traditional techniques focus on the physical characteristics of the dataset, referred to as *low-level classification*, we propose an innovative approach. In contrast, the human brain excels in discerning objects based on semantic organization, inspiring *high-level classification*. Despite progress in various techniques, challenges persist. In this context, we present solutions to address these challenges: 1) In traditional techniques, pattern confirmation tests show little variability, especially in large-scale training networks, considering conventional measures such as assortativity and clustering coefficient. We introduce innovative measures based on short cycles and the *Minimum Spanning Tree* (MST) of the network, enhancing pattern confirmation sensitivity. 2) In traditional techniques, networks corresponding to each class are built with local rules, such as the k-Nearest Neighbors (kNN) method. We propose optimizing training networks for enhanced performance. 3) We apply our innovative technique to the diagnosis of COVID-19, classifying chest X-ray images. We advance the accuracy and effectiveness of high-level data classification, particularly in the diagnosis of COVID-19 using chest X-ray images.

Keywords: Complex Networks, Data Classification, High-Level Classification, Ant Colony Algorithm, Network Construction, Genetic Algorithm, COVID-19.

LISTA DE ILUSTRAÇÕES

LISTA DE ALGORITMOS

LISTA DE TABELAS

SUMÁRIO

CAPÍTULO

1

INTRODUCTION

In today's era of continuous daily data expansion (WITTEN; FRANK, 2000; TAN; STEINBACH; KUMAR, 2005), and given the growing significance of comprehending data for businesses, public institutions, and beyond, Data Science has emerged as a popular solution to confront this ongoing challenge. Data Science (SHRESHTHA *et al.*, 2019) is a discipline that amalgamates Statistics, Computer Science, and Mathematics, and couples this with a profound understanding of various business scenarios. Its objective is to extract value from historical data and generate pertinent insights to inform decision-making. Within the realm of Computer Science, Artificial Intelligence (AI) plays a pivotal role. AI strives to replicate certain human or living being behaviors and capabilities and it encompasses numerous research domains, including Computer Vision, Machine Learning, Natural Language Processing, and Bioinspired Algorithms (MITCHELL, 1997; DUDA; HART; G.STORK, 2001; HAN; KAMBER, 2005).

To endeavor to replicate the learning abilities of humans and animals through computers, the research field known as Machine Learning has arisen. Its primary objective is to develop algorithms capable of "learning"from experiences, often derived from diverse domains (ALPAYDIN, 2014; BISHOP, 2011; MITCHELL, 1997; VAPNIK, 2008; RAEDT, 2008). Machine learning techniques have the capacity to generate models that can organize existing knowledge or even emulate the behavior of a human specialist in specific domains. Traditionally, Machine Learning is categorized into four main domains (SARKER, 2021): *supervised learning*, *unsupervised learning*, *semi-supervised learning* and *reinforcement learning* (ALPAYDIN, 2014; BISHOP, 2011; MITCHELL, 1997; VAPNIK, 2008). In supervised learning, the objective is to infer concepts from samples with known class labels. In essence, the learning process involves constructing an input-output mapping function by observing the provided training data. If the labels consist of discrete values, the problem is referred to as *classification*; if the labels are continuous values, it is termed *regression*. In unsupervised learning, the primary task is to discern the inherent characteristics of data without the guidance of labels or supervisors. One of the fundamental challenges in unsupervised learning is *data clustering*, wherein input data is segmented into groups based on established similarity criteria. This process is driven by the data itself, with no prior knowledge about the existence of specific data classes (BISHOP, 2011; MITCHELL, 1997). However, in supervised learning, manually labeling data can be a slow and expensive process, often necessitating the expertise of a human. To mitigate this challenge, *semi-supervised learning* comes into play. This approach extends labeled data samples' labels to encompass all unlabeled data samples. Typically, in constructing a classifier using semi-supervised learning, a database is composed of a substantial amount of unlabeled data alongside only a small subset of labeled data. This significantly reduces the workload of human experts in data sample labeling. Moreover, the inclusion of unlabeled data can enhance the classifier's performance (CHAPELLE; SCHöLKOPF; ZIEN, 2006). In reinforcement learning, machines act and learn like curious explorers, navigating environments through trial and error, earning rewards for successful choices and ultimately mastering complex tasks. This "learn by doing"approach unlocks fascinating possibilities for robots, self-driving cars, and more (SUTTON; BARTO, 2018) .

Data classification is a supervised learning task characterized by the use of a given training set with labeled samples to generate a model (classifier), which is later used to classify unlabeled data (ALPAYDIN, 2014; DUDA; HART; G.STORK, 2001). Due to the importance of this learning paradigm in a number of real applications, many classification techniques have been developed (ALPAYDIN, 2014; DUDA; HART; G.STORK, 2001; HAYKIN, 1998; VAPNIK, 2008), such as the K-nearest-neighbors (KNN), Discriminating Linear Analysis (LDA), Naive-Bayes, neural networks, Support Vector Machines (SVM), Decision Tree, and Deep Learning (GOODFELLOW; BENGIO; COURVILLE, 2016; LECUN; BENGIO; HINTON, 2015).

Traditional data classification techniques perform training and classification according to physical attributes of the data (for example, distance, similarity or distribution), which are called *low-level classification* techniques. However, data samples are not isolated, but tend to form certain complex patterns. The data classification that considers both the pattern formation of the input data and the physical attributes is called *high-level classification* (Carneiro; Zhao, 2013).

Essentially, traditional classification techniques follow a common heuristic: they partition the data space into distinct subspaces, each representing a specific class. The trained classifier defines the decision boundaries within this data space, and the labeling process assesses the relative position of unlabeled instances in relation to these boundaries. In crisp classification, these subspaces do not overlap, but in fuzzy classification (TUY; RIOS, 2020), there can be slight overlap. However, substantial distortions in the shapes of overlapping classes or subspaces are typically not permitted. In other words, traditional data classification techniques primarily rely on the physical features of the training data, overlooking many other inherent and semantic relationships among data elements. These semantic relationships often lead to complex class structures within the data space. Conversely, the human (and animal) brain (VIGLIOCCO *et al.*, 2009) is adept at identifying patterns based on the semantic meaning of input data. Therefore, it becomes intriguing to explore classification beyond the conventional data space partitioning strategy employed by traditional techniques. In this context, network-based techniques offer the potential to contribute to data classification on a semantic level that diverges from the conventional data space division approach.

Complex network is a large scale graph with nontrivial connectivity pattern (BARABÁSI; ALBERT, 1999; BARABÁSI *et al.*, 2016; NEWMAN, 2010; CHIRE-SAIRE, 2020). It is composed of a set of nodes and a set of edges. The nodes can represent objects and the edges define the relationships between them, such as social interaction, chemical reaction, distance, similarity, etc. Many measurements can be extracted to represent local and global relationships embedded in a network. There exist many real-world complex networks, for example, Internet (BARABÁSI; ALBERT, 1999), World Wide Web (ALBERT; JEONG; BARABÁSI, 1999), social networks (MODELS. . . , 2005), food chain (MONTOYA; SOLÉ, 2002), metabolism networks (JEONG *et al.*, 2000), blood stream (WEST; BROWN; ENQUIST, 1999), mail delivery network and electric energy distribution networks (ALBERT; ALBERT; NAKARADO, 2004).

Bringing together the preceding statements, high-level classification through the utilization of complex networks has emerged as a promising research avenue for uncovering data patterns. Recent studies have demonstrated the efficacy of representing these internal relationships, yielding commendable performance. The approach to employing complex networks for classification tasks entails the following steps: a) Construct a complex network for each class. b) Introduce the test instances individually into the complex network and assess the variances or modifications, taking into account specific pre-selected network metrics.

The original idea of high-level classification has been proposed in (SILVA; ZHAO, 2012) (SILVA; ZHAO, 2015a) and extended in (CARNEIRO; ZHAO, 2018a) (Colliri *et al.*, 2018). In this scheme, the low-level classification can be implemented by any traditional classification technique, while the high-level technique explores the complex topological properties of the network built from the input data. In the work introduced in (SILVA; ZHAO, 2012), the high-level classification is performed using three network measures: assortativity, clustering coefficient and average degree. In (SILVA; ZHAO, 2015a), the measure of the transient and the cycle lengths of a tourist walk was used for characterising network patterns. In both cases, the classification is performed by checking the conformity of the pattern formed by each network (each class) for each test data sample, i.e. a test sample is assigned to that class where its insertion in the corresponding network causes the least variation of the measures under consideration. In (CARNEIRO; ZHAO, 2018a) (Colliri *et al.*, 2018), the part of the low-level classification has been eliminated and pure high-level classification techniques have been proposed.

Although some advances have been made in high-level classification, there are still open research questions, which should to be answered. For example, how to build suitable complex networks for capturing data patterns? which network metrics are more sensitive to the insertion of test data samples, consequently, more suitable to pattern conformation checking? These issues are directly influence classification performance. In this work, we will address these questions and apply the developed high-level classification technique to COVID-19 diagnosis by classifying the chest X-ray images.

1.1 Objectives

The main objective of this work is to purpose an approach to optimize the construction of complex networks and increase the sensitivity test of high-level classification in order to improve the performance of the technique. Three specific objectives are presented as follows:

- Propose a metric with higher sensitivity to network perturbation (insertion of a testing data sample) to improve the overall performance. Remember that the process to check if one test sample belongs or not to a class is inserting it into each network (constructed by the data samples of each class) and calculating the variation of some pre-selected network measures. The test sample is classified to the class where its insertion causes the smallest variation of the measures. It means that the test sample conforms the pattern formed by that class. In the original high-level classification techniques, the following measures have been applied: assortativity, clustering coefficient, average degree, betweenness, and global efficiency. Since the networks are usually large and each time only one node (a test sample) is inserted, the variations in terms of the network measures are usually very small, which, in turn, make the classification confusing. Therefore, it is desirable to find a network measure or a set of measures, which is more sensitive to network changing caused by the test sample insertion and, consequently, to improve the classification performance. In this work, we propose to use the shortest cycle determined by Ant Colony Algorithm applied in the networks as a measure to calculate the variations caused by testing sample insertion. Ant Colony Optimization (COLORNI; DORIGO; MANIEZZO, 1991) is a bio-inspired algorithm based on ants behaviour to find the good path to get the food. It is proposed to solve optimization problems and find good solutions in a reasonable computational time. Some promising results have been obtained and the extensive numerical analyses will be presented in this work.

- Build optimal complex networks and inserting test samples in an optimized manner. The main step of high-level classification is to build a complex network from each class of the training data set. This is because the network construction process determines the connectivity pattern of the data samples. Therefore, different network construction methods generate different data patterns, which, in turn, lead to different classification results. Original high-level classification techniques use strictly local rules, such as kNN and ε-radius, to connect nodes. In this way, global patterns are ignored in the network construction process. Moreover, nowadays, we encounter many high-dimensional data sets, i.e., each data sample has a high-dimensional feature vector. Usually, the features do not have the same importance for classification, i.e., some features are more relevant than the others. Therefore, it is important to consider an optimization approach to find out the appropriate weight to each data feature. In this way, we get a better complex network to represent the pattern formation of data sets.

- Performance testing using artificial data sets and application to COVID-19 diagnosis by classifying chest X-ray images, Snoring classification by audio samples. The proposed technique must be tested to know how good it is, then, artificial data sets are build with controlled scenarios and are used to systematically study the method's performance. Besides, classical real data sets are also used to test the proposed method. Considering the high urgency and importance in combating the pandemic related to COVID-19, we apply the developed high-level technique to chest X-ray image classification for COVID-19 diagnosis. Finally, we also apply the method to Snoring classification using audio samples to show its versatility.

1.2 Organization of the Document

The rest of the document is organized as follows: in Chapter 2, an overview about complex networks and high-level classification are presented. After that, in Chapter 3, we propose two more sensitive metrics based on the shortest cycle determined by Ant Colony Algorithm and based on the Minimum Spanning Tree (MST), respectively. In the same chapter, we also present an approach to build optimal complex networks for high-level classification. In Chapter 4, we present the experimental results and the applications to validate the approach. Finally, in Chapter 5, we present concluding remarks and point out some future works.

CAPÍTULO

2

COMPLEX NETWORKS AND NETWORK-BASED HIGH-LEVEL CLASSIFICATION

In this chapter, we will give a brief overview on the relevant concepts, models and measures of complex networks and a general vision on the network-based high-level classification. These are fundamental research topics for the development of this thesis.

2.1 Relevant Concepts, Models, and Measures of Complex Networks

Complex network is an interdisciplinary research area, which has triggered much interests of various scientific communities including the machine learning community for the last decades. Due to its vast contents and continuing development, it is impossible to give a comprehensive review within this thesis. Therefore, we here just present the relevant concepts, models, and measures.

2.1.1 Complex Network Concepts

Complex networks (BARABÁSI, 2016) are large scale graphs with nontrivial connectivity patterns. Therefore, a complex network shares the same definition to a graph. A graph G is defined as a pair (V,E), where V is a finite nonempty set of vertices and E is the set of edges between the vertices $E \subseteq \{(\mathrm{u,v}) \mid \mathrm{u,v} \in \mathrm{V}\}$. There are different kinds of graphs according to the edge features:

- Undirected graph: If the relation E is symmetric, meaning that $\forall(\mathrm{u,v}) \in \mathrm{E} \implies (\mathrm{v,u}) \in E$, it is said that the graph is undirected. In other terms, if there is an edge linking vertices u

to v, so there will be a link from v to u too.

- Directed graph: If the relation E satisfies the following restriction: $\exists$ (u,v) $\in E \mid$ (v,u) $\notin E$, it is said that the graph is directed (digraph). In other terms, these kinds of graphs must have at least an arbitrary edge linking u to v, with an absence of the oppose link.

- A weighted graph G is defined as a triple (V,E,W), where V and E are the sets of vertices and edges, respectively, and W is a matrix composing the edge weights. For example, the entry $W(u,v) = k, (u,v) \in E$, fixes as k the weight of the edge linking vertices u to v. If $(u,v) \notin$ E $\implies W(u,v) = 0$.

Some relevant concepts (CHARTRAND; ZHANG, 2012) are presented below.

Adjacent Vertices: Two vertices are called adjacent if they share a common edge, in which case the common edge is said to join the two vertices. An edge and a vertex on that edge are called incident. In digraphs, it may occur that vertex v is adjacent to u, but the opposite is not true, in the case that only a directed edge from u to v is present.

Neighborhood of a Vertex: The neighborhood of a vertex $v \in V$ in a graph G is the set of vertices adjacent to v. The neighborhood is denoted by $N(v)$. Note that the neighborhood does not include v itself.

Degree of a Vertex: The degree of a vertex v is the total number of vertices adjacent to v. The degree of a vertex v is denoted by $k(v)$. We can equivalently define the degree of a vertex as the cardinality of its neighborhood set and say that, for any vertex v, $k(v) =$|N(v)|.

Graph Path: A path P is an ordered list of edges: $P = (v_1, v_2), (v_2, v_3), ..., (v_k, v_{k+1})$, such that no vertex and no edge is repeated. The first vertex of the first edge of a path is the path's origin and the second vertex of the last edge is the path's destination. Both path's origin and destination are called endpoints of the path. The length of a path is given by the cardinality of the sequence P, i.e.,$|P|$. Moreover, P is a path in G only if every entry of P is in E.

Circuit: A circuit is a path which ends at the vertex it begins. Therefore, a self-loop is a circuit of length one.

Cycle: A cycle is a circuit in which no vertex, except the first, appears more than once.

Distance between vertices: The distance $d(u,v)$ between two vertices $u \in V$ and $v \in V$ is the length of the shortest path from u to v, considering all possible paths in G from u to v. The distance between any vertex and itself is 0. If there is no path from u to v, then $d(u,v) = \infty$.

Diameter: The diameter of G is the length of the largest distance in G.

Vertex Eccentricity: The eccentricity of $v1 \in V$ is the largest distance from $v1$ to any other vertex $v \in V$.

Clique: A clique in an undirected graph is a subset of vertices such that every two vertices in the subset are connected by an edge.

Graph Component: A graph component is a subgraph in which any two vertices are connected to each other by paths.

Connected Graph: A graph is connected if there is a path connecting every pair of vertices. Therefore, a connected graph has always a single component.

Regular Graph: A graph is regular if all the vertices of G have the same degree. In particular, if the degree of each vertex is k, G is said to be k-regular.

Complete Graph: A complete graph is a graph in which every two distinct vertices are joined by exactly one edge. The complete graph with n vertices is denoted by K_n.

Tree Graph: A tree is a connected graph which has no cycles.

Spanning Tree: If G is a connected graph, the spanning tree in G is a subgraph of G which includes every vertex of G and is also a tree graph.

2.1.2 Complex Network Models

We here give a brief presentation of the main models of complex networks.

Random Networks (ERDÖS; RÉNYI, 1959), the authors proposed a model capable of generating random networks. Considering initially a network composed by n vertices totally disconnected ($m = 0$), the generation process consists in creating random edges given an arbitrary probability $p \in [0,1]$. This model disregards loops and multiple edges. At the end of the process, the distribution of the number of connections (degree) follows a Poisson distribution. This is the first large scale network model and has been extensively studied in the literature. However, most of the real networks do not follow this model because the connections are not randomly formed.

Small-World Networks (WATTS; STROGATZ, 1998), the authors proposed a model that generates networks with some properties observed in social networks. One of these properties is the transitivity. Given three people A, B and C, considering that A knows B and B knows C, so there is a high probability of A also knowing C. This property makes the network to have subgroups of highly connected vertices (communities) and a high clustering coefficient (c). Another property is the small-world effect. In social networks, the average shortest paths (l) among all pairs of people is small. It happens because people are separated from each other by a small number of people. To generate networks with this properties, the mode initially considers a regular network where the vertices are positioned under a ring equally spaced. Each vertex is connected to the nearest k neighbors on each side of the ring, that is, the $2k$ neighbors. Then each edge is randomly reconnected to two other pairs of any vertices with probability $p \in [0,1]$. When p is small, a network is generated with high c and low l.

Scale-Free Networks (BARABÁSI; ALBERT, 1999), the authors demonstrated that several real networks have a topology formed by many vertices with few connections and few vertices with many connections (hubs).They also observed that the degree distribution of these networks follows a power law $p(k) \sim k^{-\lambda}$, where λ is a constant that varies between 2 and 3. In random networks, the degree distribution follows a Poisson and the mean degree $\langle k \rangle$ indicates the scale of the network, since a randomly selected vertex will have degree $k = \langle k \rangle \pm \langle k \rangle^{\frac{1}{2}}$. On the other hand, in networks studied by Barabasi and Albert, the mean degree can not be used as a scale to predict the degree of a vertex taken at random. Therefore, these networks are called Scale-Free networks.

2.1.3 Complex Network Measures

There are many complex network measures and each one describes the network in a particular way. Therefore, the measures are important tools to understand the underlying network from different point of view. The following measures are relevant to this work:

Average degree $\langle k \rangle$: Average number of the degrees of all the vertices in the network. Considering A the adjacency matrix of the network, the degree of a vertex i can be defined as $k_i = \sum_{j=1}^{n} A_{ij}$. The average degree of the network is defined as:

$$\langle k \rangle = \frac{1}{n} \sum_{j=1}^{n} A_{ij} \tag{2.1}$$

Degree distribution $p(k)$: Distribution of degree probability of all vertices of the network. Considering that there are n_k vertices of degree k, then $p(k) = n_k/n$. Considering $p(k)$ of all vertices ($k = \{0, \ldots, n-1\}$), then we have the degree distribution. This distribution brings global information about the connection of the vertices.

Average shortest paths $\langle l \rangle$: The mean of the shortest paths between all the vertex pairs is defined by:

$$\langle l \rangle = \frac{1}{n(n-1)} \sum_{i \neq j}^{n} d(i,j) \tag{2.2}$$

where $d(i,j)$ is the distance between vertices i and j. It is an important measure of information flow in the network. For example, considering that we want to transmit information from a vertex i to a randomly chosen vertex j, if $\langle l \rangle$ is small, then $d(i,j)$ is also small in the average. Therefore, the information will be transmitted quickly in average.

Clustering coefficient c: This measure is used to verify the existence of highly connected subgroups in the network. More specifically, c measures the transitivity of the network, i.e., the fraction of the cycles of size 3 (N_Δ) among all paths of size 3 (N_3):

$$c = \frac{3N_\Delta}{N3} \tag{2.3}$$

Efficiency:The efficiency of a network is a measure of how efficiently it exchanges information. It is inversely proportional to the distance:

$$\varepsilon_{i,j} = \frac{1}{d_{i,j}}, \tag{2.4}$$

where $\varepsilon_{i,j}$ is the pairwise efficiency between nodes i and j and $d_{i,j}$ is the distance between nodes i and j.

The average efficiency of a network is defined as the average over the pairwise efficiencies:

$$E = \frac{1}{N(N-1)} \sum_{i \neq j \in V} \frac{1}{d_{i,j}}, \tag{2.5}$$

Betweenness: The betweenness measure of a vertex v evaluates the fraction of shortest paths between all pairs of vertices s and t that passes through v, denoted as η_{st}^{v}, over the total number of shortest paths between s and t, denoted as η_{st}. Then, the betweenness of vertex x_v is given by:

$$B_v = \sum_{s \neq v \in V} \sum_{t \neq v \in V} \frac{\eta_{st}^{v}}{\eta_{st}}, \tag{2.6}$$

Communicability (ESTRADA; HATANO, 2008): Communicability is defined for every pair of vertices $p \in V$ and $q \in V$. It quantifies how easily vertex p can communicate with q by means of a combination of shortest paths and random walks with varying lengths. Mathematically, the communicability of vertex p to q is given by:

$$\mathscr{C}_{pq} = \sum_{k=0}^{\infty} \frac{(\mathbf{A}^k)_{pq}}{k!}$$

where $\mathscr{C}_{pq}$ is the communicability between vertices p and q, $\mathbf{A}$ is the adjacency matrix of the network, and $(\mathbf{A}^k)_{pq}$ is the (p,q)-th element of the k-th power of matrix $\mathbf{A}$.

PageRank: PageRank is a well-known measure used by Google to rank web pages. It is supposed to simulate the behavior of a user browsing the Web. Most of the time, the user visits pages just by surfing, i.e., by clicking on hyperlinks of the page she is on. Another manner is to jump to another page by typing its URL on the browser, or going to a bookmark, etc. It can be described by the following formula (LI, 2002):

$$p(i) = \frac{q}{V} + (1-q) \sum_{j \in \mathscr{V} : j \to i} \frac{p(j)}{k_j^{(\mathrm{out})}}, \tag{2.7}$$

where V is the number of vertices of the graph, $p(i)$ is the PageRank value of vertex i, $k_j^{(\text{out})}$ the out-degree of vertex j, and the sum runs over the vertices pointing toward (direct connection to) i. The damping factor $q \in [0,1]$ is a probability that weighs the mixture between the realized random walk and random jumps.

In modeling high-level data classification techniques, the above network measures have been used to characterize data patterns in the format of networks.

2.2 High Level Classification with Complex Networks

Traditional data classification algorithms consider only low-level features (physical features) to build the classifier, but human brain is able to perform both low and high-level learning. One study (Silva; Zhao, 2012) proposes a hybrid classification technique with two components: A low-level term and a high-level classification term. The former can be any traditional technique, such as SVM, MLP, CNN. The latter seeks to construct a network from the input data set to find out the existing data patterns (high-level features). Then, the technique is tested on handwritten digits recognition. Specifically, the hybrid classification technique is represented by the following equation:

$$M_x^c = (1-\lambda)C_x^c + \lambda H_x^c \tag{2.8}$$

where M_x^c represents the combined classification pertinence level, $M_x^c \in [0,1]$, evaluated on instance x for the class c. If M_x^c is a high value (near 1), it is most probably that x is classified to class c; on the other hand, if M_x^c is a small value (near 0), it is most probably to be classified to other classes instead of c. $C_x^c \in [0,1]$ and $H_x^c \in [0,1]$ are pertinence levels returned by the low-level and high-level classifiers, respectively. $\lambda \in [0,1]$ is an user-controllable parameter to balance the contribution of the low and high classifiers to the final decision on instance x.

The estimated label $\hat{y}_x$ of the test instance x is given by:

$$\hat{y}_x = \arg\max_{c \in \mathscr{L}} M_x^{(c)}. \tag{2.9}$$

The low-level term in Eq. (2.8) can be any traditional classification technique. The high-level classification consists of the following steps:

- Network construction: A network is constructed from the data of each training class. In this process, each data instance is a node and the connection between a pair of nodes is determined a combined KNN and ε rule. KNN means a node i is connected to node j if j is within the K most similar data instances of i; the ε role means that a node i will establish connections to all the nodes (data instances) with the similarity beyond a predefined threshold ε. Note that the ε rule is used for dense regions, while the kNN is employed for sparse regions. The sparseness can be determined by the number data instances around the node under consideration.

- Calculating the network measures: For each constructed network, we calculate a set of predefined measures. These measures will be used to represent the data pattern of each class.
- Inserting a test instance: We insert a test instance using the same network construction rule presented in the first item.
- Calculating the network measures again: For each constructed network, we calculate the same set of predefined measures again after the insertion of the test instance.
- Checking the perturbation level. We check the perturbation level by calculating the variance of the measures before and after the insertion of the test instance.
- Classification. Finally, the test instance is classified to the class c, where its insertion generates the smallest perturbation. It means that the test instance conforms the pattern of class c.

Computer simulations shows a meaningful improvement in terms of classification accuracy, demonstrating the previous hypothesis of mixing low and high-level features. In a prior study (Silva; Zhao, 2012), the authors performed various simulations on artificial datasets with visual patterns for classification tasks showing the high importance of high-level features when the complexity increases. Still in that work, tests with many well-know datasets like iris, wine, glass and others show how the high-level features extracted by the proposed method can improve the performance of traditional well-known classification techniques, such as Bayesian Networks, Weighted kNN, Fuzzy C4.5, Multilayer Perceptron and Fuzzy M-SVM. In that work, the network measures of assortativity, clustering coefficient, average degree are used to characterize data patterns.

Later, in the following works (Silva; Zhao, 2013; SILVA; ZHAO, 2015b), the authors proposed to use the average transient lengths and cycle lengths of the tourist walks on the constructed networks as a network measure. The experiments are performed over two scenarios: networkless data and network data, with the same datasets: iris, wine, etc. The low-level classification techniques are: Bayesian Networks, Weighted k-NN, Multilayer Perceptron and Fuzzy M-SVM. The proposed method has also been applied to hand-written digit recognition.

Later, explorations were conducted by researchers who proposed the utilization of entropy as a network measure (Neto; Zhao, 2013). The analysis covered datasets such as iris and seeds, comparing the performance against Multilayer Perceptron and SVM on an artificial dataset featuring a visual square pattern, showcasing superior results.

In another study (Carneiro *et al.*, 2016), the proposal involved employing SL-PSO (Social Learning Particle Swarm Optimization) to determine the optimal combination of parameters (λ) for the hybrid model.

All the aforementioned investigations adhere to the hybrid scheme that combines both low and high-level classification techniques. In a different work (Carneiro; Zhao, 2013), the authors introduced a high-level model utilizing a novel complex network measure known as component efficiency. Comparative evaluations were conducted against Decision Tree and Support Vector Machine using datasets like iris, ecoli, opt. digits, and SpectFHeart, yielding favorable numerical outcomes.

Subsequently, in a different context (Colliri *et al.*, 2018), the authors presented a model exclusively featuring a high-level term. This study applied various complex network measurements (average degree, assortativity, average local clustering coefficient, transitivity, average shortest path, second moment of degree distribution) to assess the perturbation impact caused by the insertion of a testing instance. The model was tested across eight artificial datasets and nine well-known real-world datasets, demonstrating promising results.

Meanwhile, in a distinct application (Carneiro *et al.*, 2017), the authors delved into high-level classification for Semantic Role Labeling (SRL). This task involves identifying and classifying arguments, indicating the semantic relationship between an event and its participants. The experiment used PropBank-br (Brazilian Portuguese Corpus), which is notable for its imbalanced distributions and scarcity of annotated data. Results showcased the effectiveness of the proposed approach in such tasks.

In another contribution (CARNEIRO; ZHAO, 2018b), the authors introduced a data organizational structure for classification, employing complex networks constructed from training data. The classification of an unlabeled instance is based on the importance concept characterized by Google's PageRank measure of the data networks. This work also proposed a measure named spatio-structural differential efficiency, which combines physical and topological features of the input data. Extensive experiments demonstrated the promising predictive performance of the proposed technique, particularly in the detection of heart abnormalities.

Subsequent investigations (CARNEIRO *et al.*, 2019) delved into structural optimization in network-based data classification, applying a bio-inspired optimization framework to build an optimal network from the input dataset for high-level classification.

In a more recent study (CARNEIRO *et al.*, 2023), researchers scrutinized the performance of five network centrality measures (pagerank, betweenness, closeness, degree, and shortest path length) in a high-level data classification scheme. These measures were assessed across various real-world datasets in terms of predictive capability and robustness, with PageRank and degree often achieving superior results and statistically outperforming other measures in terms of predictive robustness.

Finally, the proposed high-level techniques found application in diverse domains, including EEG signal analysis (CARNEIRO *et al.*, 2023), X-ray image classification for COVID-19 diagnosis (LIU *et al.*, 2022).

CAPÍTULO

3

THE PROPOSED NETWORK-BASED HIGH-LEVEL CLASSIFICATION METHODS

This chapter describes the ideas and the evolution of the proposal to reach the objectives of this thesis. Firstly, in order to find out a more sensitive metric to distinguish the pattern formation for a test data sample, we propose two approaches: 1) We use Ant Colony Optimization algorithm to solve Travelling Salesman Problem and find the shortest path which joins all the nodes of the network to form a cycle. 2) We apply the Minimum Spanning Tree to join all the nodes. In both cases, we analyze the network structures before and after the insertion of a test sample to determine its class. Secondly, we purpose to use optimization technique to build an optimal network for each class of the training data. Thirdly, we present the methodology to apply the developed high-level classification technique to chest X-ray image classification and Snore classification.

3.1 Finding Out the More Sensitive Network Measures for High-Level Classification

In high-level classification, we first construct a network for each class of the training data. Then we insert test sample to each of the networks. In order to determine the class of the test sample, we calculate some pre-defined network measures before and after its insertion. The test sample will be classified to that class, which its insertion causes the smallest variation of the network measures. It means that the test sample conforms the pattern formed by that class. Since the networks constructed from the training data set can be large and each time only one test instance is inserted to a class network, the variations of the network measures can be tiny values, which make the classification confusing, i.e., we are not sure whether the test instance conforms the pattern formation to a given class. For this purpose, we propose two measures to check the pattern conformation when we classify a test sample: 1) The first measure is the shortest path, which includes all the nodes of the network to form a minimum weight cycle. We propose to determine such a measure using the Ant Colony Optimization Algorithm (DORIGO, 1992). Such a determination likes to solve a Travelling Salesman Problem (TSP) (GRAHAM; HELL, 1985) in the constructed training network. 2) We propose to use the Minimum Spanning Tree (MST) of the constructed network as a network measure for classification.

Both the Shortest Cycle and the MST measures characterize minimally connected network structures, i.e., a small portion of the original network edges is involved. At the same time, many other network measures are calculated by considering all the edges of the network. As a result, the insertion of a test sample will have a bigger change using the two proposed measures. Simulation results show their effectiveness in increasing the sensitivity.

3.1.1 The Shortest Cycle Measure Determined by Ant Colony Optimization Algorithm

In this section, we follow the idea of Dorigo to solve the famous Travelling Salesman Problem (TSP) applying Ant Colony Algorithm in the constructed network from the training data. The result is a minimum weight cycle, corresponding to the shortest distance to be travelled to visit every node only one time.

For this reason, we first announce the Travelling Salesman Problem (TSP). The TSP is around the next question: "Given a list of cities and the distances between each pair of cities, which is the shortest possible route that visits each city exactly once and returns to the origin city?"

TSP can be described as a complete graph $G = (V, E)$ where $V = 0, \dots, n$ is a node set, and $E = (i, j)\{(i, j) \in V\}$ is an edge set. Each element of the set V represents a city. Each edge (v_i, v_j) of the set E is associated to a distance d_{ij} from city i to city j. The TSP is only sequential access of a set of cities, and finally returns to the original departure city. The goal is to find out a tourist to visit each city only once minimizing the total travelling distance.

Now we apply the Ant Colony algorithm to solve the TSP in the network. Goss (GOSS *et al.*, 1989) observed the way how ants search food, he found three main steps: initially, each ant randomly searches a path; after some time, many ants follow one unique path; they have a indirect communication (pheromone) to tell each other which is the best path. One summary of the experiment performed by Goss is presented in Fig. 1.

Dorigo(DORIGO, 1992) realized the ants' behaviour to find the path for food and proposed the following algorithm:

- Each artificial ant starts in one node of a graph G.
- Each ant finds a path (build a solution) following the pheromone deposited in each edge

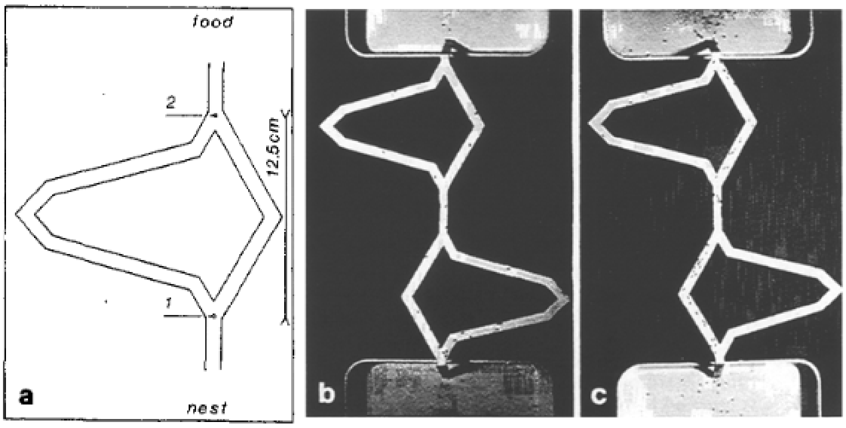


Fig. 1. A colony of *I. humilis* selecting the short branches on both modules of the bridge; a) one module of the bridge, b) and c): photos taken 4 and 8 min after placement of the bridge

Figure 1 – Illustration (GOSS *et al.*, 1989) of the results of Goss' experiment.

using Eq. 3.1.

$$p_{i,j}^{k} = \begin{cases} \frac{\tau_{i,j}^{\alpha}(t)\cdot\eta_{i,j}^{\beta}}{\sum_{j\in J^k}\tau_{i,j}^{\alpha}(t)\cdot\eta_{i,j}^{\beta}}, & if\, j \in J^k \\ 0, & otherwise \end{cases} \tag{3.1}$$

where $p_{i,j}^{k}(t)$ is the probability of ant k follows the edge (i,j) in the iteration t, J^k is the list of edges not visited yet, $\tau_{i,j}$ is the quantity of pheromone in the edge (i,j), $\eta_{i,j}$ is the information of quality of this edge, here, $\eta_{i,j} = \frac{1}{d_{i,j}}$, which is the inverse of the distance between node i and j, and α,β are parameters which represent the influence of τ,η, respectively.

- After the creation of every path, the quantity of pheromone is modified according the quality of the generated solution using Eq. 3.2 and 3.3:

$$\tau_{i,j}(t+1) = (1-\rho)\cdot\tau_{i,j}(t) + \rho\cdot\delta\tau_{i,j} \tag{3.2}$$

$$\delta\tau_{i,j} = \begin{cases} \frac{1}{f(S)}, & if(i,j) \in S \\ 0, & otherwise \end{cases} \tag{3.3}$$

where ρ is the pheromone evaporation rate parameter, $\delta\tau_{i,j}$ is the current quantity of pheromone deposited in the edge (i,j) and $f(S)$ is the total cost of the solution S, here, it is the sum of the distances along the path in the network found so far.

At each iteration, the quantity of pheromone $\tau_{i,j}$ is modified by Eq. (3.2) and Eq. (3.3) by increasing or decreasing the level of pheromone corresponding to move to "good"or "bad"solutions, respectively. Then, at the next iteration, a new path may be chosen by Eq. 3.1, which approaches further to the "good"solutions. After a number of iterations, the process converges to an optimal solution, a minimum weight (distance) cycle of the TSP.

3.1.2 The Minimum Spanning Tree Measure

Given an undirected graph $G = (V, E)$, where V are the vertices and E are the edges which connect the vertices, i.e. $(u, v) \in E$ and there are a associated cost for each edge $w(u, v)$, a path, which connects all the vertices and the total weight is minimum, with no presence of cycle, is called as Minimum Spanning Tree (MST) (CORMEN *et al.*, 2001).

The proposed measure is the sum of the weights om the edges of the MST:

$$w(T) = \sum_{(u,v)\in T} w(u, v) \tag{3.4}$$

Two well-known algorithms for finding the MST in a graph are Kruskal's and Prim's algorithms:

1. **Kruskal's Algorithm:** Kruskal's algorithm (KRUSKAL, 1956) builds the MST by iteratively adding the shortest edge that does not form a cycle with the edges already selected. It does this until all vertices are connected. The algorithm employs a disjoint-set data structure to efficiently determine whether adding an edge creates a cycle.

2. **Prim's Algorithm:** Prim's algorithm (PRIM, 1957) starts with an arbitrary vertex and incrementally grows the MST by adding the shortest edge that connects a vertex in the MST to a vertex outside of it. It maintains a priority queue to efficiently select the shortest edge at each step.

3.1.3 Training and Classification Using the Proposed Measures

Here, we present the main steps of the high-level classification using the two proposed measures.

3.1.3.1 Building the Complex Network

This subsection explains how to construct the complex networks from the input data. This step is important because the complex networks will be used to extract internal patterns of the data and to check if the structure can represent appropriately them for a testing step. For the supervised learning purpose, a split step of the dataset is done, we use 75 % of the input dataset as training data, to build complex networks. Other 25 % data samples are use for testing.

A network is constructed for each class of the training data. Each data sample is a node and an edge between each pair of nodes is created with a weight determined by the Euclidean distance between the two samples. Finally, we get a set of weighted networks and each of them represents a class of the training data.

3.1.3.2 Classification by Analyzing the Variation of the Network Measures

The objective is to measure how the purposed measures changes when a test sample is inserted to each of the constructed networks. For this purpose, the shortest cycle measure extracted by Ant Colony Algorithm and the MST measure are used. Specifically, it checks the perturbation by performing a number of experiments to get the median of this behavior and check the difference before and after the insertion of the test sample to the network of each class.

Following the assumption, the elements of a class form a pattern represented by network measures. Then, if a test sample is inserted and it does not belong to the class, the internal structure of the corresponding network will change meaningfully. In contrast, it will preserve the measure which describes the network.

- Firstly, calculate the Shortest Cycle or MST from each network of a data class.
- Insert a test sample from the $0.25{*}size\_dataset$ test set to each network.
- Calculate the MST or the Shortest Cycle after each insertion.
- Check the difference in terms of the MST measure or the Shortest Cycle measure before and after the insertion.
- The test sample is classified to the class with the smallest variation of the MST or the Shortest Cycle measure.

3.1.3.3 Network-Based High-Level Classification - Algorithm

The algorithms of classification using complex network following our proposal (SAIRE; ZHAO, 2023) is presented in Alg. 1.

3.2 Building Optimal Complex Networks

Considering the work of (ZHU, 2005), the representation of a dataset using network is the most important step of network-based learning. Here, a network is built from a set of nodes which represent each instance of the dataset, and the edges are the distance or similarity between the nodes.

Fig. 2 presents some basic method to build networks, which have been applied in the original high-level classification techniques. Now, v_i, v_j, v_l represent the instances or samples x_i, x_j, x_l from the dataset. Then, the network building process considers the neighborhood relationships and is performed by ε and kNN rules. In a ε-network, the connection is made if the distance between two samples is minor than ε, see Fig. 2 (b). A similar approach considers a fully connected network, where the edges get the weight using a similarity function, i.e. Gaussian function. This kind of learning through a differentiable function of the weights (LUXBURG,

Algoritmo 1 – Classification Algorithm with Complex Networks

Building Complex Networks

- Firstly, generate a fully connected and weighted network for each class in the dataset. Each data sample is a node and the weight between a pair of nodes is the Euclidean distance between them.
- Calculate the MST for each network and store in a list. Then, calculate the MST-based measure using Eq. (3.4).

Using Complex Networks for Classification

- Insert the element/sample to each complex network.
- Calculate again the MST-based measure for each network after insertion.
- Calculate the variation of the MST-based measure before and after the insertion for each network.
- Classify the test sample to the class, in which its insertion generates the smallest variation.

2007). In KNN networks, the K value represents the number of neighbors will be connected to a given node, see Fig. 2 (c). Besides, the relationships of similarity are asymmetric, therefore, it is not suitable to be used for constructing undirected graph. Other sophisticate methods are: b-matching (JEBARA; WANG; CHANG, 2009), linear neighborhood (Wang; Zhang, 2008) and methods based on single linkage (CUPERTINO; HUERTAS; ZHAO, 2013).

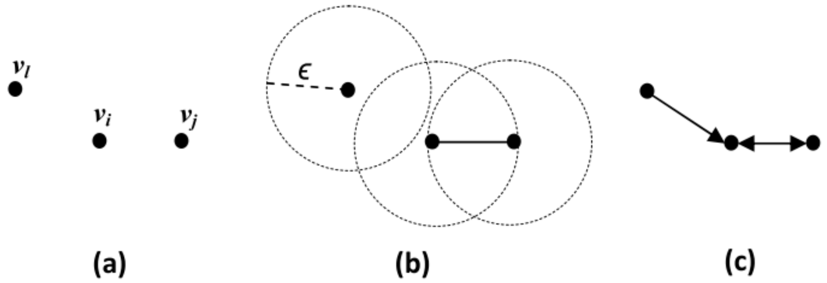


Figure 2 – Examples of graph building from vector-base data. a) initial data samples, b) using radius ε c) using K-NN

However, when calculate the distance or similarity between a pair of data samples (nodes), all the above mentioned methods consider that all the features in the feature vector of a node have the same level of importance. Really, different features have different relevance to characterize the node.

Therefore, in this thesis, we propose to construct the optimal network considering that each feature has a different level of contribution or has a different importance to represent the data pattern. The proposal includes to consider a weight for each feature of a node representing the level of importance, following an optimization approach. The objective is to find out a set of weights for each node to maximize the precision of classification using only the training set. The optimization is carried out using the Genetic Algorithms.

Our method is inspired by the structural optimization of complex networks presented in (CANCHO, 2001), in which an evolutionary algorithm is applied to generate an optimal network by minimizing link density and average distance, following the next equation:

$$E(\lambda) = \lambda * d + (1 - \lambda) * \rho \tag{3.5}$$

where d and ρ are average distance of the nodes and the link density of the network, respectively. λ is a control parameter of the linear combination.

In this work, we will construct network for each class of training set by optimizing some criteria like Eq. 3.5. In this way, the global feature of the network is also considered at the network construction phase. The insertion of a test sample follows the same optimization process.

3.2.1 Network-Based High-Level Classification - Algorithm 2

One of the important questions in network-based learning is how suitable is the constructed network from the vector-based data set for generating good learning results. In high-level classification, we are not sure whether the constructed network can really represent the data pattern by traditional network building methods, such as the above mentioned methods.

In this work, we will treat this problem. Specifically, we propose to use Genetic Algorithm for feature selection. At the end, each feature of each data sample receives a weight to represent the level of relevance in the network construction process. The proposed technique (SAIRE; ZHAO, 2023) is described by Algorithm 2.

The search of the weights related to feature importance will be done using using Alg. 2. According to PYGAD (GAD, 2021) (python package with Genetic Algorithms), the following parameter values are used in our study:

- $number\_generation$: 50, $size\_population$: 20
- $crossover\_rate$: 0.8, $mutation\_rate$: 0.2

3.3 Methods for the Applications

This section presents the method to solve real-world problems by applying the proposed high-level classification technique. Specifically, we consider the tasks of classification of chest X-ray images for COVID-19 diagnosis and classification of snore audio.

Algoritmo 2 – Optimal Network Construction Using Genetic Algorithms

Building Complex Networks

- Consider a training data set $X_c = \{\mathbf{x}_1^c, \mathbf{x}_2^c, ..., \mathbf{x}_{n_c}^c\}$, where n_c is the number of data samples in class c, $c = 1, 2, .., l$ is the class index and the number of classes is l. The ith data sample of class c is a m-dimensional vector $\mathbf{x}_i^c = [{x_{i1}}^c, {x_{i2}}^c, ..., {x_{im}}^c]^T$, where ${x_{ij}}^c$ is the jth attribute of the ith data sample of class c. In order to make feature selection and, consequently, construct the optimal network for each class of the training set, we put a different weight to each feature of the data sample, i.e., $\mathbf{x}_i^c = [w_1 {x_{i1}}^c, w_2 {x_{i2}}^c, ..., w_m {x_{im}}^c]^T$, where w_j is the weight of the jth data element ${x_{ij}}^c$. This step is similar to Alg. 1, but considering a weight vector representing the relevance of the feature. In the previous algorithms, all the weight values are 1.

Finding the relevance of features by Genetic Algorithm

- Divide the training set into two subsets, one continues be the training set $X_{training}$ and another is the validation set $X_{validation}$.

- Randomly generate N weight vectors, $W = \{W_1, W_2, ..., W_N\}$, where each element of the weight vector W_i is $w_j \in [0, 1]$. The set of weight vectors W is the initial population of the Genetic Algorithm.

- Use each weight vector W_i together with the feature vectors of the training set to construct a network for each class of the data using Alg. 1.

- Perform the classification task on the validation set $X_{validation}$ using Alg. 1.

- The fitness function is the classification accuracy of the high-level technique on the validation set $X_{validation}$.

  WHILE (stopping criteria not met)

  - Randomly select parents for reproduction from the individuals (weight vectors), which lead to the highest classification accuracy.
  - Perform crossover to create offspring.
  - Apply mutation to some offspring.
  - Use each weight vector W_i of the offspring together with the feature vectors of the training set to construct a network for each class of the data using Alg. 1.
  - Perform the classification task on the validation set $X_{validation}$ using Alg. 1.
  - Evaluate the fitness of the new offspring by simply calculating the classification accuracy on the validation set.
  - Select individuals for the next generation
  - Update the population with the selected individuals

- After finish the iterations, pick the best individual (weight vector).

3.3.1 Covid-19 Classification Application

COVID-19 is a breakthrough in human history. It destroyed economies and collapses emerge countries. During its early stage, the virus had reached a high reproduction number of 4.22 in Germany and the Netherlands. In countries with poverty and weak health systems, COVID-19 caused severer problem. By consequence, many efforts were focused on finding a vaccine and automatic tools to support prognosis of the illness.

Oriented on this direction, many groups are working with tomography, x-ray images using Artificial Intelligence techniques, i.e. Deep Learning (APOSTOLOPOULOS; MPESIANA, 2020) (OZTURK *et al.*, 2020)(MAHMUD; RAHMAN; FATTAH, 2020) for diagnosis and prognosis. Deep Learning algorithms are based in Artificial Neural Networks with many layers and where each layer or groups has a function to capture a specific kind of image features. But, one of the limitations of deep learning could be the need of large quantity of labeled images for training. This can be a big problem specially at the initial stage of the pandemic, since we still have limited access to the images related to COVID-19 patients at that time.

High-level classification captures data patterns through complex networks (CARNEIRO; ZHAO, 2018) (FADAEE; HAERI, 2019) (CHIRE-SAIRE, 2020) (VILCA; ZHAO, 2020). In other words, it works at a higher abstract level than traditional classification techniques. At the same time, COVID-19 images present certain well formed patterns. Therefore, we hope high-level technique can perform well with a small training set. Previous works have already showed the strength of the approaches using network representation in comparison to classical machine learning algorithms (CARNEIRO; ZHAO, 2018) (FADAEE; HAERI, 2019) (CHIRE-SAIRE, 2020) (VILCA; ZHAO, 2020).

Although the pandemic of COVID-19 is officially over, this research is still meaningful for diagnosis and prognosis of other kinds of problems, such as pneumonia, and an eventual future pandemic.

3.3.1.1 Feature Extraction

Before applying the high-level classification technique, we need to extract features from the original images.

Frequency Histogram

Frequency histogram is calculated to get a lower dimensional representation. Then, the statistical features of the histogram are calculated, i.e., median, mean, standard deviation, kurtosis and skew. This histogram considers the three channels of classical RGB image representation.

Grey Level Co-Occurrence Matrix Features

Grey Level Co-Occurrence Matrix (GLCM) algorithm (YUNUS, 2020) is a second order statistical method used for texture feature extraction. From the co-occurrence matrix, the following features are extracted:

- $contrast : \sum_{i,j}^{levels-1} P_{i,j}(i-j)^2$
- $dissimilarity : \sum_{i,j}^{levels-1} P_{i,j}||i-j||$
- $homogenity : \sum_{i,j}^{levels-1} \frac{P_{i,j}}{1+(i-j)^2}$
- $ASM(AngularSecondMoment) : \sum_{i,j}^{levels-1} P_{i,j}^2$
- $energy : \sqrt(ASM)$
- $correlation : \sum_{i,j}^{levels-1} P_{i,j} \frac{(i-\mu_i)(j-\mu_j)}{sqrt(\sigma_i^2 \sigma_j^2)}$

These features considers 4 orientations: 0, 45, 90 and 135 degrees. Besides, a transformation from RGB representation to gray-scale is performed using the following equation:

$$Image(i,j) = 0.3 * R + 0.59 * G + 0.11 * B, \tag{3.6}$$

where RGB are the red, green, and blue channels of the image, respectively.

3.3.1.2 Classification

After feature extraction, we will build a network for each class of the image features. In this case, we use optimization technique to generate optimal networks using Algorithm 2. After that, we will insert test samples one by one. Then we calculate the variation of the measures obtained by Ant Colony Optimization and Minimum Spanning Tree before and after the insertion. A test sample is classified to that class, whose insertion causes the smallest variation. It means that the test sample conforms the pattern of that class.

3.3.2 Snore Classification Application

Having a good quality of sleep is important to maintain health body. Snoring is a breathing disorder during the stage of sleeping of people. Therefore, snoring can interfere the quality of sleep, i.e., who snores and who is next to the snorer can create mental and physical problems.

The dataset for this experiments is a set of 500 audio files of snoring and 500 of audio with no snoring (KHAN, 2019).

Feature extraction is performed using MFCC (Mel-Frequency Cepstral Coefficients), considering previous related works (SREERAM *et al.*, 2020), (RAHMANDANI; NUGROHO; SETIAWAN, 2018). A total of 20 features are extracted and two classes are considered.

Finally, classification step is performed similarly to the previous application related to Covid-19, considering minimal variation of the metric before and after the insertion of element to the constructed complex networks from the training data set.

CAPÍTULO

4

EXPERIMENTAL RESULTS

Our experiments delve into assessing the efficacy of our proposed method in capturing and characterizing changes following the introduction of a new element to the initial class structures. The evaluation spans both artificial datasets, designed to mimic various scenarios, and real-world datasets. Alongside these experiments, we scrutinize the effectiveness of an optimization approach in enhancing the construction of complex networks, unraveling potential improvements in efficiency.

In addition to the intricate evaluations, we provide insights into the performance of our proposed methods in applications dedicated to COVID-19 detection and snoring classification. These real-world applications serve as practical showcases of the adaptability and effectiveness of our high-level classification approach, further solidifying its potential across diverse domains.

4.1 Checking the Sensitivity of the Proposed Network Measures

This section delves into the experimental setup and outcomes of the initial approach employing the Ant Colony Algorithm to delineate the intricate network's internal structure. We start by scrutinizing the sensitivity of the proposed measures when confronted with the insertion of elements, whether they belong to a class or not. The experiments aim to show the algorithm's response to variations induced by the introduction of new elements within distinct class contexts.

4.1.1 Datasets

The following datasets are proposed to test the efficiency of the proposal:

- Artificial datasets: a dataset created from a normal distribution considering a mean μ and standard deviation σ. Besides, other dataset composed of random samples generated from geometric shapes.

- Real datasets: Iris and Wine dataset (DUA; GRAFF, 2019).

4.1.2 Artificial Dataset

We generate the following two kinds of artificial datasets to test the sensitivity of the proposed measure.

4.1.2.1 Artificial Dataset 1: Normal distribution

Samples are generated using normal distribution, following Eq. 4.1.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.1}$$

where:

$f(x)$ is the probability density function.
x is the value of the random variable.
μ is the mean of the distribution.
σ is the standard deviation of the distribution.

A generated dataset of two classes with normal distribution each is shown by Fig. 3, and the parameter values are:

- First class: μ = [1,1], σ = [0.5, 0.5]

- Second class: μ = [5,5], σ = [0.4, 0.4]

4.1.2.2 Artificial Dataset 2: Dataset with Geometric Patterns

Considering that high-level classification can deal with various forms of patterns, an artificial dataset considering a circle and a line are presented in Fig. 4.

- a circle with center in (5,5) and radius, r = 2.0, with sample number = 200

- a line, with x $\in$ [0,9] and a fixed y = 1, sample number = 20

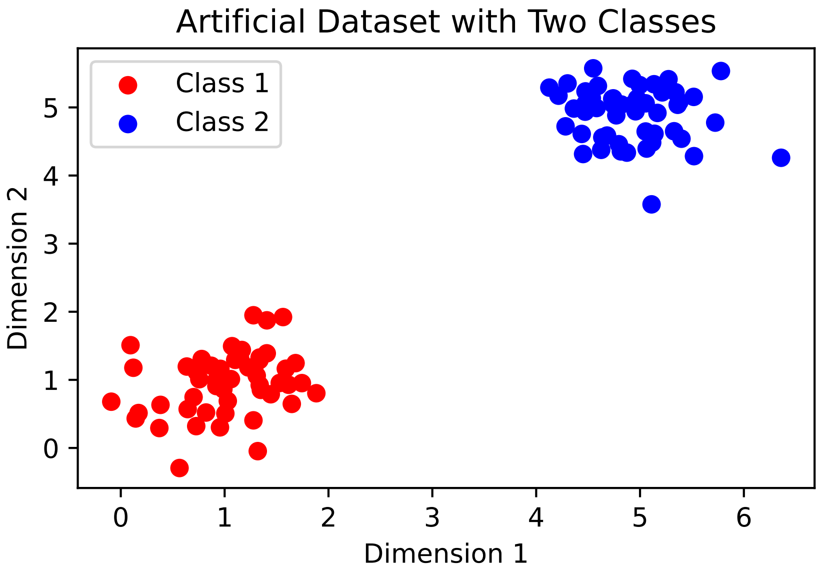


Figure 3 – Artificial dataset following a Normal Distribution with two classes, 50 samples

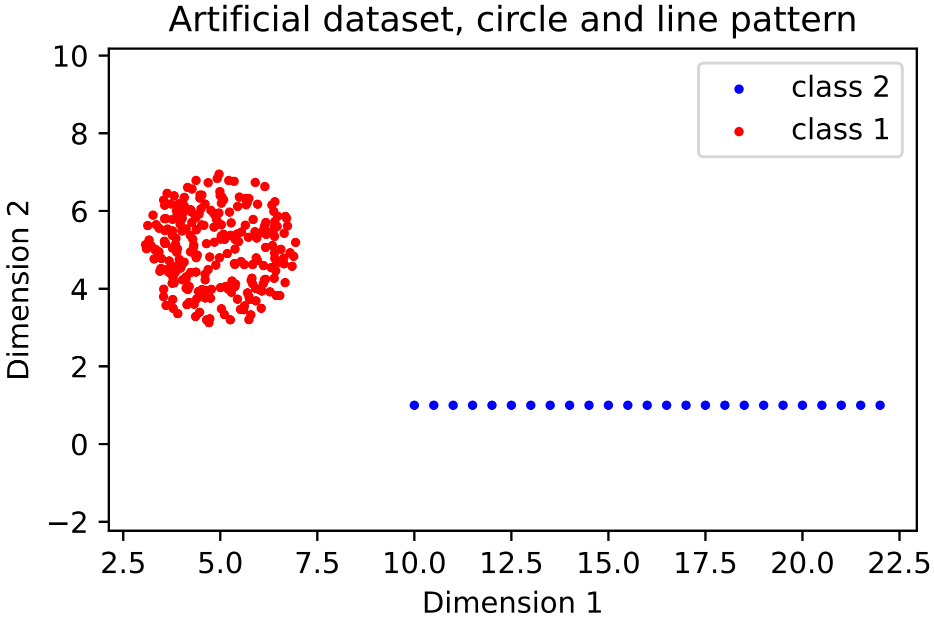


Figure 4 – Artificial dataset with two classes, circle and line forms

4.1.3 Iris and Wine Datasets

The Iris dataset (DUA; GRAFF, 2019) serves as a benchmark in machine learning, offering three distinct classes (setosa, versicolor, and virginica), each comprising 50 instances. The dataset's four features, including sepal length, sepal width, petal length, and petal width, make it a valuable resource for classification tasks and pattern recognition. Notably, the linear separability of the first class from the others adds complexity to classification challenges.

Meanwhile, the Wine dataset (DUA; GRAFF, 2019) presents chemical analyses of wines from various Italian regions, featuring 13 essential attributes. With three classes representing different wine cultivars, this dataset facilitates the exploration of wine composition through factors like alcohol content, phenols, and color intensity. Its diverse features make the Wine dataset suitable for classification tasks aiming to discern distinct wine varieties based on their chemical profiles.

4.1.4 Experimental Results

We here present the experimental results on checking the sensitivity of the proposed measures.

4.1.4.1 Experimental Results Using Artificial Datasets

The experimental results using the normal distribution dataset are presented in Figure 5. Figure 5(a) illustrates the calculated measurement values of the two classes before the insertion of the testing sample. In Figure 5(b), we observe the variations in the measurements after the insertion of an element from the same class. Here, it is possible to discern any meaningful changes. In Figure 5(c), we observe the variations after the insertion of an element that does not belong to the class. It's noteworthy that the variations are more pronounced in the third figure.

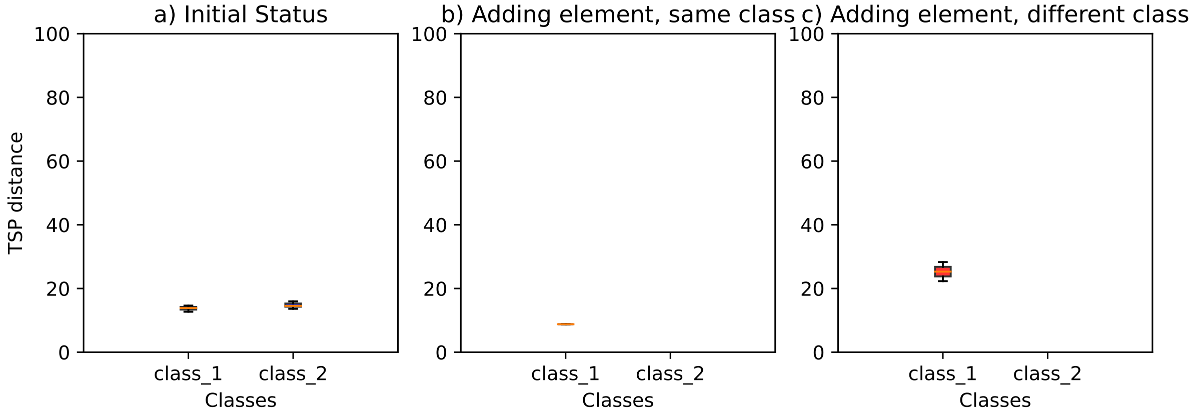


Figure 5 – Results with Artificial Dataset 1. The simulations results are obtained from 20 runs.

The simulation results performed on Artificial Dataset 2 are displayed in Figure 6. In a similar scenario to the previous dataset, we observe small variations for the insertion of an element into the class to which it belongs and high variations for the insertion of the same element into the other class to which it does not belong.

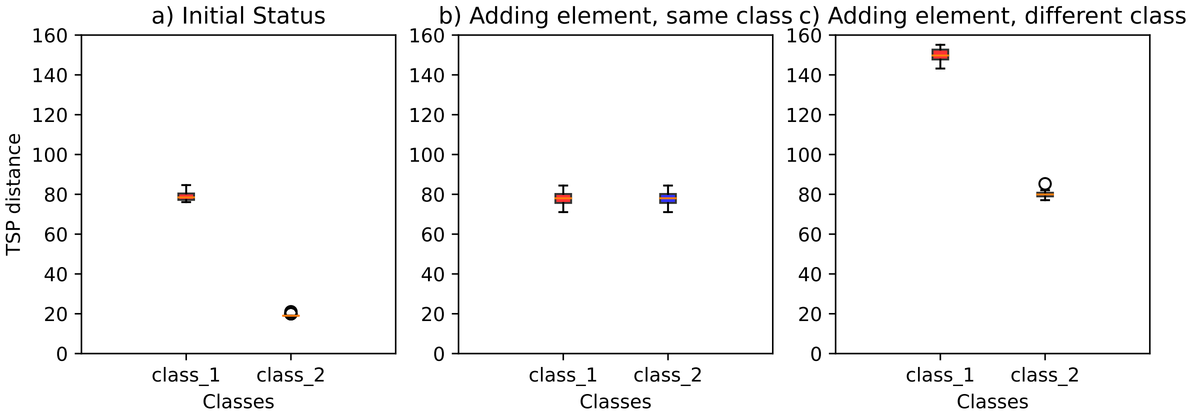


Figure 6 – Results with Artificial Dataset 2. The simulation results are obtained from 20 runs.

This pattern continues here, where we notice that the variation is higher when inserting an element from a different class and lower when inserting an element from the same class.

4.1.4.2 Experimental Results Using Real Datasets

The results of the experiments performed using the Iris dataset are presented in Figure 7. In Figure 7(a), we observe the initial status of the metric. Subsequently, Figures 7(b) and 7(c) provide insights into the results after the insertion of one element into each class.

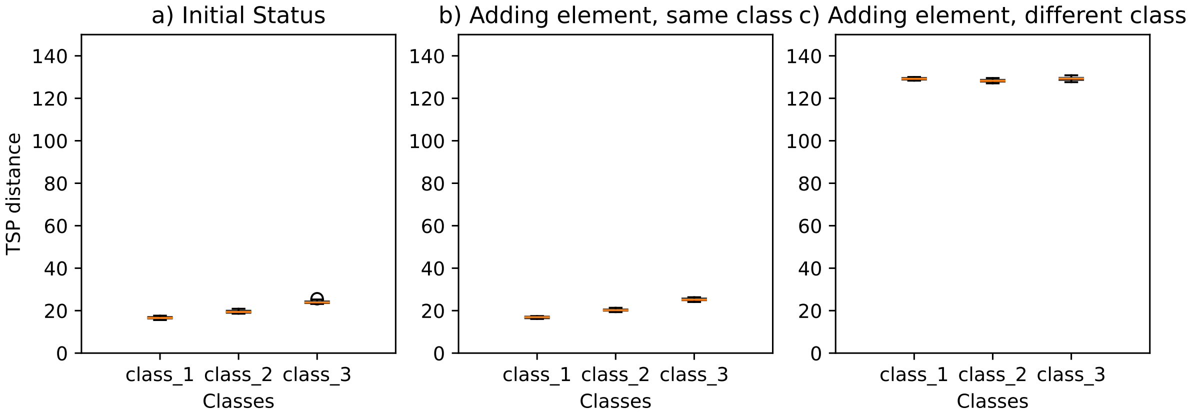


Figure 7 – Results with the Iris Dataset. The simulation results are obtained from 20 runs.

Figure 7(b) displays the metric variation after the insertion of an element into the same class, while Figure 7(c) illustrates the variation following the insertion of an element into a different class. These visualizations provide valuable insights into the dataset's behavior and the impact of class-specific insertions on the metric.

Similarly, the results of the experiments conducted with the Wine dataset are displayed in Figure 8:

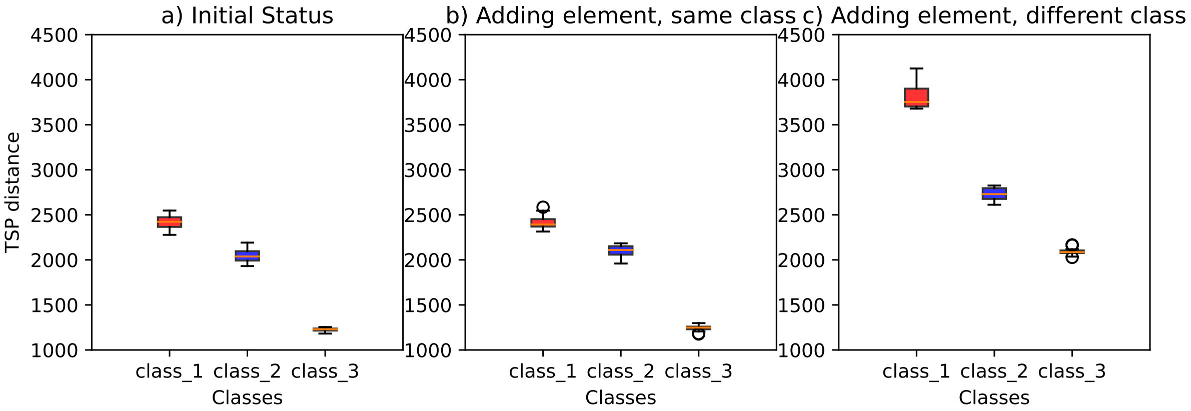


Figure 8 – Results with the Wine Dataset. The simulation results are obtained from 20 runs.

The outcomes from both real datasets underscore the high sensitivity of the proposed measures. Given the utilization of Ant Colony Optimization, it becomes essential to gain a deeper understanding of their behavior through repeated runs, in these cases, a total of 20 runs.

4.1.4.3 How about the behavior of assortativity and clustering coefficient?

In the context of all the previous datasets, we employed fully connected complex networks to represent the structural characteristics of each class. The analysis aimed to investigate whether the assortativity and clustering coefficient of these networks change before and after the insertion of elements.

Figure 9 provides a visual representation of the results obtained from the application of traditional measures to complex networks.

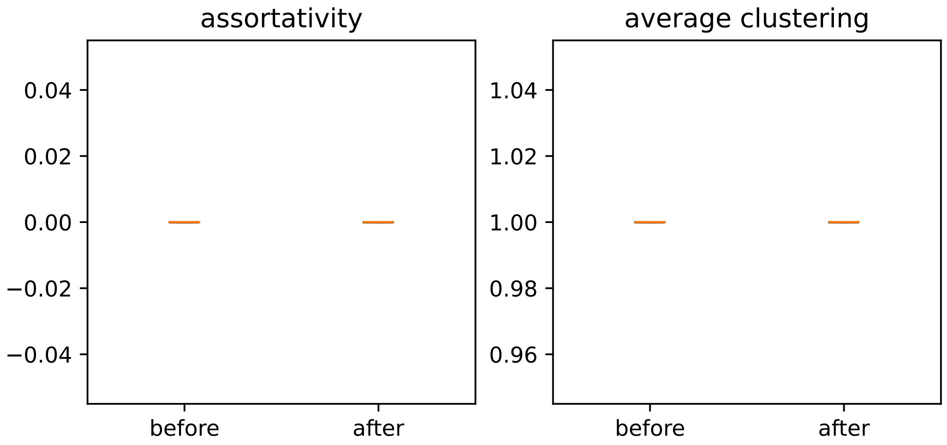


Figure 9 – Results with Traditional Measures of Complex Networks

The observations from this analysis help us assess the sensibility of network metrics when new elements are introduced into the class structure. Specifically, we focus on assortativity and clustering coefficient metrics to gain insights into how class-specific insertions may impact the network structure and we notice no impact in the metrics.

4.1.4.4 Can we find a better way to find the shortest cycle?

In our preceding experiments, we employed Ant Colony Optimization to tackle the Traveling Salesman Problem (TSP) with promising outcomes. However, it's crucial to acknowledge that the execution time of this method may escalate with an increasing number of samples per class. Additionally, due to the necessity for multiple runs to ensure reliable path estimations, the Ant Colony Optimization method exhibits high time complexity, potentially reaching $O(n^3)$ (SUDHOLT; THYSSEN, 2012).

In response to these challenges, we propose an alternative approach that leverages the Minimum Spanning Tree (MST) to represent the skeleton or pattern of the underlying network. This alternative boasts a lower time complexity of $O((V+E)\log V)$ (BARBEHENN, 1998). Consequently, we advocate for the adoption of the Minimum Spanning Tree (MST) feature, steering away from the use of the shortest cycle with Ant Colony Optimization (ACO).

As a result, we present the initial status of complex networks related to each of the previously presented datasets in Table 1, using MST as metric considering two or three classes respectively:

Dataset	Class 1	Class 2	Class 3
Artificial Dataset 1	9.8732	6.6932	x
Artificial Dataset 2	36.4401	9.5	x
Iris Dataset	10.5757	13.0241	16.1464
Wine Dataset	1274.6618	1144.5473	733.8778

Tabela 1 – Minimum Spanning Tree (MST) Base Values

After the insertion of 5 samples, belonging to both the same class and a different class, the results are presented in Fig. 10 and Fig. 11, respectively. This provides insight into the performance of the proposal using MST, indicating the sensitivity required for future classification tasks.

Finally, the simulation results of the sensitivity check using the MST measure on real datasets are shown in Fig. 12 and Fig. 13. Upon analyzing these results, it becomes evident that the MST measure is highly sensitive to the insertion of a test sample into a class to which it does not belong. In such cases, the variation in the MST measure is significant. Conversely, when a test sample is inserted into the class to which it belongs, the variation is consistently small.

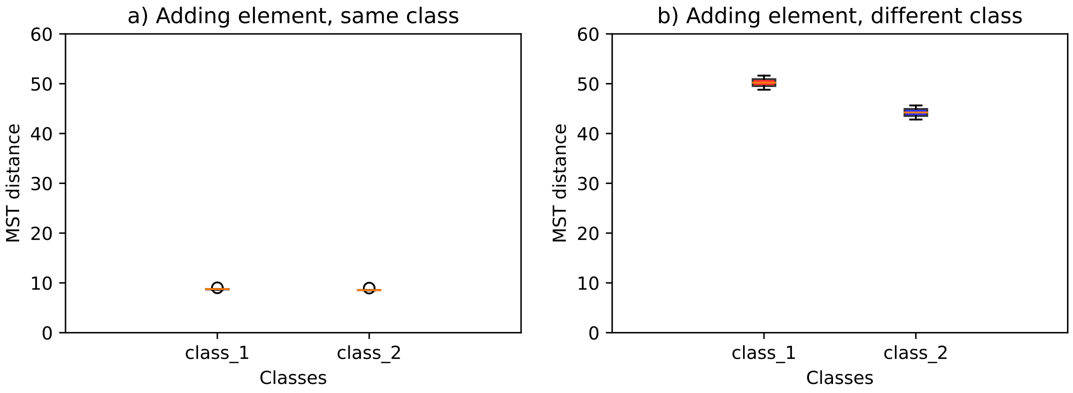


Figure 10 – Results of Artificial Dataset with Normal Distribution

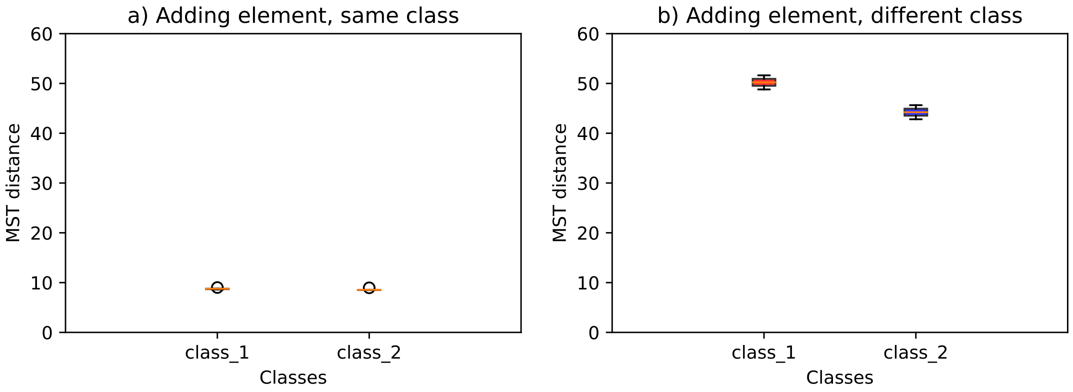


Figure 11 – Results of Artificial Dataset with Circle and Line Pattern

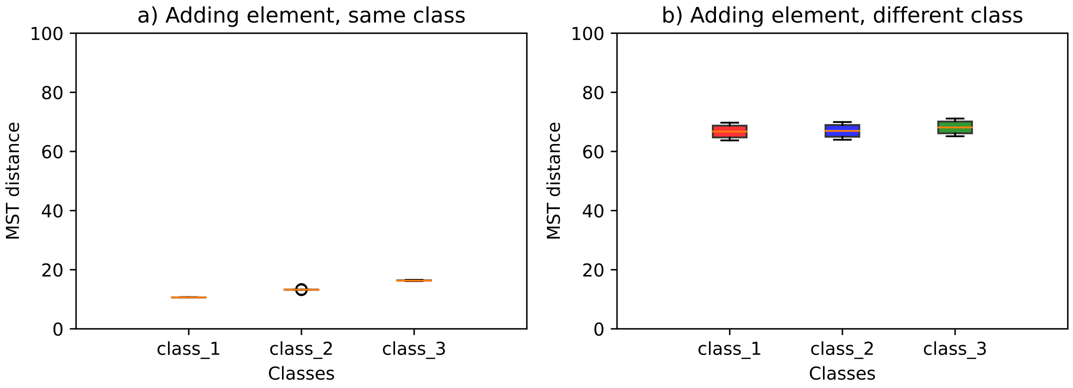


Figure 12 – Results of Iris Dataset

Due to its high sensitivity, efficiency and time complexity found in the experiments, the MST measure is used to perform all the classification tasks in this work.

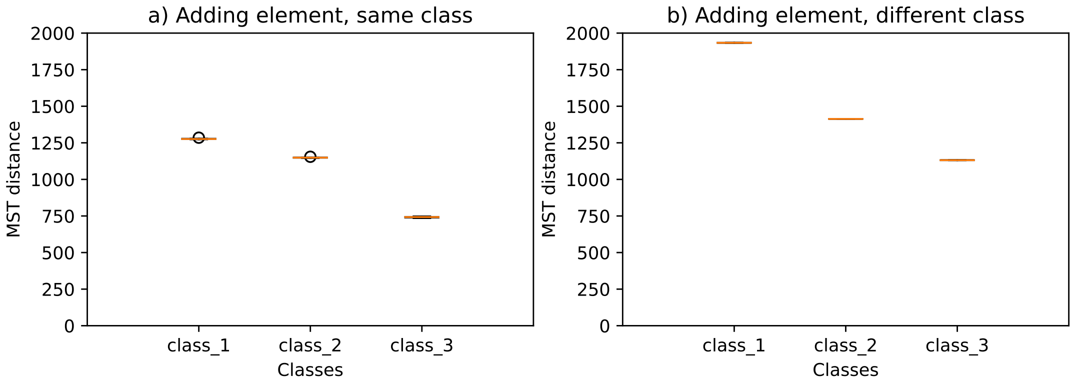


Figure 13 – Results of Wine Dataset

4.2 The Proposed High-Level Classification Method Without Optimization

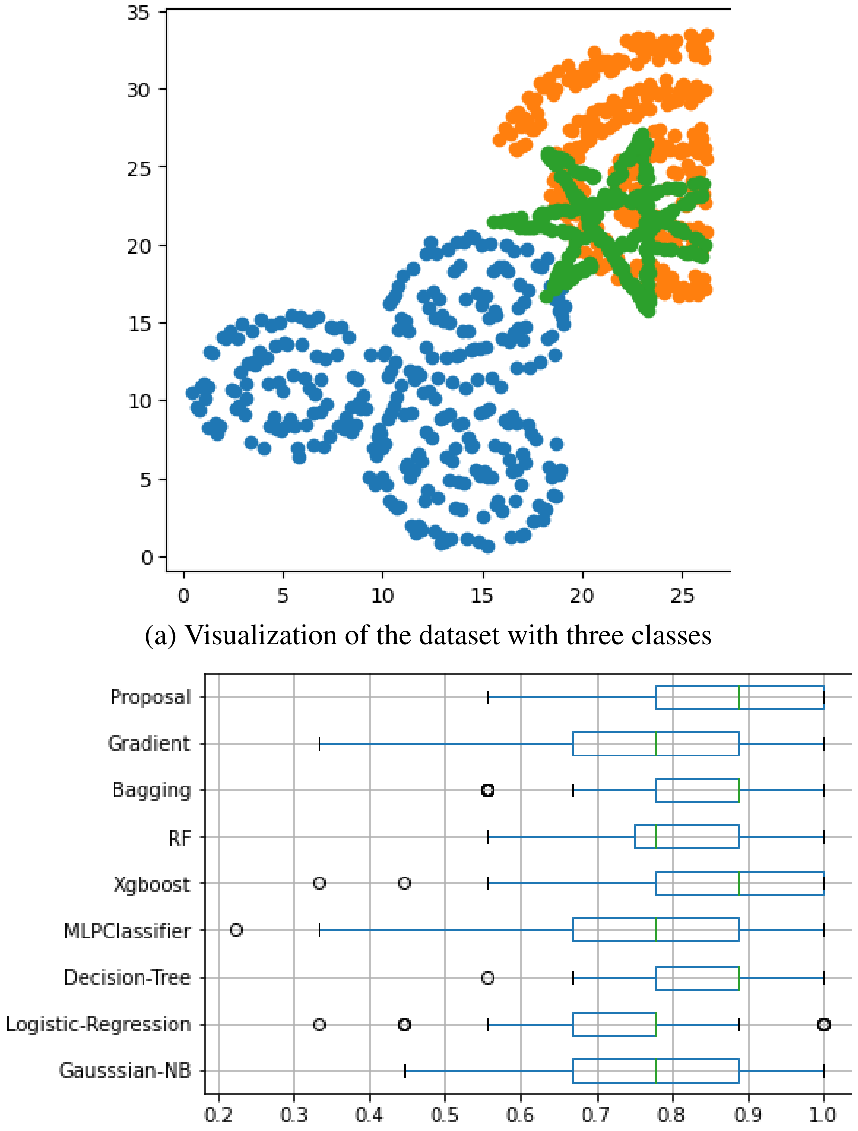
This section delves into the comprehensive analysis of experimental results, showcasing the performance of the proposed high-level classification method without optimization, delineated in Algorithm 1. The datasets employed in these experiments are meticulously crafted, encompassing a diverse array of geometrical structures. To elevate the complexity of the evaluation, intentional overlapping is introduced, posing a significant challenge to the efficacy of the proposed method.

A comparative examination is conducted, juxtaposing the outcomes derived from the proposed method against those emanating from classical machine learning algorithms integrated into Scikit-learn (PEDREGOSA *et al.*, 2011). This comparative analysis serves to elucidate the distinctive strengths and potential areas of improvement inherent in the proposed high-level classification approach. In addition to the straightforward comparison, a robustness assessment is performed through the execution of K-Folding procedures, employing a value of K=10.

To provide a concrete illustration, our initial exploration involves a dataset comprising three discernible classes, as vividly portrayed in Fig. 14(a). Herein, the orange hue denotes the spiral class, the green hue signifies the star class, and the blue hue represents the amalgamation of three interwoven spirals. Noteworthy is the pronounced overlapping between the green and orange classes, contributing to the intricacies of the classification task and underscoring the necessity for a robust and sophisticated high-level classification approach.

Figure 14(b) visually illustrates the superior performance of the proposed classification method, as evidenced by a higher median accuracy and a more favorable distribution compared to other classical algorithms.

To intensify the complexity of the classification challenge, an additional class is introduced. Referencing Fig. 15, we encounter a dataset featuring Four Classes. Notably, a red class is seamlessly integrated, exhibiting subtle overlap with the existing orange class. This deliberate



(a) Visualization of the dataset with three classes

(b) Results of the experiment - Accuracy

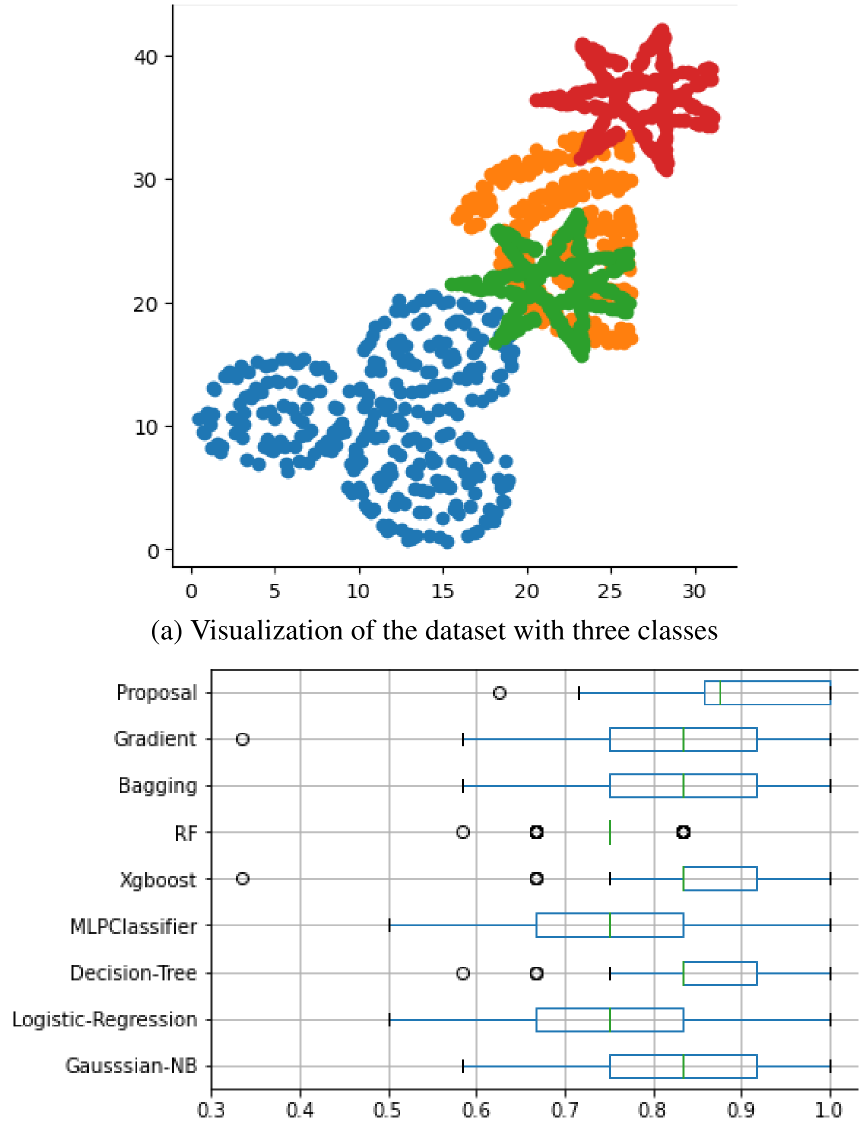
Figure 14 – Visualization of Dataset (3 classes) and results, showing the performance of the proposal - Accuracy

augmentation aims to assess the adaptability and discriminative capability of the proposed classification method in scenarios with increased class diversity and overlapping patterns.

Once more, the outcomes yielded by the proposed method emerge as the optimal approach when evaluating both the minimum and maximum values within the boxplot, as depicted in Fig. 15(b). Expanding the complexity further, we direct our attention to a dataset comprising Five Classes, showcased in Fig. 16(a).

Introducing an additional purple class characterized by a spiral form, this dataset accentuates the capacity of the proposed classification method to contend with an increased number of classes and intricate geometric patterns.

Concluding our explorations, we delve into a dataset featuring Seven Classes, vividly



(a) Visualization of the dataset with three classes

(b) Results of the experiment - Accuracy

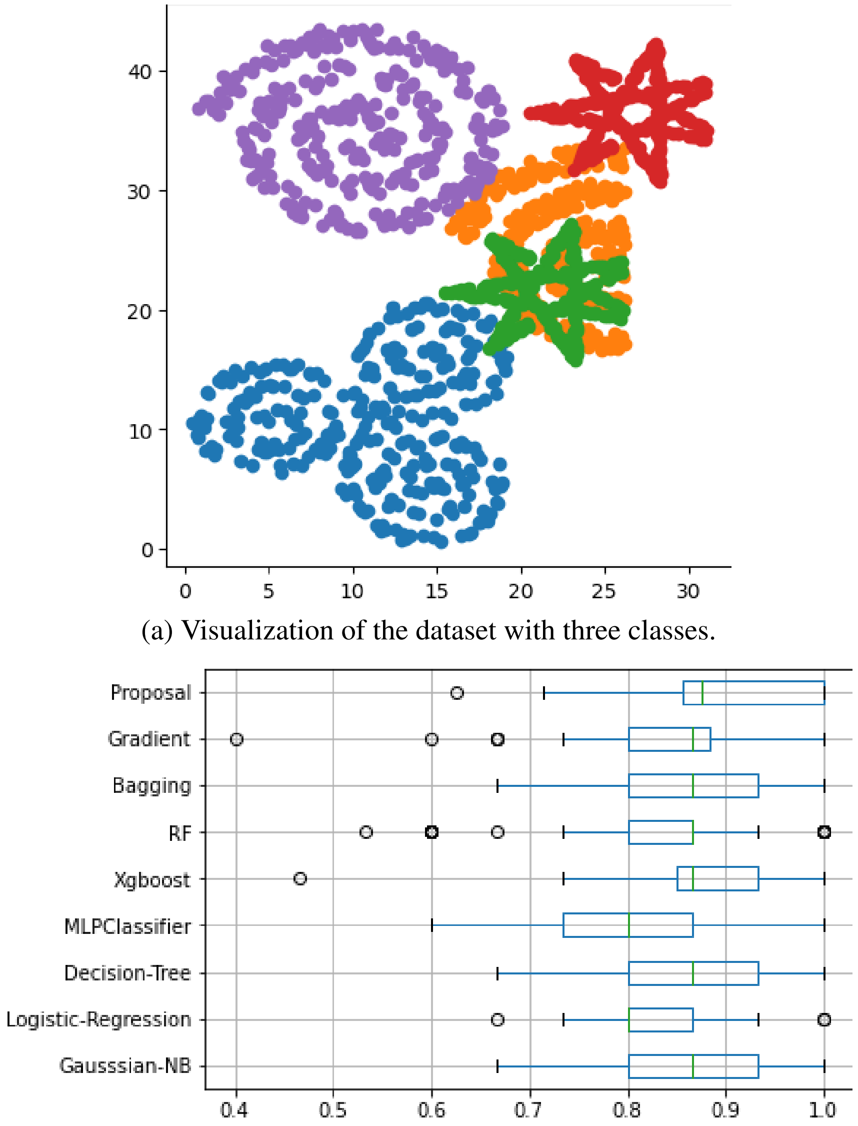
Figure 15 – Visualization of Dataset (4 classes) and results, showing the performance of the proposal - Accuracy

depicted in Fig. 17(a). This dataset introduces a new layer of complexity with the inclusion of a brown class positioned in proximity to the purple and blue classes, while a pink class finds placement near the blue class.

The deliberate insertion of these classes aims to scrutinize the robustness and adaptability of the proposed classification method in scenarios with an expanded class count and intricate spatial relationships.

Again, the proposed method achieves the best classification result in comparison to a set of machine learning algorithms. Considering all the previous experiments, it is possible notice the median of the actual proposal is better than the other algorithms.

Given the evident success of our proposed classification method across various tasks,



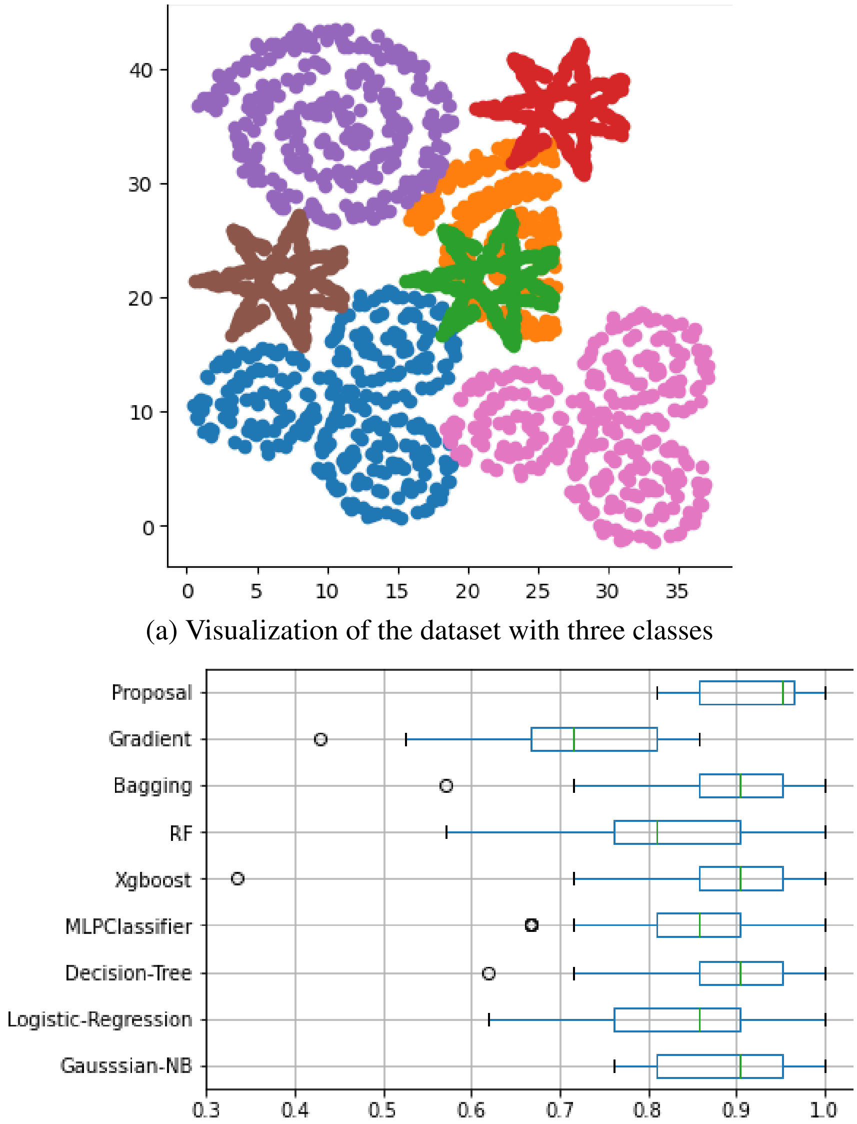
(a) Visualization of the dataset with three classes.

(b) Results of the experiment - Accuracy.

Figure 16 – Visualization of Dataset (5 classes) and results, showing the performance of the proposed classification method.

we've opted to continue with this reliable approach for our upcoming endeavors.

In the next section, we introduce an optimization strategy that involves utilizing a weight vector. This vector helps characterize the importance levels of different features in our model. The goal is to enhance the overall performance of our classification model, making it more effective and adaptable to diverse datasets and scenarios.



(a) Visualization of the dataset with three classes

(b) Results of the experiment - Accuracy

Figure 17 – Visualization of Dataset (7 classes) and results, showing the performance of the proposal

4.3 The Proposed High-Level Classification Method With Optimization

Considering that features contribute in different levels or have different importance, the proposed method includes to consider a weight for each feature of a node to get the level of importance or contribution following an optimization approach.

Figure 18 shows the classification accuracy results for the seven classes dataset with various geometric patterns presented in the last subsection. So, it is possible to notice that the results of the proposed method using an optimization approach are better.

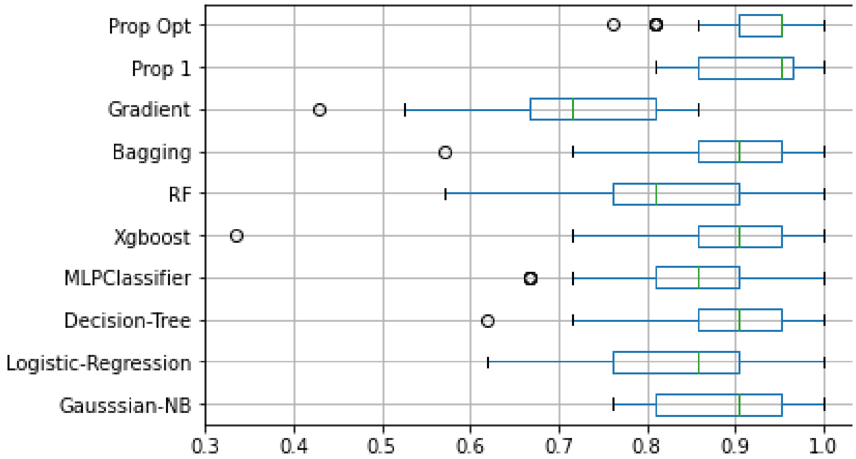


Figure 18 – Results following an Optimization Approach

Now we test the proposed high-level classification methods with more datasets. Specifically, we do simulations on some baseline artificial datasets. The chosen dataset are available in (ROMANO *et al.*, 2021). We select the following datasets: magic, satimage, sleep and phoneme and a brief description of the selected datasets is present in Tab. 2, all the features are numerical. The datasets used for these experiments have different number of features and classes.

Dataset PMLB			
Name	**Samples Class**	**Variables**	**Classes**
magic	0: 12332, 1: 6688	10	2
satimage	1: 1533, 2: 703, 3: 1358, 4: 626, 5 707, 7: 1508	36	6
sleep	0: 21359,1: 9052,2: 52698, 3: 10832,5: 11967	13	5
phoneme	0: 3818, 1: 586	5	2

Tabela 2 – Dataset description.

After the experiments using a cross validation of $k = 10$, the results are presented in Fig 19. The proposed technique without optimization is named *Proposal* and with optimization is *PropOpt*.

The weights found by the optimization procedure for each dataset are:

- magic: 0.408, 0.872, 0.248, 0.899, 0.691, 0.913, 0.359, 0.0869, 0.783, 0.823
- satimage: 0.0998, 0.186, 0.566, 0.63, 0.517, 0.445, 0.155, 0.167, 0.394, 0.0776, 0.326, 0.532, 0.285, 0.816, 0.733, 0.232, 0.803, 0.821, 0.743, 0.682, 0.169, 0.422, 0.017, 0.906, 0.851, 0.269, 0.102, 0.511, 0.649, 0.657, 0.745, 0.0268, 0.586, 0.984, 0.668, 0.403
- sleep: 0.603, 0.614, 0.711, 0.107, 0.914, 0.965, 0.23, 0.578, 0.0801, 0.711, 0.0342, 0.244, 0.792

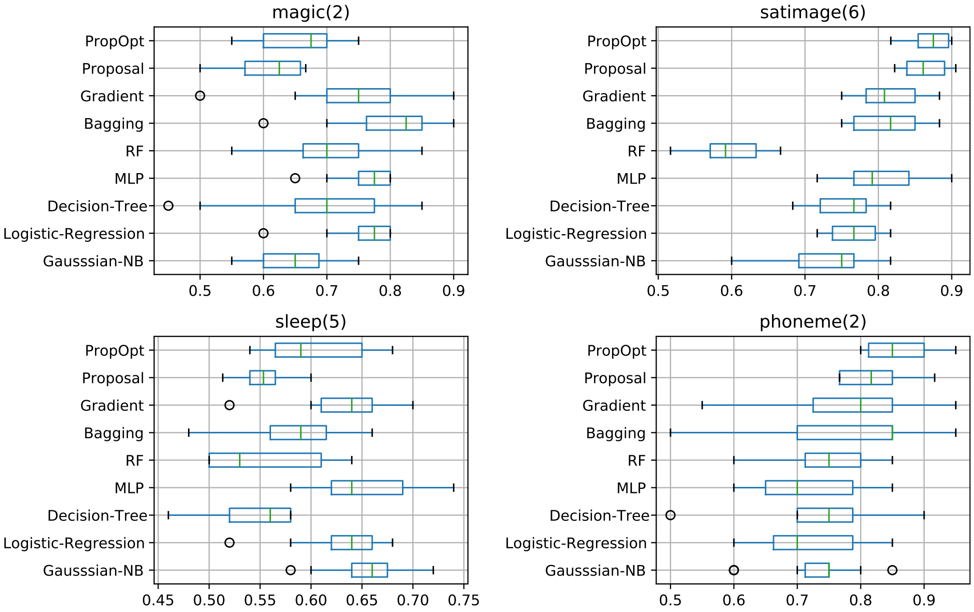


Figure 19 – Classification accuracy results.

- phoneme: 0.47, 0.208, 0.381, 0.974, 0.0497

The weights can be values between zero and one, then values closer to zero mean less relevance and closer to one, otherwise.

The experimental findings in this section underscore the efficacy of the proposed method, particularly in the satimage and phoneme datasets, where optimal results are achieved. The optimization approach showcases its potential by enhancing results by up to 10%. However, with the magic and sleep datasets, the proposed methods fall into the lower three positions. Nonetheless, it's noteworthy that, across all scenarios, the *PropOpt* method with optimization consistently outperforms the non-optimized *Proposal* method by up to 10

4.4 Experimental Results of the Applications

Our high-level classification method proves its mettle in two crucial healthcare applications (MILLER; GEE; KATZ, 2021; DORAISWAMY *et al.*, 2020; ALAKUIJALA; SALMI, 2016) . In the realm of COVID-19 detection, it showcases its adaptability by effectively discerning virus-related abnormalities within X-ray images. The method's versatility is evident as it navigates through diverse radiographic representations, contributing meaningfully to the global fight against the pandemic.

Switching gears to audio analysis, our method exhibits commendable prowess in snore classification, demonstrating its ability to distinguish subtle nuances within a diverse dataset of audio samples. Operating seamlessly in varied acoustic environments, it efficiently categorizes different snoring patterns, offering valuable insights into sleep-related disorders. These results position our high-level classification method as a good contender for healthcare diagnostics and audio signal processing applications, marking a significant stride in scientific exploration and technological advancements in these domains.

4.4.1 Covid-19 Detection

After a searching using the keywords, such as COVID-19 tomography dataset. Many datasets have been found but these datasets are too big to be downloaded and processed later. Besides this, a variety of image formats are presented in the datasets, such as nii, dicom. One available dataset[1] is chosen because the png format of the images is ready for processing. Fig. 20 presents 16 samples of positive and negative cases of COVID-19, respectively.

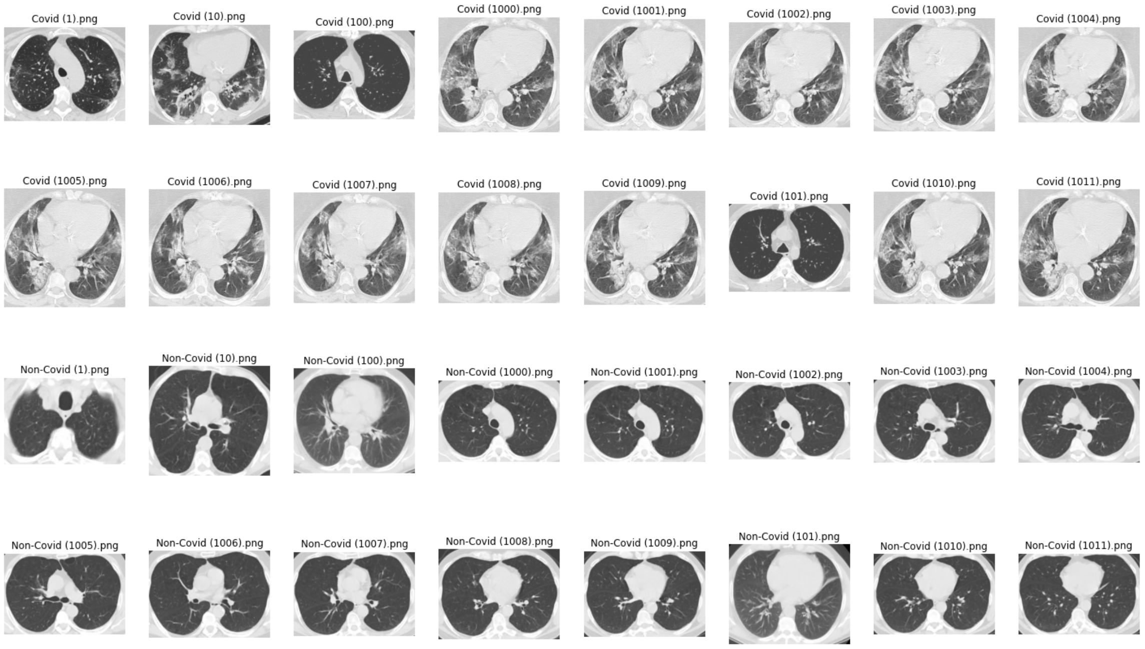


Figure 20 – Sample of Dataset - X ray Images

A dataset with 100 images is selected, positives and negative classes are balanced. By consequence, it is necessary to find a transformation to extract features of the images. A first proposal is using Frequency Histogram, because it can reduce dimensionality and represent the distribution of pixels. Previously, a transformation of color images is performed to get grayscale images. Later, a proposal using GLCM is done to get neighborhood features considering texture analysis. The classification results are shown in Fig. 21 and the good results in term of accuracy are reached comparing with other well-known classification algorithms.

4.4.2 Snore Classification

Following the extraction of features using Mel-Frequency Cepstral Coefficients (MFCC), we meticulously curated a dataset comprising 1000 samples, evenly distributed with 500 samples

[1] https://www.kaggle.com/plameneduardo/sarscov2-ctscan-dataset/notebooks

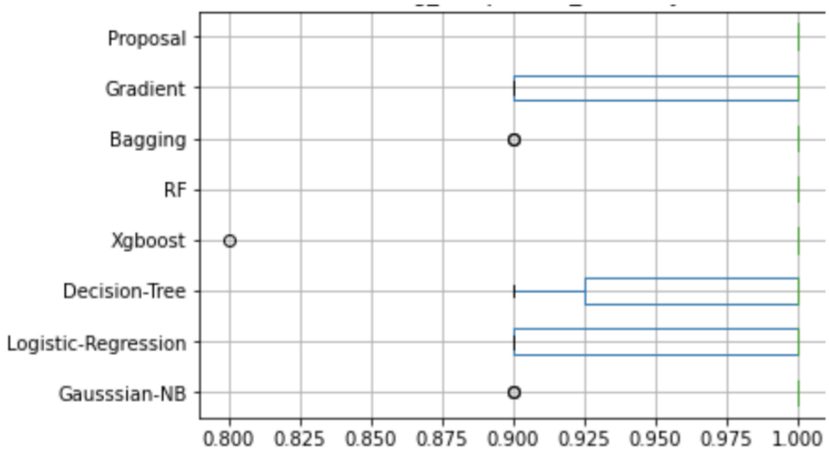


Figure 21 – Results of experiment - Accuracy

per class. Subsequently, we conducted a series of experiments to rigorously compare and validate the efficacy of our proposed method.

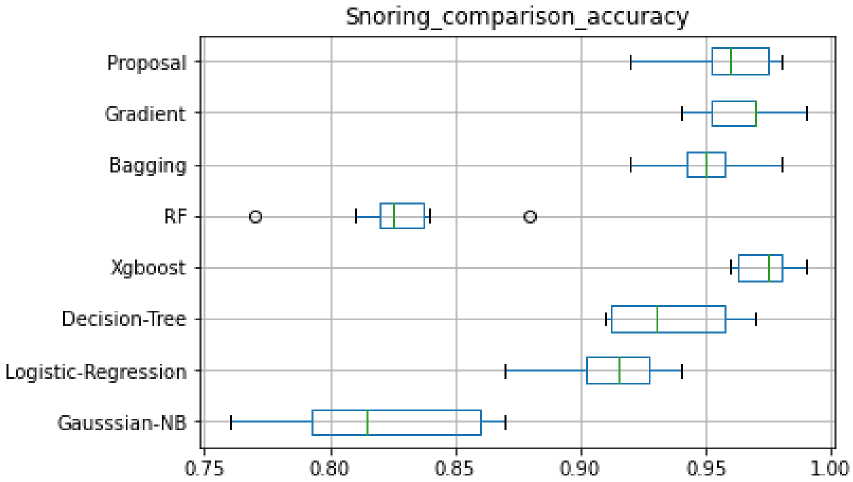


Figure 22 – Signal classification accuracy results.

The results, showcased in Fig. 22, unequivocally highlight the remarkable performance of our proposed technique. Demonstrating consistently high classification accuracy, these findings underscore the robustness and reliability of our approach in handling the intricate task of classifying audio samples based on snoring patterns.

In-depth examination of the experimental results reveals the remarkable adaptability and reliability of our proposed method. Its robust performance across artificial datasets, traditional academic datasets, and real-world applications such as COVID-19 detection and Snoring Classification attests to its versatility and effectiveness. The consistent and promising outcomes across diverse scenarios underscore the potential impact and applicability of our approach, showcasing its relevance in academic research and its potential for real-world problem-solving.

CAPÍTULO 5

CONCLUSIONS

In this work, we have developed new high-level classification methods improving the original ones in various aspects. High-level classification is pattern-based learning instead of feature-based learning. Its salient advantage lies in the semantic-level learning mimicking the behavior of human beings. A "data pattern"means the organizational structure of the data and one of the intuitive forms to characterize such kind of structures is by describing the relationships among data, which is the main property of complex networks. Therefore, we use complex networks to characterize data patterns to realize pattern-based learning.

5.1 Concluding Remarks

The original high-level classification methods present several limitations. One of them is the selection of network measures. In the previous works (Silva; Zhao, 2012; SILVA; ZHAO, 2015b; Carneiro *et al.*, 2017; CARNEIRO; ZHAO, 2018b; COLLIRI *et al.*, 2018; CARNEIRO *et al.*, 2019; LIU *et al.*, 2022), two types of network measures are applied: 1) local measures, such as degree, clustering coefficient, assortativity, and PageRank. 2) global measures, such as efficiency, the average lengths of the transient and attractor of tourist walks, communicability. This first kind of measures does not present a global view of the pattern formation of each class. The second kind of measures considers all the edges of the network. Therefore, when we insert a test sample for classification, only a very small number of edges is inserted compared to the network of a given class. Consequently, the difference of the network measures before and after the insertion is always small. As a result, it is not clear whether the the test sample conforms or not to the class pattern, i.e., the classification decision is confusing. Another limitation of high-level classification methods is the network construction. In the previous methods, all the features in the feature vector of each data sample (node) are considered to have the same importance. In practice, each feature may contribute at different level to characterize the data pattern. In order to overcome the two shortages of the original high-level classification techniques, we have made the following improvements:

- Propose two network measures with higher sensitivity to the test sample insertion. The first approach reaches a higher sensitivity using the Travelling Salesman Problem to get a cycle which connects all the nodes through Ant Colony Optimization. Another approach is to get a measure based on the calculation of the Minimum Spanning Tree of the underlying data networks. Then, we use these metrics to estimate the structure of the complex network. Good results have been obtained.
- Build optimal complex networks. The approach considers that features contribute in different levels to the classification, then it is possible to add a weight for each feature and determine the values following an optimization approach. Again, promising results have been obtained.
- Performance experiments to test the proposal and an application for COVID-19 diagnosis. The results presented in the last chapter show the good performance of the proposed methods. Besides, the application makes a good contribution considering the actual context of pandemic.

While the study did not explicitly delve into time complexity, a current limitation arises concerning the computation of Minimum Spanning Trees (MST) for each element in the test set. The need to recalculate MST for every inserted element could pose a time challenge, particularly with larger datasets. To address this, threading has been implemented to parallelize the cross-validation calculation, mitigating the potential time increase associated with the MST computation.

5.2 Future Works

As future works, we would like to do the following tasks.

- Usually, each data pattern consists of a set of sub-patterns. A test sample may match well to one or more sub-patterns, but it may not match the whole pattern. Therefore, we will divide the data network of each class into sub-networks and we will check the conformation of the test sample to each sub-network. Each sub-network may be identified by community detection methods.
- We will apply the high-level method to explain the decisions made by other classification techniques, for example, the decision made by Graph Neural Networks (GNNs). Our hypothesis is that high-level classification is pattern-based learning, while neural networks are basically feature-based learning. Therefore, the high-level methods are suitable to give explanations of neural network decisions through pattern formation of the data.

- We may also apply the high-level classification idea to develop pattern-based outlier detection techniques.
- We will use the developed high-level classification methods in new applications, such as the gene or protein pattern detection in some organisms.
- Optimize the time complexity of the proposed algorithm to handle larger datasets more efficiently.

REFERÊNCIAS

ALAKUIJALA, A.; SALMI, T. Predicting obstructive sleep apnea with periodic snoring sound recorded at home. **Journal of Clinical Sleep Medicine**, v. 12, n. 07, p. 953–958, 2016. Disponível em: <https://jcsm.aasm.org/doi/abs/10.5664/jcsm.5922>. Citado na página 63.

ALBERT, R.; ALBERT, I.; NAKARADO, G. L. Structural vulnerability of the north american power grid. **Phys. Rev. E**, American Physical Society, v. 69, p. 025103, Feb 2004. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.69.025103>. Citado na página 25.

ALBERT, R.; JEONG, H.; BARABÁSI, A.-L. Diameter of the world-wide web. **Nature**, v. 401, n. 6749, p. 130–131, Sep 1999. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/43601>. Citado na página 25.

ALPAYDIN, E. **Introduction to Machine Learning**. 3. ed. Cambridge, MA: MIT Press, 2014. (Adaptive Computation and Machine Learning). ISBN 978-0-262-02818-9. Citado nas páginas 23 e 24.

APOSTOLOPOULOS, I. D.; MPESIANA, T. A. Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks. **Physical and Engineering Sciences in Medicine**, v. 43, n. 2, p. 635–640, Jun 2020. ISSN 2662-4737. Disponível em: <https://doi.org/10.1007/s13246-020-00865-4>. Citado na página 47.

BARABÁSI, A.-L. **Network Science**. [S.l.]: Cambridge University Press, 2016. Citado na página 29.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **Science**, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999. ISSN 0036-8075. Disponível em: <https://science.sciencemag.org/content/286/5439/509>. Citado nas páginas 25 e 31.

BARABÁSI, A.-L. *et al.* **Network science**. [S.l.]: Cambridge university press, 2016. Citado na página 25.

BARBEHENN, M. A note on the complexity of dijkstra's algorithm for graphs with weighted vertices. **IEEE Transactions on Computers**, v. 47, n. 2, p. 263–, 1998. Citado na página 55.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2011. Citado nas páginas 23 e 24.

CANCHO, R. V. S. R. F. Optimization in complex networks. 2001. Citado na página 45.

CARNEIRO, M.; ZHAO, L. Organizational data classification based on the importance concept of complex networks. **IEEE Transactions on Neural Networks and Learning Systems**, v. 29, p. 3361–3373, 2018. Citado na página 47.

CARNEIRO, M. G.; CHENG, R.; ZHAO, L.; JIN, Y. Particle swarm optimization for network-based data classification. **Neural Networks**, v. 110, p. 243–255, 2019. Citado nas páginas 36 e 67.

CARNEIRO, M. G.; RAMOS, C. D.; DESTRO-FILHO, J.-B.; ZHU, Y. tao; JI, D.; ZHAO, L. High-level classification for eeg analysis. In: **The 2013 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2023. p. 1–8. Citado nas páginas 36 e 37.

Carneiro, M. G.; Rosa, J. L. G.; Zheng, Q.; Liu, X.; Zhao, L. Improving semantic role labeling using high-level classification in complex networks. In: **2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)**. [S.l.: s.n.], 2017. p. 2206–2212. Citado nas páginas 36 e 67.

Carneiro, M. G.; Zhao, L. High level classification totally based on complex networks. In: **2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence**. [S.l.: s.n.], 2013. p. 507–514. Citado nas páginas 24 e 36.

CARNEIRO, M. G.; ZHAO, L. Organizational data classification based on the importance concept of complex networks. **IEEE Transactions on Neural Networks and Learning Systems**, v. 29, n. 8, p. 3361–3373, 2018. Citado na página 25.

____. Organizational Data Classification Based on the Importance Concept of Complex Networks. **IEEE Transactions on Neural Networks and Learning Systems**, v. 29, n. 8, p. 3361–3373, ago. 2018. ISSN 2162-2388. Conference Name: IEEE Transactions on Neural Networks and Learning Systems. Citado nas páginas 36 e 67.

Carneiro, M. G.; Zhao, L.; Cheng, R.; Jin, Y. Network structural optimization based on swarm intelligence for highlevel classification. In: **2016 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2016. p. 3737–3744. ISSN 2161-4407. Citado na página 36.

CHAPELLE, O.; SCHöLKOPF, B.; ZIEN, A. **Semi-supervised learning.** [S.l.]: The MIT Press, 2006. Citado na página 24.

CHARTRAND, G.; ZHANG, P. **A First Course in Graph Theory (Dover Books on Mathematics)**. [S.l.]: Dover Publications, 2012. Citado na página 30.

CHIRE-SAIRE, J. E. **New feature for Complex Network based on Ant Colony Optimization for High Level Classification**. 2020. Citado nas páginas 25 e 47.

Colliri, T.; Ji, D.; Pan, H.; Zhao, L. A network-based high level data classification technique. In: **2018 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2018. p. 1–8. Citado nas páginas 25 e 36.

COLLIRI, T.; JI, D.; PAN, H.; ZHAO, L. A Network-Based High Level Data Classification Technique. In: **2018 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2018. p. 1–8. Citado na página 67.

COLORNI, A.; DORIGO, M.; MANIEZZO, V. **Distributed Optimization by Ant Colonies**. 1991. Citado na página 26.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms, Second Edition**. [S.l.]: The MIT Press and McGraw-Hill Book Company, 2001. ISBN 0-262-03293-7. Citado na página 42.

CUPERTINO, T. H.; HUERTAS, J.; ZHAO, L. Data clustering using controlled consensus in complex networks. **Neurocomput.**, Elsevier Science Publishers B. V., NLD, v. 118, p. 132–140, out. 2013. ISSN 0925-2312. Disponível em: <https://doi.org/10.1016/j.neucom.2013.02.026>. Citado na página 44.

DORAISWAMY, S.; ABRAHAM, A.; MAMTANI, R.; CHEEMA, S. Use of telehealth during the covid-19 pandemic: Scoping review. **J Med Internet Res**, v. 22, n. 12, p. e24087, Dec 2020. ISSN 1438-8871. Disponível em: <https://www.jmir.org/2020/12/e24087>. Citado na página 63.

DORIGO, M. **Optimization, Learning and Natural Algorithms**. Tese (Doutorado) — Politecnico di Milano, Italy, 1992. Citado na página 40.

DUA, D.; GRAFF, C. **UCI Machine Learning Repository**. 2019. Disponível em: <http://archive.ics.uci.edu/ml>. Citado nas páginas 50 e 51.

DUDA, R.; HART, P.; G.STORK, D. Pattern classification. In: ____. [S.l.: s.n.], 2001. xx. ISBN 0-471-05669-3. Citado nas páginas 23 e 24.

ERDÖS, P.; RÉNYI, A. On random graphs i. **Publicationes Mathematicae Debrecen**, v. 6, p. 290–297, 1959. Citado na página 31.

ESTRADA, E.; HATANO, N. Communicability in complex networks. **Physical Review E**, v. 77, n. 3, p. 036111, mar. 2008. Publisher: American Physical Society. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.77.036111>. Citado na página 33.

FADAEE, S. A.; HAERI, M. A. Classification using link prediction. **Neurocomputing**, v. 359, p. 395 – 407, 2019. Citado na página 47.

GAD, A. F. **PyGAD: An Intuitive Genetic Algorithm Python Library**. 2021. Citado na página 45.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. Citado na página 24.

GOSS, S.; ARON, S.; DENEUBOURG, J. L.; PASTEELS, J. M. Self-organized shortcuts in the argentine ant. **Naturwissenschaften**, v. 76, n. 12, p. 579–581, 1989. ISSN 1432-1904. Disponível em: <https://doi.org/10.1007/BF00462870>. Citado nas páginas 15, 40 e 41.

GRAHAM, R.; HELL, P. On the history of the minimum spanning tree problem. **Annals of the History of Computing**, v. 7, n. 1, p. 43–57, 1985. Citado na página 40.

HAN, J.; KAMBER, M. **Data mining : concepts and techniques**. San Francisco [u.a.]: Kaufmann, 2005. ISBN 1558604898 9781558604896. Disponível em: <http://www.amazon.com/Data-Mining-Concepts-Techniques-Management/dp/1558604898>. Citado na página 23.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. [S.l.]: Prentice Hall, 1998. Citado na página 24.

JEBARA, T.; WANG, J.; CHANG, S.-F. Graph construction and <i>b</i>-matching for semi-supervised learning. In: **Proceedings of the 26th Annual International Conference on Machine Learning**. New York, NY, USA: Association for Computing Machinery, 2009. (ICML ’09), p. 441–448. ISBN 9781605585161. Disponível em: <https://doi.org/10.1145/1553374.1553432>. Citado na página 44.

JEONG, H.; TOMBOR, B.; ALBERT, R.; OLTVAI, Z. N.; BARABÁSI, A.-L. The large-scale organization of metabolic networks. **Nature**, v. 407, n. 6804, p. 651–654, Oct 2000. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/35036627>. Citado na página 25.

KHAN, T. A deep learning model for snoring detection and vibration notification using a smart wearable gadget. **Electronics**, v. 8, n. 9, 2019. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/8/9/987>. Citado na página 48.

KRUSKAL, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. **Proceedings of the American Mathematical society**, JSTOR, v. 7, n. 1, p. 48–50, 1956. Citado na página 42.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, p. 436–444, 2015. Citado na página 24.

LI, Y. Toward a qualitative search engine. **Internet Computing**, v. 2, n. 4, p. 24–29, 2002. Citado na página 33.

LIU, W.; YAN, J.; ZHU, Y. tao; PEREIRA, E. J. de F.; LI, G.; ZHENG, Q.; ZHAO, L. Analysis of radiographic images of patients with covid-19 using fractal dimension and complex network-based high-level classification. In: **The 2012 International Conference on Complex Networks and Applications (CNA))**. [S.l.: s.n.], 2022. p. 1–10. Citado nas páginas 37 e 67.

LUXBURG, U. von. **A Tutorial on Spectral Clustering**. 2007. Citado na página 44.

MAHMUD, T.; RAHMAN, M. A.; FATTAH, S. A. Covxnet: A multi-dilation convolutional neural network for automatic covid-19 and other pneumonia detection from chest x-ray images with transferable multi-receptive feature optimization. **Computers in Biology and Medicine**, v. 122, p. 103869, 2020. ISSN 0010-4825. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0010482520302250>. Citado na página 47.

MILLER, L. M. S.; GEE, P. M.; KATZ, R. A. The importance of understanding covid-19: The role of knowledge in promoting adherence to protective behaviors. **Frontiers in Public Health**, v. 9, 2021. ISSN 2296-2565. Disponível em: <https://www.frontiersin.org/articles/10.3389/fpubh.2021.581497>. Citado na página 63.

MITCHELL, T. M. **Machine Learning**. 1. ed. USA: McGraw-Hill, Inc., 1997. ISBN 0070428077. Citado nas páginas 23 e 24.

MODELS and Methods in Social Network Analysis. [S.l.]: Cambridge University Press, 2005. (Structural Analysis in the Social Sciences). Citado na página 25.

MONTOYA, J. M.; SOLÉ, R. V. Small world patterns in food webs. **Journal of Theoretical Biology**, v. 214, n. 3, p. 405 – 412, 2002. ISSN 0022-5193. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0022519301924609>. Citado na página 25.

Neto, F. A.; 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. Citado na página 35.

NEWMAN, M. E. J. **Networks: an introduction.** [S.l.]: Oxford University Press, 2010. Citado na página 25.

OZTURK, T.; TALO, M.; YILDIRIM, E. A.; BALOGLU, U. B.; YILDIRIM, O.; Rajendra Acharya, U. Automated detection of covid-19 cases using deep neural networks with x-ray images. **Computers in Biology and Medicine**, v. 121, p. 103792, 2020. ISSN 0010-4825. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0010482520301621>. Citado na página 47.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. *et al.* Scikit-learn: Machine learning in python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado na página 57.

PRIM, R. C. Shortest connection networks and some generalizations. **The Bell System Technical Journal**, v. 36, n. 6, p. 1389–1401, 1957. Citado na página 42.

RAEDT, L. D. Logical and relational learning. In: ZAVERUCHA, G.; COSTA, A. L. da (Ed.). **Advances in Artificial Intelligence - SBIA 2008**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 1–1. ISBN 978-3-540-88190-2. Citado na página 23.

RAHMANDANI, M.; NUGROHO, H. A.; SETIAWAN, N. A. Cardiac sound classification using mel-frequency cepstral coefficients (mfcc) and artificial neural network (ann). In: **2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)**. [S.l.: s.n.], 2018. p. 22–26. Citado na página 48.

ROMANO, J. D.; LE, T. T.; CAVA, W. L.; GREGG, J. T.; GOLDBERG, D. J.; CHAKRABORTY, P.; RAY, N. L.; HIMMELSTEIN, D.; FU, W.; MOORE, J. H. Pmlb v1.0: an open source dataset collection for benchmarking machine learning methods. **arXiv preprint arXiv:2012.00058v2**, 2021. Citado na página 62.

SAIRE, J. C.; ZHAO, L. Complex network-based data classification using minimum spanning tree metric and optimization. In: **2023 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2023. p. 1–7. Citado nas páginas 43 e 45.

SARKER, I. H. Machine learning: Algorithms, real-world applications and research directions. **SN Computer Science**, v. 2, n. 3, p. 160, 2021. Disponível em: <https://doi.org/10.1007/s42979-021-00592-x>. Citado na página 23.

SHRESHTHA, S.; SINGH, A.; SAHDEV, S.; SINGHA, M.; RAJPUT, S. A deep dissertion of data science: Related issues and its applications. In: **2019 Amity International Conference on Artificial Intelligence (AICAI)**. [S.l.: s.n.], 2019. p. 939–942. Citado na página 23.

SILVA, T. C.; ZHAO, L. Network-based high level data classification. **Neural Networks and Learning Systems, IEEE Transactions on**, v. 23, n. 6, p. 954–970, 2012. Citado na página 25.

Silva, T. C.; Zhao, L. Network-based high level data classification. **IEEE Transactions on Neural Networks and Learning Systems**, v. 23, n. 6, p. 954–970, June 2012. Citado nas páginas 34, 35 e 67.

____. Pattern-based classification via a high level approach using tourist walks in networks. In: **2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence**. [S.l.: s.n.], 2013. p. 284–289. Citado na página 35.

SILVA, T. C.; ZHAO, L. High-level pattern-based classification via tourist walks in networks. **Information Sciences**, v. 294, p. 109–126, 2015. Citado na página 25.

____. High-level pattern-based classification via tourist walks in networks. **Information Sciences**, v. 294, p. 109–126, 2015. ISSN 0020-0255. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0020025514009608>. Citado nas páginas 35 e 67.

SREERAM, A.; RAVISHANKAR, U.; SRIPADA, N. R.; MAMIDGI, B. Investigating the potential of mfcc features in classifying respiratory diseases. In: **2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)**. [S.l.: s.n.], 2020. p. 1–7. Citado na página 48.

SUDHOLT, D.; THYSSEN, C. Running time analysis of ant colony optimization for shortest path problems. **Journal of Discrete Algorithms**, v. 10, p. 165–180, 2012. ISSN 1570-8667. StringMasters 2009 2010 Special Issue. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1570866711000682>. Citado na página 55.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. [S.l.]: MIT Press, 2018. Citado na página 24.

TAN, P.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. [S.l.]: Addison-Wesley, 2005. Citado na página 23.

TUY, P. G. da S. E.; RIOS, T. N. Summarizer: Fuzzy rule-based classification systems for vertical and horizontal big data. In: **2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. [S.l.: s.n.], 2020. p. 1–8. Citado na página 24.

VAPNIK, V. N. **Statistical Learning Theory**. [S.l.]: Wiley-Interscience, 2008. Citado nas páginas 23 e 24.

VIGLIOCCO, G.; METEYARD, L.; ANDREWS, M.; KOUSTA, S. Toward a theory of semantic representation. **Language and Cognition**, v. 1, n. 2, p. 219–247, 2009. Disponível em: <https://doi.org/10.1515/LANGCOG.2009.011>. Citado na página 24.

VILCA, E.; ZHAO, L. A network-based high-level data classification algorithm using betweenness centrality. In: . [S.l.: s.n.], 2020. Citado na página 47.

Wang, F.; Zhang, C. Label propagation through linear neighborhoods. **IEEE Transactions on Knowledge and Data Engineering**, v. 20, n. 1, p. 55–67, 2008. Citado na página 44.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of “small-world"networks. **Nature**, Nature Publishing Group, v. 393, n. 6684, p. 440, 1998. Citado na página 31.

WEST, G. B.; BROWN, J. H.; ENQUIST, B. J. A general model for the structure and allometry of plant vascular systems. **Nature**, v. 400, n. 6745, p. 664–667, Aug 1999. ISSN 1476-4687. Disponível em: <https://doi.org/10.1038/23251>. Citado na página 25.

WITTEN, I. H.; FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation**. San Francisco: Morgan Kaufmann, 2000. Citado na página 23.

YUNUS, M. 2020. Disponível em: <https://medium.com/@yunusmuhammad007/feature-extraction-gray-level-co-occurrence-matrix-glcm-10c45b6d46a1>. Citado na página 47.

ZHU, X. **Semi-Supervised Learning Literature Survey**. [S.l.], 2005. Disponível em: <http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf>. Citado na página 43.

ICMC USP
SÃO CARLOS