UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Agent-Based Modeling for the Analysis of Complex Networks

Alex Josué Flórez Farfán

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

ICMC USP
SÃO CARLOS

Alex Josué Flórez Farfán

Agent-Based Modeling for the Analysis of Complex Networks

Thesis submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP – in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Odemir Martinez Bruno

USP – São Carlos
January 2024

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

```
        Farfan, Alex Josue Florez
F222a      Agent-Based Modeling for the Analysis of Complex
        Networks / Alex Josue Florez Farfan; orientador
        Odemir Martinez Bruno. -- São Carlos, 2024.
           151 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, 2024.


           1. Computational Modeling. 2. Pattern
        Recognition. 3. Agent-Based Models. 4. Complex
        Networks. 5. Machine Learning. I. Bruno, Odemir
        Martinez, 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

Alex Josué Flórez Farfán

Modelagem Baseada em Agentes para Análise de 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. Odemir Martinez Bruno

USP – São Carlos
Janeiro de 2024

This work is dedicated to my parents Claudio and Benigna.

Thank you for your loving support throughout my life.

ACKNOWLEDGEMENTS

Thank you to:

God by bringing me here to Brazil, a wonderful country that I learn to love.

My parents, Claudio and Benigna for their unconditional love and continuous encouragement.

My brother Samuel and his wife Sandra, who are a source of inspiration in my life, and my niece Julieta, who is learning and exploring the wonders of this world.

My sisters Amparo and Miriam, for being a constant example in my life.

My advisor Odemir Bruno, for his guidance, support and patience during this time.

My friends at São Carlos, at university and church.

The Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP) for the opportunity to study in an institution that is full of many talented people.

The Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code PROEX-9527567/D for providing the financial means to pursue the doctorate.

“Is it possible for a man
to dream continuously
for seventy years?”
– Honi the Circle Maker

RESUMO

FARFÁN, A. J. F. **Modelagem Baseada em Agentes para Análise de Redes Complexas**. 2024. 152 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, 2024.

A modelagem baseada em agentes é uma abordagem dentro da modelagem computacional que se concentra na simulação do comportamento e das interações de agentes individuais para entender os padrões emergentes em sistemas complexos. Esta tese discute uma abordagem desenvolvida em modelagem baseada em agentes para estudar e analisar redes complexas. As características inerentes dos modelos baseados em agentes fornecem o contexto apropriado para explorar redes complexas. Ao identificar, analisar e compreender as propriedades emergentes que surgem da dinâmica e do comportamento dos agentes, podemos obter e reconhecer padrões dentro de redes complexas. A caracterização de rede é uma tarefa importante de reconhecimento de padrões. A modelagem de um processo sobre o espaço fornecido pelas redes gera padrões em diferentes níveis, individualmente nos agentes, bem como globalmente em todo o modelo. Para atingir o objetivo, é proposta uma abordagem baseada em agentes da qual são extraídas características que servem para categorizar as redes. É importante destacar que na literatura modelos baseados em agentes não têm sido utilizados para categorizar redes. O modelo proposto, denominado modelo de Crescimento, fornece uma nova consideração para caracterizar redes complexas. A análise realizada em conjuntos de dados de redes sintéticas e do mundo real indica que os resultados da classificação são semelhantes aos métodos da literatura. A acurácia da classificação mostra que em quatro conjuntos de dados, *Actinobacteria*, *Fungi*, *Kingdom* e *Plant* os resultados são melhores que os trabalhos anteriores na literatura, demonstrando o potencial desta abordagem.

Palavras-chave: Modelagem Computacional, Reconhecimento de Padrões, Modelagem Baseada em Agentes, Redes Complexas, Aprendizado de Máquina.

ABSTRACT

FARFÁN, A. J. F. **Agent-Based Modeling for the Analysis of Complex Networks**. 2024. 152 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, 2024.

Agent-based modeling is an approach within computational modeling that focuses on simulating the behavior and interactions of individual agents to understand emerging patterns in complex systems. This thesis discusses an approach developed in agent-based models in order to study and analyze complex networks. The inherent characteristics of agent-based models provide the appropriate context for exploring complex networks. By identifying, analyzing and understanding the emergent properties that arise from the dynamics and behavior of the agents we can obtain and recognize patterns within complex networks. Network characterization is an important task of pattern recognition. The modeling of a process over the space provided by networks generate patterns at different levels, individually in the agents, as well as globally in the entire model. In order to achieve the objective, an agent-based approach is proposed from which features are extracted that serve to characterize networks. It is important to highlight that in the literature agent-based models have not been used to categorize networks. The proposed model, called the Growth model, provides a novel consideration to characterize complex networks. The analysis performed on synthetic and real-world network datasets indicate that the classification results are similar with methods of the literature. The classification accuracy shows that in four datasets, *Actinobacteria*, *Fungi*, *Kingdom*, and *Plant* the results are better than the previous work in the literature, demonstrating the potential of this approach.

Keywords: Computational Modeling, Pattern Recognition, Agent-Based Models, Complex Networks, Machine Learning.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ABM	Agent-Based Model
ABM	Agent-Based Modeling
CA	Cellular Automata
CA	Cellular Automaton
CN	Complex Networks
D-TEP	Density Time-Evolution Pattern
DL	Deep Learning
DTW	Deterministic Tourist Walk
ECA	Elementary Cellular Automata
kNN	k-Nearest Neighbor
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LLNA	Life-Like Network Automata
LLNA-BP	Life-Like Network Automata descriptor based on Binary Pattern
ML	Machine Learning
NN	Neural Network
PR	Pattern Recognition
SEM	Spatially Explicit Model
SVC	Support Vector Classifier

CONTENTS

CHAPTER

1

INTRODUCTION

1.1 Context

Humans seek to explain the world through science. To obtain this scientific knowledge we make use of data that allows testing and reproducing our understanding of the world. Scientific knowledge is important both individually and collectively, since it benefits society in general (FUNK; RAINIE; PAGE, 2015), being paramount for the health, safety and prosperity of a nation (BUSH, 2020).

An important aspect of scientific knowledge, essential to demonstrate our understanding of the world, involves the use of theories and models. Modeling holds significant importance within science. Models, in conjunction with relevant data, help in explaining phenomena. The representation of natural phenomena through models stands as a key practice in science (NRC, 2012).

In particular, computational modeling is a great tool for analyzing complex systems (HINSEN, 2023). Complex systems are systems composed of a large number of elements that interact with each other in not obvious ways. The elements arise and evolve through the relationships between them. Thus, the important information resides in the relationships that are formed between the elements.

So it is important to understand and analyze how such interactions form and operate in order to understand the behavior that emerges at a large-scale level (SAYAMA, 2015). Many real world systems can be modeled as complex systems, such as political organizations, human cultures/languages, national and international economies, stock markets, the global climate, food webs, brains, physiological systems, regulatory networks, etc.

There are many ways to model complex system, including: dynamical systems, cellular automata, agent-based modeling, network science, and chaos theory (SIEGENFELD; BAR-YAM, 2020).

Cellular Automata (CA) consists of a set of elements typically arranged on a regular structure, each element has a state that is updated according to predefined rules, and they are all updated at the same time in a synchronous way (WOLFRAM, 1984).

Agent-Based Modeling (ABM) consists of autonomous agents interacting with one another and their environment, each agent has a set of rules and the activation of the agents happen in various ways, random activation, random activation by type, etc (WILENSKY; RAND, 2015).

Pattern recognition in complex networks has become increasingly relevant in academic research and real-world applications (MIRANDA; MACHICAO; BRUNO, 2016b). Pattern recognition seeks to discover meaningful structures and relationships within complex networks. The applications of pattern recognition in complex networks are vast and diverse, from social network analysis and disease spread prediction to recommendation systems and network security (BISHOP, 2006).

In the literature there are several approaches that focus on the categorization of networks, among them are those that use the measurements extracted from the network (COSTA *et al.*, 2007) and methods that use processes on the network through the use of some type of element, such as automata (MIRANDA; MACHICAO; BRUNO, 2016b) or walkers (MERENDA, 2023). In the case of automata, a simulation is performed on the network and some patterns that are generated over time are extracted, these patterns serve as feature vector. On the other hand, when using walkers on a network, they move during the simulation and their movement allows to extract patterns that serve as feature vector.

The work by Costa *et al.* (2007) presents several measurements that can be obtained from a complex network in order to create a feature vector that serves to differentiate each network. Walkers, and tourists, use the spatial information of the network to update their position, from the path they generate it is extracted a feature vector (MERENDA, 2023).

Cellular automata use the spatial information of the network to update their state, changes of state over time generate patterns that are used as a feature vector, one model of cellular automata in networks is the Life-Like Network Automata (LLNA) that uses the density of elements to update the state of the cells, this method is based on the Game of Life automata (MIRANDA; MACHICAO; BRUNO, 2016b), another version uses Local Binary Patterns (LBP) to complement the patterns generated to obtain the feature vector, the LLNA-LBP (RIBAS; MACHICAO; BRUNO, 2020), another variation is the Density-Time Evolution Pattern (D-TEP) method which uses the density of elements of temporal patterns to obtain the feature vector (ZIELINSKI *et al.*, 2022).

Using methods based on automata implies having a wide search space. The well-known cellular automata Game of Life works in a regular grid, in which the state of each cell depends on the quantity of living neighboring cells. A cell can be in either the alive or dead state, and it has 8 neighbors. If the cell is currently in the dead state, it can be born by the presence of the right number of living neighboring cells. Similarly, if the cell is in the alive state, it can survive if it maintains a specific number of living neighboring cells. These conditions can be represented as B/S (“Birth” and “Survival”), for example B3/S23 means that a cell requires 3 living neighboring cells to be born, and 2 or 3 living neighboring cells to survive. Extending this space to all possibilities of birth and survival, it means that a cell can have 9 possible values (0 to 8) of living cells to be born, and similarly 9 possible values (0 to 8) of living cells to survive. Which means that an automata that follows the rules of the game of life has a search space of $2^{(9+9)} = 262144$, which represents the number of configurations to be explored to find the rules that allow the classification of a network dataset.

Since we have processes generated by one type of element, we can ask if using a computational model to simulate a process on a network that uses more than one type of element, the interaction generated by the elements can be used to obtain characteristics and classify the networks?

By applying processes on a network that use a single type of element, it can be extracted patterns that characterize the network, then using another type of modeling on networks where various types of elements interact would have more data sources to generate patterns, these patterns in turn can contribute to categorize the networks.

In the literature there is no explicit mention of the use of Agent-Based Modeling for pattern recognition and network categorization. This can be explained because cellular automata are considered a specific type of ABM, by restricting the number of states, transition rules, and activation scheme. Therefore, we can make use of Agent-Based Modeling that uses Complex Networks as an environment in an exploratory way to study the generated patterns and use them as a feature vector that allow us to classify the underlying environment, that is, classify networks by means of agents.

This works aims to explore the field of Pattern Recognition in Complex Networks, where the focus lies on modeling processes and interactions to extract valuable information from the generated data. By studying real-world network datasets and performing necessary experiments, we seek to enhance our understanding of the underlying structure of complex networks.

For the above reasons, network characterization is an important task in the field of Network Science. Despite the notable achievements of learning-based approaches, there remains a necessity for model-based solutions. This requirement becomes particularly evident in scenarios where the available training data is insufficient, which is often encountered in various practical domains. In light of these considerations, we propose a method for network characterization that takes advantage of the complex interactions that happens in an Agent-Based Model.

1.2 Objectives

The main objective of this thesis is the study, proposal and analysis of agent-based models focused on problems related to the characterization of complex networks.

The secondary objectives of this thesis are as follows:

- Investigate and analyze existing methods in the pattern recognition literature that deal with network characterization.
- Formulate and develop pattern recognition methods utilizing agent-based modeling.
- Design an agent-based model that can work in discrete environments, such as image textures or complex networks.
- Assess and compare the efficacy of the proposed methods in comparison to existing works in the literature.

1.3 Document organization

We organize this document as follows.

Chapter 1 *INTRODUCTION* presented the context, the motivation and the objectives of this work.

Chapter 2 *FOUNDATIONS* presents background knowledge important to the understanding of this work, including computational models, pattern recognition, complex networks, cellular automata, and agent-based modeling.

Chapter 3 *RELATED WORK* shows related works in the area, which have as objective the use of patterns to categorize complex networks.

Chapter 4 *GROWTH MODEL: AN AGENT-BASED MODEL FOR CATEGORIZING COMPLEX NETWORKS* describes the methodology used to model and simulate the processes that are performed in complex networks using agent-based models.

Chapter 5 *EXPERIMENTS IN NETWORKS* presents the classification results of the model into complex networks in synthetic and real-world datasets

Chapter 6 *CONCLUSIONS* presents the conclusions, and contributions of this work.

CHAPTER 2

FOUNDATIONS

2.1 Introduction

This chapter describes concepts related to computational modeling, and pattern recognition, using image textures, complex networks, cellular automata, and agent-based models. The chapter aims to provide an explanation of each topic, as well as their importance and relevance for this work.

Computational modeling is a powerful tool used to understand real-world complex systems across various scientific disciplines. It complements traditional approaches of theory and experiments. It involves creating mathematical or algorithmic representations of complex phenomena to simulate their behavior and study their dynamics. A model is designed for specific purposes, so it contains attributes and variables that help answer particular research questions (BIELIK *et al.*, 2021).

A pattern is everything that can be perceived containing a structure. Pattern recognition is the process of recognizing patterns by statistical or algorithmic approaches. The goal of pattern recognition is to extract meaningful information from the data by identifying structures, relationships, or trends. Pattern recognition can be applied to the task of classification and clustering. When applied to classification, pattern recognition seeks to categorize data in appropriate classes. Meanwhile, in clustering, pattern recognition seeks to group the data according to their properties (BISHOP, 2006).

Image textures refer to the visual patterns and structures found in images, such as the arrangement of pixels or the variations in color and intensity. In pattern recognition, textures provide valuable information for distinguishing between different classes or categories of images. By analyzing the spatial distribution and statistical properties of textures, it is possible to develop algorithms and techniques to automatically extract texture features from images (FARFÁN; SCABINI; BRUNO, 2019). These features can be used in tasks like object recognition, image classification, and anomaly detection, which have widespread applications in fields such as automated surveillance, environmental monitoring, and disease diagnosis.

Complex networks are representations of systems which contains elements that are connected by specific relationships. Networks are prevalent across numerous scientific disciplines, ranging from social sciences to biology, computer science, and physics. Complex networks provide a framework to analyze and model the structure and dynamics of complex systems. By representing objects as nodes and their relationships as edges, it is possible to capture the underlying structure and connectivity of complex data. This information is useful in real-life application, such as the spread of diseases in social networks, the flow of information on the internet, the structure of protein interaction networks, and the resilience of transportation networks.

Cellular automata are models consisting of elements, called cells, that by following simple rules can yield complex behaviors. Cellular automata have been applied in various areas, including physics, biology, computer science, and social sciences. They can be used to generate synthetic patterns, simulate pattern growth or evolution, and analyze the emergence of complex behaviors.

Agent-based models are computational simulations used to study the behavior and interactions of autonomous agents within a system. Each agent can perceive and respond to other agents and its environment according to a set of rules. Agent-based models are widely used in various domains, including ecology, economics, social sciences, and computer science. By employing agent-based models, we can gain a better understanding of emergent properties, collective behavior, and the impact of individual decisions on system-level dynamics.

2.2 Computational Modeling

Computational modeling is a tool for the simulation and understanding of complex systems. Complex systems contain interrelated elements that exhibit emergent behavior and patterns that are not easily predictable from the behavior of the individual elements alone (COSCIA, 2021). Computational modeling, also known as computer simulation or mathematical modeling, has been successfully applied to the study and analysis of complex systems in various scientific disciplines.

Computational models are employed in diverse fields, including physics, biology, economics, social sciences, engineering, and many others. They play a crucial role in scientific research, problem-solving, and decision-making by providing a means to study complex systems in a controlled and computationally efficient manner (DOWNEY, 2018).

Complex systems comprise a large number of interacting components with non-linear relationships and emergent behavior. The interaction between the elements of a computational model is crucial to understand certain phenomena. In a computational model, multiple variables are typically employed to characterize the complex system under study. These variables are adjusted individually or collectively, taking into account the final objective of the system.

This approach allows researchers to explore and analyze complex systems that are often challenging to comprehend through traditional analytical methods alone (GAO; XU, 2021).

In recent times, an important application of computational models has been in the modeling and prediction of the COVID-19 epidemic in the world (CURRIE *et al.*, 2020) and in particular in Brazil (SCABINI *et al.*, 2021). The applications of computational models are diverse and their capabilities continue to be explored, for example, in Chemistry to model realistic cell membranes (MARRINK *et al.*, 2019), in drug discovery (ADELUSI *et al.*, 2022), in Medicine to treat patients with heart-related problems (NIEDERER; LUMENS; TRAYANOVA, 2019), having big-data to provide personalized care (SOROUSHMEHR; NAJARIAN, 2022). It has also been applied to analyze soil fertility in Agriculture (HELFER *et al.*, 2020), in the Aerospace industry to design structural components (D'MELLO *et al.*, 2020), in conjunction with Machine Learning to address mechanical engineering applications (NGUYEN *et al.*, 2023).

In summary, using computational models in the form of computer programs facilitates the execution of experiments that simulate the behavior of complex systems. In addition, these systems can be simulated under different conditions in a controlled way.

2.3 Pattern Recognition

Pattern recognition is defined as the field of research that seeks the automatic discovery of important parts in data using computational models with a defined purpose (BISHOP, 2006).

Pattern recognition uses machine learning as there are now a variety of data sources, data processing power, and new approaches to study them. One type of pattern recognition is classification which seeks to categorize data by assigning it an identifier or label among a predetermined set of labels. This is a supervised learning approach in which the inputs with their respective identifications are provided.

The other type of pattern recognition is the grouping or clustering of similar data based on a measure of similarity. This is an unsupervised approach in which the input data does not contains labels to group the data.

Pattern recognition has been applied in various areas, in the statistical analysis of imbalanced data (ZHAI; QI; SHEN, 2022), to identify eye movements using signal processing (AHMED; NASRIN, 2022), in the analysis of textures in images (FARFÁN; SCABINI; BRUNO, 2019), in agriculture for the detection of plant diseases using computer vision (HARAKANNANAVAR *et al.*, 2022), to determine dominant movements in the behavior of crowds (MATKOVIC; IVASIC-KOS; RIBARIC, 2022), among others.

In network science, pattern recognition takes into account the understanding of the underlying network structure and their dynamic properties. Previous works in the literature (MERENDA, 2023; MIRANDA; MACHICAO; BRUNO, 2016a; RIBAS; MACHICAO; BRUNO, 2020; ZIELINSKI *et al.*, 2022) have demonstrated that networks can be categorized using computational models by using a dynamic process over the network to extract data that identifies them.

2.4 Image Textures

We can recognize the texture of an object through touch and vision. In image processing *texture* represents a local pattern that is repeated across an image. Texture provides useful information about regions of interest that share similar characteristics (GONZALEZ, 2016).

Texture analysis seeks to characterize regions in an image based on their structural content. It provides ways to quantitatively describe certain attributes that serve to describe the image in a qualitative way. Thus the texture in images comprises a series of descriptors that quantify the perceived texture of an image. In the image this information is in the spatial distribution of the pixels.

The study of texture is important in research areas of image processing and computer vision. It has been applied in task of classification and segmentation. Some approaches to study textures include the use of gray level co-occurrence matrices (GLCM) (HALL-BEYER, 2017), Gabor-based filters (OU *et al.*, 2016), the local binary pattern (LBP) (RAHIM *et al.*, 2013), etc.

In fact, applications of texture are present in diverse fields of science and industry, such as, in medicine to categorize X-rays medical images (NARAYAN *et al.*, 2023), to determining the severity of COVID-19 patients by computed tomography images (AMINI; SHALBAF, 2022), in agriculture to diagnose the nutritional status in coffee leaves (TUESTA-MONTEZA; MEJIA-CABRERA; ARCILA-DIAZ, 2023), to measure the habitat heterogeneity on tropical forest birds using satellite images (SUTTIDATE *et al.*, 2023), to classify different species of apples (ROPELEWSKA, 2021), to characterize and segment materials (HAN; YANG; CHEN, 2022), among others.

2.5 Complex Networks

Many complex phenomena can be studied and represented using networks. Network Science is an important and relevant field of research, which seeks to understand complex systems by analyzing the relationships and interactions between its components (SAYAMA, 2015).

Network science has become a data modeling tool for complex systems. In the industry, this way of thinking about data in an interconnected way has given rise to the network model known as the graph data model (BECHBERGER; PERRYMAN, 2020). Network analysis and graph theory concepts have profound importance in our world. Networks provide a different way of thinking about data. Since it is necessary to think about the data in terms of relationships, these relationships give each network its particular characteristics.

The theory of graphs started with the intuition of Leonhard Euler, in the 18th century, who studied the famous problem of the Königsberg bridges that crossed the Pregel River, where there was heavy ship traffic and a prosperous trade (BARABÁSI; WATTS; NEWMAN, 2006). The problem was to determine whether or not it was possible to cross the seven bridges starting from the same point and without crossing the same bridge more than once. Euler demonstrated analytically that this path was not possible using a graphical representation that consisted of a set of points, corresponding to the land bands, and a set of curves or lines, which connected these points.

After Euler, this area had important contributions from several mathematicians regarding the discoveries of graph properties. More recently, in addition to focusing on the study of these properties, other questions have arisen regarding the formation of graphs and how real networks are structured.

A network is a collection of points linked together in pairs. Many objects of interest in the physical, biological, and social sciences can be thought of as networks and thinking of them in this way can often lead to new and useful insights (NEWMAN, 2010).

Complex networks are an active area of research that uses concepts from graph theory. It applies methods from statistics and physics to characterize, model, and analyze its static and dynamic structure (CASANOVA, 2013). The term complex networks appears as a result of the work by Erdos and Rényi (ERDOS; RÉNYI, 1960) in small-world networks (WATTS; STROGATZ, 1998), free scale networks (BARABÁSI; ALBERT, 1999), and identification of communities in networks (GIRVAN; NEWMAN, 2002).

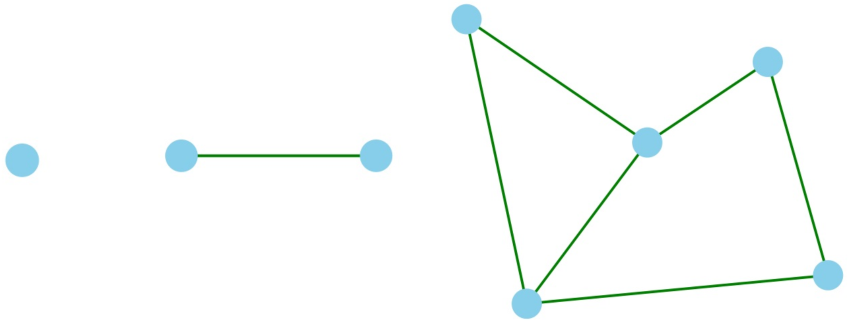
Complex networks have been applied in many research areas, such as to evaluate urban public transportation systems (WANG; WANG; SHEN, 2020), in physics to analyze a nuclear reactor (CERVI; CAMMI; ZIO, 2019), and to analyze images as complex networks (BAPTISTA; BACCO, 2021), in hydrology to assess the importance of stations (DEEPTHI; SIVAKUMAR, 2023), to identify influential nodes to spread information (GUPTA; MISHRA, 2021), tools to understand biological complex networks (BOGDAN *et al.*, 2021), This diversity of applications with different perspectives makes the complex network science an area of continuous growth.

2.5.1 Basics of Complex Networks

The study of networks, in the form of mathematical graph theory, is a fundamental pillar for the study of complex networks (NEWMAN, 2003). It all started with the study of graphs (DIESTEL, 2017).

A graph G is defined as a set of pairs (V,E), in which, V is a finite and non-empty set of vertices and E is a set of edges that represent the connections between these vertices, that is, $E \subseteq \{(u,v)|u,v \in V\}$. Vertices are usually called nodes, and edges are usually called links, or connections. Figure 1 shows the common visual representation of a node, and edge and a simple graph.

Figure 1 – Representation of (a) a node, (b) an edge, and (c) a simple graph.

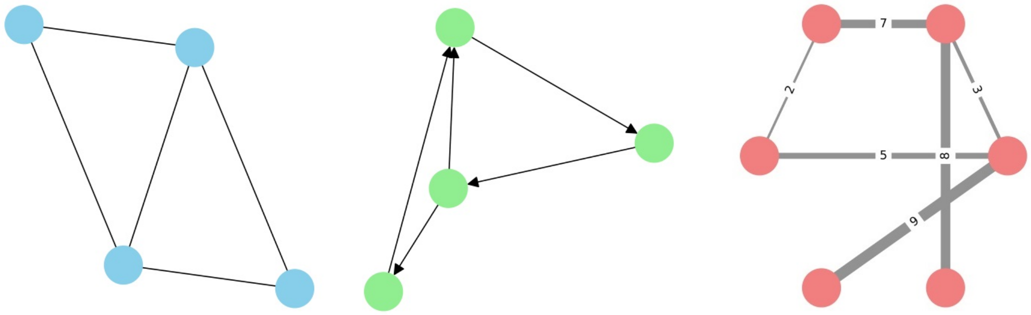


(a) (b) (c)

Source: Elaborated by the author.

Figure 2 presents basic types of graphs: undirected, directed and weighted. In an undirected graph an edge has no direction, and is represented as a single line. In a directed graph, the connections are not symmetric, edges have a direction so the relationship is in one way, and edges are represented with an arrow. In a weighted graph, the relationships have a strength, a numeric quantity that represents the weight of the edge. Weights can be interpreted as proximity or distance.

Figure 2 – Types of graphs: (a) undirected, (b) directed and (c) weighted



(a) (b) (c)

Source: Elaborated by the author.

The nature of complex networks to represent complex systems showing different structures, topologies, and features, makes possible to build a rich source of measures to characterize, compare, classify, and model them. There are measures and methods based on real properties of the system that characterize complex networks (COSTA *et al.*, 2007).

2.5.2 Network measures

For network analysis, several metrics can be used to study the structure and topology of complex networks. Metrics are quantitative measures that provide insights into various aspects of the network. The measures help in the understanding of characteristics, comparison and classification of complex networks. Among the measures based on connectivity, we have:

- **Degree** is the number of edges connected to a node. The degree only considers immediate neighbors. The degree k_{v_i} of a vertex v_i is calculated as the number of edges incident to other vertices, and expressed as:

  $$k_{v_i} = \sum_{j=1}^{N} A(i,j)$$

  in which $A(i,j)$ is the representation of a network as an adjacency matrix. The rows and columns of the adjacency matrix correspond to nodes in the network, and the entries in the matrix indicate the presence of an edge between pairs of nodes. In weighted networks, the weighted degree or strength s_{v_i} is obtained by the sum of the weights of the edges.

- **Average degree** is the average value of the degree of all vertices. It is defined as:

  $$\langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_{v_i}$$

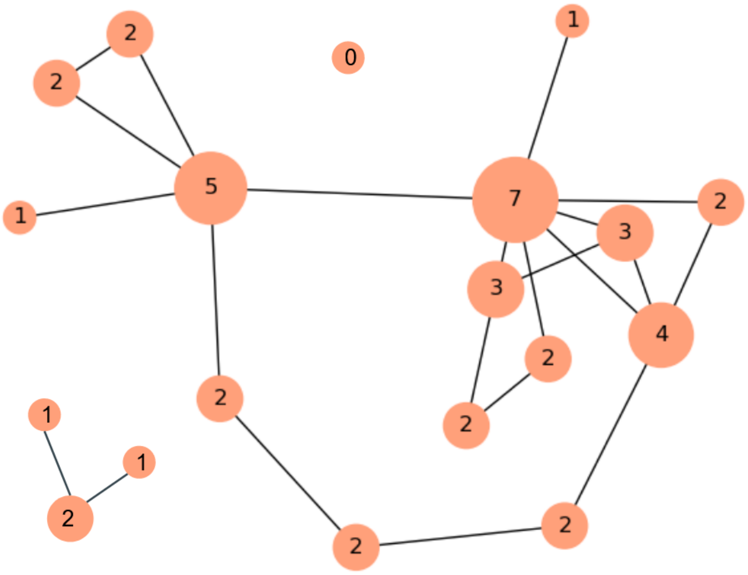
  In directed networks it can be computed the out-degree and the in-degree: the out-degree is the sum of edges leaving a vertex, and the in-degree is the number of edges entering a vertex. In directed and weighted networks it can be computed the out-strength and the in-strength: the out-strength is the sum of the weights of the outgoing connections and the in-strength is the sum of the weights of the incoming edges.

- **Degree distribution** describes the probability distribution of node degrees in the network. It is the probability $P(k)$ of a vertex to have degree k. Many real-world networks follow a power-law distribution, in which the probability $P(k)$ decreases as a power of k.

- **Degree difference** or degree correlation, is the difference in degrees between two connected nodes (FARZAM; SAMAL; JOST, 2020). If we have two nodes i and j connected by an edge, the degree difference δ between them is defined as:

  $$\delta = |k_{v_i} - k_{v_j}|$$

Figure 3 – A network where the numbers inside the vertices represent the degree of the node.



Source: Elaborated by the author.

Figure 3 represents a network, in which the vertices contains numbers that represents the degree. It is worth noting that there is one vertex with a degree value of zero, meaning that this vertex does not have any relation to another vertex.

Measures of centrality seek to identify the most important nodes in a network. Among the measures of centrality, we have:

- **Degree Centrality**: It measures the importance of a node based on the number of its connections.
- **Betweenness Centrality**: It quantifies how often a node acts as a bridge along the shortest paths between other nodes in the network. Its values lie between 0 and 1, where 0 means that there are no shortest paths and 1 that all shortest paths go through this node.
- **Closeness Centrality**: It measures how quickly a node can reach other nodes in the network. It is based on the average shortest path lengths.
- **Eigenvector Centrality**: This measure takes into account the importance of the neighbors of a node and is often used to find influential nodes.

2.5.3 Network Models

Networks were first studied in a deterministic way. The mathematicians Paul Erdös and Alfred Rényi started to consider them as stochastic objects. Subsequently, several studies have found patterns in many networks that were considered as structural characteristics.

Based on these structural characteristics and statistical properties, networks can be classified into different models, the three main models of complex networks in the literature are random networks, small-world networks, and scale-free networks.

- **Random networks** are models generated from random connections between vertices. One way to build a random network is starting with a set V of vertices initially disconnected and a set E of edges. Then, at each iteration, an edge of E is drawn at random, thus connecting a pair of vertices. A second way consists of the definition of a probability p of connection between the vertices of the network. Let N be the total number of vertices of the network, for each vertex $i \in V$, there are $N-1$ distinct possibilities of connection with other vertices. (ERDOS; RÉNYI, 1960).
- **Small-world networks** are models characterized by small paths between vertices. Many vertices can be reached by others through a small number of edges, for example in social networks. It presents cycles in more proportion than in random networks (WATTS; STROGATZ, 1998).
- **Scale-free** are models in which few vertices have a high degree compared to the rest of the network. Few vertices tend to have many connections, and most vertices have few connections. The degree distribution in scale-free networks follows a power law (BARABÁSI; ALBERT, 1999)

2.6 Cellular Automata

Cellular automata (CA), the plural form of cellular automaton, is a computational modeling approach that involves simulating the behavior of a system composed of individual cells or units, each following a set of predefined rules. These cells are arranged in a grid or lattice, and their state evolves over discrete time steps based on the rules and the states of their neighboring cells. Typically, all cells of a cellular automata update their states simultaneously.

Cellular automata can be defined as a discrete, spatially distributed computational model comprising a grid of cells, each of which can take on a finite set of states. The evolution of the system occurs in discrete time steps, where the state of each cell at a given time depends on its own state and the states of its neighboring cells, according to a predefined set of rules (SAYAMA, 2015).

CA is a powerful tool used to study the emergence of complex patterns and behaviors from simple local interactions, making it relevant across diverse scientific disciplines (WOLFRAM, 2002). These emerging patterns can range from simpler structures, such as homogeneous, stable, or periodic, to chaotic patterns (WOLFRAM, 1984). Even looking simple, CAs often display interesting behaviors and have been applied to modeling biological phenomena (MORDVINTSEV *et al.*, 2020), to recognize handwritten patterns (WALI; SAEED, 2018), to model the spatial dynamics of urban growth (CAO *et al.*, 2020), it even has applications in cryptography (BHARDWAJ; BHAGAT, 2018).

The concept of cellular automata was first introduced by mathematician John von Neumann and his colleague Stanislaw Ulam in the 1940s (MITCHELL, 2009). However, it was John Conway's "Game of Life" introduced in 1970, that popularized cellular automata and led to extensive research in the field. The Game of Life demonstrated how simple rules could lead to the emergence of complex and fascinating patterns, capturing the attention of researchers from various domains (MARTÍNEZ; ADAMATZKY; SECK-TUOH-MORA, 2022).

A cellular automata consists of the following components (HOEKSTRA; KROC; SLOOT, 2010):

1. Cells: The fundamental units of a cellular automata are the cells, which are typically arranged in a regular grid or lattice. Each cell represents an individual entity and can take on one of a finite number of discrete states.

2. Neighborhood: The neighborhood of a cell refers to a set of cells that are considered neighbors when determining the next state of a central cell. In two-dimensional CA, the most common types of neighborhoods include the Moore Neighborhood, in which all eight cells surrounding the central cell are considered, the von Neumann Neighborhood, in which only the four cardinal direction cells adjacent to the central cell are considered, and extended neighborhoods like an hexagonal neighborhood.

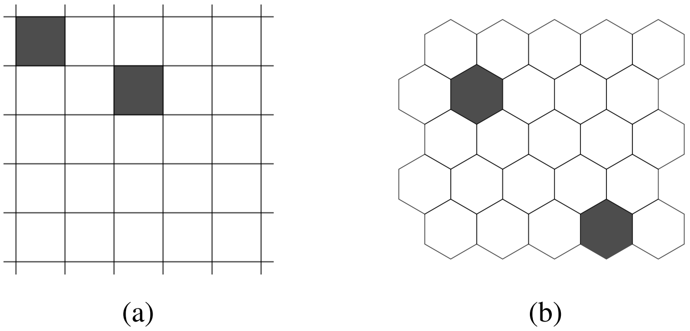
3. State Transition Rules: Cellular Automata are governed by a set of rules that determine how the state of each cell changes over time. These rules are typically specified in the form of a transition function, which maps the current state of a cell and the states of its neighbors to its new state at the next time step.

4. Time Steps: Cellular Automata evolve over discrete time steps, where the state of the entire grid is updated simultaneously according to the transition rules. The states of the cells at time $t+1$ depend on their states at time t and the states of their neighbors at time t.

5. Boundary Conditions: The behavior of cells at the edges of the grid needs to be defined to avoid edge effects. Various boundary conditions, such as periodic boundary conditions or zero-padding, can be employed.

A CA is a model of the world, in which the world is divided into discrete spaces or cells. This space or tessellation T can be composed of cells of any shape: triangular, square, hexagonal, or even irregular. Implicitly, it is assumed that the cells are homogeneous, that is, they have the same properties and follow the same transition rules of states (SHALIZI, 2006). Figure 4 shows a representation of a cellular automata in a regular and a hexagonal grid. The color of the cells represent their state, with two states: black as "alive" and white as "dead". The image also represents the boundary conditions, the regular grid is unbounded, and the hexagonal grid is bounded.

Figure 4 – Representation of a cellular automata in a space: (a) two cells in a regular grid, and (b) two cells in a hexagonal grid



Source: Elaborated by the author.

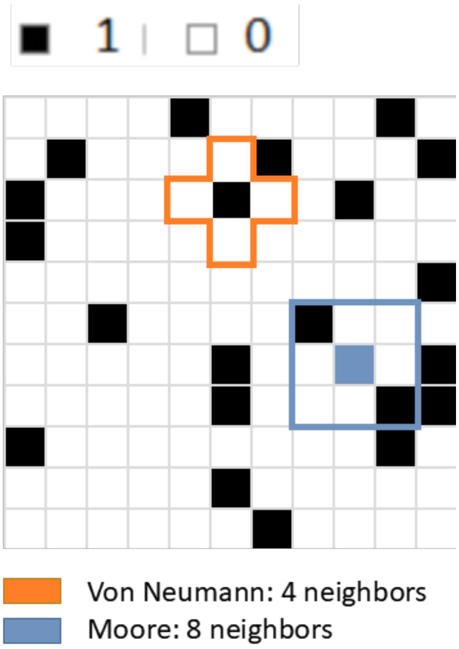
A formal definition of a cellular automata (BAETENS; BAETS, 2012) can be given by the tuple $C = \langle T, S, f, N, \Phi \rangle$, in which the elements are:

- T is an tessellation of dimension n in the Euclidean space $\mathbb{R}^n$, which consists of cells c_i, $i \in \mathbb{N}$
- S, is a set of k states, and $S \in \mathbb{N}$
- f, is the transition function, such that $f : \mathbb{N} \to S$. This function determines the state of the cell c_i at time t, that is, $s(c_i,t)$
- N, is a function that defines the neighborhood of each cell: $N : T \to \bigcup_{p=1}^{\infty} T^p$. This function maps each cell c_i to a finite sequence $N(c_i) = (c_{ij})_{j=1}^{|N(c_i)|}$ consisting of $|N(c_i)|$ distinct cells
- $\Phi = (\phi_i), i \in \mathbb{N}$, is a set of functions such that $\phi_i : S^{|N(c_i)|} \to S$. Each ϕ_i governs the dynamic of the cell $c_i : f(c_i,t+1) = \phi_i(f(c_{ij},t)_{j=1}^{|N(c_i)|})$.

Figure 5 shows a cellular automata with its characteristics, the tessellation T is a bi-dimensional grid, the states that each cell can have is 1 or 0, each cell can have 4 or 8 neighbors, von Neumann or Moore neighborhood, respectively. The initial configuration and transition rules are specific for each CA.

With their apparent simplicity, certain CAs possess universal computational capabilities, enabling them in principle to perform any computation achievable by a computer (COOK, 2004). Among the types of cellular automata we have elementary cellular automata, two-dimensional cellular automata, and continuous cellular automata. These will be described in the following sections.

Figure 5 – A cellular automata defined over a bidimensional grid, with states 1 or 0, black cells are alive and white cells are dead, showing possible neighborhoods von Neumann and Moore.



Source: Elaborated by the author.

2.6.1 Elementary Cellular Automata

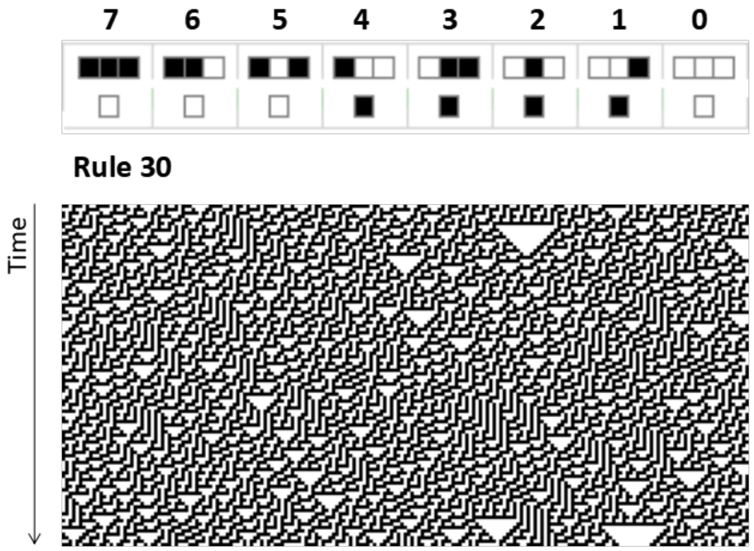
The elementary cellular automata (ECA) represents the simplest class of CAs. Defined over a one-dimensional space, each element or cell can be in one of two possible states: 0 or 1. Given an initial random configuration, at each time step, the state of each cell is updated according to local rules that consider the state of its two closest neighbors and the state of the cell itself.

An ECA is defined by the quintuple: $C = \langle Z^1, S = \{0,1\}, f, N = (c_{i-1}, c_i, c_{i+1}), \Phi : S^3 \to S \rangle$, in which $r = 1$ is the radius of the neighborhood. In this way, combining the three analyzed cells and the number of states, there is a total of $|S|^N_{(c_i)} = 2^3 = 8$ possible neighborhood configurations to be analyzed with each evolution of the CA. Thus, there are a total of $2^8 = 256$ transition rules. The rules can be summarized by a table, it maps the state of the neighborhood to the next state in one time step, each cell influences the state of one neighbor.

The evolution of the CA is normally visualized in bidimensional space-time diagrams whose columns represent the cells of the CA and the lines the configuration of each cell state in each time. Figure 6 shows the transition table for rule 30 of an ECA, and the space-time pattern generated over time. Rule 30 is a specific rule set for elementary cellular automata, and it is particularly famous for its complex and seemingly random behavior (WOLFRAM, 2002).

An interesting feature is the pattern that emerge from the dynamics of automata evolution. Wolfram categorized these patterns into four distinct classes: (1) Stable: when the initial configurations of CA cells quickly converge to a single state 0 or 1, (2) Periodic: when the states

Figure 6 – ECA: Rule 30 and space-time pattern evolution.



Source: Elaborated by the author.

of the cells fluctuate with each evolution, (3) Chaotic: in this case, the cells show apparently random and non-periodic behavior, with a strong dependence on the initial states of each cell, and, (4) Complex: the emerging patterns of this class of CAs are characterized by presenting stable structures that can survive for a long period of time, but there are also structures that interact in a more complex way, and from this interaction, new patterns can be created, destroyed or propagated.

2.6.2 Two-dimensional Cellular Automata

The two-dimensional cellular automata are defined on tessellations of the Euclidean space $\mathbb{R}^2$. A well-known class of two-dimensional CAs are Life-Like cellular automata, which are inspired by the rules of the Game of Life. They are defined on the same topology and also in the same neighborhood.

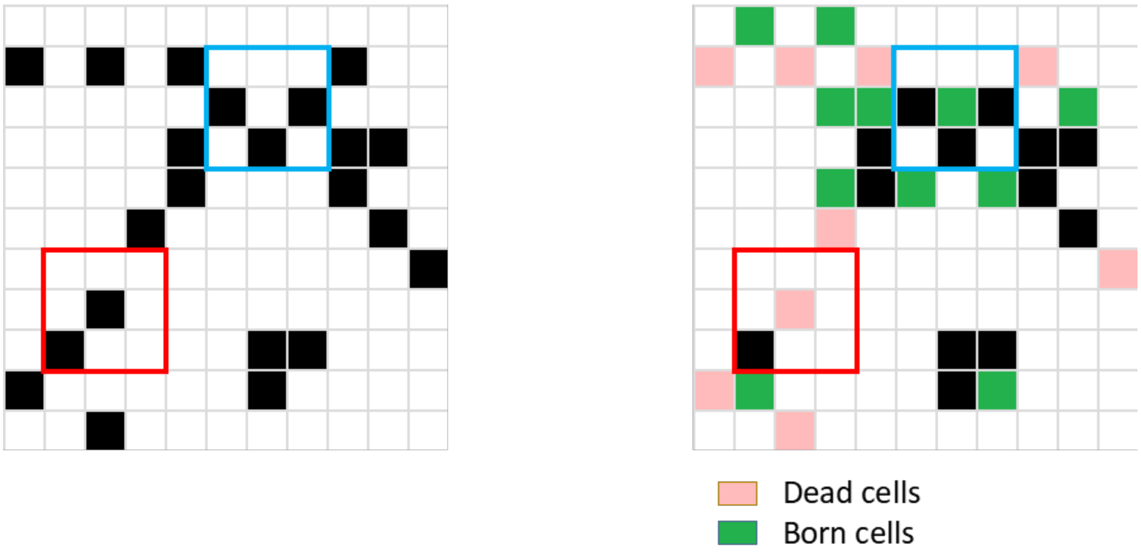
John Conway developed the Game of Life, it follows simple rules and can produce interesting patterns. Gardner (1970) popularized the Game of Life. The Game of Life takes place on a two-dimensional orthogonal grid. A cell in the grid has two states, alive or dead, and eight neighbors. Each cell interacts with its neighbors and at each time step, the new states are defined by transition rules related to the density of neighbors. The evolution of these transitions gives rise to interesting patterns and particular properties.

In Game of Life, the next state of each cell depends on its current state and the number of living neighbors. If a cell is alive, it remains alive if it has 2 or 3 living neighbors. If a cell is dead, it remains dead unless there are exactly three living neighbors. Given the Game of Life transition rules, we can describe them by the following dynamic system (DEMONGEOT; GOLÈS; TCHUENTE, 1985):

$$x_i(t+1) = \begin{cases} 1, & \text{if } \begin{cases} x_i(t) = 0 & \text{and} \quad \Sigma_{j=1}^{8} x_j = 3 \\ x_i(t) = 1 & \text{and} \quad 2 \leq \Sigma_{j=1}^{8} x_j \leq 3 \end{cases} \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

The following notation is also widely used to represent this system: B3/S23, that is, a cell is born (B) if it has 3 alive neighbors, and, a cell survives (S) if at least 2 or 3 neighbors are alive. Figure 7 shows two instants of time Conway's Game of Life. From the configuration on the left, the blue square shows that the dead cell is surrounded by 3 living cells, so in the following time, it will come to life. On the other hand, the red square shows a living cell that only has one living neighbor, so in the next instant of time the cell will die.

Figure 7 – Conway's Game of Life. Black cells are alive, and white cells are dead. A dead cell requires 3 living cells to be born, and a living cell requires 2 or 3 living cells to survive.



Source: Elaborated by the author.

2.6.3 Continuous Cellular Automata

Continuous cellular automata (CCA) differ from their discrete counterparts as they allow cells to assume continuous values instead of discrete states. CCA models are commonly used in fields like physics and biology to simulate continuous processes. Unlike the discrete case, CCA is defined by partial differential equations that describe the local interactions between neighboring cells. CCA is particularly useful for modeling systems with continuous spatial distributions, such as fluid dynamics or reaction-diffusion processes.

A continuous cellular automata consists of the following components (WOLFRAM, 2002):

- **Space**: The system is represented by a continuous spatial domain rather than a discrete grid. Each cell now corresponds to a location in continuous space.
- **Variables**: Cells can have continuous values, often representing physical attributes like temperature, density, concentration, etc. These values can change smoothly over space and time.
- **Dynamics**: Instead of discrete time steps, time is treated as continuous. Cell values evolve gradually and continuously based on differential equations or other mathematical models that describe how the variables change.
- **Neighborhood**: Neighboring cells in the continuous space are defined based on proximity, usually using a kernel function or a distance metric.

2.7 Agent-Based Models

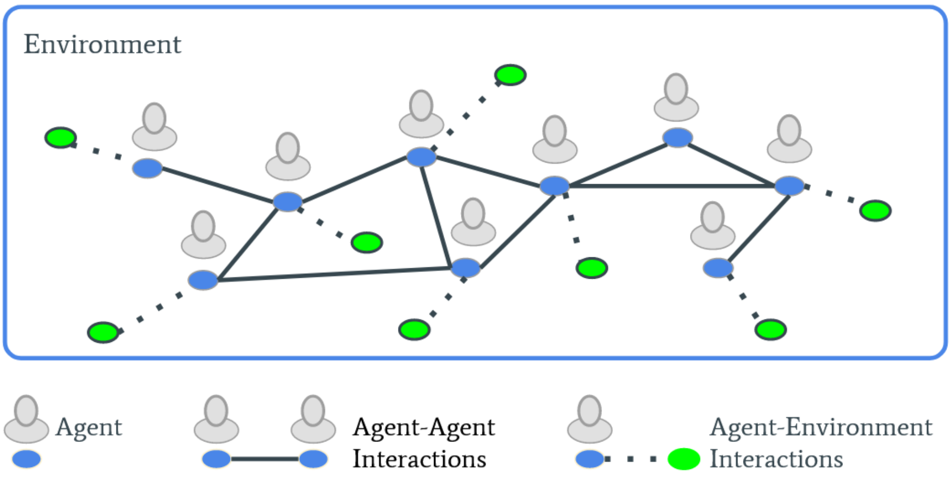
A characteristic of complex systems is the emergent behavior that results from collective and self-organizing dynamics and the absence of a central controller (BOCCARA, 2010). Among the mathematical tools for modeling complex systems are methods based on differential equations and methods based on agents. The former uses population densities, the latter deals with spatially distributed agents, representing the elements of the system.

Agent-based modeling (ABM) is a computer simulation that consists of agents interacting with one another, in order to study an overall system. Especially in ecology, ABMs are also called individual-based models (IBM). In the computational simulation, the agents are autonomous, they react to their environment and other agents using predefined rules (GRIMM *et al.*, 2006). In a simulation, the interactions of agents are monitored to see their behavior over time. At each moment, every agent acts according to its current state, the state of the world around it, and the rules that govern its behavior. The agents can be as diverse as needed. Agents can be used to represent living cells, animals, individual humans, even entire organizations, or abstract entities. Figure 8 shows a representation of an ABM, in which the solid lines represent the interaction between agents and the dashed lines represent interaction of agents with the environment.

ABMs can handle a wider range of non-linear behaviors than conventional models. ABMs are a recognized approach to studying complex systems and have been applied in various areas of research. ABMs are useful modeling tools to study processes involving stochasticity, nonlinear interactions, and/or heterogeneous spatial structures (BODINE *et al.*, 2020).

The applications of ABMs are diverse, among them we can mention the following: To simulate the migratory movement of birds to determine critical migration sites and unexpected movement patterns (AURBACH *et al.*, 2020). To model the complex and seemingly unpredictable dynamics of a stock market (VANFOSSAN; DAGLI; KWASA, 2020). To model the spread and impact of diseases, such as the case of malaria transmission (GHARAKHANLOU; MESGARI;

Figure 8 – The elements of an agent-based model and its interaction between them and with the environment.



Source: Elaborated by the author.

HOOSHANGI, 2019). To evaluate the effects of the lockdown of the COVID-19 epidemic on the population (KHALIL; FATMI, 2022). To model and propose solutions for the evacuation of people in the event of fires in buildings (KASEREKA *et al.*, 2018) or public services like the metro system (LO *et al.*, 2014). It has applications in fisheries science (HAASE *et al.*, 2023). In image processing to enhance the contrast of images (LUQUE-CHANG *et al.*, 2023). To model and simulate airport flights (MA *et al.*, 2023).

In the literature the Schelling model holds significant importance in agent-based modeling for being one of the first models for understand social dynamics and patterns of segregation. It was proposed by the economist Thomas C. Schelling in 1971 (SCHELLING, 1971), the model has been applied in several domains, including urban planning (GONZÁLEZ-MÉNDEZ *et al.*, 2021), sociology, and economics, even now it continues to be the basis for other models, such as approaches to fairness and altruism (FLAIG; HOUY, 2019), or intolerance in the face of resource scarcity (JANI, 2020).

The importance of the Schelling model lies in its ability to illustrate how individual preferences and behaviors can lead to macro-level patterns of segregation. The model explores the emergence of segregation based on simple decision-making rules of individuals, without assuming any inherent bias or explicit discrimination.

There are many interesting ABMs in the literature such as:

- Predator-prey dynamics, the agents are sheep and wolves, sheep eat grass, wolves eat sheep, and they both reproduce.
- Political dynamics, the agents are the states that are fighting each other in alliances.
- Migration behavior for displaced people

- Epidemiological simulation, in which an infectious disease spread over a road of networks and the way people move.

2.7.1 Designing ABMs

The main challenge in using ABM is to define the behavior of the agents and select the rules they use for making decisions. This is done in many cases using common sense and guesswork, which is not enough to imitate true behavior. Trying to model all of the details of a real problem can lead to complicated solutions in which the effect of agents on the environment is difficult to determine. To effectively use ABM, it is needed to proceed in a systematic way to check assumptions of the model and implement additional complexity only when appropriate.

There are ways to design and develop an ABM, the steps according to Sayama (2015) are:

- Define the environment in which the agents will interact. The environment can be modeled over a discrete or a continuous space, a discrete one can be an image, or a network, a continuous environment can be an euclidean space, or a street map space (DATSERIS; VAHDATI; DUBOIS, 2022).
- Define the agent types that will be placed in the environment. The agents contain attributes that help to identify and locate it in the environment, as well as any other attribute according to the purpose of the model.
- Define the model properties, like the scheduler, that controls when the agent will be active (COMER, 2014).
- Define the functionality of agents along the simulation. It can be needed to provide functions that act on each agent, and a function that acts on the model as a whole.
- Measure and visualize any specific data generated from the model, this, in order to check the assumptions of the model.
- Collect data generated during the simulation of the model

2.7.2 Impact of the scheduler in ABM

The “activation scheme”, also known as “scheduling” or “updating”, is a very important decision in the design of agent-based models. The **scheduler** is the component of ABMs that controls the order of activation of the agents. The “activation scheme” determines the conditions and the sequence of interaction of the agents.

There can be many types of activation schemes. The type of activation chosen has a significant impact on the emergent population patterns and dynamics of the model. The activation can be synchronous or asynchronous, and the latter can be uniform (activated by type and random), or domain specific (COMER, 2014). Therefore, it is important to specify the activation scheme in the design of agent-based models (COMER; LOERCH, 2013). The following activation types are usually implemented in specialized software:

- **Sequential activation**: The agents are activated according to the order they were created in the model.
- **Simultaneous activation**: A synchronous activation, the simulation behaves as if all agents have been activated simultaneously, as in the case of cellular automata.
- **Random activation**: The agents are activated following a random order, usually there is a shuffle in the list of agents before they are activated.
- **Random activation by type**: The agents are activated according to their type, a type of agent runs first, and then a second type of agent runs.
- **Domain specific**: The agents can also be activated using rules that are specific to an application domain.

Different activation schemes can lead to diverse model outcomes. The study by Comer (2014) analyzed the effects of various activation schemes on the models' behavior of three agent-based models, the authors found that in some cases the activation scheme produces notorious variations in outcomes or overall model behavior. Alizadeh and Cioffi-Revilla (2015) analyzed an agent-based opinion dynamics model, and Mudigonda *et al.* (2022) analyzed a variant of the standard agent-based Susceptible-Infected-Recovered-Deceased (SIRD) contagion model. Both of these studies demonstrate qualitative and quantitative differences between the activation regimes. Consequently, the choice of the activation scheme can help to explore and discover behaviors and patterns in agent-based models, which allows a better understanding of the complex system under study.

2.7.3 Importance of the ABM framework

A simple ABM can be implemented using the default tools provided by a programming language; however, when more sophisticated ABM models are required, it is better to use specialized tools that can handle the intrinsic complexities of ABMs. Generally, tools for implementing ABMs offer components for modeling, analysis, and visualization (ANTELMI *et al.*, 2023).

There are various tools that provide building blocks for ABMs. Among the various general purpose tools we have *NetLogo*, *Mesa*, *Agents.jl* and *krABMaga*.

The well-known *NetLogo* framework is useful to learn and explore concepts of ABM, but as the complexity increases it is necessary to look for another solutions. *NetLogo* provides a random activation scheme (WILENSKY; RAND, 2015).

Python has various implementations of agent-based models frameworks, the one that is more consolidated is *Mesa* (KAZIL; MASAD; CROOKS, 2020). *Mesa* provides different activation schemes.

Agents.jl is a framework written entirely in the Julia language, which means that its execution is usually faster than *NetLogo*, or *Mesa*. It has capabilities to simulations in geographical maps (DATSERIS; VAHDATI; DUBOIS, 2022).

krABMaga is another framework to develop ABMs, it is written in the Rust programming language. Being written in a modern and low-level programming language, *krABMaga* becomes one of the frameworks that best uses the computational resources in addition to its efficient execution, being the fastest of the frameworks previously mentioned (ANTELMI *et al.*, 2019).

2.8 Considerations

The methods described of walkers, cellular automata, and agent-based models, have been applied in the analysis of complex networks. Table 1 presents a comparison of the methods described.

Table 1 – Comparing computational methods used to characterize complex networks.

Method	**Element name**	**Activation type**	**Element type**	**Elements in a position**	**Interaction between elements**
Walks	Tourist	Random	Unique	Multiple	No
CA	Cell	Simultaneous	Unique	Single	No
ABM	Agent	Sequential Simultaneous Random Random by type Domain specific	Multiple	Multiple	Yes

Source: Elaborated by the author.

The tourist in a walk and the cells in a cellular automata are elements of the same type, while the agents in agent-based models are elements that can be of different types. The methods based on walkers and automata have no interaction between them. On the other hand, agent-based models work with more than one type of agent, which interact with each other and the environment. In addition, agents have various kinds of activation.

In a complex network, walkers use the degree measure of a node to find its new position in the network according to some rules. Cells in cellular automata use node degree information to update their state without changing their position. Agents in ABMs can use the degree information as well as the environment information to interact with other agents, thereby updating their status and position in the network.

CHAPTER

3

RELATED WORK

3.1 Introduction

Pattern recognition is an area of research that includes several tasks in which a machine learns to identify patterns or regularities within data. Pattern recognition can involve classification, as well as as clustering, anomaly detection, and regression.

Networks are useful for modeling interconnected data, providing a powerful framework to represent and analyze complex relationships, dependencies, and interactions among various entities or components in a system. The data stored in networks is not structured, so it is needed proper approaches to find, analyze, and get important information from it. Since a method is designed for a specific task that depends on the type of data under consideration, performing pattern recognition in networks requires specific and usually novel approaches.

Classification is a task where the goal is to categorize input data into predefined classes or categories. The classification of networks using pattern recognition techniques can provide ways to get insights into the inherent patterns associated with the structure of the network.

3.2 Pattern Recognition in Networks

Recognizing patterns in a network involves extracting a set of features to form a feature vector that, when given as input to a machine learning algorithm, enables the classification of that network.

There have been various approaches to categorize complex networks. These approaches can be differentiated between those based on the use of measurements extracted from the networks and those based on the use of dynamic information generated by executing a process on the networks.

The nodes and the edges determine the relationships of a network, therefore they influence the dynamic processes that are executed on the network. Thus, it is necessary to obtain measurements of the specific structural characteristics of each network in order to analyze and classify the networks (COSTA *et al.*, 2007). In this case, the feature vector comes from the measurements of the network.

We can use a dynamic process on the network to obtain a feature vector of the network. The process depends on the type of elements that interact on the network. In walks,random and deterministic, the elements are walkers that move in the space according to rules and certains conditions (SILVA; ZHAO, 2016). In cellular automata, the elements are cells that change their state according to the state of neighboring cells. In agent-based models, the elements are autonomous agents that move in the environment, interacting with other agents and with the state of the environment.

The processes that use information from the environment to obtain the next state of the elements are known as spatially explicit models (SEM). Random walkers generally do not use information from the environment for their movement, they only move according to a certain probability. In each case, the feature vectors are obtained by studying the dynamics that arise from the activation of the elements on the network.

Several methods have been proposed in the literature, and the following sections provide a description of some relevant methods for this work.

3.2.1 Structural Measures

Networks can be analyzed using their own characteristics in a static way. Just as networks can be analyzed taking advantage of their dynamic properties. There are several measurements that can be extracted from a network (COSTA *et al.*, 2007). These measures are used as a feature vector in classification tasks. Appropriately combined, these measures make up the set of feature vectors in classification tasks.

Structural measures used to analyze networks can refer to the connectivity of the network, such as average degree, degree distribution, and degree difference. In the context of pattern recognition, several structural measures can be correlated (COSTA *et al.*, 2010).

It is also possible to use measurements related to distance, in networks distance is important because reflects the overall network structure. Among the distance measures we have the number of edges in a path, the geodesic path, and average geodesic path.

3.2.2 Life-Like Network Automata

The work by Miranda, Machicao and Bruno (2016a) presents the Life-Like Network Automata (LLNA). It is a cellular automata modeled on a network, where the transition rules are based on the rules of the Game of Life. A regular structure considers a maximum number of neighbors to apply the transition rules, while, a network has an irregular structure, this means that some nodes can have a large number of neighbors, and many nodes can have few neighbors, so in LLNA they use the density of neighbors when applying the transition rules. The feature vector was defined from the pattern formed in each node, which corresponds to the changes in its state over time, which they called Time Evolution Patterns (MIRANDA; MACHICAO; BRUNO, 2016b).

The LLNA works on a network and each node of the network represents a cell or an automaton. In formal terms, the LLNA is formed by an irregular tessellation T, and each node of the network is considered a cell with k_i neighbors, which a set of initial states s_0, only varies in a certain set of states S and follows a rule of transition Φ that determines the dynamics of the cells, in the form of a tuple:

$$C = (T, S, s_0, N, \Phi)$$

Let $S(c_i,t)$ be the state of cell c_i at time t, each cell can be in one of two states, $S = 0$ for dead and S= 1 for alive. The transition function at time $t+1$ is defined by $\Phi : S(c_i,t) \to S(c_i,t+1)$. Each cell can have 3 transitions, and each transition depends on the density σ of alive neighbors using the adjacency matrix A. Each cell is influenced by its neighbors, and their density is the main factor in the transition rules. There are separate rules for birth and survival. If it has the appropriate neighbor density to be born and is currently dead, it will come to life. If it has the appropriate neighbor density to survive and is currently alive, it will remain alive. However, if it lacks the neighbor density to either be born or survive, it will remain dead. In other words, if it is dead in time t, it will be born in time $t+1$, if it is alive in time t will survive in time $t+1$, in other case the cell dies.:

$$\sigma(c_i,t) = \frac{1}{k_i}\sum_{j=1}^{n} A_{ij}S(c_j,t)$$

The work by Ribas, Machicao and Bruno (2020), LLNA-LBP, adapts the work of LLNA and adds the concepts of the Local Binary Pattern (LBP) descriptor used in texture image analysis. Another work, related by their use of a Spatially Explicit Model in Cellular Automata is presented by Florindo and Metze (2021) that uses the LBP descriptor that provides the transition function for the automata. The work by Zielinski *et al.* (2022), D-TEP, uses LLNA and provides different ways to extract the time evolution patterns formed by the simulation

3.2.3 Random Walks

This type of walkers has its origin in the well-known Traveling Salesman Problem (TSP) (STANLEY; BULDYREV, 2001), in which the salesman must determine the shortest possible route that allows him to travel a set of cities spread over a space.

A random walk in networks is a stochastic process where a “walker” moves from one node to another following a set of rules determined by probabilities. At each step, the walker chooses a neighboring node to move to based on predefined probabilities associated with the edges of the network.

Random walks uses walkers that moves over a network. In this case, each automaton walks over the network going from an initial node to a neighboring node at random according to a transition probability. As it is a special case of Markov chain, the generated stochastic process can be reversed. Due to the nature of the process, the random walk can move to already visited nodes. When this behavior is restricted so as not to repeat nodes, the random walk avoids nodes already visited, this type of walk is called self-avoiding walk.

3.2.4 Deterministic Walks

Deterministic walks in a network refer to sequences of movements of a walker across nodes that is determined by specific rules and deterministic processes, as opposed to random walks that involve probabilistic choices. In deterministic walks, the next node to visit is uniquely determined based on the current node and the defined rules for movement. Lima, Martinez and Kinouchi (2001) present a local optimization problem named Traveling Tourist Problem, in this case, the tourist has the restriction of minimizing the distance to go to the next city, and not obtaining the shortest possible route from all the cities like in TSP.

A Deterministic Tourist Walk (DTW) models the movement of a tourist through a network traversing the nodes in a deterministic manner. The movement of the tourist is affected by its environment. Over an image a tourist considers the difference among neighboring pixel values, it can be the minimum or maximum difference (BACKES *et al.*, 2010). In networks the movement can be based in the difference of the degree of neighboring nodes (GONÇALVES; MARTINEZ; BRUNO, 2012).

The activation of the tourists is in a random fashion, because a tourist does not interact with neighboring tourists. For this reason, there can be many tourist in a single position, but their behavior is independent of every other tourist in the environment. The work by Merenda (2023) uses deterministic tourist walks to analyze and categorize complex networks.

3.2.5 Spatially explicit Agent-Based Models

Agent-based models are computational models that simulate the behavior and interactions of autonomous entities, referred to as agents, within a given environment. Each agent has its own set of rules and behaviors, which collectively produce emergent phenomena at the system level. Several ABMs have been modeled using networks as the agent communication structure.

An spatially-explicit model, reflects the effect of the environment and the interaction of the elements, therefore the dynamic is constrained by the environment. The spatially explicit data of the environment can be incorporated into the model, this could reveal the effects of the spatial characteristics of the agents (DEANGELIS; GRIMM, 2014).

Agent-based models naturally use networks as their environment for agent interaction. It has been seen that ABM have real applications such as the examples presented in the previous chapter. ABMs are usually modeled to simulate a real phenomenon. Other works uses ABMs with specific purposes to analyze the network, for example the work by Jin *et al.* (2009) uses ABMs to clustering Complex Networks.

3.3 Considerations

Several methods have been proposed in the literature that serve as the basis for the present work. This work seeks to explore the capabilities of agent-based models to classify networks. Unlike the methods based on tourists and cellular automata, the methods based on ABMs can have more than one type of agent interacting on the environment, as well as there can be more than one agent in the same position, and the activation of agents can occur in various ways.

The works in the literature, walkers and cellular automata, use only one type of element to explore and analyze a network. ABMs do not have restrictions on the type of elements, number of states, transition rules and type of activation. Besides that, in the literature, agent-based models have not been used for pattern recognition and network categorization.

CHAPTER

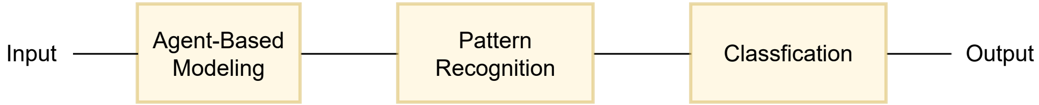
4

GROWTH MODEL: AN AGENT-BASED MODEL FOR CATEGORIZING COMPLEX NETWORKS

4.1 Introduction

This chapter presents the methodology employed in this work. Figure 9 shows a visual representation of the outline of this work. The purpose is to model a dynamic process that progresses within a given environment in order to obtain valuable information that aids in characterizing the environment. It consists of three stages: Agent-Based Modeling, Pattern Recognition, and Classification.

Figure 9 – The outline for this work consists of 3 stages: (1) Agent-Based Modeling, (2) Pattern Recognition, and (3) Classification.



Source: Elaborated by the author.

In this work, the primary input consists of complex networks datasets. In pattern recognition works, complex networks are modeled to form the underlying structure for computational models, like cellular automata and agent-based models. The initial stage of agent-based modeling, focuses on the design and modeling of the interactive process. The objective of this phase is to establish a comprehensive understanding of the underlying dynamics within the environment.

The second stage of pattern recognition, includes the execution of simulations and the generation of patterns based on the proposed model. In this stage, patterns are extracted, and features are constructed from the generated data. The process involves identifying meaningful structures within the data.

Finally, the third stage of classification, involves the evaluation and application of the generated features in the context of a classification problem. With the aim to assign appropriate labels to the analyzed patterns, to characterize the underlying environment of complex networks. The main metric to evaluate the model performance is accuracy. Other useful measures for the classification process are also presented, such as the confusion matrix, precision, recall, and AUC-ROC curve. The Validation Strategy is the Repeated Stratified k-fold Cross Validation.

This work is motivated by the lack of scientific work exploring the use of agent-based models for network classification. ABMs is a promising area to explore and research. An ABM can capture the inherent properties of the environment under study, i.e. complex networks. By using agent-based models in the simulation, we can explore the characteristics of the environment and discover valuable data that contributes to the characterization and understanding of the underlying environment.

The ABM was initially modeled on texture images, then a transformation was done to adapt it to work with complex networks. The modeling process was conducted on images due to their inherent two-dimensional structure and the ability to easily visualize the values of each pixel. This allowed for intuitive interaction with the image, enabling the addition or removal of data and providing visual feedback of the ongoing process. However, networks, lacking a fixed position in space, posed a distinct challenge. Consequently, to model this process in another environment, the modeling approach was adapted to incorporate the characteristics of complex networks.

An image can be interpreted as a surface, where the values assigned to each pixel represent the height of the surface, the height represents the metric in texture images. In the case of gray scale images, these values range from 0 to 255. From this concept of elevation, it becomes possible to place objects onto the surface, where their behavior is influenced by their respective height. This notion gave rise to the idea of simulating water, which naturally flows towards regions of lower altitude. Building upon this water movement, another object was modeled: plants. The growth of these plants depend on the presence and quantity of water in their neighborhood.

Several questions were considered to develop the model in texture images: How to represent the agents that will interact in this environment? What will be the specific attributes for the water and plant agents on this environment? How can we capture the dynamics between plants and the surrounding water? In this way, we explored methods to represent the interaction of water and plants influenced by the environment.

Then, we proceed to transform this model applied in images into a model applied in complex networks. In order to make the desired modifications, it is necessary to establish a correlation between the metric used in texture images to a corresponding altitude value within a network context. This mapping enables the integration of the same concept of altitude information across both domains.

For the model applied in networks, we use the degree information as a measure analogous to the "height" in relation to pixel levels in images. Among the metrics used to characterize complex networks, the degree of a node plays a crucial role in the study of complex networks. The degree of a node refers to the number of connections it has with its neighbors. The information of the degree of a node has been effectively applied in social networks (ZHANG; LUO, 2017), psychiatry (ZHOU *et al.*, 2017), and medicine (WANG *et al.*, 2019).

There are other several metrics that are used to characterize complex networks. These metrics serve as quantitative measures to assess and describe the fundamental properties and attributes of complex networks (WILLS; MEYER, 2020). The selection of appropriate metrics depends on the specific research objectives and the nature of the network being analyzed. These metrics provide valuable insights into the structural, functional, and topological aspects of the networks, facilitating a comprehensive understanding of their characteristics and behavior (COSTA *et al.*, 2007).

In our work, we aim to explore the interaction between the agents using the spatially explicit information of the environment. Through this dynamic process, we aim to discover patterns that helps us in the understanding of the underlying structural organization of the environment. The following sections describe the input network datasets, the stage of agent-based modeling, the pattern recognition stage, and the classification process.

4.2 Network datasets

A network dataset for classification is a collection of labeled networks used to train, evaluate, and test a classification model. The dataset comprises the network data and the class labels. The datasets we used in the experiments were synthetic and from the real world. These datasets were also used in related works of network categorization, therefore they are useful for establishing comparative performances with the classification results of their methods.

4.2.1 Synthetic network datasets

Synthetic datasets are those specifically created to test the performance of the proposed models in the literature. Most synthetic datasets are balanced, that is, they have the same number of samples per class. We use the synthetic datasets presented in the work by Miranda, Machicao and Bruno (2016a): Synthetic 4-models, Synthetic 4-models plus $\langle k \rangle$, Scale-free, and a set of Noisy datasets.

- **Synthetic 4-models**: This dataset is composed of synthetic networks generated from the combination of the average degree $\langle k \rangle$ and the network size N, that is equal to the number of nodes. It was used 7 values of the average degree, and 4 values of the network size, specifically $\langle k \rangle = [4,6,8,10,12,14,16]$, and $N = [500,1000,1500,2000]$, applied to the following network models:

- An Erdös-Rényi model (ERDÖS, 1959). The model generates random networks where the edges have a connection probability of $p = \langle k \rangle / N$.
- A Watts–Strogatz small-world model (WATTS; STROGATZ, 1998). Each edge in the networks have a rewiring probability of $p = 0.1$.
- A Barabási-Albert scale-free model (BARABÁSI; ALBERT, 1999). The networks exhibit linear and non-linear preferential attachments.
- A Waxman or geographical model (WAXMAN, 1988). These networks are considered as the spatial generalization of the Erdös-Rényi networks. Nodes are connected according to a probability that is a function of their spatial distance.

The purpose of the combination of the average degree $\langle k \rangle$ and the network size N is to make the synthetic dataset heterogeneous, thus the dataset presents a wide range of features. For each pair of $\langle k \rangle$ and N there are 100 networks, which adds up to a total of 2800 networks for each model, resulting in a dataset containing 11200 networks. Overall, this dataset has 4 classes, and each class has 2800 samples.

- **Synthetic 4-models + $\langle k \rangle$**: This synthetic dataset contains networks that come from the Synthetic 4-models previously described. The networks result from the combination of 7 $\langle k \rangle - N$ values with the 4 models. The values of $\langle k \rangle$ are $[4,6,8,10,12,14,16]$ and keeping N constant equal to 500. The $model - \langle k \rangle - N$ combination represents a class, so we have 28 classes in this dataset, and each class is made up of 100 networks. In total, this dataset has 2800 networks.

- **Scale-free**: This dataset comprises scale-free networks based on the models by Albert and Barabási (2000), and Dorogovtsev and Mendes (2002), with values of average degree $\langle k \rangle = 8$ and size network $N = 1000$. In the Barabási-Albert networks, the parameter α represents the preferential attachment of the edges (BARABÁSI, 2015). A value of α equal to 1 means that the edges present linear preferential attachment, while values other than 1 the preferential attachment is non-linear. Using the Barabási-Albert model, 100 networks were built for each of the 4 values of $\alpha = [0.5,1.0,1.5,2.0]$. Using the Dorogovtsev-Mendes model 100 networks were built. Thus, the dataset is composed of five distinct classes, and each of these classes has 100 networks.

- **Noisy datasets**: They correspond to 4 datasets whose networks come from the synthetic *4-models* dataset and the *Scale-free* dataset to create a noise-free dataset, referred as *Noise-0*. There are 400 networks from the synthetic *4-models* and 400 networks from the *Scale-free* dataset, making a total of 800 networks. From this base *Noise-0* dataset, a noise rate was applied to the dataset in order to modify the topological structure of the networks. In this way, it was created 3 other datasets. The applied noise rate, ρ, was of 10%, 20% and 30%, the higher the value of ρ, the more changes the network undergoes. The changes consisted in the removal of $\rho/2$ edges and the addition of $\rho/2$ edges. The alterations introduced to the networks were made to assess the robustness of the method under consideration. To summarize, each noise dataset, noise-0, noisy-10%, noisy-20% and noisy-30%, has 8 classes with 100 networks per class.

4.2.2 Real-world network datasets

Real-world datasets are specific to an area of application, for example social networks, chemical molecules, or coming from bioinformatics. In most of the cases, real-world datasets are imbalanced, that is, the number of samples per class is not equal. We use the following datasets: A set of Metabolic datasets comprising seven datasets: *Actinobacteria*, *Animal*, *Firmicutes-Bacillis*, *Fungi*, *Kingdom*, *Plant*, *Protist*. A *Social* dataset, and a *Stomata* dataset.

- **Metabolic**: this dataset was constructed using the substrate-product network model (ZHAO *et al.*, 2006). The vertices corresponds to metabolites, and the relationship between them are the products per each reaction. Seven datasets were obtained from the Kyoto Encyclopedia of Genes and Genomes database (KANEHISA *et al.*, 2015):

  - **Actinobacteria**: This dataset contains the classes *Mycobacterium*, *Corynebacterium* and *Streptomyces*. This is an imbalanced dataset containing 3 classes with 60, 86, and 53 networks per class respectively, with 199 networks in total.
  - **Animal**: this dataset comprises the classes: mammals, birds, fishes and insects. This dataset is composed of 4 classes with 14 networks per class, with 56 networks in total.
  - **Firmicutes-Bacillis**: this dataset contains the classes *Bacillus*, *Staphylococcus*, *Streptococcus* and *Lactobacillus*. This is an imbalanced dataset containing 4 classes with 122, 76, 133 and 83 networks per class respectively, with 414 networks in total.
  - **Fungi**: this dataset contains the classes: *saccharomycetes*, *sordariomycetes*, and *basidiomycetes*. This dataset is composed of 4 classes with 15 networks per class, with 60 networks in total.
  - **Kingdom**: this dataset contains *eukaryotes*, that is, a diverse domain of organisms whose cells have a nucleus such as *animals*, *plants*, *fungi* and *protist*. This dataset is composed of 4 classes with 40 networks per class, with 160 networks in total.
  - **Plant**: this dataset contains the classes: *Monocots*, *Green algae* and *Eudicots*. This dataset is composed of 3 classes with 9 networks per class, with 27 networks in total.

  - **Protist**: this dataset contains the classes *Amoebozoa*, *Alveolates*, *Stramenopiles* and *Euglenozoa*. This dataset is composed of 4 classes with 5 networks per class, with 20 networks in total.

- **Social**: This dataset comes from the Stanford Network Analysis Project (LESKOVEC; KREVL, 2014). Two networks were used: Google+ and Twitter. Each network corresponds to an "ego-network", an ego-network refers to the subnetwork or neighborhood surrounding a specific node of interest, known as the ego node. In this dataset the ego node is not included. There was a selection of 50 samples of each network, therefore, this dataset has 2 classes with 50 networks per class with 100 networks in total.

- **Stomata**: This dataset is presented in the work by Miranda, Machicao and Bruno (2016a), they come from the transformation of binary images to networks of the distribution of stomata on plant leaves of *Tradescantia zebrina*, this plant is commonly known as *wandering dude*. To form the network, they used the spatial distribution of the stomata, thus connecting two stomata according to an established radius. As the radius increases, the stomata connect with more stomata, thus the density of the network increases. The data set consists of 3 classes, that represent 3 different lighting conditions, with 96 networks per class.

4.3 Agent-Based Model

4.3.1 Description of the model

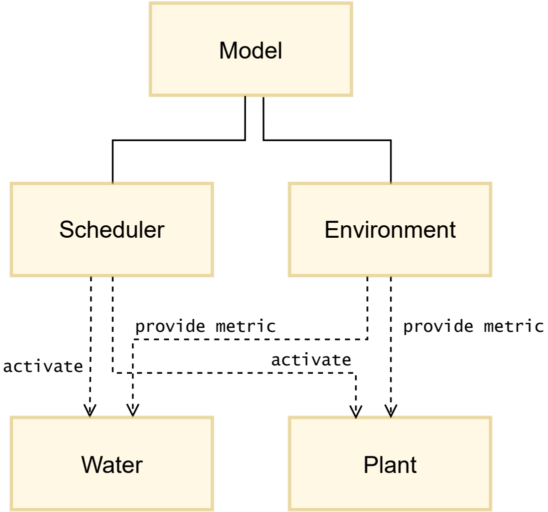
With the objective of finding characteristics that help to categorize networks, we model a dynamic process using agent-based models, the interaction of the agents on the environment provides data and patterns, these patterns are used to classify the complex networks.

If we interpret an image as an environment where water and plants interact. The environment provides a tangible representation of how water naturally flows towards lower altitudes and how plants dynamically grow in response to their surrounding environment. Thus, providing a visual feedback that aids in the understanding and validation of the behavior of the agents within the model.

The proposed model is called Growth Model, it simulates the growing of plant agents through the interaction with water agents over an environment. Figure 10 represents the main components of the model. It consists of the Model, the Scheduler, the Environment, the Water and Plant agents.

The Model is the main component that includes the other components, it serves as the interface of the parameters and the behavior of the agents. The Scheduler is the component that used to define the order in which the agents will be activated. The Environment represents the space in which the agents interact, the space can be either a texture image or a complex network. The two types of agents in the model, the plant agent and the water agent are related to the Scheduler and the Environment. The Scheduler activates the agents by type in random order, and the Environment provides the spatially explicit information that the agents need to adapt its behavior.

Figure 10 – The Growth model: an Agent-Based Model for categorizing complex networks



Source: Elaborated by the author.

The model consists of the structure, the dynamics, and data collection of the agent based model:

- Environment: An image or a network.
- Agents: Water agents and Plant agents.
- Behavior of agent Water: the Water moves to lower places of the environment. There can be more than one Water agent in the same position.
- Behavior of agent Plant: the Plant growth according to the water around its neighborhood. There is only one Plant agent in a position.
- Neighborhood: In an image the neighborhood are the positions around a pixel. In a network the neighborhood represents the neighbors around a node.
- Activation scheme: During the simulation, first the water is activated then the plants are activated.
- Data collection: The data generated is collected after both agents were activated.

4.3.2 Growth Model

In this agent-based model interacts two types of agents, water and plant. The water moves over the environment, and the plant grows according to the number of water around its vicinity region. The concrete environment to explore the interaction between these two agents is an image or a network.

In the following paragraphs, to describe the agents behavior of the Growth model, we refer to them as “plant” and “water”, in uppercase or in lowercase, where necessary to help in the understanding of the model, instead of writing “plant agent” and “water agent”.

4.3.2.1 The environment

The environment provides the spatially explicit information necessary to model the interaction of the agents, as it establish a position for the agents and helps to identify its neighbors. Its structure provides a criterion to determine the movement of water and the growth of plants. The environment defines the metric for the agents, with this metric being represented by the term “height”. In the context of images, it corresponds to the pixel value, while in networks, it refers to the degree of the node.

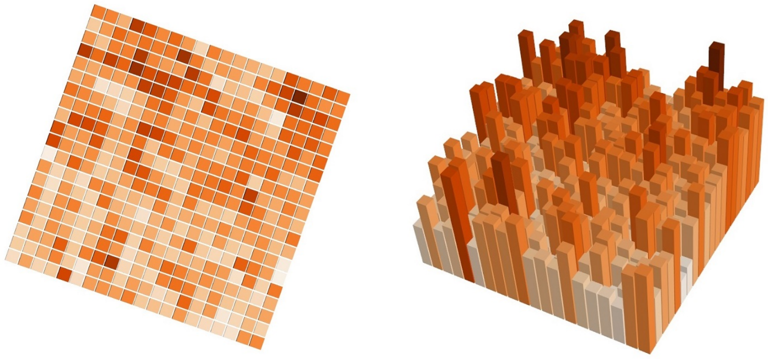
In this work, the environments used for the interaction of agents are complex networks. In images a pixel is related to its neighbors according to the Moore neighborhood. In networks a node its related to its closest neighbors, and we work with undirected networks.

A first modeling approach was performed in images because its structure helps us to visualize the position and height of the agents. To model the environment in images, we use gray scale images, in this way the image can be represented as a surface, in which the highest values represent peaks, and lower values represent valleys in the surface. Thus, the criterion for the movement of water through the environment are the values of the pixels. Figure 11 depicts an image and its representation as a tridimensional surface. The image is represented with an orange-color scale to improve visibility and clarity of the surface. The colors represent the pixel values of the image, lighter values are lower pixel values, conversely darker values correspond to higher pixel values.

The agents can be placed in specific positions in the environment, for example, the plants in lower altitude locations, and the water in higher elevate locations, with the objective to facilitate the movement of water and the growing of plants, or the agents can be placed in random positions in the environment. During the simulation, the water moves across the surface exploring lower regions of altitude, and in turn the plant has an allowed region of operation, which means that its scope is limited in its range of neighbors, and therefore interacts with the water and neighboring plants that are within a certain height.

Figure 12 shows the scope of a plant. The plant is represented by a green square in the center of the region and the numbers represent the relative height of the plant agent with respect

Figure 11 – An image and its representation as a surface: (a) a gray scale image of size 20×20 pixels, and (b) the gray scale image viewed as a tridimensional surface.

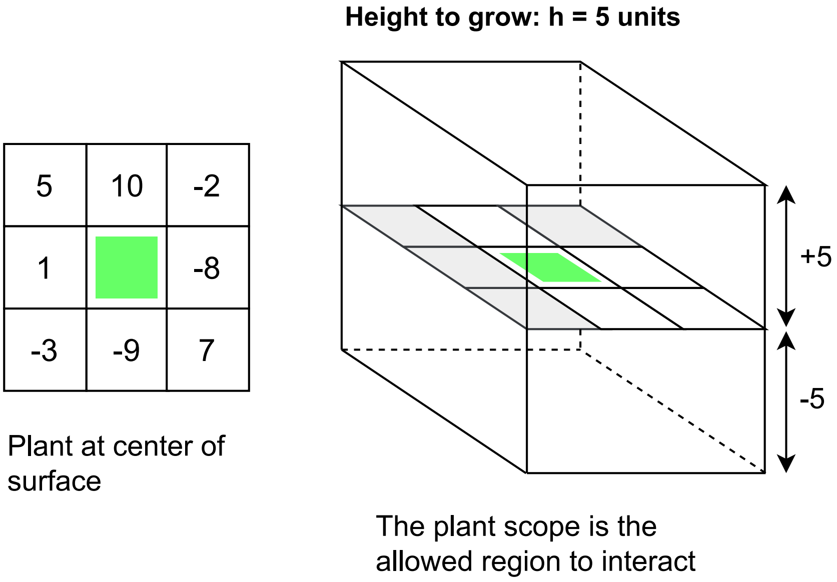


(a) (b)

Source: Elaborated by the author.

to its neighborhood. The scope is defined as the height h in which the plant can interact within the environment, and therefore with other agents. The image represents two regions, on the left a 3×3 area, and on the right a $3 \times 3 \times h$ surface, in the image $h = 5$ units. If the height is not considered, the plant agent can reach all neighboring positions, however when considering a height, the plant agent interacts only with those positions that are above or below the height h, on the surface they are the gray squares .

Figure 12 – Scope of the plant agent: Comparison of a 3×3 region with a $3 \times 3 \times h$ volume region



5	10	-2
1	🟩	-8
-3	-9	7

Height to grow: h = 5 units
Plant at center of surface
+5
-5
The plant scope is the allowed region to interact

Source: Elaborated by the author.

4.3.2.2 Water agent

Each water agent represents a drop of water that can move around the environment and can be consumed by a plant agent. A water can be in one of two states: alive or dead. In the alive status a water is scheduled to move freely through the environment, however in the dead status it is removed from the scheduler and the environment.

From its starting position the water moves through the environment towards lower altitude regions. The rule of motion is by the relative height in the environment, the water agents will move to positions with lower altitude. As a consequence of the structure of the environment, the movement of water can result in having more than one water agent in the same position.

A water moves freely over the environment along all the execution of the simulation, meaning that its status of alive does not change and only it can change its status to dead when it is consumed by a plant. The water may also exhibit an evaporation phenomenon, meaning that it has a lifetime, a time during the simulation in which the water can be used and its status is alive. With a lifetime the water moves a maximum number of steps, and if it is not consumed by a plant it evaporates, that is, its status change to dead, so it is remove from the simulation. Every time a water moves, its lifetime decreases, if this value drops to zero, the water changes its status to dead, and it is removed from the model.

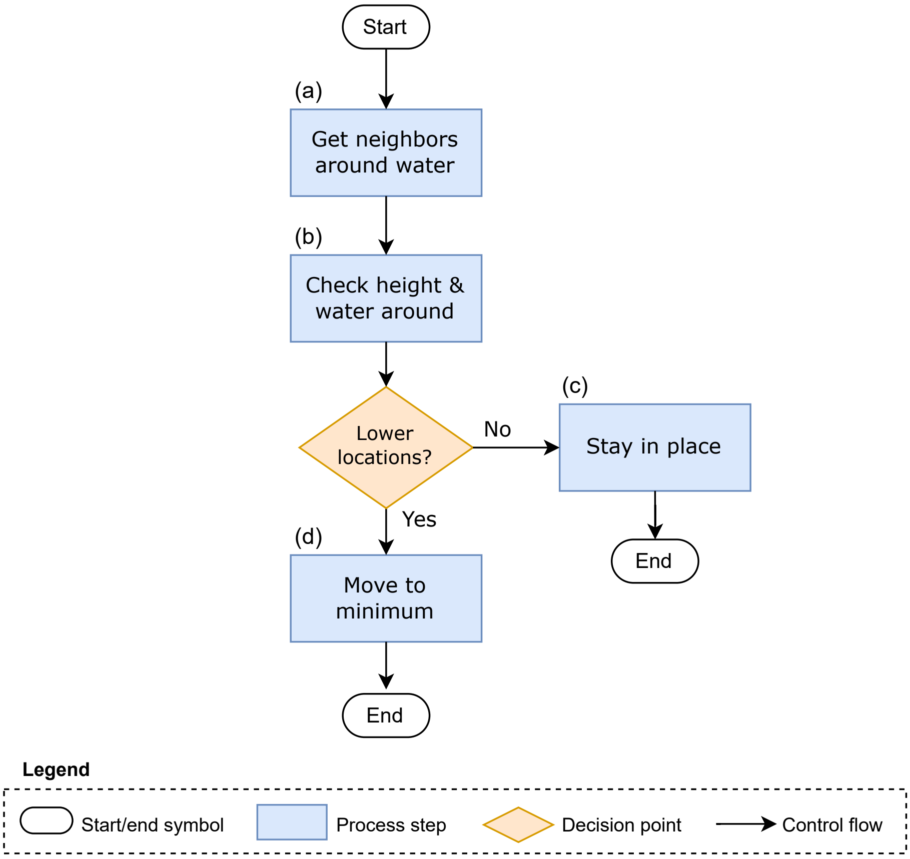
4.3.2.3 Movement of Water

Figure 13 presents the algorithm for the water movement over the environment. The purpose of the algorithm is to find a new position for the water. The searching for lower positions resembles the movement of butterflies trying to find higher positions in an environment creating virtual corridors, the movement is known as hill-topping in butterflies that was described by Pe’er, Saltz and Frank (2005). From its current position, the water verifies its neighborhood, looking for the lowest positions in the environment taking into account the existing water of its neighbors, and will move to another position, one with an equal or lower altitude, or will stay in its position because it already is in a local minima.

At the start, the water gets the neighbors around its current position, with this information the water checks the height of the neighbors, for each neighboring position this value is the sum of the height of the environment and the amount of water in that position. At the decision point the water checks if there is a lower position to which it can move, if it finds a new position with lower height the water moves, which means that the scheduler updates its position, otherwise the water remains in its position.

Figures 14, 15, and 16 illustrate the water movement algorithm in three scenarios: the movement of one water agent in a surface with different levels of altitude, the movement of one water agent in a surface with equal lower level of altitude, and the sequence of movements of two water agents that start in the same position.

Figure 13 – Algorithm for the movement of water



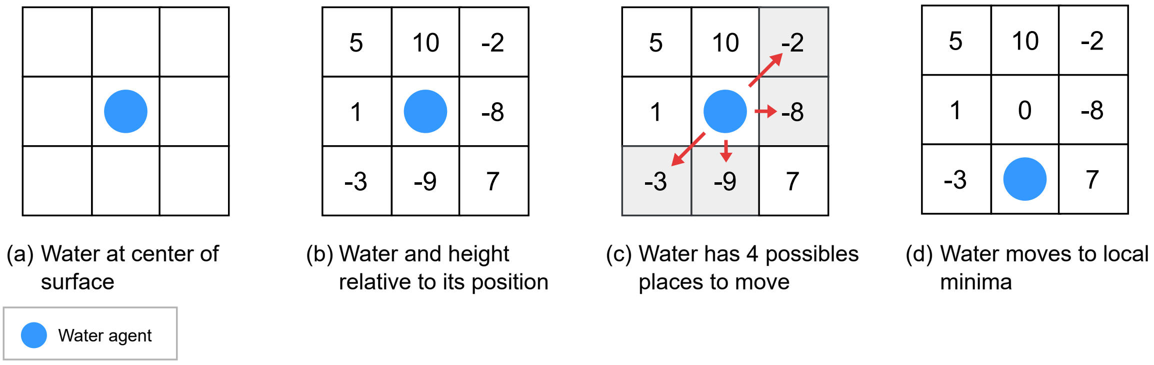
Source: Elaborated by the author.

Figure 14 illustrate the movement of a water agent. The light blue dot in the center represents a Water agent. In the first image (a) the water agent is located in the middle of a 3×3 surface. The numbers inside the squares, in the second image (b), represent the height of the surface relative to the water agent. A positive value means that the surface altitude is higher a number of units, similarly, a negative value means that the surface altitude is lower a number of units. The third image (c) shows that the water agent is located around a surface with 4 places with higher altitude and 4 places with lower altitude than itself, the red arrows point to lower possible places where the water agent can move. And finally, in the fourth image (d), after evaluating its surrounding, the water agent moves to the local minima.

In Figure 15 shows the movement of a water agent, in this case the surface has 5 positions of lower altitude with respect of the agent position, all the 5 positions have equal altitude. In (a) the Water agent is in the middle of the neighborhood, in (b) the water and the height of the surface, in (c) the water agent checked that has 5 possible places to move, all of equal altitude, in (d) the water agent moves to a new position, as the altitude is equal, the selection of the position is in random form.

In Figure 16 there are two water agents in the same position in the environment. The first water agent is of light blue color, and the second is of turquoise color. In (a) the two water agents

Figure 14 – Sequence of the movement of a water agent in a neighborhood with different altitudes.



(a) Water at center of surface

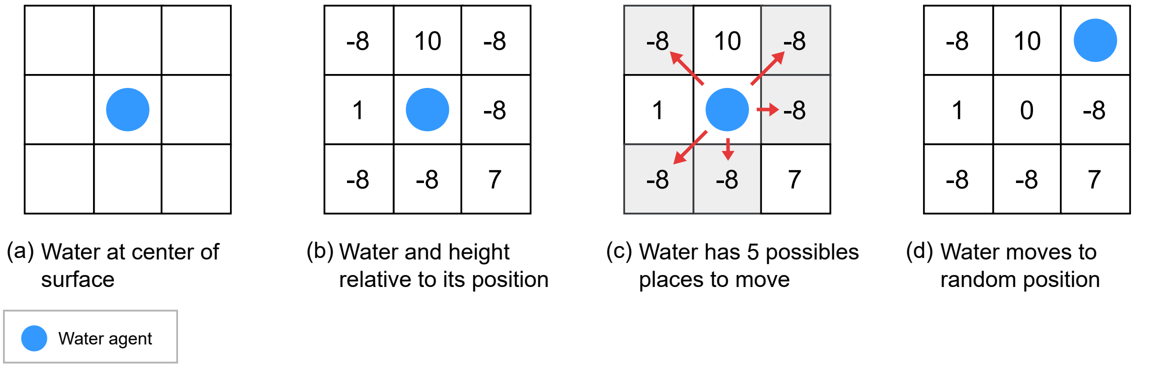
(b) Water and height relative to its position

(c) Water has 4 possibles places to move

(d) Water moves to local minima

Source: Elaborated by the author.

Figure 15 – Sequence of movement of one water agent with 5 possibles places to move with equal height



(a) Water at center of surface

(b) Water and height relative to its position

(c) Water has 5 possibles places to move

(d) Water moves to random position

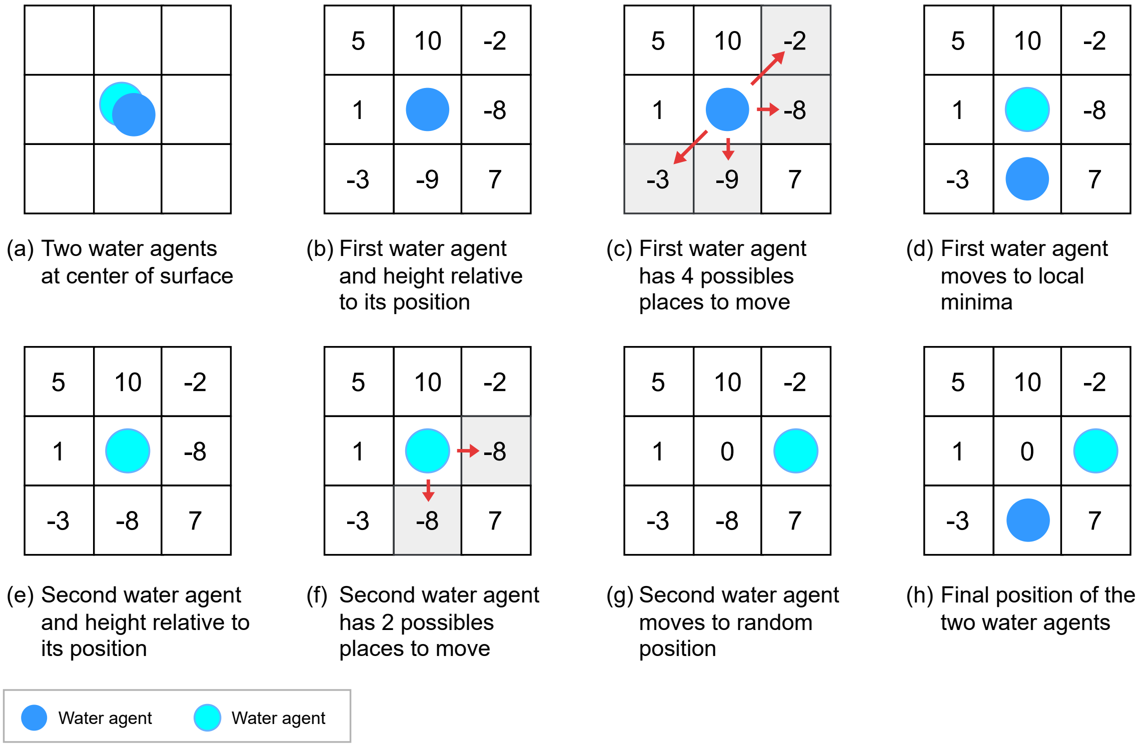
Source: Elaborated by the author.

are at the center of the surface. In (b) the first Water agent and the height relative to its position, in (c) the first water agents checks that it has 4 possibles places to move, and, in (d) the first water agent move to the lowest altitude position, in this case there is only one possible place, the position marked with -9. For the second agent to move, the environment has changed, since the previous water agent adds the height of the pixel to the quantity of water, in this case, from -9 goes to -8. In (e) appears the second water agent and its height relative to its position. In (f) the second water agent checks its environment and has two possible places of equal altitude to move, in (g) the agent moves, this time the selection of the new position is at random. And finally, in (h) appears the positions of the two water agents after their movements.

4.3.2.4 Plant agent

In the growth model, the plant is an agent that starts in an established position and can grow and expand creating other plant agents. Its dynamic depends on the surface and the amount of water agents around it. A plant starts in the alive state and with a time to live, while being alive it can interact with other agents in the environment. If during the execution of the model

Figure 16 – Sequence of movement of two drops of water starting at the same position



(a) Two water agents at center of surface

(b) First water agent and height relative to its position

(c) First water agent has 4 possibles places to move

(d) First water agent moves to local minima

(e) Second water agent and height relative to its position

(f) Second water agent has 2 possibles places to move

(g) Second water agent moves to random position

(h) Final position of the two water agents

Source: Elaborated by the author.

the plant does not get water to grow, its time to live decreases, and if its time to live reaches zero, it changes its state to dead, in that case it is removed from the environment.

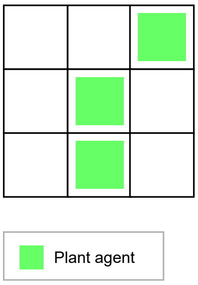
The model considers that each plant occupies a unique position in the environment. Thus in Figure 17, there are three plants located in closely related positions which interact independently. At the start of the simulation, the Plant agents can be placed randomly or in specific places of altitude in the environment. In order to allow the interaction between the agents, the water is placed in the highest positions, while the plants are placed in the lowest positions, so that the water can find a way to the plants and the plants can grow in the environment.

4.3.2.5 Plant growth

Figure 18 shows the algorithm for the growing of plants. The objective of the algorithm is the interaction of the plants with the environment and water agents, so that they can grow and expand towards neighboring positions. In this way, the growth of plants can be viewed both as vertical and horizontal. In the vertical growth the plant increases its height, in the horizontal growth, new plants are created. The effect that emerges during the simulation of the model is that the plants grow uniformly in height from their initial position and expand to occupy a greater area.

In the algorithm, the altitude considered correspond to the sum of the surface height and

Figure 17 – The plant agent occupies unique positions in the environment. In this space there are three plant agents.



Source: Elaborated by the author.

the plant height. In images the surface height is equal to the pixel value, and in networks the surface height is equal to the degree of the node.

In Figure 18 the start symbol means that the Scheduler has activated a plant. The letters next to block processes identify the progression of the algorithm. At first (a), the plant observes its neighborhood and obtains the water that is within its scope. The plant obtains water from the surface that is in a specific range of height, and does not consider water in positions higher or lower than its scope. The model defines a parameter that corresponds to the amount of water needed to grow. If the amount of water is not enough to grow (b), the lifetime of the plant decreases and it is checked if the lifetime has dropped to zero. At this point the plant disappears from the environment, its state is set to dead and it is removed from the Scheduler.

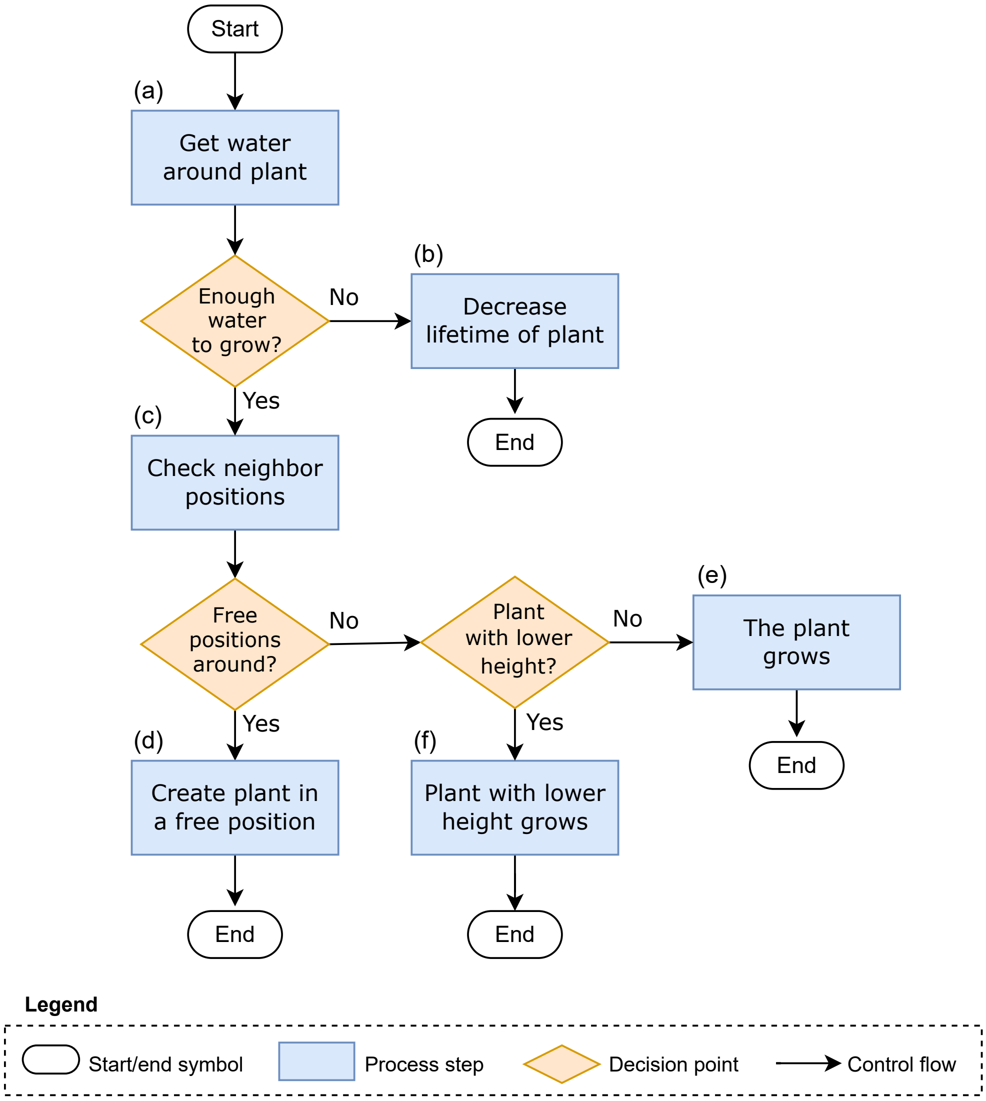
When the plant has enough water in its environment, the algorithm checks the neighborhood (c) to find positions that do not contain plants. If there are one or more positions without plants, the algorithm chooses the position with the lower altitude and creates a plant at that position (d). On the contrary, if all the neighboring positions are occupied by plants, the algorithm checks which of those plants has lower altitude, and choose the one with lower altitude (f) to increase its height. Finally, if all the surrounding plants have higher altitude, the plant grows (e).

Figures 19, 20, and 21 illustrate the operation of the algorithm in three scenarios: a plant and 7 water agents in the environment, a plant that does not have enough water in its scope, and a plant with enough water to grow. The green square represents a Plant agent and the light blue dots represent Water agents. At the beginning the plant agent is located in the middle of the surface, and the numbers represent the height relative to its position.

In Figure 19, there is a plant and seven water agents in the environment. The amount of water that can be used by the plant is restricted to the height of ± 5 pixels, this means that of the 7 water agents, the plant can only consume 4 water agents, 3 from the left column and one in the upper right corner.

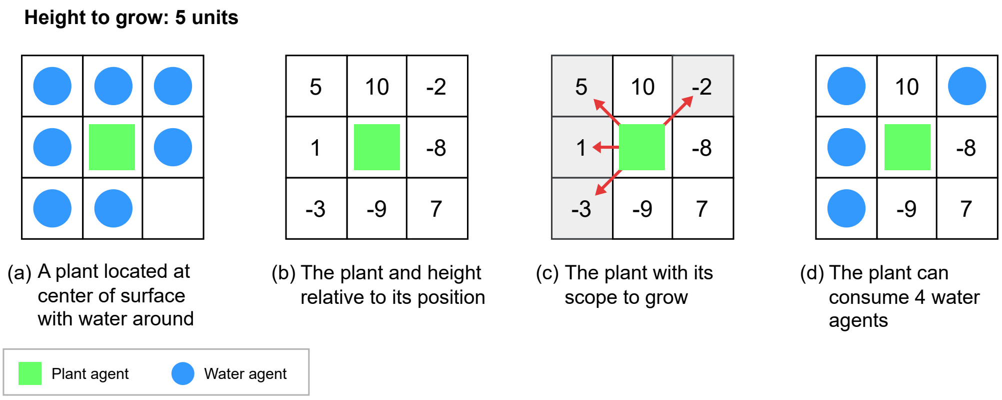
In Figure 20, the plant has a lifetime of 2 and it requires 5 water agents to grow. In (a) is

Figure 18 – Algorithm for plant growth



Source: Elaborated by the author.

Figure 19 – Effect of the scope of the plant agent in an environment. The plant can only consume 4 water agents of the 7 that are around it.



(a) A plant located at center of surface with water around

(b) The plant and height relative to its position

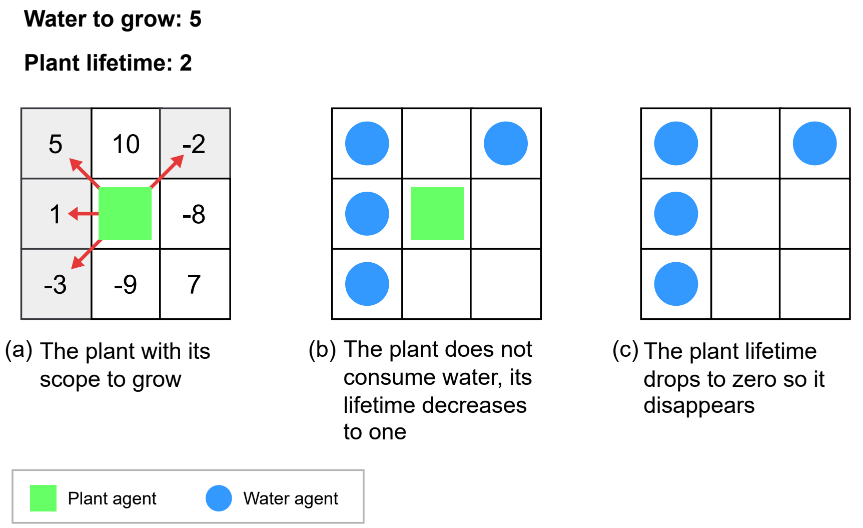
(c) The plant with its scope to grow

(d) The plant can consume 4 water agents

Source: Elaborated by the author.

shown the scope of the plant, there are 4 positions from which it can consume water and where it can grow. In (b) the plant has 4 water agents within its reach, but it does not consume them and its lifetime decreases. At (c) again the plant does not consume the water around it, its lifetime decreases to zero and it is removed from the model.

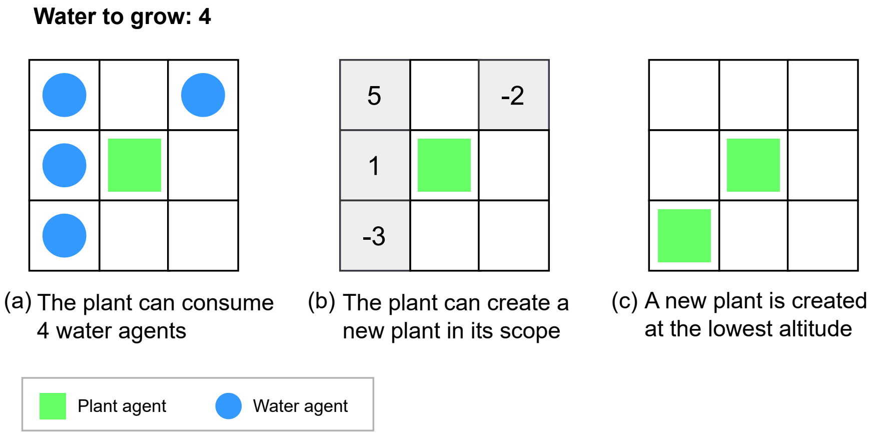
Figure 20 – A plant agent with not enough water to grow


Water to grow: 5
Plant lifetime: 2
5
10
-2
1
-8
-3
-9
7
(a) The plant with its scope to grow
(b) The plant does not consume water, its lifetime decreases to one
(c) The plant lifetime drops to zero so it disappears
Plant agent
Water agent

Source: Elaborated by the author.

In Figure 21, the plant requires 4 water agents to grow. In (a) the plant has 4 neighboring water agents in its scope. In (b), the plant consumed the water agents and it can create another plant in its scope. In (c), the plant chooses the position with the lowest altitude and creates a new plant.

Figure 21 – Creating a new plant in the plant scope


Water to grow: 4
5
-2
1
-3
(a) The plant can consume 4 water agents
(b) The plant can create a new plant in its scope
(c) A new plant is created at the lowest altitude
Plant agent
Water agent

Source: Elaborated by the author.

4.3.2.6 Parameters of the model

During the execution of the growth model, various parameters are used to influence the behavior of the model in simulations. These parameters serve as input values affecting the model, the environment, and the agents.

- Model parameters:
  - Percentage of water: Initial percentage of water placed in the environment
  - Percentage of plant: Initial percentage of plants placed in the environment
  - Number of iterations: Number of times the model is executed.
  - Iterations to add water: Number of iterations that need to pass to add water while running the simulation.
- Environment parameters:
  - Plant position: Initial position of plants in the environment. Plants can be placed at lower places or at random places in the environment.
  - Water position: Initial position of water in the environment. Water can be placed at upper places or at random places in the environment.
  - Metric: The value used to define the height of the environment. In images is the pixel value, meanwhile in networks is the degree of the node.
- Plant parameters:
  - Quantity of water to grow: Define the quantity of water needed by a plant to grow.
  - Height to grow: The height variation that a plant checks to obtain water, in this way defines its scope.
  - Time alive of plant: The lifetime of plants. Indicates how many iterations the plant is alive during the simulation.
- Water parameters:
  - Time alive of water: The lifetime of water. Indicates how many iterations the water is alive during the simulation.

4.3.2.7 The Growth model in complex networks

Table 2 presents the main characteristics and differences of the growth model applied in images and complex networks. From the model proposed in images, it was needed some adaptations to apply the model in networks. In complex networks, the environment is an undirected network, the metric used is the degree of the node, unlike images, it can have *n* neighboring nodes, and the scope is defined as the degree difference.

Table 2 – Characteristics and differences of the growth model in images and complex networks.

Texture images	**Complex networks**
Environment: scale-gray image	Environment: undirected network
Metric: Pixel values	Metric: Node degree
Neighborhood: 8 neighboring pixels	Neighborhood: *n* neighboring nodes
Plant scope: considers pixel difference	Plant scope: considers degree difference

4.4 Pattern Recognition

During the simulation of the model, we collect data from the interactions of the agents and the environment. The data is analyzed in this stage of pattern recognition. The patterns are extracted and a feature vector is built. Both global features and local features can be used to construct the feature vector.

4.4.1 Global features

It is possible to use global features to evaluate the performance of the model. A global feature captures the overall dynamics of the simulation. In each iteration, the Scheduler executes the behavior of every agent within the model. After each iteration, data is collected from all the active agents, and we use these data as feature vectors for the classification stage.

Table 3 presents the description of the names of the features that can be collected from the model, their abbreviations for later reference, and their descriptions. The features are separated according to the agent that originates it. These features were evaluated in order to choose those that provided better performance in the classification stage.

Table 3 – Global features obtained during the simulation of our Agent-Based Model

Features	**abbr.**	**Description**
Plant area	*pa*	It represents the number of nodes with plant agents in the environment.
Plant time	*pt*	It is the lifetime of the plant agents.
Plant volume	*pv*	To obtain the plant volume, we consider the area and the height of the plants.
Water area	*wa*	It represents the number of nodes that contain water agents in the environment.
Water quantity	*wq*	It is the amount of alive water agents in the environment
Water steps	*ws*	It represents the number of times that the water agents changed its position.
Water time	*wt*	It is the lifetime of the water agents

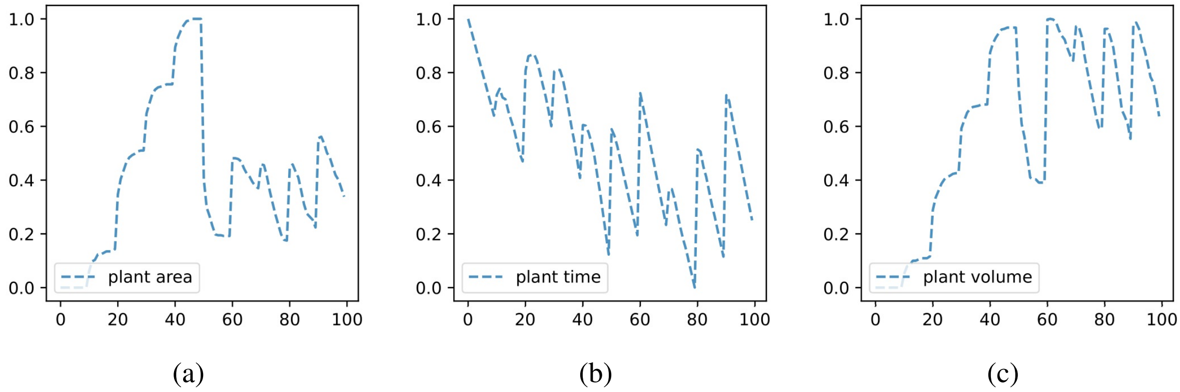
Source: Elaborated by the author.

From the plant agents can be extracted the features: plant area *pa*, plant volume *pv*, and plant time *pt*. The plant area *pa* represents the quantity of plant agents during the simulation, because a plant does not move during the simulation, so its position does not change. The plant time *pt* is the sum of the time alive. The plant volume *pv* represents the number of plants considering their height.

From the water agent can be extracted the features: water area *wa*, water steps *ws*, water time *wt*, water quantity *wq*. The water area *wa* is the area occupied by the water agents in the environment, since multiple agents can be in the same position, the water area is smaller than the number of water agents. The water quantity *wq* represents the amount of alive water agents in the environment, and there can be multiple water agents in the same position. The water time *wt* is the sum of time alive of water agents. The water steps *ws* is the number of times the water agents has moved to another position in the environment.

Figure 22 shows a plot of the data generated with the plant agent. The x-axis represents the number of iterations of the simulation, and the y-axis the normalized data. Table 4 presents the correlation data between the characteristics of the plant agent. *pa* is not correlated with *pt*, and *pa* has a correlation value with *pv* that indicates that there is a moderate positive correlation between the two characteristics. *pt* has a moderate negative correlation with *pt*.

Figure 22 – Plant features generated in a network of the *Actinobacteria* dataset: (a) plant area, (b) plant time, and (c) plant volume



Source: Elaborated by the author.

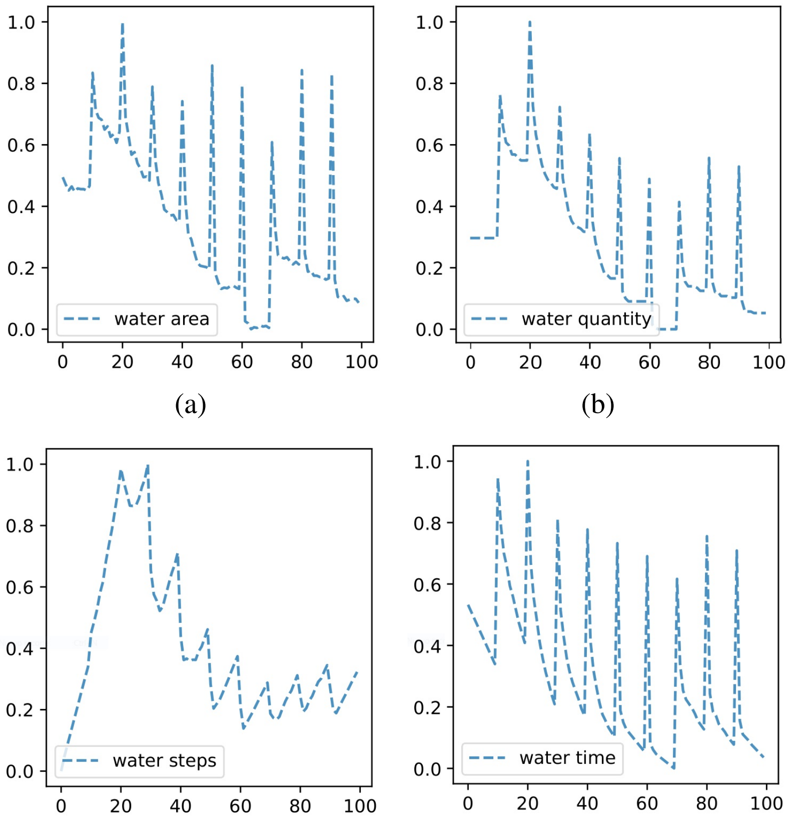
Table 4 – Data of correlation among plant features

	pa	*pt*	*pv*
pa	1	-0.06	0.672
pt		1	-0.498
pv			1

Source: Elaborated by the author.

Figure 23 shows a plot of the data generated with the agent water. The x-axis represents the number of iterations, and the y-axis the normalized data. Table 5 presents the correlation data between the characteristics of the water agent. wa is correlated with wq and wt, and wa presents a moderate positive correlation with ws. wq is moderately correlated with ws, and wq is correlated with wt. ws is not correlated to wt.

Figure 23 – Water features generated in a network of the *Actinobacteria* dataset: (a) water area, (b) water quantity, (c) water steps, and (d) water time.



(a)

(b)

(c)

(d)

Source: Elaborated by the author.

Table 5 – Data of correlation among water features

	wa	wq	ws	wt
wa	1	0.958	0.527	0.943
wq		1	0.713	0.876
ws			1	0.288
wt				1

Source: Elaborated by the author.

From the global characteristics we choose 2 characteristics of each agent. From the plant agent we choose plant area pa and plant volume pv. The plant time pt depends on the value of the parameter time alive of plant. When this value is set to its special value, pt would have a constant value throughout the simulation, not being a suitable value as a characteristic in the classification task. From the water agent we choose the water quantity wq and the water steps

ws. Water area wa and water quantity wq are correlated, so we decide on one. And we do not choose the water time wt for the reasons described above in the plant agent. Thus, we have the 4 features of plant area, plant volume, water quantity, and water steps $\{pa, pv, wq, ws\}$ as feature vector to apply in the stage of classification. These features are measured at each iteration so the size of the feature vector depends on the number of iterations that the model is simulated.

4.4.2 Local features

The local features used are the metrics of Lempel-Ziv complexity and the Shannon Entropy. The process to obtain the local features is the same for both metrics. For example, to use the Shannon Entropy, the approach involves initially calculating the Shannon entropy for individual nodes within the network, and computing the minimum and maximum entropy value. Then, the calculated entropy value of each node is assigned to an interval in an histogram, obtaining the entropy distribution of the entire network. This way, the size of the feature vector, for each network, will be the number of bins in the histogram. The work by Miranda, Machicao and Bruno (2016b) defined 20 bins to create the histogram. Therefore, the feature vector does not depend on the number of nodes in the network.

4.4.2.1 Lempel-Ziv Complexity

The Lempel-Ziv (LZ) complexity is a measure of the complexity or compressibility of a string of symbols. The basic idea behind Lempel-Ziv complexity is to quantify the amount of information or redundancy in a sequence of symbols (LEMPEL; ZIV, 1976). The algorithm works by identifying repeated patterns or substrings in the input string and encoding them more efficiently. The complexity is then calculated based on the length of the compressed representation compared to the original string (DOĞANAKSOY; GÖLOĞLU, 2006).

The Lempel-Ziv complexity is expressed as:

$$C_{LZ} = \frac{\text{length of compressed representation}}{\text{length of original string}}$$

A lower value of C_{LZ} indicates higher complexity or less compressibility, implying more information and less redundancy in the string.

4.4.2.2 Shannon Entropy

Shannon entropy, named after Claude Shannon, is a measure of the uncertainty or information content associated with a random variable. It quantifies the average amount of unpredictability associated with the possible outcomes of a random process (OMAR; PLAPPER, 2020). Shannon entropy is a fundamental concept in information theory and has various applications, including its use in analyzing complex networks.

The formula for Shannon entropy $H(X)$ for a discrete random variable X with probability distribution $P(X)$ is given by:

$$H(X) = -\sum_i P(x_i) \log_2(P(x_i)) H(X)$$

Here, x_i represents each possible outcome of the random variable, and the sum is taken over all possible outcomes. The logarithm is base 2, which results in entropy measured in bits.

In network characterization, Shannon entropy provides a quantitative measure of a network's heterogeneity. High entropy often suggests scale-free networks with hubs and many low-degree nodes, while low entropy might indicate regular or random networks.

4.5 Classification process

For the classification process, we define the classifiers used in the experiments, the validation strategy to assess the performance of the model, and the metrics used.

4.5.1 Classifiers

In this part, we define the classifiers used for the data generated by the agents. The objective of classifiers is to build models that can predict the category of a given input data point based on its features. There are many types of classifiers, including decision trees, support vector machines, logistic regression, and neural networks. As we work with structured data, we employ machine learning classifiers that provide good results for this type of data type (GRINSZTAJN; OYALLON; VAROQUAUX, 2022; SHWARTZ-ZIV; ARMON, 2022): k-Nearest Neighbors, Linear Discriminant Analysis and Linear Support Vector Classifier. These classifiers were used with the default parameters offered by *Scikit-learn* (PEDREGOSA *et al.*, 2011).

- **k-Nearest Neighbors, kNN**: It is an algorithm that makes predictions based on the similarity of data instances in the feature space. kNN is a method that classifies or predicts new data points based on the majority vote or average of the k-nearest neighbors in the training set. It is a simple and effective classification algorithm that uses a distance metric to classify data. An instance of the testing set is compared against each instance of the training dataset, then according to the k nearest neighbors, a label is assigned for this instance (MARSLAND, 2015). In *Scikit-learn* the number of neighbors by default is set to 5.

- **Linear Discriminant Analysis, LDA**: It is a dimensionality reduction and classification technique. It is used for supervised classification tasks and is suited when dealing with high-dimensional data. It seeks to find a linear transformation of the data that can help to discriminate the classes. After the transformation, the classification is made using some metrics such as Euclidean distance (LI; ZHU; OGIHARA, 2006).

- **Linear SVC (Linear Support Vector Classifier)** is a linear classifier based on the Support Vector Machine (SVM) algorithm. It is designed for binary and multiclass classification tasks and is particularly useful when dealing with large-scale datasets. LinearSVC aims to find the optimal hyperplane that best separates the different classes in the feature space. The implementation of LinearSVC in *Scikit-learn* uses $L2$ regularization by default, which helps to prevent overfitting and improve the generalization of the model (FAN *et al.*, 2008).

4.5.2 Validation Strategy

Cross-validation is a statistical method to evaluate the generalization performance of a machine learning algorithm on a dataset. It involves splitting the available data into multiple subsets, or folds, to simulate the performance of the model on unseen data by iteratively training and evaluating the model on different subsets of the available data.

A common approach of cross-validation is **k-fold Cross-validation**, in which k defines the number of splits of the data. Using *k-fold cross-validation*, the data is split into k parts such that they are of equal or approximate size. Then a classifier uses $k-1$ folds of the data for training and 1 fold for testing, this process is repeated k times, in this way all the data is part of the testing set exactly once.

Another approach to split the data is by using **Stratified k-fold cross-validation**, this means that, in addition to split the data into equal sizes, each fold contains the same proportion of the classes as the entire data set. Thus, the value of k also depends on the number of samples per class in the dataset. By setting k equal to 10, the dataset must have at least 10 samples per class, so each sample will be part of the test set (MÜLLER; GUIDO, 2017).

A third approach that allows to obtain a better evaluation of the model is the **Repeated Stratified k-fold cross-validation**. This means repeating the cross-validation procedure n times. In each repetition, before obtaining a stratified division, the data is shuffled. Thus, the total number of cross-validation runs is $n \times k$ (KUHN; JOHNSON, 2013).

To evaluate the performance and generalization ability of our model, we use the *Repeated Stratified k-fold cross-validation* strategy. In order to get comparable result to previous works (MIRANDA; MACHICAO; BRUNO, 2016a; ZIELINSKI *et al.*, 2022), we set the number of repetitions n equal to 100.

In the case of datasets with at least 10 samples per class, for example the synthetic *Scale-free* dataset, we set $k = 10$ for cross validation, however, in datasets with less than 10 samples per class, such as the *Plant* dataset, which has 9 samples per class, we use $k = 3$ for cross-validation. In addition, when working with imbalanced datasets, we make use of random subsampling of the data (LEMAîTRE; NOGUEIRA; ARIDAS, 2017), before applying the cross-validation procedure.

4.5.3 Data preparation steps

The representation of the data can impact the performance of a machine learning classifier. Thus, before applying a classifier, it is necessary to chain a series of processing steps. These preparation steps are usually implemented with a tool called *pipeline*. A pipeline allows chaining multiple steps of the machine learning process. This includes data preprocessing (scaling, encoding, etc.), feature selection, and the classifier itself (BUITINCK *et al.*, 2013).

The main benefit of pipelines in a cross-validation setting is that it helps to avoid data leakage issues. This happens when data transformations (e.g., scaling, encoding) are applied to the entire dataset before cross-validation, thus the test set influence the training set giving a false impression of the performance of the model and can result in models that fail to generalize.

In this work, for balanced datasets, the pipeline consists of two steps of scaling and classification. In turn, for imbalanced datasets an initial random subsampling step is added. Random subsampling takes an equal number of samples in each run of the classification process. Then, the pipeline is applied to the repeated stratified k-fold cross validation procedure.

4.5.4 Classification metrics

It is important to evaluate the performance of the proposed model. To evaluate the results of the experiments, we use a set of classification metrics. These metrics serve to quantify the performance of the classification results. The metrics considered are: accuracy, confusion matrix, precision, recall, f-measure, true positive rate, false negative rate, and area under the ROC curve.

- Accuracy: It represents the overall performance of the classifier. It is the ratio of correctly predicted instances to the total number of instances in the dataset. In mathematical terms, accuracy is defined as:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}}$$

- Confusion matrix: A confusion matrix is a table that provides a comprehensive breakdown of the performance classifier. It presents the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the model. The confusion matrix helps to understand the performance of the model on each class. It is usually represented as follows:

- Precision: It evaluates the proportion of true positive predictions out of all positive predictions made by the classifier. The precision indicates how many of the predicted positive

	Predicted Positive	Predicted Negative
Actual Positive	**TP**	**FN**
Actual Negative	**FP**	**TN**

instances were actually positive. It is calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- Recall: It is also known as Sensitivity or true positive rate (TPR), it measures the proportion of true positive instances that were correctly identified by the classifier out of all actual positive instances. The recall shows how well the model captures the positive instances. It is calculated as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- f-measure: It is also known as the F1-score, is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall. It is calculated as:

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- False Positive Rate (FPR): It represents the proportion of negative instances that were incorrectly classified as positive by the classifier. It is calculated as:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

- Area Under the ROC Curve (AUC-ROC): The Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of the classifier across different discrimination thresholds. The AUC-ROC metric quantifies the overall performance of the classifier by calculating the area under the ROC curve. AUC-ROC ranges from 0 to 1, where a value of 1 represents a perfect classifier, and a value of 0.5 indicates a random classifier.

- Matthews Correlation Coefficient (MCC): It is a metric used to evaluate the performance of a binary or multiclass classification model. It is particularly useful when dealing with imbalanced datasets where one class may dominate the other in terms of the number of instances. MCC takes into account true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by the classifier and is defined as follows:

$$\text{MCC} = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{\sqrt{(\text{TP} + \text{FP}) * (\text{TP} + \text{FN}) * (\text{TN} + \text{FP}) * (\text{TN} + \text{FN})}}$$

The MCC ranges from -1 to +1, where:

- **+1** indicates a perfect classifier.
- **0** indicates a classifier that performs no better than random.
- **-1** indicates a classifier that performs in exactly the opposite way of the desired classification.

4.6 Considerations

The model was designed to explore the interaction of more than one type of element in an environment, starting from simpler models like walkers and CA. The proposed model have two agents that interact in an environment, the water agent, which moves freely in the environment, and the plant agent, which grows according to the environment and the amount of water. The model was designed with the aim of applying it to classification problems, as various types of data can be generated through the simulation of the model, these patterns allow for the characterization of complex networks,

By using spatially explicit agent-based models we can explore the characteristics of the environment that result from the interaction of agents, and thus take advantage of the rich nature of texture images or the structural complexities inherent in complex networks. The following chapter presents the application of the growth model in complex networks datasets.

CHAPTER 5

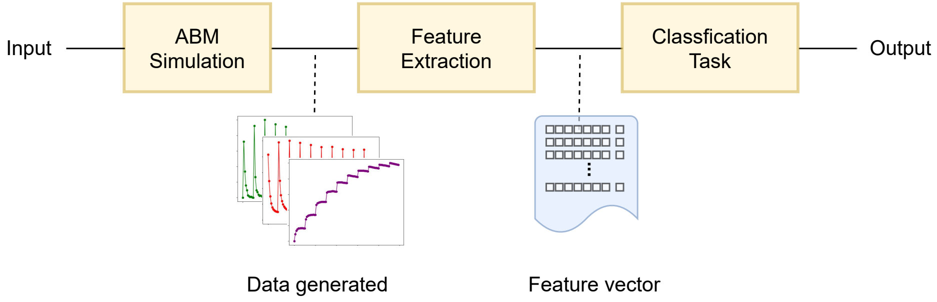
EXPERIMENTS IN NETWORKS

5.1 Introduction

This chapter presents the experiments performed using the Growth Model to categorize networks. The Growth Model was applied on real and synthetic network datasets. In each experiment presents the parameters and the analysis of the results obtained using the Growth Model.

Figure 24 represents the three processes that constitutes the experimental design: the ABM Simulation, the Feature Extraction, and the Classification Task.

Figure 24 – Experiment setup to classify networks with the Growth Model



Source: Elaborated by the author.

In the ABM Simulation process is executed the Growth Model simulation, since we are dealing with a supervised learning problem, the input of this process are network datasets with their corresponding labels, as well as, the set of input parameters used in the model. The output of this process is the data generated by the interaction of the agents. In the feature extraction process is performed the extraction of the data generated during the simulation. This data is selected and organized as feature vectors. A feature is a numerical value that quantifies some important characteristic of the agents. These feature vectors constitutes the input for the third process, the classification task, in which it is performed the network classification, which involves the classifiers and the validation strategy. The output of the experiments are the classification results comprising the classification accuracy, its standard deviation, the confusion matrix, among other metrics.

The rest of the chapter is organized as follows: The section 5.2, Input network datasets, presents a summary of the networks used in the experiments. In section 5.3, Model simulation, are detailed the parameters and data generated in the experiments. In section 5.4, Pattern recognition in synthetic datasets, are detailed the classification results and their metrics. In section 5.5, Pattern recognition in real-world datasets, are detailed the classification results and their metrics. The section 5.6, Comparison of classification results with previous works, summarizes the results obtained and presents results of related methods from the literature. The section 5.7, Considerations, is commented the importance of the model.

5.2 Input network datasets

Table 6 summarizes the synthetic network datasets used in the experiments to assess our model. It is observed that all the datasets are balanced, the only one that presents the largest number of samples per class is the *Synthetic 4-models* with 2800 samples per class, and the network that has the most classes is the *Synthetic 4-models* + $\langle k \rangle$, with 28 classes, the other datasets have from 4 to 8 classes with 100 samples per class.

Table 6 – Synthetic network datasets used in this work

Name	Networks	Classes	Samples / Class	Type
4-models	11200	4	2800	Balanced
4-models + $\langle k \rangle$	2800	28	100	Balanced
Scale-free	500	5	100	Balanced
Noise-0	800	8	100	Balanced
Noisy-10%	800	8	100	Balanced
Noisy-20%	800	8	100	Balanced
Noisy-30%	800	8	100	Balanced

Source: Elaborated by the author.

Table 7 summarizes the real-world network datasets used in the experiments to assess our model. In this case the datasets are balanced and imbalanced, the dataset with more networks is *Firmicutes-Bacillis* with 414 elements. The range of classes is small, varying from 2 to 4 classes. The dataset with the fewest elements is *Protist* with 20 networks in total. It can be seen, that in contrast to synthetic datasets, that real-world networks present variation in the quantity of elements per class and the representatives of some classes appear much more frequently.

In the experiments, each dataset will be utilized individually. This implies that the model

Table 7 – Real-world network datasets used in this work

Name	Networks	Classes	Samples / Class	Type
Actinobacteria	199	3	60, 86, 53	Imbalanced
Animals	56	4	14	Balanced
Firmicutes-Bacillis	414	4	122, 76, 133, 83	Imbalanced
Fungi	60	4	15	Balanced
Kingdom	160	4	40	Balanced
Plant	27	3	9	Balanced
Protist	20	4	5	Balanced
Social	100	2	50	Balanced
Stomata	288	3	96	Balanced

Source: Elaborated by the author.

will be applied independently to each dataset, allowing for a focused examination and evaluation of its performance on each specific dataset.

5.3 Model simulation

The Growth Model was designed to analyze networks using a dynamic process, in order to take advantage of the agent interactions on the network structure. As explained in the methodology chapter, there are several parameters that are used for the simulation of the model. Therefore, in order to establish the values of the parameters it is necessary to carry out an exploratory analysis of the data. Each dataset has been analyzed and the properties of average nodes, average edges, average degree, and range of nodes have been obtained.

Table 8 present data from the synthetic datasets. All the networks in the *Scale-free* dataset are connected. The other datasets presents networks that are connected and disconnected, the proportion of networks connected is greater than those that are disconnected. In datasets with noise, there is a slightly change in the number of connected networks, in comparison to the noise-free dataset *Noise-0*. The average number of nodes is similar in all datasets, except the *4-models* + $\langle k \rangle$ that has 500 nodes. The average edges is close to 4000 in the *Scale-free* and *Noise* datasets, in comparison the *4-models* has 50% less edges, and *4-models* + $\langle k \rangle$ has 50% more edges on average. The average degree is the same in the *4-models* datasets with 10.07, the other datasets have average degrees around 8. And, almost all networks in each dataset have the same size, with the exception of the *4-models*.

The exploratory data of the real-world dataset is shown in Table 9. All the real-world metabolic datasets are disconnected. The *Social* and *Stomata* datasets contains connected and disconnected networks. The average number of nodes is similar in all datasets, except the *Stomata* that has the least number of nodes. The average edges in the real-world metabolic datasets vary from 200 to 400 edges. The *Social* dataset has the highest number of edges on average of all the

Table 8 – Exploratory data of the synthetic complex network datasets

Dataset	Connected	Avg nodes	Avg edges	Avg degree	Range nodes
4-models	Yes: 9102, No: 2098	1250	6292	10.07	500 - 2000
4-models + $\langle k \rangle$	Yes: 2348, No: 452	500	2516	10.07	500
Scale-free	Yes: 500	1000	3980	7.96	1000
Noise-0	Yes: 796, No: 4	999	4000	8.00	997 - 1000
Noise-10%	Yes: 798, No: 2	999	3999	8.00	995 - 1000
Noise-20%	Yes: 798, No: 2	999	3998	8.00	993 - 1000
Noise-30%	Yes: 796, No: 4	996	3997	8.02	991 - 1000

Source: Elaborated by the author.

other datasets, in a ratio close to 50 to 1. The dataset *Stomata* also has a high number of edges on average, about half that of *Social*. This disparity of values is also reflected in the average degree of the networks, with the real-world metabolic having between 4 and 5 of average degree, however, *Social* has 93 and *Stomata* has 246. The number of nodes indicates that in each dataset the networks have a similar size, with the exception of the *Social* dataset because the network size varies considerably from 15 to 4938 nodes.

Table 9 – Exploratory data of the real-world complex network datasets

Dataset	Connected	Avg nodes	Avg edges	Avg degree	Range nodes
Actinobacteria	No: 199	1127	2862	5.07	803 - 1533
Animals	No: 56	1725	4086	4.73	1392 - 1919
Firmicutes	No: 414	901	2192	4.82	530 - 1267
Fungi	No: 60	1329	3182	4.78	624 - 1540
Kingdom	No: 160	1487	3538	4.70	448 - 2095
Plant	No: 27	1680	4111	4.89	1241 - 1870
Protist	No: 20	921	2024	4.33	598 - 1325
Social	Yes: 76, No: 24	1107	127412	93.87	15 - 4938
Stomata	Yes: 270, No: 18	406	51203	246.40	292 - 486

Source: Elaborated by the author.

5.3.1 Feature selection for evaluation of classification accuracy

To assess the influence of the features $\{pa, pv, wq, ws\}$ we evaluate how each feature individually and in combination contributes to the classification results. The model was run on the *Actinobacteria* dataset setting the values of the parameters to $wg = 4, hg = 50, pw = 60, tw = 0, pp = 30, tp = 0, it = 200,$ and $iw = 10$.

Table 10 shows the classification results of the combination of the features in the *Actinobacteria* dataset. The lines are used to separate the number of features used in the classification process. Using the features individually, the kNN classifier got the best result with plant volume.

The other good results were obtained with a combination of plant area, and plant volume with a feature of water. In the LDA classifier, the best result was when using the 4 features together, and the other good results are product of the combination of the features including water steps. However using only one feature got lower values of accuracy. And in the case of the Linear SVC classifier, the classification results are similar to the LDA classifier, getting the best classification result using the combination of plant features with water quantity, the other good results were product of the combination of three or four features. When using less features the results are lower. In general, the classification accuracy was better with the LDA classifier, followed by the Linear SVC and then the kNN classifier.

Table 10 – Classification results for the *Actinobacteria* dataset using the features of plant area, plant volume, water quantity, and water steps. The evaluation metric used is accuracy. For each classifier, the highest accuracy is shown in bold, and the cells with the top 5 values are colored in light blue.

	Classifiers		
Features	**kNN**	**LDA**	**SVC**
pa [a]	94.64 ± 4.73	87.84 ± 7.08	94.57 ± 4.84
pv [b]	**95.65** ± **4.26**	88.40 ± 7.30	95.40 ± 4.38
wq [c]	91.05 ± 6.31	87.30 ± 7.50	95.39 ± 4.61
ws [d]	90.78 ± 6.73	89.66 ± 6.97	94.92 ± 4.53
pa, pv	94.54 ± 4.59	94.82 ± 4.97	96.45 ± 4.18
wq, ws	90.83 ± 6.61	98.23 ± 3.06	96.05 ± 4.26
pa, wq	92.76 ± 5.72	95.90 ± 4.28	95.52 ± 4.52
pa, ws	92.53 ± 5.70	96.85 ± 4.07	95.19 ± 4.54
pv, wq	93.16 ± 5.13	96.03 ± 4.36	96.05 ± 4.24
pv, ws	92.96 ± 5.39	98.06 ± 3.17	95.75 ± 4.39
pa, pv, ws	93.41 ± 5.18	98.24 ± 2.80	96.27 ± 4.22
pa, pv, wq	93.39 ± 5.16	96.17 ± 4.20	**97.16** ± **4.11**
pa, wq, ws	92.16 ± 6.01	98.82 ± 2.35	96.55 ± 4.01
pv, wq, ws	92.41 ± 5.83	98.51 ± 2.75	96.25 ± 3.80
pa, pv, wq, ws	93.01 ± 5.43	**99.32** ± **2.00**	96.73 ± 3.82

[a] *pa*: plant area
[b] *pv*: plant volume
[c] *wq*: water quantity
[d] *ws*: water steps

Source: Research data.

5.3.2 Study of model parameters

In order to appreciate the impact of the parameters on the classification results, we ran a series of experiments on the *Actinobacteria* dataset with different parameter values, using the kNN, LDA and Linear SVC classifiers with repeated stratified cross-validation scheme. We select the *Actinobacteria* dataset because it is the second dataset in terms of number of networks and has 3 classes. The other metabolic datasets have between 3 to 4 classes.

According to the best accuracy result shown in Table 10 we select the combination of the 4 features of plant area, plant volume, water quantity, and water steps, $\{pa, pv, wq, ws\}$, to present the results for the following experiments in the exploration of the parameters.

To set the initial parameters of the model, we look into the properties of the networks shown in Table 8 and Table 9 to establish the initial set of values for the parameters. A grid search strategy to test every possible combination of the values of the parameters is not feasible, another approach is to set values related to the properties of the network, then gradually vary one parameter, with the objective to maximize the accuracy result.

The following texts present the plots with the classification results by varying one parameter and fixing the others. The parameters of the model that were studied are: water to grow wg, height to grow hg, percent of water pw, time alive water tw, percent of plant pp, time alive plant tp, iterations to add water iw, and number of iterations it for the model simulation.

In Figure 25 the parameter water to grow wg is varied from 1 to 10 in steps of 1. This parameter defines how many water agents a plant needs to consume to grow.

The LDA classifier presents their best accuracies with 98.54% and 98.49%, these with wg around 3 to 5 , with wg less than 3 and greater than 5 the accuracy decreases. The Linear SVC classifier got its best result with 97.94% and $wg = 5$, its second best result was when wg is 8. In the case of the kNN classifier presents a different behavior, since the results it shows that as wg increases, the accuracy got better, having their best results of 95.43% and 95.46% with wg equal to 8 and 9 respectively.

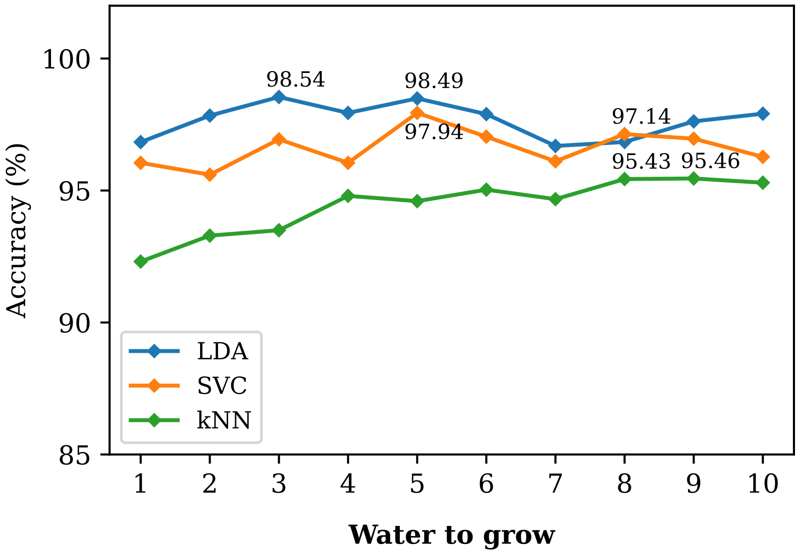
In Figure 26 the parameter height to grow hg is varied from 2 to 200 in steps of 20. This parameter determines the scope of a plant agent, and it is based on the degree difference between a node and its neighbors.

The LDA classifier obtains its best accuracies values with 99.17% and 99.02% around hg of 20 to 60, using higher values than 60 the accuracy decreases. Similarly, the Linear SVC classifier obtains its best accuracy values with hg of 20 and 60, higher values tend to decrease the accuracy. An opposite effect occurs with the kNN classifier in which, as hg increases the precision also increases, reaching 96.44% and 96.40% with hg of 160 and 200 respectively.

In Figure 27 the parameter percent of water pw is varied from 10 to 100 in steps of 10. This parameter determines the quantity of water with which the simulation starts, according to table 9 on average the *Actinobacteria* dataset has 1127 nodes and if we set pw to 50, the quantity of water in each network on average is $1127 \times 50\% \approx 564$ water. As in the case of percent plant, greater values of pw indicate that the water occupies more nodes in the network.

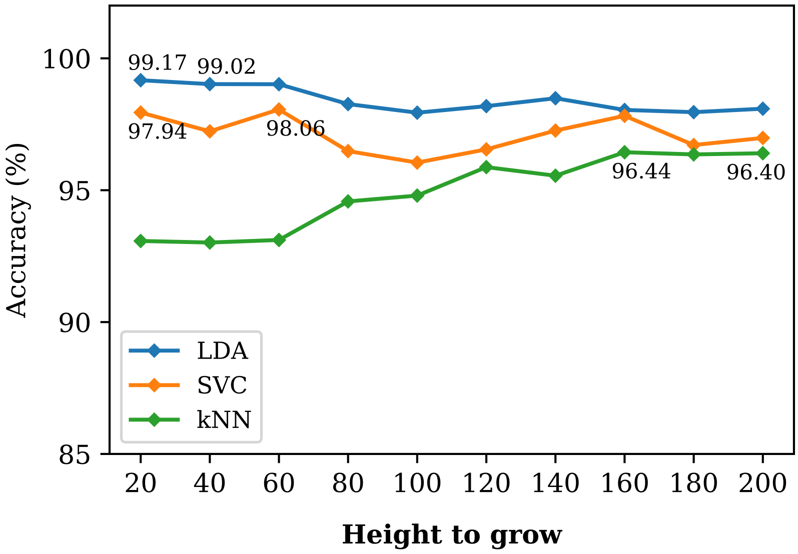
The LDA classifier presents the best results with 99.02% and 99% in pw equal to 60%

Figure 25 – Classification accuracy in the *Actinobacteria* dataset varying the parameter water to grow $wg = [1, 10]$ in steps of 1. The values of the rest of the parameters are: $hg = 100$, $pw = 60$, $tw = 0$, $pp = 30$, $tp = 0$, $iw = 10$, $it = 200$. The feature vector used is $\{pa, pv, wq, ws\}$



Source: Elaborated by the author.

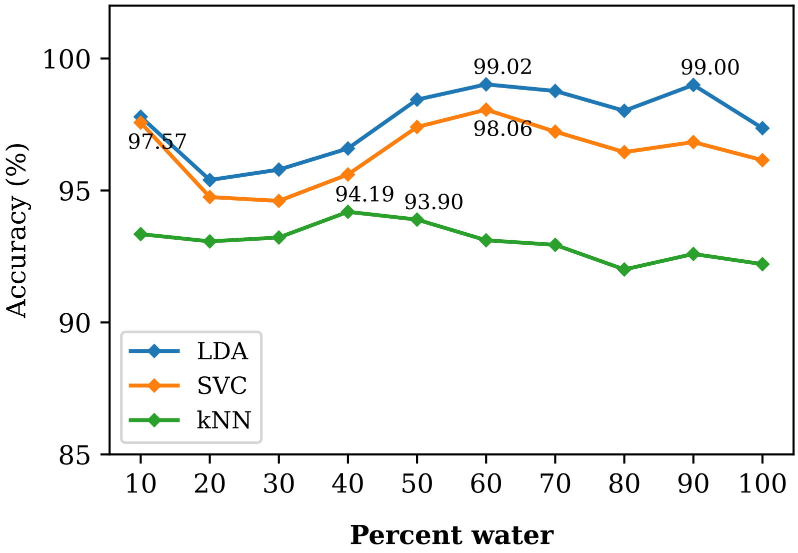
Figure 26 – Classification accuracy in the *Actinobacteria* dataset varying the parameter height to grow $hg = [20, 200]$ in steps of 20. The values of the rest of the parameters are: $wg = 4$, $pw = 60$, $tw = 0$, $pp = 30$, $tp = 0$, $iw = 10$, $it = 200$. The feature vector used is $\{pa, pv, wq, ws\}$



Source: Elaborated by the author.

and 90% respectively. With values less than 60% the accuracy decreases. The Linear SVC classifier visually presents a similar behavior to the LDA, its best value 98% is also with $pw = 60$. The kNN classifier presents its best results 94.19% and 93.90% with pw equal to 40 and 50 respectively, adding more water to the model negatively influences its performance. In general, increasing the value of pw influences the processing time of the model, since a greater number of agents need to be processed.

Figure 27 – Classification accuracy in the *Actinobacteria* dataset varying the percent of water $pw = [10, 100]$ in steps of 10. The values of the rest of the parameters are: $wg = 4$, $hg = 60$, $tw = 30$, $pp = 30$, $tp = 0$, $iw = 10$, $it = 200$. The feature vector used is $\{pa, pv, wq, ws\}$



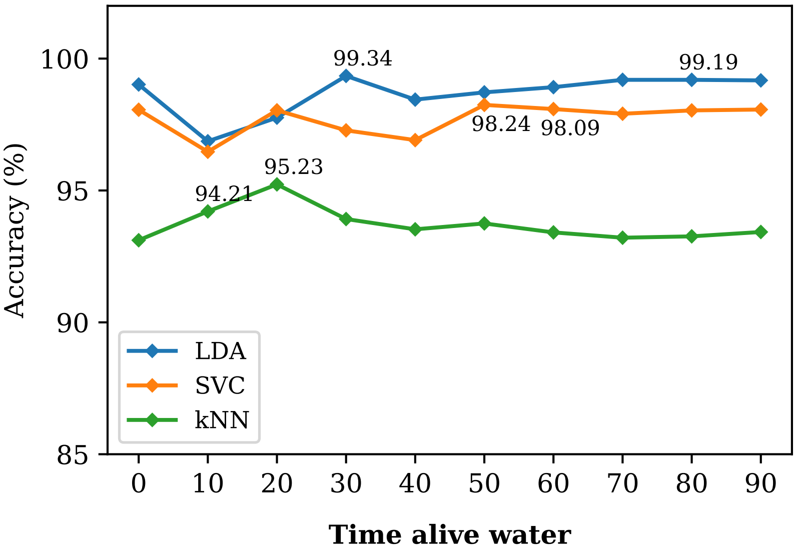
Source: Elaborated by the author.

In Figure 28 the parameter time alive of water tw is varied from 0 to 90 in steps of 10. This parameter defines how long a water agent remains in the simulation, as in the case of the parameter time alive of plant tp, the special value $tw = 0$ indicates that the water does not have a limit on its time alive, it is present during all the simulation.

The LDA classifier presents the best accuracy with 99.34% and $tw = 30$, less than 30 the accuracy decreases, and with values greater than 50 the accuracy increases, getting its second best value with 99.19% and $tw = 80$. Considering that water is being added every $iw = 10$ iterations, these results indicate that a proportional value of time alive water tw influences the classification results, and that it is better for the model to have water present longer during simulations. The Linear SVC classifier got its best result of 98.24% and 98.09% with tw equal to 50 and 60 respectively. The KNN classifier obtain its best results with 95.23% and $tw = 20$, and increasing the time alive of water does not improve the performance of the kNN classifier. It is worth mentioning that when tw is 0 (present throughout the simulation) the classification values in LDA and Linear SVC are close to their best values.

In Figure 29 the parameter percent of plant pp is varied from 10 to 100 in steps of 10. This parameter determines the number of plants with which the simulation starts, according to table 9 on average the *Actinobacteria* dataset has 1127 nodes and if we set pp to 30, the number of plants in each network on average is 1127 × 30% ≈ 338 plants. Greater values of pp indicate that it occupies more nodes in the network.

Figure 28 – Classification accuracy in the *Actinobacteria* dataset varying the time alive of water $tw = [0,90]$ in steps of 10. The special value $tw = 0$ indicates that the water agent is present during all the simulation. The values of the rest of the parameters are: $wg = 4$, $hg = 60$, $pw = 60$, $tp = 0$, $pp = 30$, $iw = 10$, $it = 200$. The feature vector used is $\{pa, pv, wq, ws\}$



Source: Elaborated by the author.

The LDA classifier presents the best accuracy results around 10% to 40% with 99.34% with $pp = 30$, and when increasing pp the accuracy results decrease. The Linear SVC classifier presents a different behavior, with their two best results with pp in 20% and 70%, and adding more plants does not improve the accuracy. The KNN classifier obtain its best results around 50% to 70% getting 95.27% with $pp = 50$, other values do not improve the accuracy.

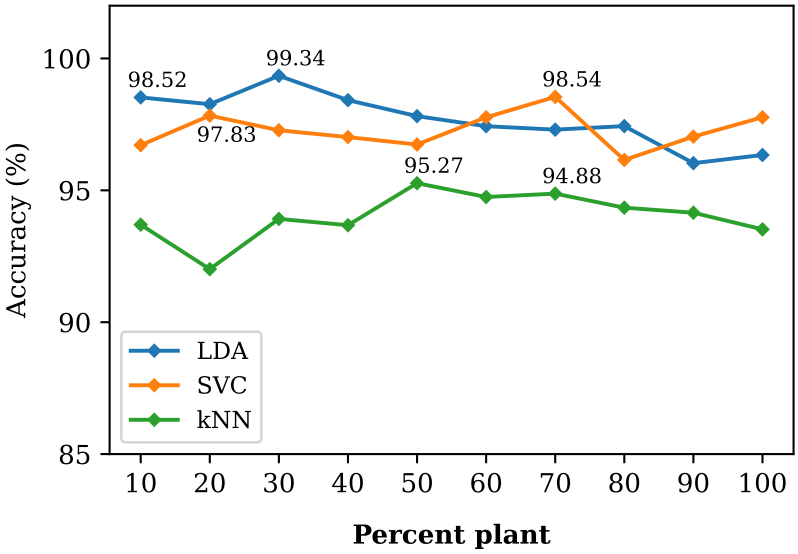
In Figure 30 the parameter time alive of plant tp is varied from 0 to 180 in steps of 20. This parameter defines how long a plant remains in the simulation, it is important to note that in this case $tp = 0$ is a special value that indicates that the plant does not have a limit on its time alive, it is present during all the simulation.

The LDA classifier presents the best accuracy with 99.34% and $tp = 0$, and its second best value with 99.09% and $tp = 120$, this indicates that as the time alive of the plant increases the accuracy results are better. The Linear SVC classifier got its best result with 98.14% and $tp = 140$, when tp is less than 60 the accuracy results decrease. The KNN classifier obtain its best results with 94.41% and $tp = 100$, and in general its accuracy results vary little, so this parameter does not affect the performance of the kNN classifier.

In Figure 31 the parameter number of iterations to add water iw is varied from 2 to 38 in steps of 2. When the number of iterations for adding water is greater than the time alive water, this parameter affects the amount of water present during the simulation.

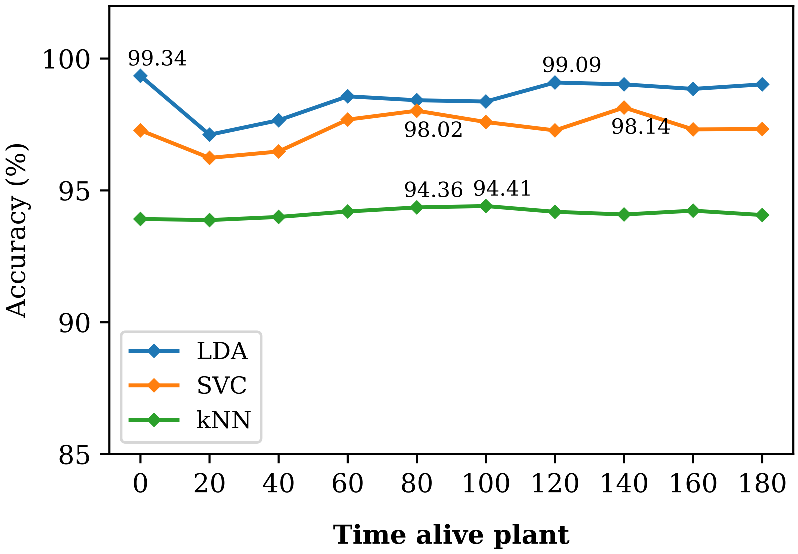
The LDA classifier presents accuracy results above 95% when iw is less than 30, pro-

Figure 29 – Classification accuracy in the *Actinobacteria* dataset varying the percent of plant $pp = [10, 100]$ in steps of 10. The values of the rest of the parameters are: $wg = 4$, $hg = 60$, $pw = 60$, $tw = 30$, $tp = 0$, $iw = 10$, $it = 200$. The feature vector used is $\{pa, pv, wq, ws\}$



Source: Elaborated by the author.

Figure 30 – Classification accuracy in the *Actinobacteria* dataset varying the time alive of plant $tp = [0, 180]$ in steps of 20. The special value $tp = 0$ indicates that the plant is present during all the simulation. The values of the rest of the parameters are: $wg = 4$, $hg = 60$, $pw = 60$, $tw = 30$, $pp = 30$, $iw = 10$, $it = 200$. The feature vector used is $\{pa, pv, wq, ws\}$

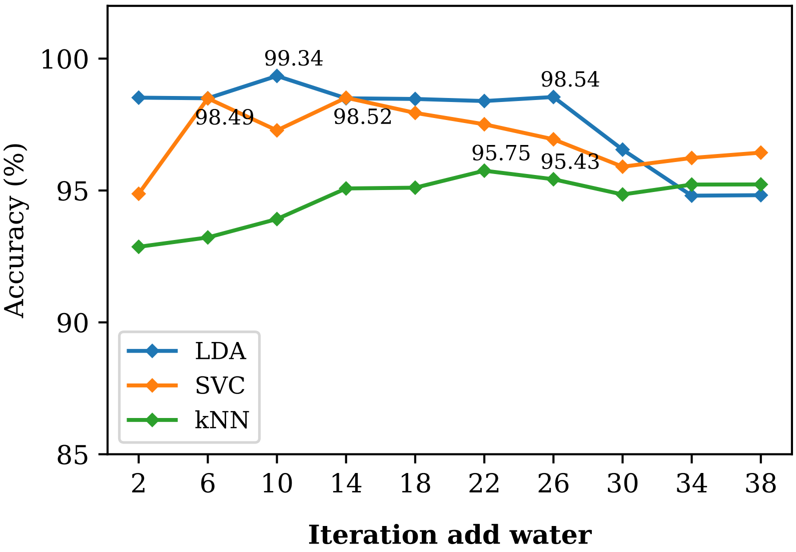


Source: Elaborated by the author.

viding the best result of 99.34% with 10 iterations. The Linear SVC classifier presents the best result with $iw = 14$ with 98.52%, from which its results decrease. On the other hand, the kNN classifier improves as the value of iw increases, obtaining a maximum of 95.75% at $iw = 22$. The

3 classifiers begin to present lower results when the number of iterations to add water is greater than or equal to the lifetime of the water $tw = 30$. When iw is greater than 30, LDA provides the lowest accuracy results of the 3 classifiers.

Figure 31 – Classification accuracy in the *Actinobacteria* dataset varying the number of iterations to add water $iw = [2,38]$ in steps of 4. The values of the rest of the parameters are: $wg = 4$, $hg = 60$, $pw = 60$, $tw = 30$, $pp = 30$, $tp = 0$, $it = 200$. The feature vector used is $\{pa, pv, wq, ws\}$



Source: Elaborated by the author.

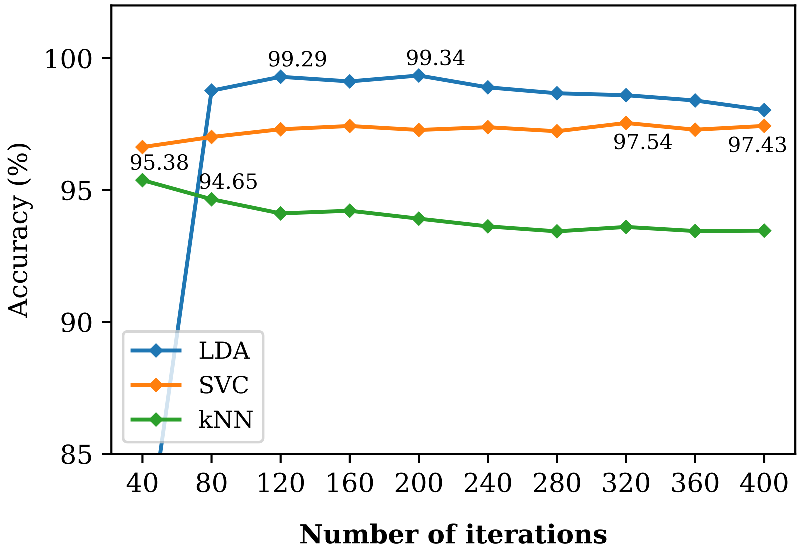
In Figure 32 the parameter number of iterations it determines the number of times the simulation is run. It is varied from 40 to 400 in steps of 40. With 40 iterations the LDA classifier has less than 85% of accuracy, and does not appear in the plot, then its accuracy improve with more iterations, giving the best value of 99.34% with 200 iterations. The Linear SVC classifier improves consistently with the number of iterations. But, in the case of the kNN classifier, its classification results decrease with more iterations.

Table 11 presents a summary of the values employed to study the parameters of the model in the *Actinobacteria* dataset. The table shows the parameters and the range of values used. In each row is highlighted in light blue two values that provided the best classification results using the LDA classifier. According to the classification results of the plots shown previously, the values colored in light blue indicate the range of values in which the model has good performance, with accuracy between 98.49% to 99.34%.

5.4 Pattern recognition in synthetic datasets

The following values of parameters were chosen because provided the best accuracy values with the LDA classifier using the 4 four features $\{pa, pv, wq, ws\}$.

Figure 32 – Classification accuracy in the *Actinobacteria* dataset varying the parameter number of iterations $it = [40, 200]$ in steps of 40. The values of the rest of the parameters are: $wg = 4$, $hg = 60$, $pw = 60$, $tw = 30$, $pp = 30$, $tp = 0$, $iw = 10$. The feature vector used is $\{pa, pv, wq, ws\}$



Source: Elaborated by the author.

Table 11 – Range of values used to study the parameters of the Growth Model in the *Actinobacteria* dataset. The values that provided the two best accuracies are colored in light blue

Parameter	Values									
wg	1	2	3	4	5	6	7	8	9	10
hg	20	40	60	80	100	120	140	160	180	200
pw	10	20	30	40	50	60	70	80	90	100
tw	0	10	20	30	40	50	60	70	80	90
pp	10	20	30	40	50	60	70	80	90	100
tp	0	20	40	60	80	100	120	140	160	180
iw	2	6	10	14	18	22	26	30	34	38
it	40	80	120	160	200	240	280	320	360	400

Source: Elaborated by the author.

According to the exploration made, we can vary the parameters water to grow wg and height to grow hg, and we can fix the rest of parameters. In the exploration these values allowed the model to perform consistently. The water to grow wg varies between 2, 4 and 6. The height to grow hg varies between 40, 60, and 100. The parameters rest of the parameters are fixed. The percent of water $pw = 60$. The time alive water $tw = 30$. The percent of plant $pp = 30$. The time alive plant $tp = 0$, the special value, that indicates that the plant is present during all the simulation. The sum of the values of percent water pw and percent plant pp is 90%, it means that they occupy 90% of the size of the network, giving to all the nodes the probability of containing agents at a moment of the simulation The number of iterations to add water $iw = 10$. The number of iterations of the simulation $it = 200$.

The parameters use for the pattern recognition experiments in the synthetic datasets are in Table 12. Each column shows a set of values used, with these parameters we have 5 different executions of the model in each dataset.

Table 12 – Parameters selected to execute the Growth Model in the synthetic network datasets

	Set of values				
Parameter	**First**	**Second**	**Third**	**Fourth**	**Fifth**
wg	2	2	2	4	6
hg	40	60	100	100	100
pw	60	60	60	60	60
tw	30	30	30	30	30
pp	30	30	30	30	30
tp	0	0	0	0	0
iw	10	10	10	10	10
it	200	200	200	200	200

Source: Research data.

5.4.1 4-models dataset

Table 13 presents the classification results in the *4-models* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset, the extracted features enable the categorization of the 4 network classes, achieving an accuracy score of 99.13%. This result indicates that out of the 11200 networks, only 98 are misclassified.

Table 14 presents the classification metrics for the best result obtained in the *4-models* dataset. The values of precision and recall close to 1 for each of the 4 classes indicates that the classifier is accurately identifying positive instances and effectively capturing the majority of them. The other metrics also verify these results.

Figure 33 presents the confusion matrix for the best result obtained in the *4-models* dataset. Computing the accuracy value from the confusion matrix, the values shown vary slightly from the previous results, this is due to the fact that the implementation of the confusion matrix in *Scikit-Learn* uses the standard k-fold Cross-validation, unlike the results shown in Table 13 where *Scikit-learn* uses Repeated Stratified k-fold cross-validation.

Figure 34 shows the area under the curve for each class for the best result obtained in the *4-models* dataset.

Table 13 – Classification results using the LDA classifier for the *4-models* dataset. The highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=6*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=100*	*hg=100*
pa	84.89 ± 1.03	84.79 ± 1.08	84.63 ± 1.14	89.51 ± 0.87	84.71 ± 0.87
pv	91.25 ± 0.79	91.35 ± 0.81	91.40 ± 0.87	92.16 ± 0.83	87.89 ± 0.94
wq	87.79 ± 0.92	87.91 ± 0.97	88.02 ± 0.82	82.56 ± 1.02	81.81 ± 0.97
ws	92.20 ± 0.85	92.31 ± 0.74	92.11 ± 0.85	83.95 ± 0.98	82.69 ± 1.05
pa, pv	98.54 ± 0.36	98.57 ± 0.34	98.57 ± 0.35	95.95 ± 0.61	91.85 ± 0.70
wq, ws	95.75 ± 0.62	95.62 ± 0.59	95.56 ± 0.58	94.78 ± 0.66	95.78 ± 0.52
pa, wq	96.27 ± 0.61	96.28 ± 0.54	96.18 ± 0.52	94.59 ± 0.63	90.08 ± 0.84
pa, ws	98.01 ± 0.41	98.02 ± 0.41	97.97 ± 0.44	93.32 ± 0.72	93.30 ± 0.62
pv, wq	96.93 ± 0.51	96.93 ± 0.48	96.95 ± 0.50	94.99 ± 0.65	90.03 ± 0.81
pv, ws	97.78 ± 0.43	97.81 ± 0.37	97.80 ± 0.42	95.87 ± 0.58	92.56 ± 0.69
pa, pv, ws	99.03 ± 0.29	99.06 ± 0.26	98.97 ± 0.27	97.24 ± 0.45	95.47 ± 0.53
pa, pv, wq	98.99 ± 0.31	99.00 ± 0.27	98.96 ± 0.26	97.39 ± 0.45	94.49 ± 0.62
pa, wq, ws	98.55 ± 0.37	98.55 ± 0.33	98.51 ± 0.34	97.36 ± 0.45	96.83 ± 0.49
pv, wq, ws	98.55 ± 0.34	98.53 ± 0.32	98.54 ± 0.32	97.72 ± 0.43	96.58 ± 0.52
pa, pv, wq, ws	**99.12 ± 0.28**	**99.13 ± 0.26**	**99.07 ± 0.27**	**98.28 ± 0.39**	**97.62 ± 0.44**

Source: Research data.

Table 14 – Classification metrics for the best result of classification in the *4-models* dataset in Table 13

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
barabasi	0.999	0.985	0.000	0.992	0.990	1.000	2800
erdos	0.978	0.994	0.008	0.986	0.981	1.000	2800
geo	0.987	0.987	0.004	0.987	0.983	1.000	2800
watts	1.000	0.997	0.000	0.998	0.998	0.999	2800

Source: Research data.

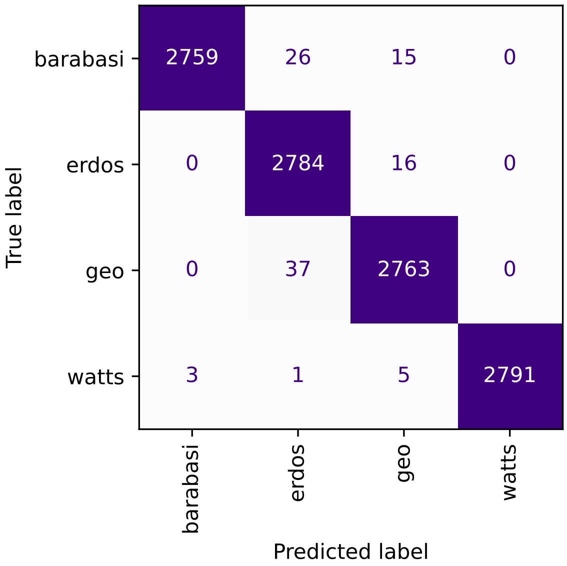
5.4.2 4-models + $\langle k \rangle$ dataset

Table 15 presents the classification results in the *4-models* + $\langle k \rangle$ dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset, the extracted features were not able to effectively categorize the 28 network classes, achieving an accuracy score of 84.97%. This result indicates that out of the 2800 networks, 420 networks were misclassified.

Table 16 presents the classification metrics for the best result obtained in the *4-models* + $\langle k \rangle$ dataset. The low precision scores for the barabasi networks indicate that the model failed to identify the positive instances.

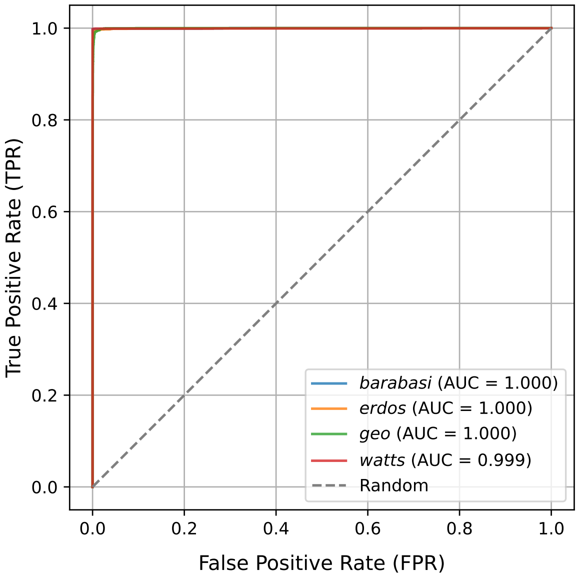
Figure 35 presents the confusion matrix for the best result obtained in the *4-models*

Figure 33 – Confusion matrix obtained for the best results of classification in the *4-models* dataset in Table 13



Source: Elaborated by the author.

Figure 34 – AUC-ROC obtained for the best results of classification in the *4-models* dataset in Table 13



Source: Elaborated by the author.

+ $\langle k \rangle$ dataset. The confusion matrix shows us which are the classes that presented a higher misclassification rate, which in this case corresponds to the networks of the *barabasi* model, the extracted characteristics were not able to differentiate their intra-class variations.

Table 15 – Classification results using the LDA classifier for the *4-models* + $\langle k \rangle$ dataset. The highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=6*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=100*	*hg=100*
pa	73.10 ± 2.58	74.37 ± 2.23	73.90 ± 2.19	73.08 ± 2.21	72.24 ± 1.96
pv	79.00 ± 1.92	78.56 ± 2.00	79.03 ± 1.90	79.09 ± 1.84	76.38 ± 2.22
wq	64.50 ± 2.44	64.42 ± 2.37	63.98 ± 2.49	60.83 ± 2.57	64.59 ± 2.79
ws	66.72 ± 2.34	66.66 ± 2.23	65.80 ± 1.98	60.47 ± 2.51	65.30 ± 2.38
pa, pv	84.24 ± 1.78	84.84 ± 1.67	**84.86 ± 1.68**	82.43 ± 1.84	80.96 ± 2.30
wq, ws	71.73 ± 2.26	71.01 ± 2.10	69.55 ± 2.13	70.93 ± 2.57	76.80 ± 2.05
pa, wq	79.42 ± 2.01	79.80 ± 1.95	79.55 ± 2.05	76.26 ± 2.22	75.81 ± 2.51
pa, ws	79.72 ± 2.10	81.15 ± 1.93	80.48 ± 2.15	75.91 ± 2.28	76.47 ± 2.09
pv, wq	81.56 ± 1.68	81.04 ± 1.86	81.15 ± 1.66	81.09 ± 1.90	81.18 ± 2.01
pv, ws	80.52 ± 1.76	80.81 ± 1.82	80.68 ± 1.67	80.22 ± 1.82	82.70 ± 1.80
pa, pv, ws	83.66 ± 1.77	84.50 ± 1.80	84.67 ± 1.58	81.78 ± 1.96	82.39 ± 1.79
pa, pv, wq	**84.94 ± 1.57**	**84.97 ± 1.68**	84.74 ± 1.64	**82.75 ± 1.88**	82.36 ± 2.01
pa, wq, ws	80.04 ± 2.11	80.08 ± 2.11	78.90 ± 2.02	78.32 ± 2.21	80.46 ± 1.91
pv, wq, ws	81.48 ± 1.75	80.83 ± 1.91	80.71 ± 1.84	81.49 ± 1.90	**83.81 ± 1.83**
pa, pv, wq, ws	83.69 ± 1.66	83.86 ± 1.84	83.52 ± 1.74	81.75 ± 1.90	82.44 ± 1.86

Source: Research data.

5.4.3 Scale-free dataset

Table 17 presents the classification results in the *Scale-free* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset, the extracted features enable the categorization of the 5 network classes, achieving an accuracy score of 99.82%. This result indicates that only 1 network was misclassified out of the 500 networks.

Table 18 presents the classification metrics for the best result obtained in the *Scale-free* dataset. The different metrics and its values close to 1 indicate that the model is effectively distinguishing the networks in each class.

Figure 36 presents the confusion matrix for the best result obtained in the *Scale-free* dataset. In this case, the confusion matriz shows that there are 3 incorrectly misclassified instances, 1 of *mendes* and 2 of *nonlinear15*.

Figure 37 shows the area under the curve for each class for the best result obtained in the *Scale-free* dataset. The AUC-ROC of the 5 classes are 1 or values close to 1, this indicates that the model effectively discriminate the 5 classes.

Table 16 – Classification metrics for the best result of classification in the *4-models* + $\langle k \rangle$ dataset in Table 15

Class	precision	recall	FPR	f1-score	MCC	AUC	support
barabasi_k=10	0.452	0.420	0.019	0.435	0.415	0.962	100
barabasi_k=12	0.342	0.400	0.029	0.369	0.345	0.955	100
barabasi_k=14	0.340	0.330	0.024	0.335	0.311	0.955	100
barabasi_k=16	0.562	0.590	0.017	0.576	0.560	0.978	100
barabasi_k=4	1.000	0.840	0.000	0.913	0.914	0.983	100
barabasi_k=6	0.780	0.780	0.008	0.780	0.772	0.990	100
barabasi_k=8	0.644	0.670	0.014	0.657	0.644	0.974	100
erdos_k=10	0.961	0.990	0.001	0.975	0.975	1.000	100
erdos_k=12	0.943	1.000	0.002	0.971	0.970	1.000	100
erdos_k=14	0.906	0.960	0.004	0.932	0.930	0.999	100
erdos_k=16	0.969	0.930	0.001	0.949	0.947	1.000	100
erdos_k=4	0.915	0.970	0.003	0.942	0.940	1.000	100
erdos_k=6	0.952	0.990	0.002	0.971	0.970	1.000	100
erdos_k=8	0.943	1.000	0.002	0.971	0.970	1.000	100
geo_k=10	0.948	0.910	0.002	0.929	0.926	0.999	100
geo_k=12	0.928	0.900	0.003	0.914	0.911	0.999	100
geo_k=14	0.794	0.810	0.008	0.802	0.795	0.995	100
geo_k=16	0.871	0.810	0.004	0.839	0.834	0.997	100
geo_k=4	0.968	0.900	0.001	0.933	0.931	1.000	100
geo_k=6	0.989	0.930	0.000	0.959	0.958	1.000	100
geo_k=8	0.979	0.940	0.001	0.959	0.958	1.000	100
watts_k=10	0.980	1.000	0.001	0.990	0.990	1.000	100
watts_k=12	0.961	0.990	0.001	0.975	0.975	1.000	100
watts_k=14	0.894	0.930	0.004	0.912	0.909	0.999	100
watts_k=16	0.957	0.890	0.001	0.922	0.920	0.999	100
watts_k=4	1.000	1.000	0.000	1.000	1.000	1.000	100
watts_k=6	0.990	1.000	0.000	0.995	0.995	1.000	100
watts_k=8	1.000	0.990	0.000	0.995	0.995	0.999	100

Source: Research data.

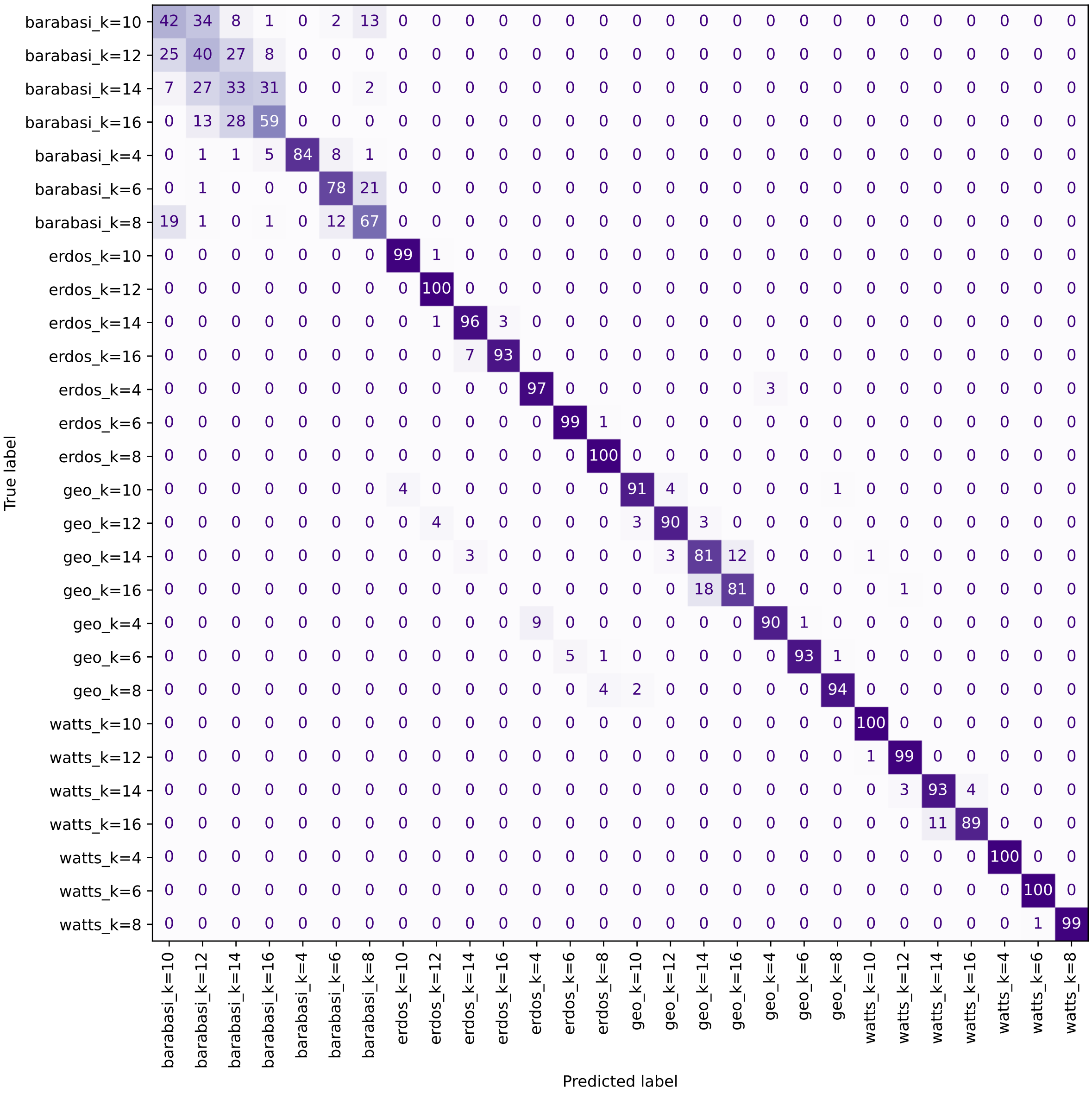
5.4.4 Noise-0 dataset

Table 19 presents the classification results in the *Noise-0* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset, the extracted features enable the categorization of the 8 network classes, achieving an accuracy score of 99.56%. This result indicates that out of the 800 networks, only 8 are misclassified.

Table 20 presents the classification metrics for the best result obtained in the *Noise-0* dataset. The recall value of 0.95 in *GEO* indicates that the classifier is performing well overall but may be having some difficulty in correctly identifying instances in this specific class.

Figure 38 presents the confusion matrix for the best result obtained in the *Noise-0* dataset.

Figure 35 – Confusion matrix obtained for the best results of classification in the *4-models* + $\langle k \rangle$ dataset in Table 15



	barabasi_k=10	barabasi_k=12	barabasi_k=14	barabasi_k=16	barabasi_k=4	barabasi_k=6	barabasi_k=8	erdos_k=10	erdos_k=12	erdos_k=14	erdos_k=16	erdos_k=4	erdos_k=6	erdos_k=8	geo_k=10	geo_k=12	geo_k=14	geo_k=16	geo_k=4	geo_k=6	geo_k=8	watts_k=10	watts_k=12	watts_k=14	watts_k=16	watts_k=4	watts_k=6	watts_k=8
barabasi_k=10	42	34	8	1	0	2	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
barabasi_k=12	25	40	27	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
barabasi_k=14	7	27	33	31	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
barabasi_k=16	0	13	28	59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
barabasi_k=4	0	1	1	5	84	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
barabasi_k=6	0	1	0	0	0	78	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
barabasi_k=8	19	1	0	1	0	12	67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
erdos_k=10	0	0	0	0	0	0	0	99	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
erdos_k=12	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
erdos_k=14	0	0	0	0	0	0	0	0	1	96	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
erdos_k=16	0	0	0	0	0	0	0	0	0	7	93	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
erdos_k=4	0	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
erdos_k=6	0	0	0	0	0	0	0	0	0	0	0	0	99	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
erdos_k=8	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
geo_k=10	0	0	0	0	0	0	0	4	0	0	0	0	0	0	91	4	0	0	0	0	1	0	0	0	0	0	0	0
geo_k=12	0	0	0	0	0	0	0	0	4	0	0	0	0	0	3	90	3	0	0	0	0	0	0	0	0	0	0	0
geo_k=14	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	3	81	12	0	0	0	1	0	0	0	0	0	0
geo_k=16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	81	0	0	0	0	1	0	0	0	0	0
geo_k=4	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	90	1	0	0	0	0	0	0	0	0
geo_k=6	0	0	0	0	0	0	0	0	0	0	0	0	5	1	0	0	0	0	0	93	1	0	0	0	0	0	0	0
geo_k=8	0	0	0	0	0	0	0	0	0	0	0	0	0	4	2	0	0	0	0	0	94	0	0	0	0	0	0	0
watts_k=10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
watts_k=12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	99	0	0	0	0	0
watts_k=14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	93	4	0	0	0
watts_k=16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	89	0	0	0
watts_k=4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
watts_k=6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
watts_k=8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	99

Source: Elaborated by the author.

The confusion matrix indicates that the *GEO* class has 5 instances that are incorrectly assigned as *ER* class.

Figure 39 shows the area under the curve for each class for the best result obtained in the *Noise-0* dataset. In addition to showing the AUC-ROC curve with its normal limits in the x-y axes from 0 to 1, a plot with the x-axis from 0 to 0.2 and the y-axis from 0.8 to 1 is shown to appreciate the details of very close AUC-ROC values. In general the AUC-ROC is 1 or close to 1 for each class, this indicates a good performance.

Table 17 – Classification results using the LDA classifier for the *Scale-free* dataset. The highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=6*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=100*	*hg=100*
pa	**99.56 ± 0.98**	**99.18 ± 1.17**	98.78 ± 1.55	98.13 ± 1.83	99.26 ± 1.15
pv	98.38 ± 1.76	98.63 ± 1.56	98.58 ± 1.61	**98.78 ± 1.48**	99.58 ± 0.81
wq	98.84 ± 1.46	98.15 ± 1.86	98.12 ± 1.89	98.63 ± 1.42	**99.82 ± 0.57**
ws	96.58 ± 2.47	95.52 ± 2.75	97.04 ± 2.34	98.19 ± 1.96	99.56 ± 1.00
pa, pv	97.69 ± 2.15	96.79 ± 2.38	96.94 ± 2.07	97.78 ± 2.06	98.78 ± 1.74
wq, ws	93.88 ± 3.58	92.16 ± 3.86	96.78 ± 2.23	91.68 ± 3.86	97.32 ± 2.53
pa, wq	97.74 ± 2.21	96.46 ± 2.76	95.28 ± 2.84	97.31 ± 2.25	97.96 ± 1.88
pa, ws	98.04 ± 1.99	97.67 ± 2.18	**99.02 ± 1.54**	98.02 ± 1.97	99.42 ± 1.14
pv, wq	96.31 ± 2.56	95.36 ± 2.51	95.84 ± 2.59	97.54 ± 2.13	98.32 ± 1.64
pv, ws	96.35 ± 2.51	95.08 ± 3.10	98.70 ± 1.45	94.95 ± 2.92	97.90 ± 2.09
pa, pv, ws	96.79 ± 2.60	95.17 ± 2.96	97.66 ± 2.10	91.72 ± 3.82	95.56 ± 3.11
pa, pv, wq	93.47 ± 3.20	92.47 ± 3.40	93.78 ± 3.04	92.54 ± 3.76	92.44 ± 3.83
pa, wq, ws	93.20 ± 3.37	92.14 ± 3.53	95.48 ± 2.62	88.95 ± 4.62	94.48 ± 3.19
pv, wq, ws	90.12 ± 4.26	90.77 ± 3.75	96.22 ± 2.48	88.81 ± 4.45	95.20 ± 3.15
pa, pv, wq, ws	91.19 ± 3.97	91.68 ± 4.06	94.10 ± 3.28	88.06 ± 4.69	92.58 ± 3.67

Source: Research data.

Table 18 – Classification metrics for the best result of classification in the *Scale-free* dataset in Table 17

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
barabasi	1.000	1.000	0.000	1.000	1.000	1.000	100
mendes	1.000	0.990	0.000	0.995	0.994	1.000	100
nonlinear05	0.990	1.000	0.003	0.995	0.994	0.999	100
nonlinear15	1.000	0.980	0.000	0.990	0.987	1.000	100
nonlinear2	0.980	1.000	0.005	0.990	0.988	1.000	100

Source: Research data.

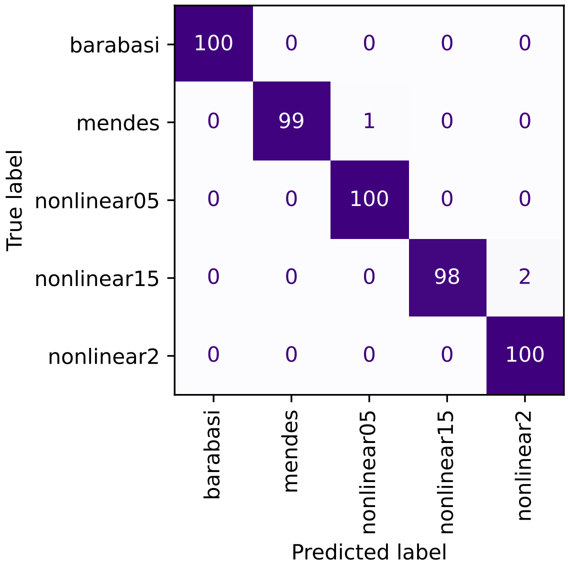
5.4.5 Noise-10% dataset

Table 21 presents the classification results in the *Noise-10%* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset, the extracted features enable the categorization of the 8 network classes, achieving an accuracy score of 98.26%. This result indicates that out of the 800 networks, only 14 are misclassified.

Table 22 presents the classification metrics for the best result obtained in the *Noise 10%* dataset. In this cases, the recall values of *BANL15* of 0.95 and *GEO* of 0.91 indicates that the classifier have some difficulty in identifying instances of these two classes.

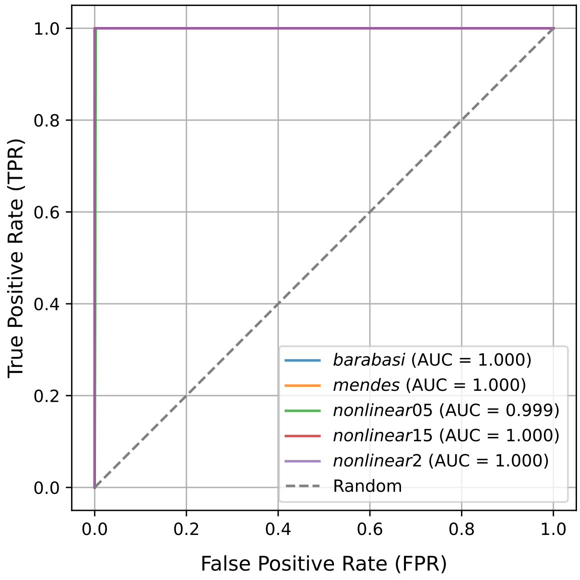
Figure 40 presents the confusion matrix for the best result obtained in the *Noise 10%*

Figure 36 – Confusion matrix obtained for the best results of classification in the *Scale-free* dataset in Table 17



Source: Elaborated by the author.

Figure 37 – AUC-ROC obtained for the best results of classification in the *Scale-free* dataset in Table 17



Source: Elaborated by the author.

dataset. The confusion matrix also show that the model has some difficulty in the identification of 5 networks of the *BANL15* class and 9 networks of the *ER* class.

Figure 41 shows the area under the curve for each class for the best result obtained in the *Noise 10%* dataset. The zoomed plot of the AUC-ROC indicate that the *BANL2* class has the most difficulty to be classified.

Table 19 – Classification results using the LDA classifier for the *Noise-0* dataset. The highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=6*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=100*	*hg=100*
pa	**98.87 ± 1.04**	98.43 ± 1.33	98.29 ± 1.18	96.96 ± 1.79	96.66 ± 1.88
pv	98.47 ± 1.23	**98.49 ± 1.31**	98.45 ± 1.30	98.36 ± 1.39	98.96 ± 1.08
wq	97.50 ± 1.60	97.24 ± 1.77	96.90 ± 1.75	97.81 ± 1.70	96.93 ± 1.74
ws	97.06 ± 1.84	96.83 ± 1.96	97.27 ± 1.73	95.76 ± 2.14	95.30 ± 2.13
pa, pv	98.84 ± 1.17	98.25 ± 1.55	98.41 ± 1.29	98.66 ± 1.22	98.64 ± 1.30
wq, ws	96.71 ± 2.00	95.92 ± 2.27	98.41 ± 1.29	98.28 ± 1.38	97.39 ± 1.61
pa, wq	98.59 ± 1.35	97.71 ± 1.71	97.15 ± 1.70	98.34 ± 1.52	97.56 ± 1.61
pa, ws	98.49 ± 1.29	98.21 ± 1.51	98.52 ± 1.28	97.66 ± 1.63	97.77 ± 1.59
pv, wq	97.95 ± 1.56	97.25 ± 1.69	98.29 ± 1.34	**99.06 ± 1.06**	**99.56 ± 0.72**
pv, ws	97.82 ± 1.56	97.69 ± 1.57	98.65 ± 1.14	98.21 ± 1.50	98.81 ± 1.14
pa, pv, ws	97.59 ± 1.66	97.02 ± 1.88	**98.79 ± 1.23**	97.75 ± 1.72	97.56 ± 1.50
pa, pv, wq	97.28 ± 1.70	96.44 ± 2.08	97.17 ± 1.89	98.53 ± 1.49	97.90 ± 1.50
pa, wq, ws	96.39 ± 2.11	93.94 ± 2.49	97.31 ± 1.50	96.88 ± 1.85	96.79 ± 1.74
pv, wq, ws	95.61 ± 2.36	95.74 ± 2.01	98.05 ± 1.55	97.35 ± 1.68	98.15 ± 1.54
pa, pv, wq, ws	95.75 ± 2.30	93.54 ± 2.83	97.15 ± 1.74	95.69 ± 2.22	95.81 ± 1.89

Source: Research data.

Table 20 – Classification metrics for the best result of classification in the *Noise-0* dataset in Table 19

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
BA	1.000	1.000	0.000	1.000	1.000	1.000	100
BANL15	1.000	0.990	0.000	0.995	0.994	1.000	100
BANL2	0.990	1.000	0.001	0.995	0.994	1.000	100
BANL5	0.990	1.000	0.001	0.995	0.994	1.000	100
ER	0.952	1.000	0.007	0.976	0.972	0.999	100
GEO	1.000	0.950	0.000	0.974	0.971	0.999	100
MEN	1.000	0.990	0.000	0.995	0.994	1.000	100
WS	1.000	1.000	0.000	1.000	1.000	1.000	100

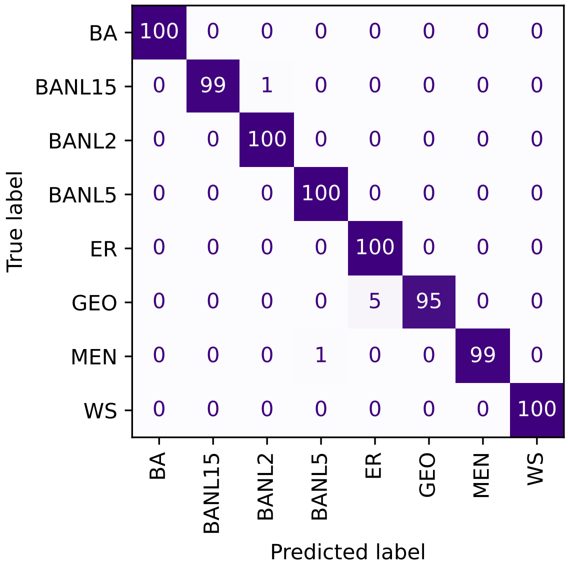
Source: Research data.

5.4.6 Noise-20% dataset

Table 23 presents the classification results in the *4-models* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset, the extracted features enable the categorization of the 8 network classes, achieving an accuracy score of 97.89%. This result indicates that out of the 800 networks, only 17 are misclassified.

Table 24 presents the classification metrics for the best result obtained in the *Noise 20%*

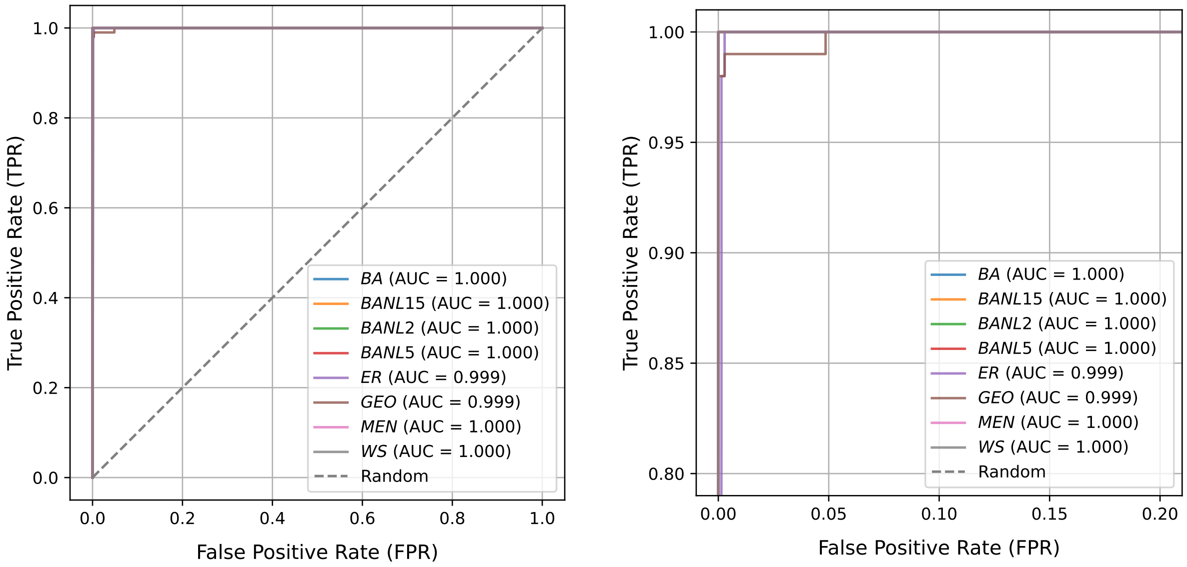
Figure 38 – Confusion matrix obtained for the best results of classification in the *Noise-0* dataset in Table 19



True label \ Predicted label	BA	BANL15	BANL2	BANL5	ER	GEO	MEN	WS
BA	100	0	0	0	0	0	0	0
BANL15	0	99	1	0	0	0	0	0
BANL2	0	0	100	0	0	0	0	0
BANL5	0	0	0	100	0	0	0	0
ER	0	0	0	0	100	0	0	0
GEO	0	0	0	0	5	95	0	0
MEN	0	0	0	1	0	0	99	0
WS	0	0	0	0	0	0	0	100

Source: Elaborated by the author.

Figure 39 – AUC-ROC obtained for the best results of classification in the *Noise-0* dataset in Table 19



(a) AUC ROC

(b) Zoom of AUC ROC

Source: Elaborated by the author.

dataset. Comparing the values obtained in the previous tables of the datasets with less noise, we can see that there is an influence on the results, which translates into slightly lower values for each metric.

Figure 42 presents the confusion matrix for the best result obtained in the *Noise 20%* dataset. In this case the *GEO* class has 10 networks that were misclassified, and confused as the

Table 21 – Classification results using the LDA classifier for the *Noise-10%* dataset. The highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=6*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=100*	*hg=100*
pa	97.34 ± 1.61	96.84 ± 1.81	95.86 ± 2.12	94.34 ± 2.44	94.65 ± 2.14
pv	95.11 ± 2.48	94.97 ± 2.22	94.61 ± 2.38	96.74 ± 1.99	97.59 ± 1.73
wq	93.27 ± 2.72	91.21 ± 3.09	92.77 ± 2.76	95.53 ± 2.03	95.38 ± 2.08
ws	94.05 ± 2.64	91.78 ± 2.46	92.65 ± 2.96	92.60 ± 2.75	93.17 ± 2.27
pa, pv	96.88 ± 1.82	**97.17 ± 1.72**	95.70 ± 2.22	95.94 ± 2.17	97.36 ± 1.69
wq, ws	96.16 ± 2.01	95.27 ± 2.07	97.67 ± 1.66	97.36 ± 1.74	97.44 ± 1.58
pa, wq	96.58 ± 1.89	95.39 ± 2.34	94.99 ± 2.03	93.88 ± 2.45	95.05 ± 2.21
pa, ws	**97.99 ± 1.55**	96.89 ± 1.94	**98.05 ± 1.45**	95.56 ± 2.18	97.24 ± 1.54
pv, wq	95.28 ± 2.13	94.49 ± 2.13	95.31 ± 2.31	**98.05 ± 1.49**	**98.26 ± 1.31**
pv, ws	96.32 ± 2.10	95.93 ± 2.07	97.16 ± 1.69	97.67 ± 1.60	97.92 ± 1.35
pa, pv, ws	96.89 ± 1.82	96.36 ± 1.87	97.30 ± 1.72	96.44 ± 2.08	97.31 ± 1.54
pa, pv, wq	95.91 ± 2.09	94.86 ± 2.33	94.85 ± 2.24	93.82 ± 2.43	96.29 ± 1.91
pa, wq, ws	96.09 ± 2.04	95.54 ± 2.37	97.41 ± 1.54	94.46 ± 2.52	96.02 ± 2.03
pv, wq, ws	95.35 ± 2.17	94.91 ± 2.27	97.36 ± 1.71	96.42 ± 2.12	97.76 ± 1.46
pa, pv, wq, ws	95.97 ± 2.19	95.38 ± 2.39	97.20 ± 1.79	94.81 ± 2.15	95.92 ± 1.90

Source: Research data.

Table 22 – Classification metrics for the best result of classification in the *Noise 10%* dataset in Table 21

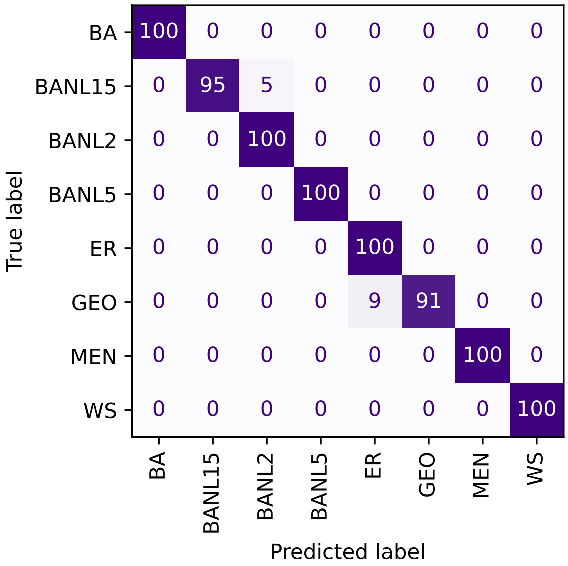
Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
BA	1.000	1.000	0.000	1.000	1.000	1.000	100
BANL15	1.000	0.950	0.000	0.974	0.971	1.000	100
BANL2	0.952	1.000	0.007	0.976	0.972	0.999	100
BANL5	1.000	1.000	0.000	1.000	1.000	1.000	100
ER	0.917	1.000	0.013	0.957	0.952	1.000	100
GEO	1.000	0.910	0.000	0.953	0.948	1.000	100
MEN	1.000	1.000	0.000	1.000	1.000	1.000	100
WS	1.000	1.000	0.000	1.000	1.000	1.000	100

Source: Research data.

ER class.

Figure 43a shows the area under the curve for each class for the best result obtained in the *Noise 20%* dataset. It can be appreciated that adding noise to the dataset influences the classification values. In this case, the curves begin to move away from their maximum value of 1.

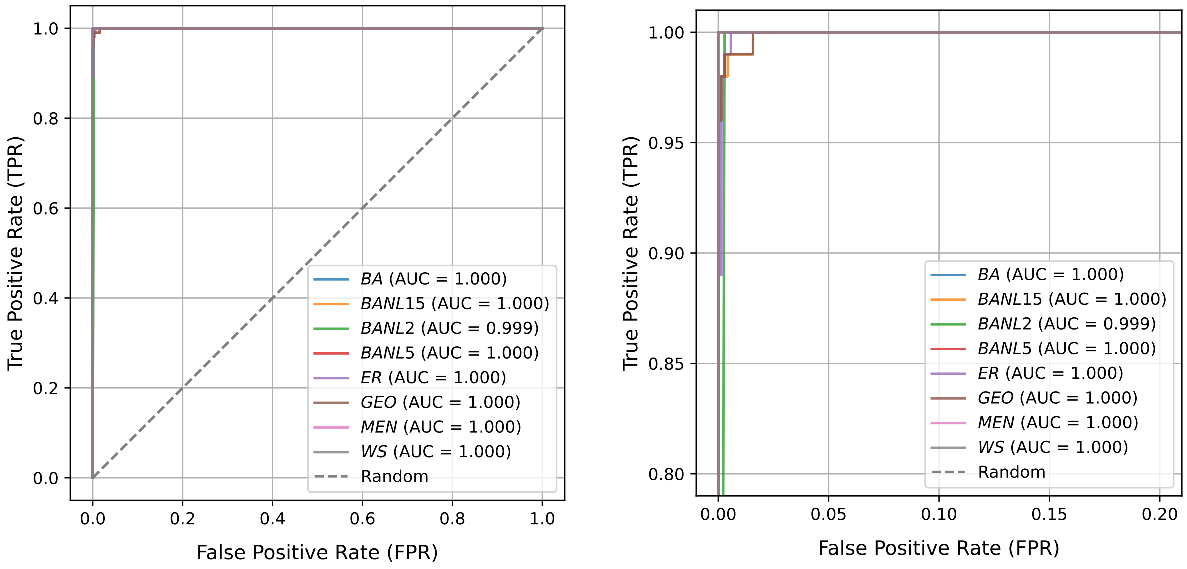
Figure 40 – Confusion matrix obtained for the best results of classification in the *Noise 10%* dataset in Table 21



	BA	BANL15	BANL2	BANL5	ER	GEO	MEN	WS
BA	100	0	0	0	0	0	0	0
BANL15	0	95	5	0	0	0	0	0
BANL2	0	0	100	0	0	0	0	0
BANL5	0	0	0	100	0	0	0	0
ER	0	0	0	0	100	0	0	0
GEO	0	0	0	0	9	91	0	0
MEN	0	0	0	0	0	0	100	0
WS	0	0	0	0	0	0	0	100

Source: Elaborated by the author.

Figure 41 – AUC-ROC obtained for the best results of classification in the *Noise 10%* dataset in Table 21



(a) AUC ROC

(b) Zoom of AUC ROC

Source: Elaborated by the author.

5.4.7 *Noise-30% dataset*

Table 25 presents the classification results in the *Noise 30%* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset, the extracted features enable the categorization of the 8 network classes, achieving an accuracy score of 96.84%. This result

Table 23 – Classification results using the LDA classifier for the *Noise 20%* dataset. The highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=6*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=100*	*hg=100*
pa	96.96 ± 1.76	95.88 ± 2.04	94.54 ± 2.35	92.02 ± 2.92	93.26 ± 2.45
pv	93.07 ± 2.26	94.96 ± 2.15	94.49 ± 2.34	96.06 ± 2.07	96.84 ± 1.87
wq	91.92 ± 2.65	91.72 ± 2.58	92.81 ± 2.58	93.17 ± 2.38	94.58 ± 2.24
ws	93.91 ± 2.48	92.59 ± 2.72	93.18 ± 2.45	90.38 ± 3.07	91.61 ± 2.20
pa, pv	96.84 ± 1.82	**96.11 ± 1.92**	95.00 ± 2.08	95.06 ± 2.21	95.29 ± 2.28
wq, ws	94.64 ± 2.32	93.92 ± 2.47	96.71 ± 1.91	96.16 ± 2.01	96.64 ± 1.77
pa, wq	96.93 ± 1.75	94.32 ± 2.30	94.75 ± 2.19	91.70 ± 2.95	94.86 ± 2.40
pa, ws	**97.43 ± 1.68**	95.60 ± 2.31	97.10 ± 1.55	94.41 ± 2.38	95.20 ± 2.40
pv, wq	93.59 ± 2.30	93.69 ± 2.26	94.76 ± 2.21	96.11 ± 2.08	**97.89 ± 1.44**
pv, ws	95.24 ± 2.17	94.71 ± 2.47	**97.25 ± 1.62**	**97.13 ± 1.78**	97.60 ± 1.58
pa, pv, ws	96.46 ± 1.87	95.12 ± 2.47	97.01 ± 1.64	95.29 ± 2.20	95.94 ± 2.11
pa, pv, wq	95.61 ± 2.07	94.69 ± 2.19	94.84 ± 1.91	92.99 ± 2.46	95.45 ± 1.93
pa, wq, ws	96.06 ± 1.94	95.65 ± 2.16	96.24 ± 1.91	93.37 ± 2.58	94.94 ± 2.75
pv, wq, ws	93.32 ± 2.57	93.99 ± 2.46	96.44 ± 2.22	95.14 ± 2.28	97.56 ± 1.56
pa, pv, wq, ws	95.06 ± 2.25	95.23 ± 2.12	96.79 ± 1.81	92.41 ± 2.62	95.51 ± 2.15

Source: Elaborated by the author.

Table 24 – Classification metrics for the best result of classification in the *Noise 20%* dataset in Table 23

Class	precision	recall	FPR	f1-score	MCC	AUC	support
BA	1.000	1.000	0.000	1.000	1.000	1.000	100
BANL15	0.990	1.000	0.001	0.995	0.994	0.999	100
BANL5	0.990	0.990	0.001	0.990	0.989	1.000	100
BAnl2	1.000	0.990	0.000	0.995	0.994	1.000	100
ER	0.908	0.990	0.014	0.947	0.941	0.998	100
GEO	0.989	0.900	0.001	0.942	0.936	0.998	100
MEN	0.990	0.990	0.001	0.990	0.989	1.000	100
WS	1.000	1.000	0.000	1.000	1.000	1.000	100

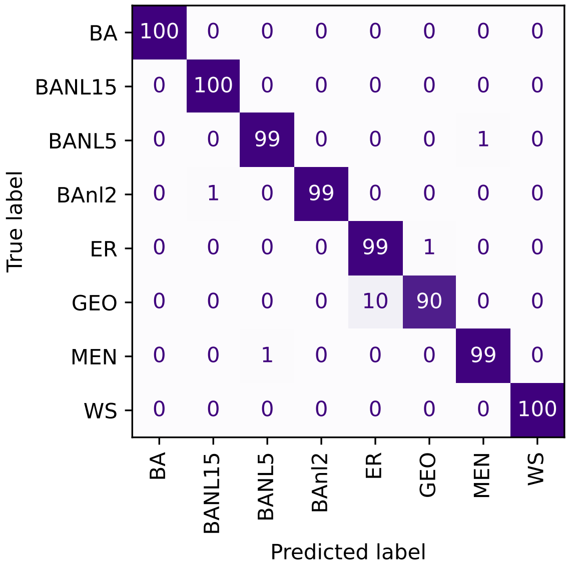
Source: Elaborated by the author.

indicates that out of the 800 networks, only 25 are misclassified.

Table 26 presents the classification metrics for the best result obtained in the *Noise 30%* dataset. The values of MCC ranges from 0.876 to 1, this means that in general the classifier has a good level of overall performance.

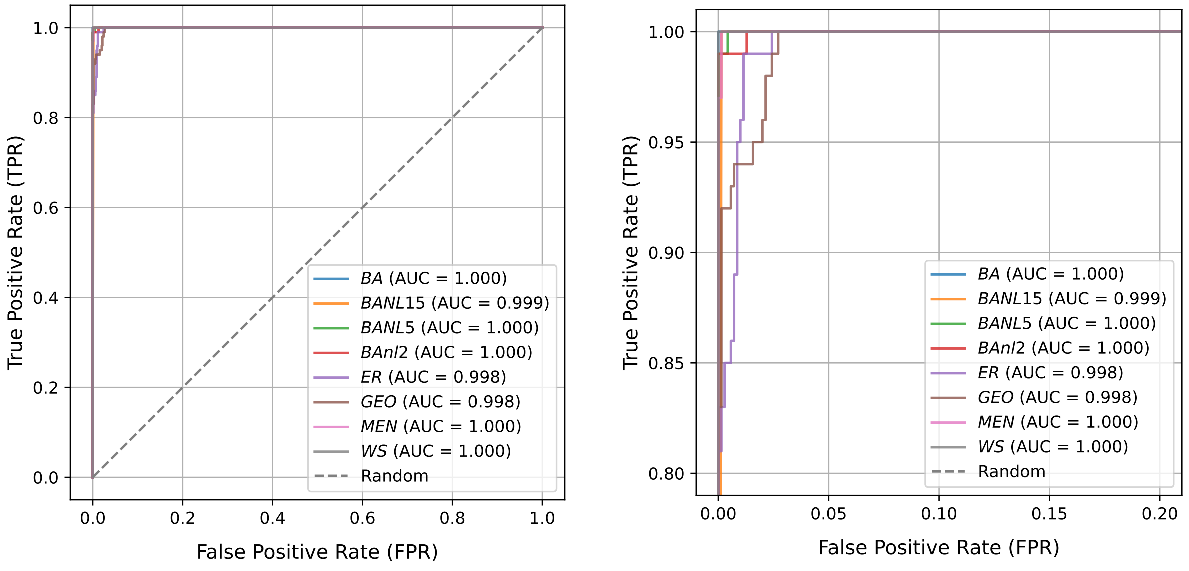
Figure 44 presents the confusion matrix for the best result obtained in the *Noise 30%* dataset. In this case the *GEO* and *ER* classes have the most instances misclassified with 15 and 6 respectively.

Figure 42 – Confusion matrix obtained for the best results of classification in the *Noise 20%* dataset in Table 23



Source: Elaborated by the author.

Figure 43 – AUC-ROC obtained for the best results of classification in the *Noise 20%* dataset in Table 23



(a) AUC ROC

(b) Zoom of AUC ROC

Source: Elaborated by the author.

Figure 45a shows the area under the curve for each class for the best result obtained in the *Noise 30%* dataset. As stated previously, the addition of noise has influence in the model, the plot shows 2 classes that in which their AUC-ROC curves are moving away from the maximum value.

Table 25 – Classification results using the LDA classifier for the *Noise 30%* dataset. The highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=6*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=100*	*hg=100*
pa	94.20 ± 2.35	93.94 ± 2.44	92.84 ± 2.58	92.24 ± 2.91	92.88 ± 2.47
pv	91.88 ± 2.68	93.36 ± 2.51	92.57 ± 2.63	94.69 ± 2.47	96.58 ± 1.93
wq	88.68 ± 2.93	90.31 ± 3.37	90.15 ± 2.80	93.31 ± 2.52	94.71 ± 2.19
ws	91.37 ± 3.11	91.64 ± 2.89	90.62 ± 3.40	89.15 ± 3.13	90.69 ± 2.68
pa, pv	**96.54 ± 1.82**	**95.36 ± 2.36**	95.19 ± 2.08	94.99 ± 2.32	95.59 ± 2.18
wq, ws	90.92 ± 3.10	91.81 ± 2.95	93.68 ± 2.44	**95.25 ± 2.24**	95.41 ± 2.29
pa, wq	94.84 ± 2.41	93.69 ± 2.57	94.39 ± 2.50	91.92 ± 2.96	94.39 ± 2.31
pa, ws	94.62 ± 2.38	93.97 ± 2.56	**95.95 ± 2.08**	93.10 ± 2.46	93.90 ± 2.50
pv, wq	93.58 ± 2.57	94.29 ± 2.56	93.42 ± 2.10	95.08 ± 2.32	**96.84 ± 1.87**
pv, ws	94.10 ± 2.41	94.53 ± 2.54	95.23 ± 2.03	94.96 ± 2.28	96.45 ± 1.99
pa, pv, ws	95.88 ± 2.14	94.74 ± 2.51	95.91 ± 2.07	94.72 ± 2.26	94.01 ± 2.62
pa, pv, wq	95.76 ± 2.21	94.11 ± 2.57	94.29 ± 2.43	93.66 ± 2.75	95.30 ± 2.22
pa, wq, ws	93.26 ± 2.81	92.56 ± 2.48	94.56 ± 2.64	92.74 ± 2.72	93.41 ± 2.74
pv, wq, ws	92.12 ± 2.95	93.75 ± 2.49	95.08 ± 2.11	94.47 ± 2.60	96.25 ± 1.90
pa, pv, wq, ws	94.56 ± 2.40	93.26 ± 2.51	94.89 ± 2.62	93.52 ± 2.73	92.90 ± 2.57

Source: Research data.

Table 26 – Classification metrics for the best result of classification in the *Noise 30%* dataset in Table 25

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
BA	1.000	1.000	0.000	1.000	1.000	1.000	100
BANL15	0.990	1.000	0.001	0.995	0.994	0.999	100
BANL2	1.000	0.990	0.000	0.995	0.994	1.000	100
BANL5	0.990	0.990	0.001	0.990	0.989	1.000	100
ER	0.862	0.940	0.021	0.900	0.886	0.995	100
GEO	0.934	0.850	0.009	0.890	0.876	0.995	100
MEN	0.990	0.990	0.001	0.990	0.989	1.000	100
WS	1.000	1.000	0.000	1.000	1.000	1.000	100

Source: Research data.

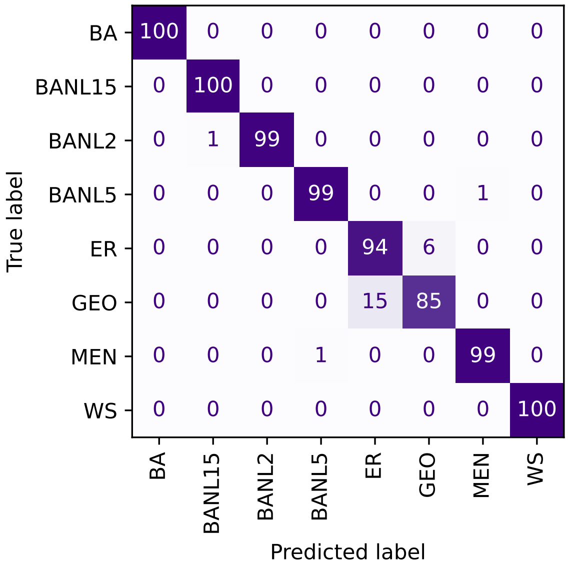
5.5 Pattern recognition in real-world datasets

According to the exploration of the model by varying the values of the parameters, we perform the pattern recognition task in the real world datasets.

For each dataset we performed a series of experiments varying the parameters wg, and hg fixing the rest of the parameters pw, pp, tp, iw, and it as discussed previously in subsection 5.3.2 in order to search for good classification results. For each dataset we present 5 variations of the parameters.

The following values of parameters were chosen because provided the best accuracy

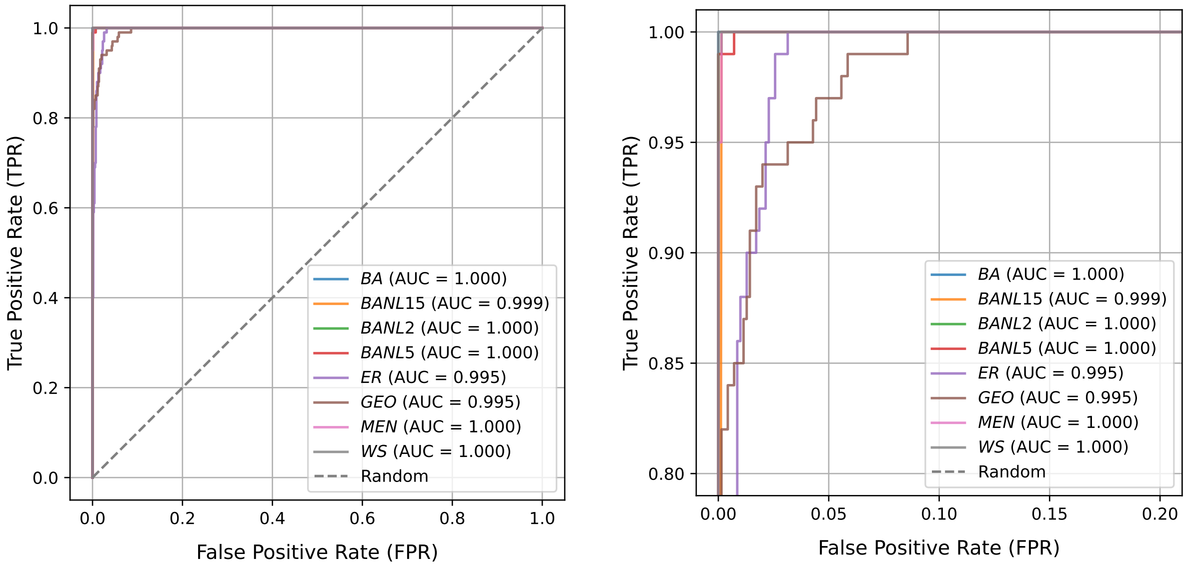
Figure 44 – Confusion matrix obtained for the best results of classification in the *Noise 30%* dataset in Table 23



True label \ Predicted label	BA	BANL15	BANL2	BANL5	ER	GEO	MEN	WS
BA	100	0	0	0	0	0	0	0
BANL15	0	100	0	0	0	0	0	0
BANL2	0	1	99	0	0	0	0	0
BANL5	0	0	0	99	0	0	1	0
ER	0	0	0	0	94	6	0	0
GEO	0	0	0	0	15	85	0	0
MEN	0	0	0	1	0	0	99	0
WS	0	0	0	0	0	0	0	100

Source: Elaborated by the author.

Figure 45 – AUC-ROC obtained for the best results of classification in the *Noise 30%* dataset in Table 25



(a) AUC ROC

(b) Zoom of AUC ROC

Source: Elaborated by the author.

values with the LDA classifier using the 4 four features $\{pa, pv, wq, ws\}$.

The water to grow wg varies between to 2 and 4, these were values that allow the model to perform well, so it will permit the model to search a better accuracy. The height to grow hg is set to 40, 60, and 100 to allow a plant to have a bigger scope to explore its neighborhood. The percent of water pw is set to 60. The time alive water tw is set to 30. The percent of plant *pp* is set to 30. The values of percent water and percent plant allow the entire environment to participate in the simulation, so each node in the network has an agent at a given moment. The time alive plant tp is set to 0, the special value, so the plant agents remain in the network during the entire simulation. The number of iterations to add water *iw* is set to 10, this means that water is added 20 times during the simulation. The number of iterations of the simulation *it* is fixed at 200.

The parameters use for the pattern recognition experiments in the real-world datasets are in Table 27. Each column shows a set of values used, with these parameters we have 5 different executions of the model in each dataset.

Table 27 – Parameters selected to execute the Growth Model in the real-world network datasets

	Set of values				
Parameter	**First**	**Second**	**Third**	**Fourth**	**Fifth**
wg	2	2	2	4	4
hg	40	60	100	40	60
pw	60	60	60	60	60
tw	30	30	30	30	30
pp	30	30	30	30	30
tp	0	0	0	0	0
iw	10	10	10	10	10
it	200	200	200	200	200

Source: Research data.

5.5.1 Actinobacteria dataset

Table 28 presents the classification results in the *Actinobacteria* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. The best accuracy value is 99.34% using the 4 features with $wg = 4$, and $hg = 60$. This result indicates that out of the 199 networks, only 2 instances were misclassified.

Table 29 presents the classification metrics for the best result obtained in the *Actinobacteria* dataset. The recall and MCC values for the *Mycobacterium* class indicate that it is the most difficult to be classified correctly.

Figure 46 presents the confusion matrix, it is observed that the *Corynebacterium* class, all the samples are classified as TP, in the *Mycobacterium* class of 60 samples, 58 are correctly classified, and the *Streptomyces* class 53 samples are correctly correctly.

Figure 47 shows the area under the curve for each class. Two of the classes have values close to 1, it indicates that the model is effective in distinguishing those classes, with a low false positive rate and a high true positive rate.

Table 28 – Classification results using the LDA classifier for the *Actinobacteria* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA **Features**	*wg=2* *hg=40*	*wg=2* *hg=60*	*wg=2* *hg=100*	*wg=4* *hg=40*	*wg=4* *hg=60*
pa	93.91 ± 4.93	96.18 ± 4.02	96.78 ± 3.70	75.10 ± 9.51	83.12 ± 8.31
pv	88.67 ± 6.90	90.10 ± 6.61	86.25 ± 7.25	88.36 ± 7.21	87.62 ± 7.19
wq	79.69 ± 8.96	67.99 ± 11.36	82.42 ± 8.36	92.46 ± 5.79	90.82 ± 6.62
ws	84.57 ± 8.00	89.85 ± 6.43	90.03 ± 6.28	94.22 ± 5.01	92.36 ± 5.79
pa, pv	81.40 ± 8.33	88.32 ± 7.64	94.22 ± 4.95	97.19 ± 3.75	95.85 ± 4.46
wq, ws	95.12 ± 4.93	98.29 ± 3.01	94.96 ± 4.71	97.29 ± 3.68	98.59 ± 2.95
pa, wq	88.93 ± 6.84	93.42 ± 5.20	94.17 ± 4.80	97.38 ± 3.77	96.73 ± 3.88
pa, ws	93.29 ± 5.94	96.01 ± 4.24	93.71 ± 5.55	97.97 ± 2.97	98.09 ± 3.40
pv, wq	87.80 ± 7.27	92.11 ± 5.90	89.79 ± 6.33	96.11 ± 4.56	96.39 ± 4.15
pv, ws	96.13 ± 4.44	97.58 ± 3.59	95.31 ± 4.73	97.11 ± 3.50	97.59 ± 3.33
pa, pv, ws	**97.19 ± 3.87**	98.01 ± 3.09	96.57 ± 3.96	**98.54 ± 2.73**	98.57 ± 2.78
pa, pv, wq	94.67 ± 4.64	96.26 ± 4.08	95.46 ± 4.36	97.81 ± 3.09	98.13 ± 3.01
pa, wq, ws	96.81 ± 4.28	98.67 ± 2.73	95.91 ± 4.20	98.06 ± 3.22	99.05 ± 2.47
pv, wq, ws	97.06 ± 3.90	98.77 ± 2.73	95.48 ± 4.46	97.63 ± 3.64	99.07 ± 2.36
pa, pv, wq, ws	97.11 ± 3.82	**98.97 ± 2.32**	**96.81 ± 3.80**	98.49 ± 2.65	**99.34 ± 2.28**

Source: Research data.

Table 29 – Classification metrics for the best result of classification in the *Actinobacteria* dataset in Table 28

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
Corynebacterium	0.977	1.000	0.018	0.989	0.980	0.995	86
Mycobacterium	1.000	0.967	0.000	0.983	0.976	0.997	60
Streptomyces	1.000	1.000	0.000	1.000	1.000	1.000	53

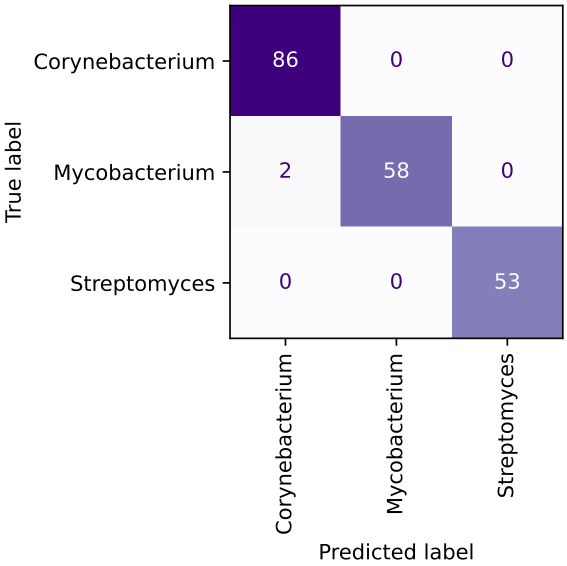
Source: Research data.

5.5.2 Animals dataset

Table 30 presents the classification results in the *Animals* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. The best accuracy value is 98.50% using 3 features with $wg = 2$, and $hg = 40$. This result indicates that out of the 56 networks, only 1 instance was misclassified.

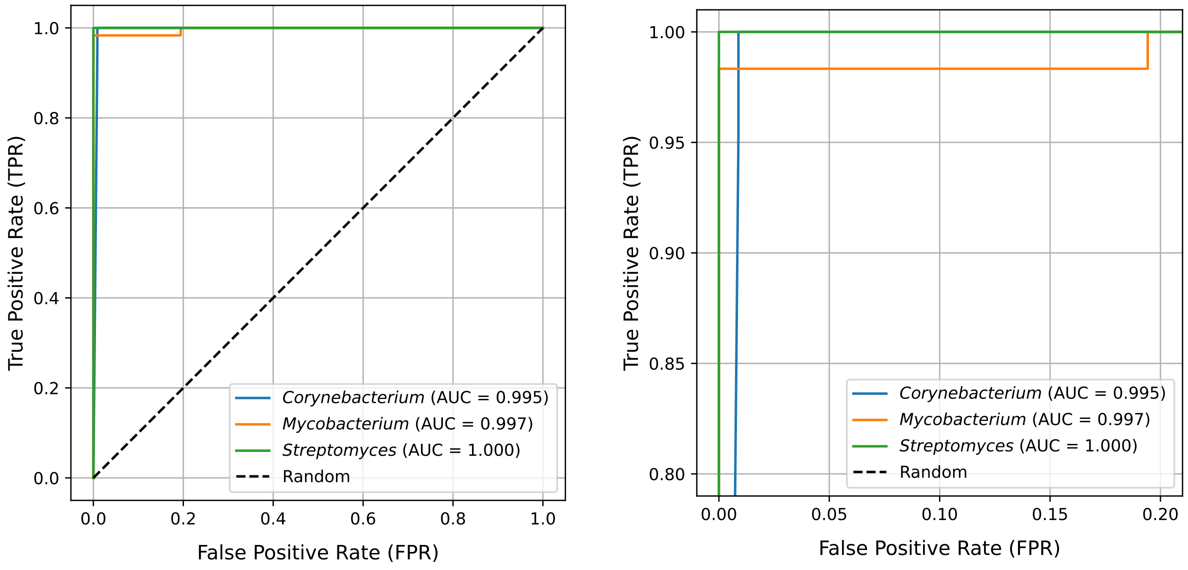
Table 31 presents the classification metrics for the best result obtained in the *Animals* dataset. The values for the recall metric from 0.929 to 1 for this dataset means that the classifier is able to identify the positive instances across all classes. The values for the f1-score metric from 0.963 to 1 means that the classifier is achieving a high balance between precision and recall

Figure 46 – Confusion matrix obtained for the best results of classification in the *Actinobacteria* dataset in Table 28



Source: Elaborated by the author.

Figure 47 – AUC-ROC obtained for the best results of classification in the *Actinobacteria* dataset in Table 28



(a) AUC ROC

(b) Zoom of AUC ROC

Source: Elaborated by the author.

across all classes.

Figure 48 presents the confusion matrix for the best result obtained in the *Animals* dataset. The confusion matrix shows that the instance misclassified correspond to the *fishes* class.

Figure 49 shows the area under the curve for each class for the best result obtained in the

Table 30 – Classification results using the LDA classifier for the *Animals* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=4*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=40*	*hg=60*
pa	76.32 ± 17.35	90.97 ± 11.66	94.45 ± 9.41	90.38 ± 11.68	88.33 ± 13.21
pv	81.82 ± 13.03	94.78 ± 9.30	96.33 ± 7.53	88.88 ± 12.95	86.62 ± 12.53
wq	96.62 ± 7.85	93.40 ± 9.99	91.55 ± 10.24	87.27 ± 12.48	87.25 ± 12.58
ws	95.17 ± 8.33	89.33 ± 12.40	90.08 ± 12.37	88.50 ± 13.09	92.30 ± 11.05
pa, pv	94.53 ± 9.17	94.45 ± 8.89	**97.43 ± 6.39**	89.78 ± 11.80	92.32 ± 10.59
wq, ws	97.68 ± 6.25	95.88 ± 8.42	91.18 ± 10.59	90.75 ± 12.14	92.75 ± 10.60
pa, wq	95.62 ± 8.42	93.43 ± 9.40	93.38 ± 9.67	90.15 ± 11.04	88.82 ± 12.05
pa, ws	93.52 ± 10.10	92.02 ± 10.84	92.67 ± 10.15	91.93 ± 11.16	92.02 ± 9.79
pv, wq	98.03 ± 5.86	96.00 ± 8.65	91.53 ± 10.77	87.47 ± 12.63	88.25 ± 11.72
pv, ws	97.57 ± 6.80	94.47 ± 9.89	92.12 ± 10.31	90.88 ± 11.13	**96.33 ± 7.45**
pa, pv, ws	**98.50 ± 4.94**	92.62 ± 10.35	94.48 ± 9.07	**92.17 ± 10.86**	92.72 ± 9.62
pa, pv, wq	97.77 ± 6.17	93.90 ± 9.32	92.88 ± 10.15	89.87 ± 11.51	88.30 ± 12.00
pa, wq, ws	97.50 ± 6.44	95.15 ± 8.46	91.30 ± 10.51	90.88 ± 11.71	91.07 ± 11.05
pv, wq, ws	98.13 ± 5.96	**96.47 ± 8.10**	91.85 ± 11.02	90.13 ± 11.94	91.05 ± 10.55
pa, pv, wq, ws	97.50 ± 6.66	94.72 ± 8.48	92.52 ± 9.82	90.55 ± 11.80	90.45 ± 10.98

Source: Research data.

Table 31 – Classification metrics for the best result of classification in the *Animals* dataset in Table 30

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
birds	1.000	1.000	0.000	1.000	1.000	1.000	14
fishes	1.000	0.929	0.000	0.963	0.952	0.998	14
insects	1.000	1.000	0.000	1.000	1.000	1.000	14
mammals	0.933	1.000	0.024	0.966	0.955	1.000	14

Source: Research data.

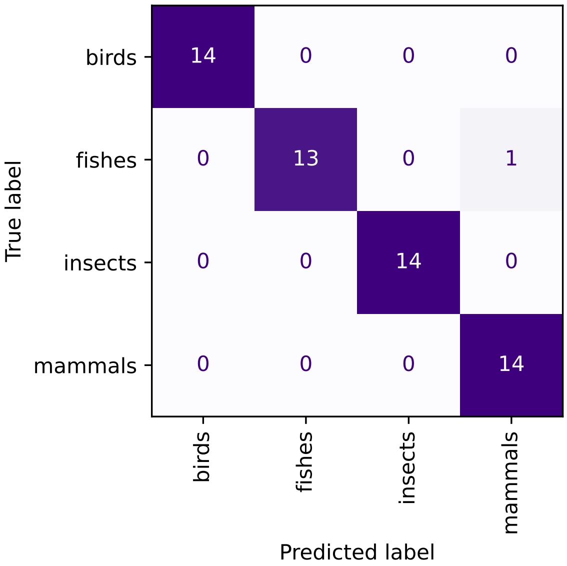
Animals dataset. The AUC-ROC curve shows how the *fishes* is moving away from the maximum value.

5.5.3 *Firmicutes-Bacillis dataset*

Table 32 presents the classification results in the *Firmicutes-Bacillis* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset, all the best accuracy values were using the 4 features, and the best accuracy of 95.84% happens with $wg = 4$ and $hg = 40$. From the 414 networks 17 instances were incorrectly classified.

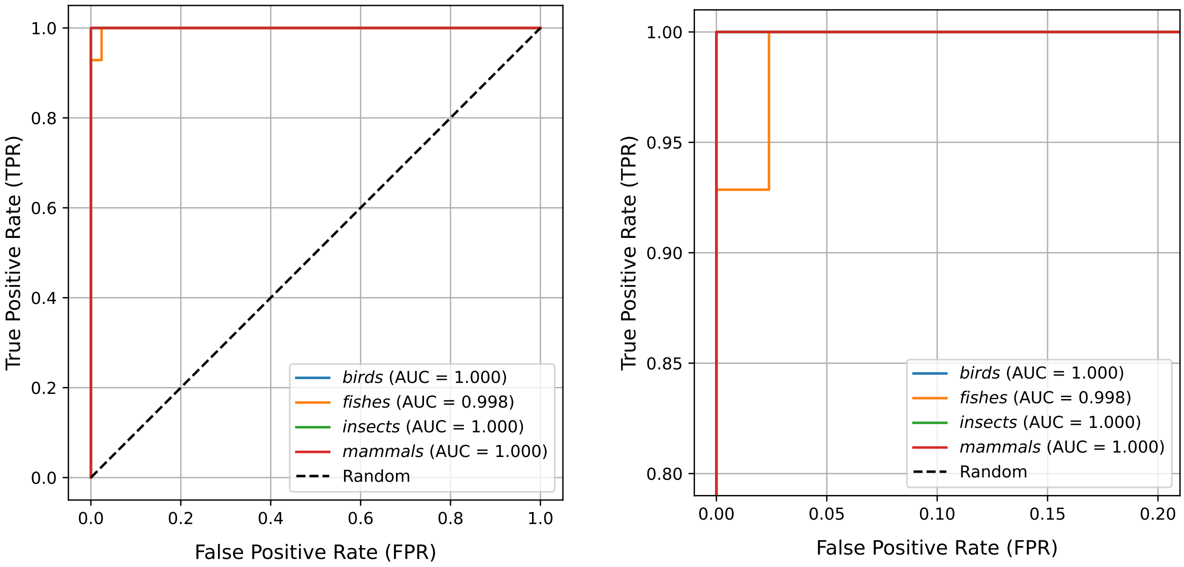
Table 33 presents the classification metrics for the best result obtained in the *Firmicutes-*

Figure 48 – Confusion matrix obtained for the best results of classification in the *Animals* dataset in Table 30



Source: Elaborated by the author.

Figure 49 – AUC-ROC obtained for the best results of classification in the *Animals* dataset in Table 30



(a) AUC ROC

(b) Zoom of AUC ROC

Source: Elaborated by the author.

Bacillis dataset. The precision value of 0.831 in the *Lactobacillus* class indicates that the classifier is not able to identify correctly the positive instances.

Figure 50 presents the confusion matrix for the best result obtained in the *Firmicutes-Bacillis* dataset. The confusion matrix also shows that there are 15 networks of class *Lactobacillus* classified as *Streptococcus*.

Table 32 – Classification results using the LDA classifier for the *Firmicutes-Bacillis* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=4*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=40*	*hg=60*
pa	91.37 ± 4.15	91.42 ± 4.44	91.43 ± 4.30	89.07 ± 4.71	88.94 ± 4.89
pv	90.67 ± 4.51	88.53 ± 4.70	87.53 ± 5.42	85.60 ± 5.97	86.63 ± 5.36
wq	90.01 ± 4.42	87.68 ± 4.99	85.74 ± 5.63	85.59 ± 5.48	85.90 ± 5.13
ws	91.80 ± 4.42	90.03 ± 4.64	87.71 ± 5.35	90.05 ± 4.78	90.56 ± 4.45
pa, pv	90.92 ± 4.20	88.61 ± 4.96	88.38 ± 4.89	79.02 ± 6.29	77.08 ± 6.80
wq, ws	78.72 ± 6.85	78.99 ± 6.68	86.95 ± 5.07	91.12 ± 3.86	91.76 ± 4.60
pa, wq	89.40 ± 4.66	81.17 ± 6.58	79.59 ± 5.99	79.99 ± 6.05	75.81 ± 6.57
pa, ws	89.39 ± 5.51	84.19 ± 5.52	82.46 ± 5.62	85.65 ± 6.30	84.66 ± 5.96
pv, wq	87.89 ± 5.11	82.25 ± 6.59	73.79 ± 7.71	71.26 ± 6.91	67.88 ± 7.35
pv, ws	88.05 ± 4.76	84.43 ± 5.76	83.20 ± 5.79	87.94 ± 5.55	85.19 ± 5.73
pa, pv, ws	88.02 ± 5.01	87.69 ± 5.27	89.42 ± 5.03	94.74 ± 3.47	92.74 ± 4.03
pa, pv, wq	81.99 ± 5.60	78.26 ± 7.00	82.49 ± 5.99	89.14 ± 4.75	85.37 ± 5.63
pa, wq, ws	88.61 ± 5.11	88.94 ± 4.99	93.06 ± 4.09	94.53 ± 3.69	93.37 ± 3.75
pv, wq, ws	91.37 ± 4.67	89.47 ± 4.53	92.31 ± 3.78	94.34 ± 3.46	92.98 ± 3.67
pa, pv, wq, ws	**93.60 ± 3.53**	**93.57 ± 4.16**	**94.73 ± 3.69**	**95.84 ± 3.08**	**94.54 ± 3.44**

Source: Research data.

Table 33 – Classification metrics for the best result of classification in the *Firmicutes-Bacillis* dataset in Table 32

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
Bacillus	0.992	0.959	0.003	0.975	0.965	1.000	122
Lactobacillus	0.831	0.892	0.045	0.860	0.825	0.990	83
Staphylococcus	0.937	0.974	0.015	0.955	0.945	0.993	76
Streptococcus	0.922	0.887	0.036	0.904	0.860	0.992	133

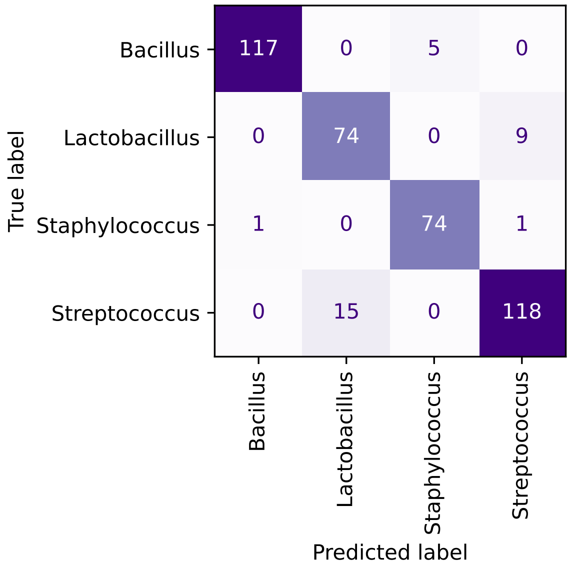
Source: Research data.

Figure 51a shows the area under the curve for each class for the best result obtained in the *Firmicutes-Bacillis* dataset. In this case, only the *Bacillus* class has the best AUC-ROC value, the other classes have lower values indicated by the curves moving away from the maximum value.

5.5.4 *Fungi dataset*

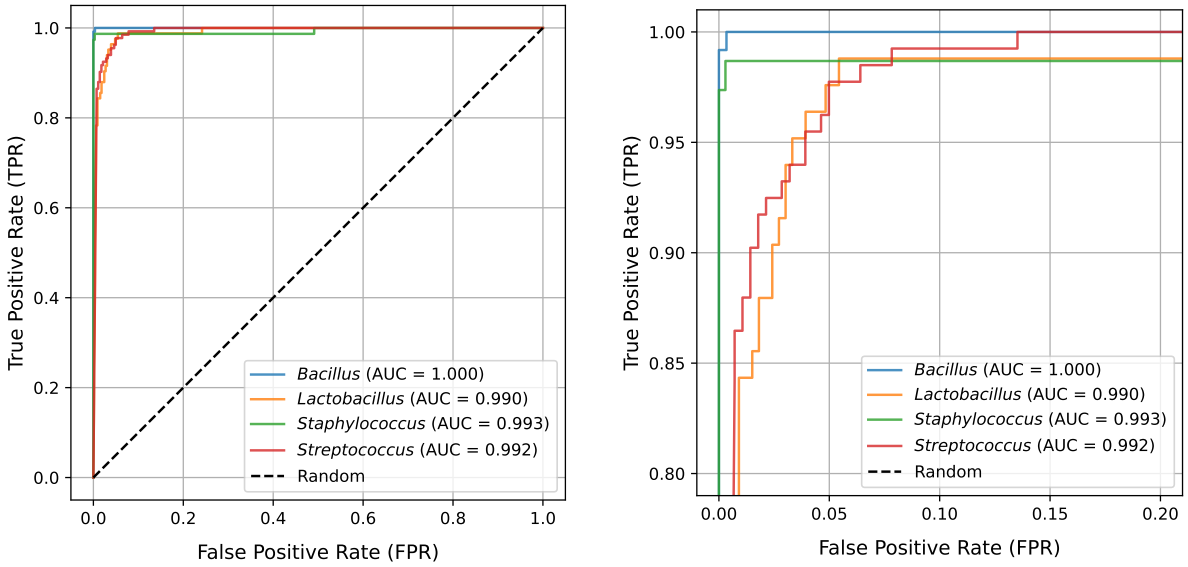
Table 34 presents the classification results in the *Fungi* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset the classifier achieved an accuracy of

Figure 50 – Confusion matrix obtained for the best results of classification in the *Firmicutes-Bacillis* dataset in Table 32



Source: Elaborated by the author.

Figure 51 – AUC-ROC obtained for the best results of classification in the *Firmicutes-Bacillis* dataset in Table 32



(a) AUC ROC

(b) Zoom of AUC ROC

Source: Elaborated by the author.

83.17%, what indicates that out of 60 networks, 10 networks were misclassified.

Table 35 presents the classification metrics for the best result obtained in the *Fungi* dataset. The lower values of the metrics indicate that the classifier is not able to correctly identify the instances in each class. Only the *saccharomycetes* class has the best values across all metrics.

Table 34 – Classification results using the LDA classifier for the *Fungi* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=4*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=40*	*hg=60*
pa	65.42 ± 17.40	67.58 ± 18.27	59.25 ± 19.27	58.50 ± 20.61	59.33 ± 18.98
pv	59.42 ± 20.25	54.25 ± 17.87	59.67 ± 18.44	54.33 ± 20.29	71.17 ± 17.05
wq	64.67 ± 19.55	49.25 ± 19.81	65.33 ± 17.82	58.92 ± 18.99	58.75 ± 18.02
ws	65.00 ± 19.29	67.67 ± 18.15	63.58 ± 18.86	63.42 ± 19.91	69.83 ± 17.19
pa, pv	73.50 ± 16.35	80.25 ± 15.10	70.25 ± 17.63	60.67 ± 17.08	70.17 ± 17.84
wq, ws	**83.08 ± 13.79**	77.00 ± 15.51	75.92 ± 17.22	55.83 ± 18.84	62.08 ± 18.25
pa, wq	72.83 ± 17.50	61.08 ± 18.05	73.83 ± 17.20	60.17 ± 18.47	62.92 ± 17.59
pa, ws	80.92 ± 16.28	75.83 ± 17.54	77.42 ± 17.07	**68.17 ± 17.57**	68.92 ± 18.23
pv, wq	72.42 ± 16.80	60.17 ± 19.42	71.08 ± 16.95	57.75 ± 18.17	59.75 ± 17.74
pv, ws	75.67 ± 17.47	76.25 ± 16.95	72.83 ± 17.82	59.33 ± 19.19	71.67 ± 17.72
pa, pv, ws	82.75 ± 15.93	76.67 ± 15.18	**83.17 ± 15.63**	61.08 ± 18.28	**73.00 ± 17.99**
pa, pv, wq	74.00 ± 16.47	65.75 ± 17.89	74.25 ± 16.57	60.67 ± 18.79	64.00 ± 16.45
pa, wq, ws	82.42 ± 15.38	80.17 ± 14.76	80.33 ± 15.43	58.08 ± 18.85	62.83 ± 16.05
pv, wq, ws	80.33 ± 15.88	77.25 ± 16.17	73.83 ± 17.36	53.08 ± 17.72	62.92 ± 17.67
pa, pv, wq, ws	80.33 ± 15.70	**80.92 ± 15.85**	82.17 ± 15.50	56.67 ± 18.86	64.25 ± 17.27

Source: Research data.

Table 35 – Classification metrics for the best result of classification in the *Fungi* dataset in Table 34

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
basidiomycetes	0.786	0.733	0.067	0.759	0.683	0.935	15
eurotiomycetes	0.800	0.800	0.067	0.800	0.733	0.941	15
saccharomycetes	0.933	0.933	0.022	0.933	0.911	0.985	15
sordariomycetes	0.812	0.867	0.067	0.839	0.783	0.941	15

Source: Research data.

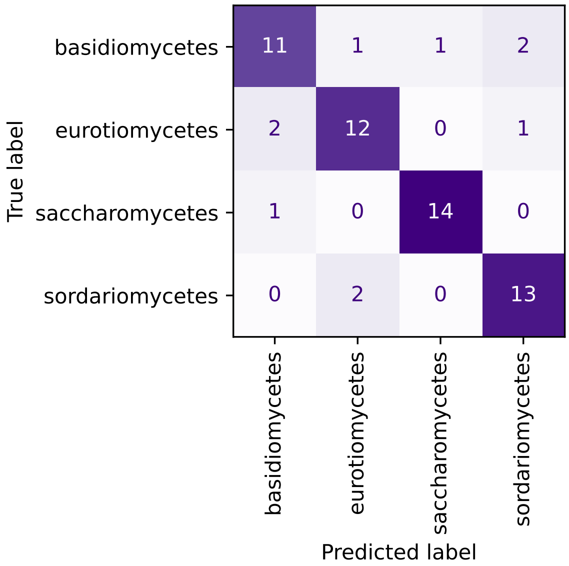
Figure 52 presents the confusion matrix for the best result obtained in the *Fungi* dataset. The confusion matrix shows that at least one instance of each class was misclassified.

Figure 53 shows the area under the curve for each class for the best result obtained in the *Fungi* dataset. The AUC-ROC values are greater than 0.93 for each class, which means that the classifier is generally able to identify the positive and negative instances of each class, which was also seen in the confusion matrix.

5.5.5 Kingdom dataset

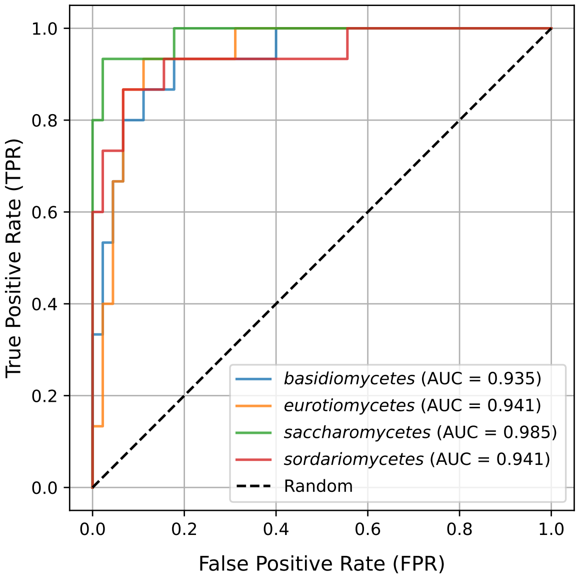
Table 36 presents the classification results in the *Kingdom* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume,

Figure 52 – Confusion matrix obtained for the best results of classification in the *Fungi* dataset in Table 34



Source: Elaborated by the author.

Figure 53 – AUC-ROC obtained for the best results of classification in the *Fungi* dataset in Table 34



Source: Elaborated by the author.

water quantity, water steps $\{pa, pv, wq, ws\}$. In this dataset the classifier achieved an accuracy of 99.75% using the 4 features with $wg = 2$ and $hg = 60$, what indicates that out of 160 networks, only 1 network was misclassified.

Table 37 presents the classification metrics for the best result obtained in the *Kingdom* dataset. The values for each metric indicate that in general the classifier is correctly identifying the positive instances, and the negative instances.

Table 36 – Classification results using the LDA classifier for the *Kingdom* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=4*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=40*	*hg=60*
pa	97.44 ± 3.65	95.31 ± 4.71	96.28 ± 4.33	83.62 ± 8.65	88.25 ± 7.06
pv	95.84 ± 4.69	96.69 ± 4.28	95.91 ± 4.82	94.41 ± 5.78	96.44 ± 4.66
wq	75.53 ± 9.58	67.28 ± 11.51	89.25 ± 6.71	96.16 ± 4.58	92.94 ± 6.32
ws	95.56 ± 5.10	93.53 ± 6.14	95.06 ± 5.29	95.59 ± 4.83	97.38 ± 3.56
pa, pv	94.88 ± 5.11	94.38 ± 5.69	96.19 ± 4.28	98.28 ± 3.36	97.38 ± 3.72
wq, ws	98.81 ± 2.53	99.03 ± 2.58	98.97 ± 2.40	97.69 ± 3.38	98.72 ± 2.60
pa, wq	94.59 ± 5.47	92.75 ± 6.36	96.91 ± 4.28	98.06 ± 3.50	95.75 ± 4.71
pa, ws	97.44 ± 3.86	97.44 ± 3.65	98.78 ± 2.77	97.88 ± 3.56	99.34 ± 2.20
pv, wq	96.00 ± 4.29	93.25 ± 5.77	94.59 ± 5.22	98.44 ± 3.17	97.25 ± 3.98
pv, ws	99.03 ± 2.35	98.25 ± 2.88	99.06 ± 2.40	98.06 ± 3.45	98.12 ± 3.19
pa, pv, ws	98.97 ± 2.40	98.94 ± 2.51	98.97 ± 2.48	97.94 ± 3.49	99.12 ± 2.42
pa, pv, wq	98.00 ± 3.47	96.00 ± 4.84	98.25 ± 3.13	**99.00 ± 2.53**	96.84 ± 4.24
pa, wq, ws	98.78 ± 2.98	99.53 ± 1.76	99.25 ± 2.12	98.16 ± 2.98	99.03 ± 2.35
pv, wq, ws	**99.06** ± **2.32**	99.38 ± 1.98	99.31 ± 1.96	97.38 ± 3.50	**99.47** ± **1.74**
pa, pv, wq, ws	98.91 ± 2.46	**99.75** ± **1.22**	**99.34** ± **1.92**	98.00 ± 3.41	99.47 ± 1.85

Source: Research data.

Table 37 – Classification metrics for the best result of classification in the *Kingdom* dataset in Table 36

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
animals	1.000	1.000	0.000	1.000	1.000	1.000	40
fungi	0.976	1.000	0.008	0.988	0.984	1.000	40
plants	1.000	1.000	0.000	1.000	1.000	1.000	40
protist	1.000	0.975	0.000	0.987	0.983	1.000	40

Source: Research data.

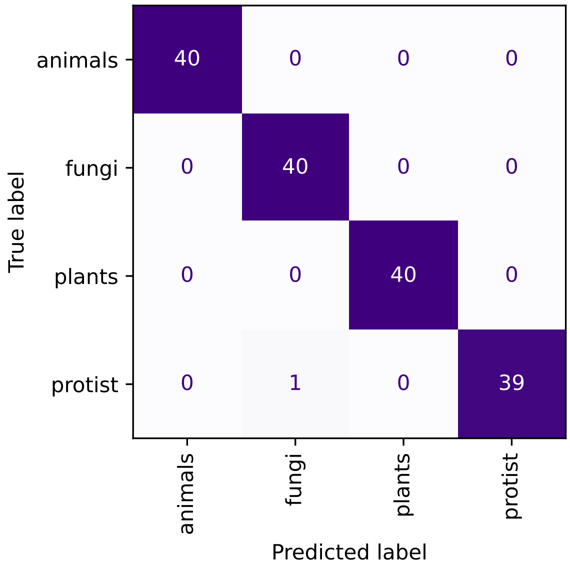
Figure 54 presents the confusion matrix for the best result obtained in the *Kingdom* dataset. The confusion matrix shows that one instance of the *protist* class is misclassified as *fungi* class.

Figure 55 shows the area under the curve for each class for the best result obtained in the *Kingdom* dataset. In this particular case, the AUC-ROC shows a value of 1 for each class.

5.5.6 Plant dataset

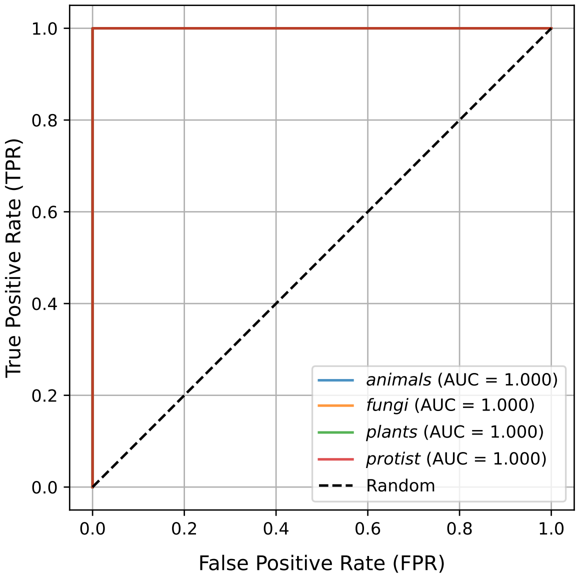
Table 38 presents the classification results in the *Plant* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. The *Plant* dataset only has 27 samples and an accuracy

Figure 54 – Confusion matrix obtained for the best results of classification in the *Kingdom* dataset in Table 36



Source: Elaborated by the author.

Figure 55 – AUC-ROC obtained for the best results of classification in the *Kingdom* dataset in Table 36



Source: Elaborated by the author.

value of 88.52% means that 3 instances were misclassified.

Table 39 presents the classification metrics for the best result obtained in the *Plant* dataset. The precision values indicate that the classifier is able to identify the positive instances in each class. The f1-score values indicate that there is a balance between the precision and recall values.

Figure 56 presents the confusion matrix for the best result obtained in the *Plant* dataset.

Table 38 – Classification results using the LDA classifier for the *Plant* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=4*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=40*	*hg=60*
pa	68.52 ± 12.86	75.93 ± 13.17	83.70 ± 12.42	71.48 ± 12.73	63.33 ± 11.88
pv	83.70 ± 10.63	85.00 ± 10.89	76.11 ± 11.08	72.41 ± 13.22	75.56 ± 11.44
wq	85.00 ± 11.08	78.33 ± 11.19	74.63 ± 14.50	77.59 ± 15.11	77.22 ± 12.74
ws	74.81 ± 12.97	82.78 ± 9.17	74.44 ± 12.88	70.93 ± 14.07	74.44 ± 13.65
pa, pv	86.30 ± 12.06	86.85 ± 12.91	**84.07 ± 11.54**	76.67 ± 9.45	77.59 ± 12.59
wq, ws	83.89 ± 11.90	77.96 ± 11.20	74.63 ± 14.07	72.22 ± 13.38	75.19 ± 12.23
pa, wq	84.26 ± 11.53	76.67 ± 10.87	75.93 ± 13.02	77.41 ± 12.66	75.56 ± 12.14
pa, ws	82.41 ± 10.98	80.74 ± 9.91	77.22 ± 11.19	75.37 ± 13.62	75.56 ± 13.58
pv, wq	85.74 ± 10.36	82.96 ± 10.44	81.48 ± 11.59	**83.52 ± 9.83**	81.67 ± 9.48
pv, ws	77.78 ± 12.00	85.00 ± 9.48	83.33 ± 10.44	80.56 ± 13.09	**84.63 ± 14.21**
pa, pv, ws	85.00 ± 12.48	**87.22 ± 8.80**	83.33 ± 11.56	79.26 ± 12.58	81.67 ± 15.16
pa, pv, wq	85.93 ± 11.27	83.33 ± 9.41	80.74 ± 12.81	81.48 ± 10.86	79.26 ± 10.63
pa, wq, ws	85.93 ± 11.45	77.22 ± 10.43	73.33 ± 12.54	74.07 ± 12.78	78.15 ± 13.30
pv, wq, ws	84.81 ± 10.53	80.19 ± 11.84	79.07 ± 13.00	79.63 ± 12.53	82.41 ± 11.53
pa, pv, wq, ws	**88.52 ± 10.34**	79.44 ± 11.97	78.52 ± 13.89	78.33 ± 13.06	80.74 ± 11.45

Source: Research data.

Table 39 – Classification metrics for the best result of classification in the *Plant* dataset in Table 38

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
eudicots	0.900	1.000	0.056	0.947	0.922	0.969	9
greenalgae	1.000	1.000	0.000	1.000	1.000	1.000	9
monocots	1.000	0.889	0.000	0.941	0.918	0.969	9

Source: Research data.

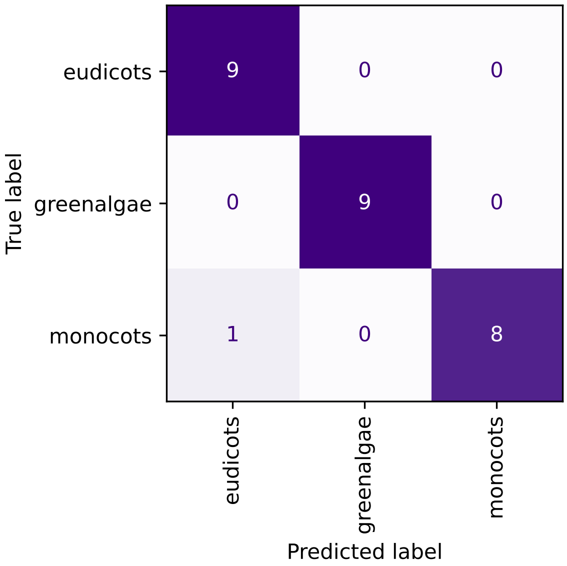
As previously stated, the results shown in the confusion matrix differ to those of the classification results because of implementation details of the cross-validation strategy in *Scikit-learn*. For this specific case, only 1 instance is misclassified, which means an accuracy value of 96%.

Figure 57 shows the area under the curve for each class for the best result obtained in the *Plant* dataset. The values of AUC-ROC for each class indicate values close to an optimal model.

5.5.7 Protist dataset

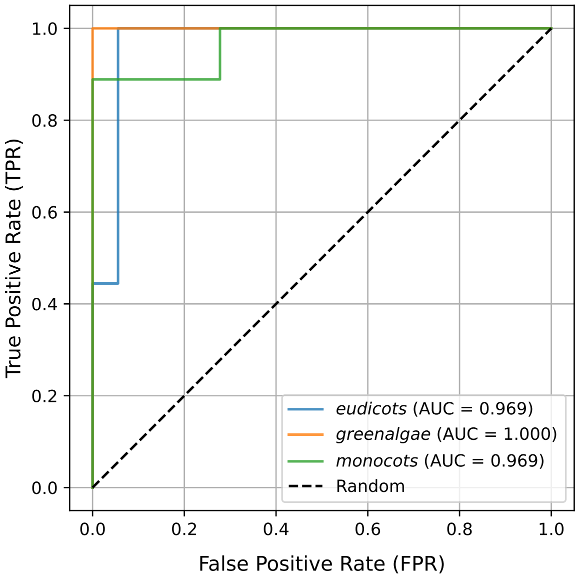
Table 40 presents the classification results in the *Protist* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. The best accuracy obtained in *Protist* was 83.13% using 2

Figure 56 – Confusion matrix obtained for the best results of classification in the *Plant* dataset in Table 38



Source: Elaborated by the author.

Figure 57 – AUC-ROC obtained for the best results of classification in the *Plant* dataset in Table 38



Source: Elaborated by the author.

features $\{pv, wq\}$ with parameter values $wg = 4$ and $hg = 60$. This dataset has the least number of instances among all the datasets studied. Out of 20 networks, 3 networks were incorrectly classified.

Table 41 presents the classification metrics for the best result obtained in the *Protist* dataset. The precision values for *Amoebozoa* and *Euglenozoa* of 1 indicates that the classifier correctly identified the positive instances of these classes. Instead, for *Alveolates* and *Stramenopiles* classes have a greater number of false positives.

Table 40 – Classification results using the LDA classifier for the *Protist* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=4*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=40*	*hg=60*
pa	65.87 ± 13.51	72.26 ± 13.83	67.58 ± 15.00	65.04 ± 15.56	69.25 ± 15.41
pv	52.70 ± 11.90	65.16 ± 13.73	60.56 ± 17.23	63.21 ± 17.54	72.38 ± 14.09
wq	56.19 ± 12.65	73.25 ± 13.59	64.56 ± 16.38	51.75 ± 12.63	82.82 ± 12.36
ws	47.70 ± 13.56	75.24 ± 10.10	68.85 ± 14.84	56.59 ± 13.77	73.25 ± 13.27
pa, pv	**66.43 ± 14.82**	65.95 ± 17.58	71.15 ± 13.78	**70.60 ± 13.30**	76.51 ± 13.04
wq, ws	52.54 ± 14.17	77.14 ± 11.79	67.30 ± 15.37	61.79 ± 13.52	80.67 ± 11.37
pa, wq	56.47 ± 14.47	73.25 ± 12.55	66.27 ± 15.87	56.27 ± 14.00	81.07 ± 11.86
pa, ws	50.63 ± 11.88	76.79 ± 10.87	69.21 ± 15.48	65.40 ± 13.74	76.71 ± 12.40
pv, wq	58.61 ± 11.16	74.96 ± 13.89	66.67 ± 15.37	58.25 ± 13.29	**83.13 ± 10.55**
pv, ws	51.87 ± 13.13	77.94 ± 11.09	**71.63 ± 13.78**	66.90 ± 11.73	79.25 ± 9.93
pa, pv, ws	52.22 ± 14.32	77.46 ± 10.74	70.60 ± 14.83	65.00 ± 14.97	78.97 ± 10.05
pa, pv, wq	54.37 ± 13.64	73.81 ± 14.21	66.43 ± 14.96	54.05 ± 14.40	79.76 ± 13.36
pa, wq, ws	50.75 ± 14.68	77.90 ± 12.33	67.38 ± 15.87	61.83 ± 13.50	81.51 ± 11.63
pv, wq, ws	53.33 ± 15.69	**78.41 ± 11.99**	65.67 ± 14.85	63.85 ± 12.92	81.15 ± 10.97
pa, pv, wq, ws	52.18 ± 15.24	77.50 ± 11.57	65.83 ± 13.52	61.94 ± 12.40	78.49 ± 11.54

Source: Research data.

Table 41 – Classification metrics for the best result of classification in the *Protist* dataset in Table 40

Class	**precision**	**recall**	**FPR**	**f1-score**	**MCC**	**AUC**	**support**
Alveolates	0.833	1.000	0.067	0.909	0.882	0.973	5
Amoebozoa	1.000	0.800	0.000	0.889	0.866	0.827	5
Euglenozoa	1.000	1.000	0.000	1.000	1.000	1.000	5
Stramenopiles	0.800	0.800	0.067	0.800	0.733	0.920	5

Source: Research data.

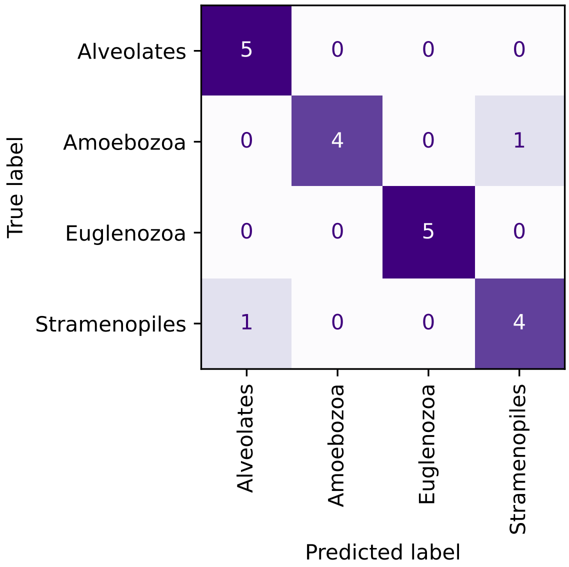
Figure 58 presents the confusion matrix for the best result obtained in the *Protist* dataset. The confusion matrix shows that 1 instance of *Amoebozoa* and 1 instance of *Stramenopiles* are misclassified.

Figure 59 shows the area under the curve for each class for the best result obtained in the *Protist* dataset. The plot shows that the *Amoebozoa* has the lowest value of AUC-ROC, this corresponds with the plot that in one point has a value lower than 0.5.

5.5.8 Social dataset

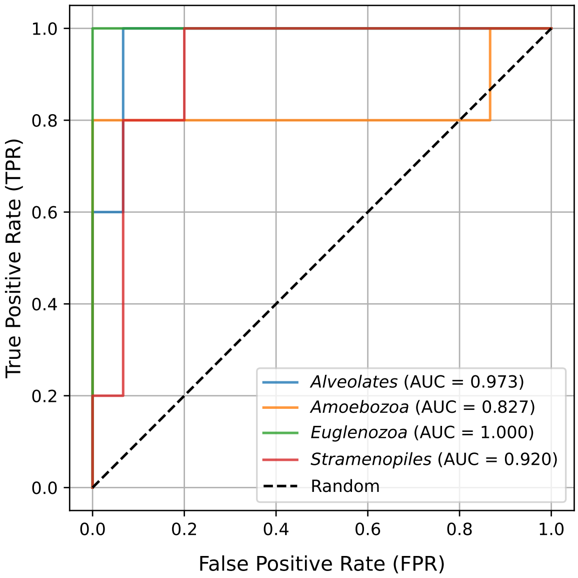
Table 42 presents the classification results in the *Social* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water

Figure 58 – Confusion matrix obtained for the best results of classification in the *Protist* dataset in Table 40



Source: Elaborated by the author.

Figure 59 – AUC-ROC obtained for the best results of classification in the *Protist* dataset in Table 40



Source: Elaborated by the author.

quantity, water steps $\{pa, pv, wq, ws\}$. The classification results using the LDA classifier shows that when the same parameters of the real-world metabolic datasets are applied to the *Social* dataset the classification results reach 77.25%.

For this dataset we also show the classification results usign the Linear SVC and kNN classifiers, as they provided similar accuracy results in Table 43 and Table 44. The best accuracy value with the Linear SVC classifier is 91.95% using the parameter values $wg = 4, hg = 40$, and the feature vector $\{wq, ws\}$, and coincidentally is the kNN classifier provided the same result

Table 42 – Classification results using the LDA classifier for the *Social* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

LDA	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=4*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=40*	*hg=60*
pa	**72.95 ± 11.57**	**72.30 ± 10.76**	75.15 ± 12.08	71.90 ± 12.02	70.15 ± 12.67
pv	69.25 ± 13.00	72.00 ± 13.15	**77.25 ± 12.16**	**72.15 ± 14.10**	**76.20 ± 12.63**
wq	66.30 ± 12.82	65.05 ± 13.30	67.05 ± 12.60	68.10 ± 14.91	64.50 ± 14.41
ws	61.85 ± 14.39	58.10 ± 15.57	64.40 ± 14.41	63.35 ± 15.14	62.65 ± 14.75
pa, pv	66.35 ± 14.25	69.75 ± 12.55	72.65 ± 12.63	71.75 ± 12.71	73.40 ± 13.51
wq, ws	55.30 ± 16.28	63.90 ± 15.16	64.40 ± 14.02	64.45 ± 13.74	65.50 ± 14.17
pa, wq	63.45 ± 16.11	64.30 ± 15.80	65.30 ± 14.73	69.50 ± 14.79	64.40 ± 14.89
pa, ws	60.40 ± 13.41	61.40 ± 13.46	66.10 ± 13.85	65.30 ± 14.93	65.35 ± 14.69
pv, wq	60.20 ± 15.56	66.10 ± 14.52	66.65 ± 13.65	63.45 ± 15.99	60.70 ± 14.88
pv, ws	60.00 ± 16.25	61.35 ± 17.05	66.45 ± 14.21	62.15 ± 15.36	64.30 ± 13.87
pa, pv, ws	57.80 ± 15.69	64.35 ± 15.35	62.50 ± 15.64	65.25 ± 13.41	67.00 ± 13.08
pa, pv, wq	59.05 ± 16.05	63.25 ± 14.83	65.75 ± 14.61	68.50 ± 15.32	58.00 ± 15.94
pa, wq, ws	57.70 ± 15.71	60.10 ± 15.46	60.10 ± 15.33	70.10 ± 13.53	67.15 ± 15.05
pv, wq, ws	59.80 ± 15.43	64.85 ± 13.96	66.25 ± 14.85	65.45 ± 14.79	67.85 ± 14.79
pa, pv, wq, ws	57.85 ± 14.28	63.55 ± 14.52	67.05 ± 12.92	71.40 ± 13.79	69.15 ± 14.66

Source: Research data.

using the same values. The best accuracy value with the kNN classifier is 92.90% using the parameter values $wg = 2, hg = 60$, and the feature vector $\{pa\}$. We have two classifiers, Linear SVC and kNN, that have similar good results of accuracy, 91.95% and 92.90% respectively, but the LDA classifier has a lower accuracy of 77.25%, it suggests that the Linear SVC and kNN classifiers perform consistently well on the *Social* dataset, while the LDA classifier might not be suitable for this specific dataset.

Table 45 presents the classification metrics for the best result obtained in the *Social* dataset. The MCC is 0.761 and the AUC-ROC is 0.880. The values of precision of 0.896 for the *gplus* class indicates that the classifier is relatively accurate in its positive predictions, with a low rate of false positives. And its recall value of 0.860 indicates that the classifier is sensitive to positive instances but is also missing some instances.

Figure 60 presents the confusion matrix for the best result obtained in the *Social* dataset using the kNN classifier. The confusion matrix incorrectly classified 5 networks of the *twitter* class and incorrectly classified 7 networks of *gplus* class.

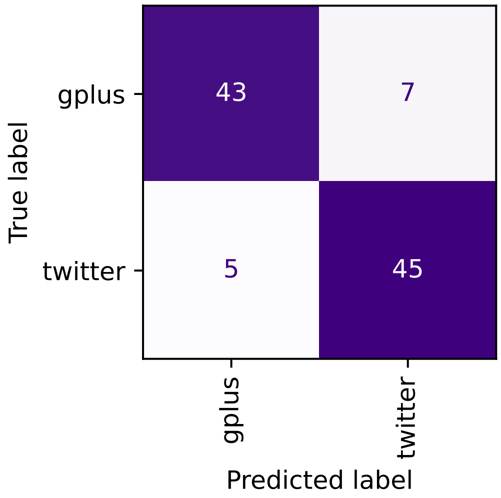
Figure 61 shows the area under the curve for the best result obtained in the *Social* dataset. A value of 0.880 of AUC-ROC suggests that on average the classifier correctly identify positive instances more times than negative instances.

Table 43 – Classification results using the Linear SVC classifier for the *Social* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

Linear SVC **Features**	*wg=2* *hg=40*	*wg=2* *hg=60*	*wg=2* *hg=100*	*wg=4* *hg=40*	*wg=4* *hg=60*
pa	**91.15** ± **8.79**	**91.30** ± **7.89**	91.65 ± 8.29	90.25 ± 9.19	**91.45** ± **8.27**
pv	88.00 ± 10.44	88.80 ± 9.67	90.00 ± 8.94	87.55 ± 10.07	88.00 ± 10.39
wq	89.00 ± 9.59	89.00 ± 9.11	89.00 ± 8.66	90.60 ± 9.68	90.05 ± 9.62
ws	88.40 ± 9.82	88.75 ± 9.32	87.95 ± 9.07	90.25 ± 9.30	87.80 ± 10.01
pa, pv	90.05 ± 9.03	90.90 ± 8.14	90.95 ± 8.34	89.80 ± 9.27	90.75 ± 8.83
wq, ws	89.55 ± 9.66	89.40 ± 9.09	89.05 ± 8.64	**91.95** ± **8.76**	90.40 ± 9.43
pa, wq	89.40 ± 9.62	89.85 ± 8.63	90.20 ± 8.48	91.30 ± 9.02	91.30 ± 8.85
pa, ws	90.65 ± 9.17	90.65 ± 8.19	90.85 ± 8.59	91.00 ± 9.06	90.30 ± 9.05
pv, wq	89.00 ± 9.59	89.65 ± 8.80	91.60 ± 8.03	90.90 ± 9.50	90.60 ± 9.41
pv, ws	88.60 ± 9.85	89.80 ± 8.36	91.00 ± 8.25	90.20 ± 9.54	87.85 ± 9.74
pa, pv, ws	90.55 ± 9.01	90.65 ± 8.31	91.35 ± 8.47	91.35 ± 8.98	90.30 ± 9.16
pa, pv, wq	89.45 ± 9.60	89.80 ± 8.66	91.25 ± 8.24	91.35 ± 8.98	91.25 ± 8.83
pa, wq, ws	90.30 ± 9.27	90.00 ± 8.54	90.20 ± 8.48	91.90 ± 8.74	90.50 ± 9.42
pv, wq, ws	89.65 ± 9.61	89.75 ± 8.74	**91.65** ± **7.92**	91.90 ± 8.74	90.00 ± 9.43
pa, pv, wq, ws	90.45 ± 9.13	90.05 ± 8.57	91.25 ± 8.30	91.85 ± 8.78	90.45 ± 9.61

Source: Research data.

Figure 60 – Confusion matrix obtained for the best results of classification in the *Social* dataset in Table 44



Source: Elaborated by the author.

5.5.9 Stomata dataset

Table 46 presents the classification results in the *Stomata* dataset using the values of the parameters selected. It is presented the results of the four features plant area, plant volume, water quantity, water steps $\{pa, pv, wq, ws\}$. We present the results using the kNN classifier, because

Table 44 – Classification results using the kNN classifier for the *Social* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

kNN	*wg=2*	*wg=2*	*wg=2*	*wg=4*	*wg=4*
Features	*hg=40*	*hg=60*	*hg=100*	*hg=40*	*hg=60*
pa	91.95 ± 8.47	**92.90 ± 7.52**	**92.90 ± 7.59**	91.80 ± 8.70	**92.80 ± 8.01**
pv	87.10 ± 10.70	87.55 ± 10.02	88.75 ± 8.94	87.90 ± 10.03	86.35 ± 10.73
wq	**92.75 ± 8.42**	92.55 ± 7.75	92.70 ± 7.92	90.90 ± 9.18	90.55 ± 8.90
ws	91.00 ± 8.89	91.05 ± 8.21	90.10 ± 8.94	90.10 ± 9.38	88.05 ± 9.88
pa, pv	90.65 ± 9.00	90.60 ± 8.22	91.25 ± 8.18	90.00 ± 9.64	91.80 ± 8.23
wq, ws	91.85 ± 8.72	92.40 ± 7.76	92.60 ± 7.89	91.95 ± 8.76	91.10 ± 8.47
pa, wq	91.95 ± 8.47	92.90 ± 7.52	92.90 ± 7.72	**92.05 ± 8.56**	91.75 ± 8.45
pa, ws	92.00 ± 8.49	92.90 ± 7.52	92.90 ± 7.72	90.50 ± 9.53	89.10 ± 9.71
pv, wq	89.95 ± 9.41	90.30 ± 8.83	90.75 ± 8.60	91.00 ± 9.33	91.90 ± 9.29
pv, ws	87.40 ± 9.71	87.95 ± 9.71	90.75 ± 8.48	89.75 ± 9.51	88.90 ± 9.48
pa, pv, ws	90.05 ± 9.19	90.30 ± 8.48	91.15 ± 8.32	89.80 ± 9.16	90.10 ± 8.60
pa, pv, wq	90.95 ± 9.09	91.60 ± 7.90	91.80 ± 7.79	91.40 ± 9.06	92.50 ± 8.35
pa, wq, ws	92.60 ± 8.38	92.90 ± 7.52	92.90 ± 7.72	90.65 ± 9.49	89.25 ± 9.79
pv, wq, ws	88.55 ± 9.46	91.05 ± 8.57	91.75 ± 7.96	90.95 ± 9.14	90.10 ± 9.11
pa, pv, wq, ws	90.50 ± 9.47	91.30 ± 8.14	92.45 ± 7.84	91.15 ± 9.12	90.95 ± 8.58

Source: Research data.

Table 45 – Classification metrics for the best result of classification in the *Social* dataset in Table 44

Class	**precision**	**recall**	**FPR**	**f1-score**	**support**
gplus	0.896	0.860	0.100	0.878	50
twitter	0.865	0.900	0.140	0.882	50

Source: Research data.

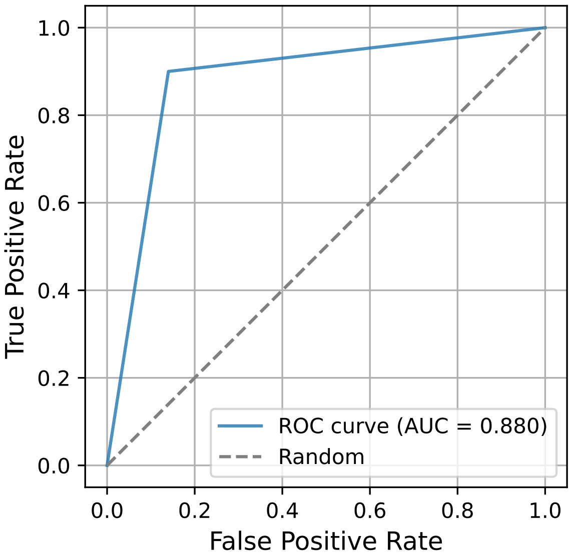
in this dataset presented better classification results. For this dataset, the best accuracy is 94.65% using only the feature $\{pa\}$. Unlike the *Social* dataset, when using more than one feature the accuracy value decreases.

Table 47 presents the classification metrics for the best result obtained in the *Stomata* dataset. In all the metrics the values are similar for each class, this indicates that in general the classifier is correctly distinguishing positive from negative instances.

Figure 62 presents the confusion matrix for the best result obtained in the *Stomata* dataset. Each of the 3 classes got confused with the other classes, the *natural* class has the least misclassified.

Figure 63 shows the area under the curve for each class for the best result obtained in the *Stomata* dataset. The AUC-ROC also shows that the 3 classes have similar values, which means

Figure 61 – AUC-ROC obtained for the best results of classification in the *Social* dataset in Table 44



Source: Elaborated by the author.

Table 46 – Classification results using the kNN classifier for the *Stomata* dataset using the values of the parameters selected. For each combination of parameters the highest accuracy is shown in bold, and the cells with the top 3 values are colored in light blue.

kNN **Features**	*wg=2* *hg=40*	*wg=2* *hg=60*	*wg=2* *hg=100*	*wg=4* *hg=40*	*wg=4* *hg=60*
pa	**81.01 ± 6.90**	**88.28 ± 5.86**	**94.65 ± 4.24**	**77.32 ± 7.62**	**85.88 ± 5.88**
pv	57.82 ± 8.17	62.63 ± 9.01	70.33 ± 7.70	54.26 ± 8.12	57.61 ± 8.63
wq	55.40 ± 9.13	57.59 ± 8.71	56.80 ± 7.85	58.02 ± 7.75	61.14 ± 7.88
ws	51.06 ± 8.87	51.81 ± 8.43	53.30 ± 8.55	50.65 ± 8.29	51.42 ± 7.85
pa, pv	66.81 ± 8.36	72.74 ± 9.14	77.53 ± 7.07	65.09 ± 7.81	72.06 ± 8.27
wq, ws	55.37 ± 8.25	56.43 ± 9.17	55.65 ± 7.73	56.64 ± 8.00	58.35 ± 7.99
pa, wq	64.36 ± 8.07	68.76 ± 7.87	69.07 ± 7.49	63.68 ± 8.05	65.95 ± 7.81
pa, ws	63.72 ± 8.22	66.88 ± 7.77	70.46 ± 7.02	64.24 ± 7.91	69.16 ± 7.74
pv, wq	52.24 ± 8.36	54.05 ± 8.10	55.23 ± 8.42	50.78 ± 8.02	52.82 ± 8.46
pv, ws	48.32 ± 7.75	51.01 ± 8.46	52.64 ± 7.95	47.90 ± 7.93	51.01 ± 8.00
pa, pv, ws	60.57 ± 8.16	61.15 ± 7.90	65.33 ± 6.95	57.56 ± 7.77	63.35 ± 7.83
pa, pv, wq	61.15 ± 8.56	63.70 ± 7.78	66.08 ± 7.28	58.63 ± 8.09	62.80 ± 7.83
pa, wq, ws	63.09 ± 7.86	65.14 ± 7.80	67.26 ± 7.04	63.27 ± 7.55	64.82 ± 8.00
pv, wq, ws	50.87 ± 8.30	51.85 ± 8.81	53.75 ± 8.03	50.02 ± 7.93	51.93 ± 8.36
pa, pv, wq, ws	60.35 ± 8.27	60.63 ± 7.77	64.43 ± 7.12	58.10 ± 7.77	62.70 ± 7.71

Source: Research data.

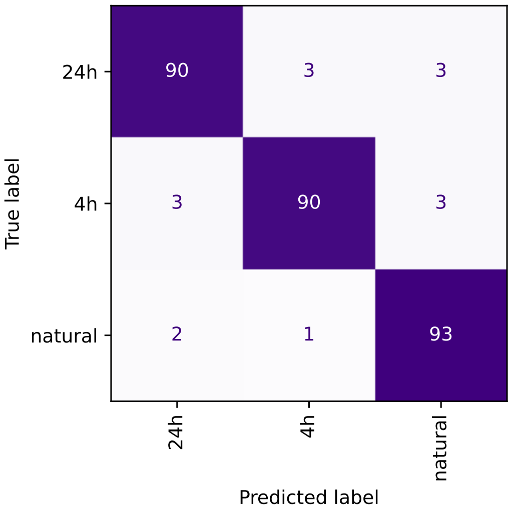
that in general the classifier discriminates the positive from the negative instances.

Table 47 – Classification metrics for the best result of classification in the *Stomata* dataset in Table 44

Class	precision	recall	FPR	f1-score	MCC	AUC	support
24h	0.947	0.938	0.026	0.942	0.914	0.956	96
4h	0.957	0.938	0.021	0.947	0.922	0.958	96
natural	0.939	0.969	0.031	0.954	0.930	0.969	96

Source: Research data.

Figure 62 – Confusion matrix obtained for the best results of classification in the *Stomata* dataset in Table 46



Source: Elaborated by the author.

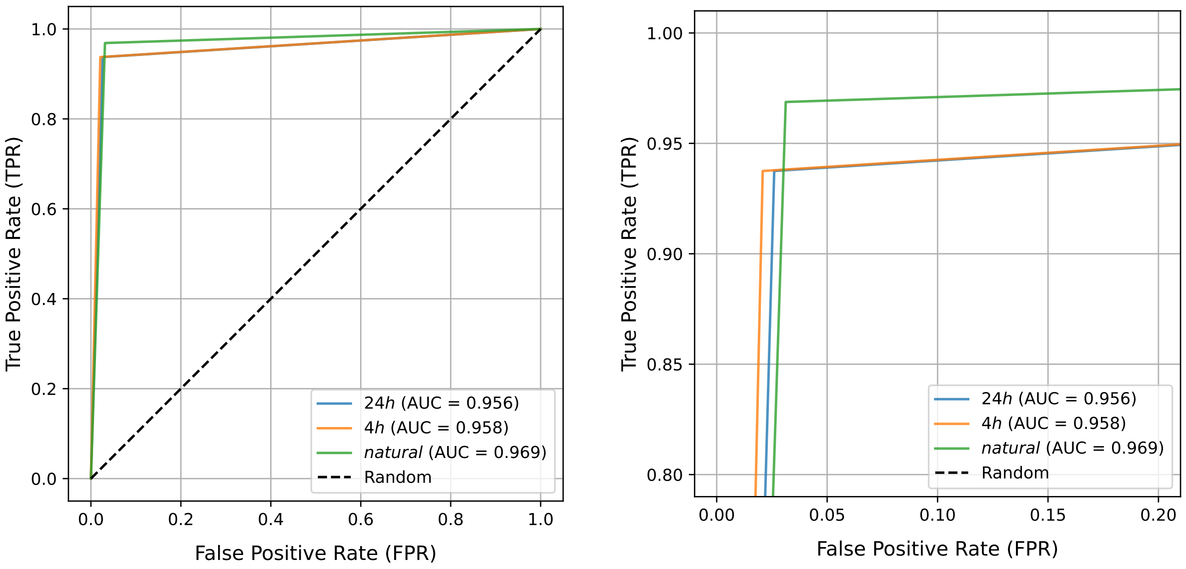
5.6 Comparison of classification results with previous works

In this section we show the classification results of our method in comparison with previous results in the literature such as, Density Time-Evolution Pattern (D-TEP) (ZIELINSKI *et al.*, 2022), Life-Like Network Automata - Binary patterns (LLNA-BP) (RIBAS; MACHICAO; BRUNO, 2020), LLNA and classical network measurements (MIRANDA; MACHICAO; BRUNO, 2016a).

D-TEP uses as features the measures of three histograms of global values, degrees and one temporal. LLNA-BP uses as features the measures of global histograms and degrees of binary patterns obtained from time evolution patterns. LLNA uses as features the measures of Shannon entropy, word length and Lempel-ziv complexity from time evolution patterns. And, the network measures used as features were the mean degree, degree distributions, correlations, distances, path lengths, hierarchical and spectral measures, transitivity, and clustering coefficient.

Table 48 presents the best classification results of the Growth Model as detailed in sec-

Figure 63 – AUC-ROC obtained for the best results of classification in the *Stomata* dataset in Table 46



(a) AUC ROC

(b) Zoom of AUC ROC

Source: Elaborated by the author.

tion 5.4 using the LDA classifier. For synthetic datasets, our approach obtains values comparable to the methods in the literature, except in the *4-models* + $\langle k \rangle$ dataset, in which obtains the lowest accuracy value.

Table 48 – Comparison of the Growth Model with previous approaches in synthetic network datasets using global features

Dataset	**Growth**	**D-TEP**	**LLNA-BP**	**LLNA**	**Structural**
4-models	99.13 ± 0.26	100.00 ± 0.00	100.00 ± 0.00	99.99 ± 0.00	100.0 ± 0.00
4-models + k	84.97 ± 1.68		98.31 ± 0.02	90.76 ± 0.07	65.20 ± 0.20
Scale-free	99.82 ± 0.57	100.00 ± 0.00	99.52 ± 0.19	98.30 ± 0.07	96.20 ± 0.20
Noise-0	99.56 ± 0.72	100.00 ± 0.00	100.00 ± 0.00		-
Noise-10%	98.26 ± 1.31	100.00 ± 0.00	100.00 ± 0.00	99.98 ± 0.00	100.00 ± 0.00
Noise-20%	97.89 ± 1.44	100.00 ± 0.00	99.98 ± 0.00	99.97 ± 0.01	100.00 ± 0.00
Noise-30%	96.84 ± 1.87	99.75 ± 0.00	99.99 ± 0.00	99.95 ± 0.01	100.00 ± 0.00

Source: Elaborated by the author.

Table 49 presents the best classification results of the Growth Model as detailed in section 5.5 using the LDA classifier. For the real-world datasets our approach obtains values comparable to the methods in the literature. Even, for the metabolic network datasets obtain better results in *Actinobacteria*, *Fungi*, *Kingdom*, and *Plant* datasets. In the *Stomata* dataset obtained a better result with the kNN classifier.

Table 50 presents a comparison of the best classification results of the Growth Model in real-world datasets using node information of degree, eigenvector, and closeness. The accuracy

Table 49 – Comparison of the Growth Model with previous approaches in real network datasets using global measures

Dataset	**Growth**	**D-TEP**	**LLNA-BP**	**LLNA**	**Structural**
Actinobacteria	99.34 ± 2.28	97.68 ± 5.00	95.13 ± 1.22	91.48 ± 1.60	93.16 ± 0.70
Animals	98.50 ± 4.94	100.00 ± 0.00	84.87 ± 15.25	77.25 ± 16.29	83.71 ± 15.29
Firmicutes	95.84 ± 3.08	96.06 ± 0.35	98.30 ± 1.17	84.63 ± 2.00	95.67 ± 0.59
Fungi	83.17 ± 15.63	81.00 ± 4.38	76.17 ± 17.45	54.58 ± 19.38	54.90 ± 15.39
Kingdom	99.75 ± 1.22	96.24 ± 0.35	97.44 ± 3.98	93.10 ± 5.38	96.61 ± 4.33
Plant	88.52 ± 10.34	81.33 ± 6.00	74.81 ± 5.64	69.70 ± 4.67	54.19 ± 9.17
Protist	83.13 ± 10.55	-	87.00 ± 5.29	68.55 ± 6.30	45.10 ± 10.02
Social	92.90 ± 7.52	92.50 ± 0.50	93.40 ± 0.92	92.00 ± 1.00	88.00 ± 2.00
Stomata	94.65 ± 4.24	-		90.00 ± 6.00	83.00 ± 4.00

Source: Elaborated by the author.

is better when using node degree information.

Table 50 – Comparison of the best classification results in real-world network datasets using node information of degree, eigenvector, and closeness

Dataset	**Degree**	**Eigenvector**	**Closeness**
Actinobacteria	99.34 ± 2.28	96.13 ± 4.08	94.47 ± 5.01
Animals	98.50 ± 4.94	91.13 ± 11.61	96.33 ± 7.14
Firmicutes	95.84 ± 3.08	88.05 ± 4.35	87.75 ± 4.34
Fungi	83.17 ± 15.63	55.17 ± 18.36	60.83 ± 15.7
Kingdom	99.75 ± 1.22	91.88 ± 6.76	91.81 ± 6.29
Plant	88.52 ± 10.34	82.96 ± 11.74	86.3 ± 10.22
Protist	83.13 ± 10.55	62.86 ± 13.55	62.38 ± 13.98

Source: Elaborated by the author.

Table 51 presents a comparison of the best classification results of the Growth Model using global and local measures in real-world datasets. The local measures used are the Shannon entropy and Lempel-Ziv complexity. The classification scheme is the same as the previous experiments.

In *Firmicutes*, and *Kingdom* datasets, using local features yields better results compared to global features. For the remaining datasets, the results are approximately the same using local or global features. The *Fungi* dataset shows a greater difference, being better the use of global features.

5.6.1 Implementation details

The source code was developed in a Windows environment using the Python programming language (version 3.11). A useful feature of Python is that it integrates modules (packages,

Table 51 – Comparison of the best classification results in real-world network datasets using local measures and global features

Dataset	Local	Global
Actinobacteria	99.13 ± 2.15	99.34 ± 2.28
Animals	97.40 ± 7.04	98.50 ± 4.94
Firmicutes	96.62 ± 2.57	95.84 ± 3.08
Fungi	69.67 ± 17.22	83.17 ± 15.63
Kingdom	99.81 ± 1.06	99.75 ± 1.22
Plant	83.33 ± 13.07	88.52 ± 10.34
Protist	79.76 ± 11.30	83.13 ± 10.55

Source: Elaborated by the author.

libraries or frameworks) from any scientific domain.

For the modeling stage and generation of patterns created in the environment we utilized *Mesa* (version 1.2.0), *Mesa* is a package designed to create Agent-Based Models in Python. To model and study the structure of complex networks we used *NetworkX* (version 3.1). For the classification part we used machine learning classifiers provided by *Scikit-learn* (version 1.3.0), and *Imbalanced-learn* (version 0.11.0) to deal with the imbalanced datasets. To track and store the classification results we used *MLflow* (version 2.4.2).

5.7 Considerations

In the synthetic network datasets studied in this work, we got similar results in most of the cases, what indicates, the potential to deal with real-world networks.

In most of the real-world network datasets studied in this work, we got comparable results and even better results than those previously presented in the literature.

This means that the model has the potential to continue to be explored and extended for applications involving pattern recognition in networks.

By analyzing the networks using global measures, obtained from all the agents after each iteration, the model is able to differentiate the classes within the bases.

CHAPTER 6

CONCLUSIONS

This work explored agent-based models, network science, and pattern recognition, which are active fields of research with real-world applications. Upon reviewing the literature on complex network categorization, a noticeable gap was identified: the lack of studies employing agent-based models for the analysis of complex networks, so the main focus of this thesis was to explore the use of Agent-Based Modeling (ABM) to develop methods for pattern recognition in complex networks. While agent-based models (ABMs) are frequently used in the literature to simulate real-world systems, This work demonstrates the potential to be employed with an exploratory approach in the context of categorizing complex networks.

The proposed model is called the Growth Model, it utilizes two types of agents, water, and plant, which interact with each other and the environment. Water agents move based on the environment's height, while plant agents grow according to the amount of water in their neighborhood. The Growth Model can work either with texture images or complex networks.

Methods from the state of the art for categorizing complex networks were studied, including deterministic walks and cellular automata. These methods only work with a single type of element, lacking interaction between them, and the activation of their elements is restricted to random and simultaneous forms. In contrast, agent-based models allow for multiple elements in their environment, which can interact with each other, and offer various activation methods for agents, including sequential, simultaneous, random, random by type of element, and domain-specific activation.

Initially, agent-based models in images were explored, as their spatial structure allows representation as a surface in three dimensions. This surface served as the environment for the agents that required the concept of "height" as a metric for their behavior in the simulation. This approach facilitated the visualization of interactions among agents and with the environment, contributing to a more comprehensive understanding of their dynamics. To extend the application of the model to complex networks, known for their irregular structure, a transformation from the image-based model was required. The natural relationship between nodes and edges was employed for this purpose. Consequently, the model takes advantage on the intrinsic characteristics of the underlying network space, serving as the environment for the agents.

During the simulation of the model, global and local characteristics can be collected and utilized as a feature vector for the classification stage. Global characteristics include the *plant area*, *plant volume*, *water quantity*, and *water steps*. These features were obtained in each iteration of the simulation, and the vector size depends on the number of model iterations. The local characteristics employed were the Lempel-Ziv complexity and the Shannon Entropy, these values were computed at each node and the size of the feature vector does not depend on the number of iterations.

The Growth Model was applied to synthetic and real datasets. The classification results for real networks are similar to those found in literature methods. It is worth noting that the accuracy improved in four datasets — *Actinobacteria*, *Fungi*, *Kingdom*, and *Plant* — when utilizing global features. Better results were obtained using local features for *Firmicutes* and *Kingdom* compared to the use of global features.

The results demonstrate the promising potential of this approach. The method opens numerous avenues for exploration and presents significant opportunities for the development of simulations using agent-based modeling. The proposed method indicates that agent-based models serve as tools for analyzing complex networks, providing valuable insights into their structure and behavior.

6.1 Production

The work "Social interaction layers in complex networks for the dynamical epidemic modeling of COVID-19 in Brazil", uses a computational model with Agent-Based Modeling in which a discrete process is simulated on a network, for the analysis of the contemporary COVID-19 epidemic.

Leonardo F.S. Scabini, Lucas C. Ribas, Mariane B. Neiva, Altamir G.B. Junior, **Alex J.F. Farfán**, Odemir M. Bruno (2021). Social interaction layers in complex networks for the dynamical epidemic modeling of COVID-19 in Brazil. *Physica A: Statistical Mechanics and its Applications*.

The work "Analyzing Patterns in Agent Based Models for Network Classification", submitted to the International Conference of Pattern Recognition and Applications that shows the applicability of the model for Pattern Recognition in Complex Networks.

BIBLIOGRAPHY

ADELUSI, T. I.; OYEDELE, A.-Q. K.; BOYENLE, I. D.; OGUNLANA, A. T.; ADEYEMI, R. O.; UKACHI, C. D.; IDRIS, M. O.; OLAOBA, O. T.; ADEDOTUN, I. O.; KOLAWOLE, O. E.; XIAOXING, Y.; ABDUL-HAMMED, M. Molecular modeling in drug discovery. **Informatics in Medicine Unlocked**, v. 29, p. 100880, 2022. ISSN 2352-9148. Available: <https://www.sciencedirect.com/science/article/pii/S235291482200034X>. Citation on page 35.

AHMED, N. I.; NASRIN, F. Reducing error rate for eye-tracking system by applying svm. In: SKALA, V.; SINGH, T. P.; CHOUDHURY, T.; TOMAR, R.; BASHAR, M. A. (Ed.). **Machine Intelligence and Data Science Applications**. Singapore: Springer Nature Singapore, 2022. p. 35–47. ISBN 978-981-19-2347-0. Citation on page 35.

ALBERT, R.; BARABÁSI, A.-L. Topology of evolving networks: Local events and universality. **Phys. Rev. Lett.**, American Physical Society, v. 85, p. 5234–5237, Dec 2000. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.85.5234>. Citation on page 62.

ALIZADEH, M.; CIOFFI-REVILLA, C. Activation regimes in opinion dynamics: Comparing asynchronous updating schemes. **Available at SSRN 2830325**, 2015. Available: <https://papers.ssrn.com/sol3/papers.cfm?abstract%5Fid=2830325>. Citation on page 50.

AMINI, N.; SHALBAF, A. Automatic classification of severity of covid-19 patients using texture feature and random forest based on computed tomography images. **International Journal of Imaging Systems and Technology**, v. 32, n. 1, p. 102–110, 2022. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/ima.22679>. Citation on page 36.

ANTELMI, A.; CORDASCO, G.; D'AMBROSIO, G.; VINCO, D. D.; SPAGNUOLO, C. Experimenting with agent-based model simulation tools. **Applied Sciences**, v. 13, n. 1, 2023. ISSN 2076-3417. Available: <https://www.mdpi.com/2076-3417/13/1/13>. Citation on page 50.

ANTELMI, A.; CORDASCO, G.; D'AURIA, M.; VINCO, D. D.; NEGRO, A.; SPAGNUOLO, C. On evaluating rust as a programming language for the future of massive agent-based simulations. **Communications in Computer and Information Science**, v. 1094, p. 15–28, 2019. ISSN 18650929. Conference of 19th Asia Simulation Conference, AsiaSim 2019 ; Conference Date: 30 October 2019 Through 1 November 2019; Conference Code:233729. Citation on page 51.

AURBACH, A.; SCHMID, B.; LIECHTI, F.; CHOKANI, N.; ABHARI, R. Simulation of broad front bird migration across western europe. **Ecological Modelling**, v. 415, p. 108879, 2020. ISSN 0304-3800. Available: <http://www.sciencedirect.com/science/article/pii/S0304380019303874>. Citation on page 47.

BACKES, A. R.; GONçALVES, W. N.; MARTINEZ, A. S.; BRUNO, O. M. Texture analysis and classification using deterministic tourist walk. **Pattern Recognition**, v. 43, n. 3, p. 685–694, 2010. ISSN 0031-3203. Available: <https://www.sciencedirect.com/science/article/pii/S0031320309002969>. Citation on page 56.

BAETENS, J. M.; BAETS, B. D. Cellular automata on irregular tessellations. **Dynamical Systems**, Taylor & Francis, v. 27, n. 4, p. 411–430, 2012. Citation on page 43.

BAPTISTA, D.; BACCO, C. D. Principled network extraction from images. **Royal Society Open Science**, v. 8, n. 7, p. 210025, 2021. Available: <https://royalsocietypublishing.org/doi/abs/10.1098/rsos.210025>. Citation on page 37.

BARABÁSI, A.-L. **Network Science**. [S.l.]: Cambridge University Press Cambridge, UK:, 2015. Citation on page 62.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **Science**, v. 286, n. 5439, p. 509–512, 1999. Available: <https://www.science.org/doi/abs/10.1126/science.286.5439.509>. Citations on pages 37, 41, and 62.

BARABÁSI, A.-L.; WATTS, D. J.; NEWMAN, M. **The structure and dynamics of networks**. [S.l.]: Princeton University Press, 2006. Citation on page 37.

BECHBERGER, D.; PERRYMAN, J. **Graph databases in action**. [S.l.]: Manning Publications, 2020. Citation on page 37.

BHARDWAJ, R.; BHAGAT, D. Two level encryption of grey scale image through 2d cellular automata. **Procedia Computer Science**, v. 125, p. 855–861, 2018. ISSN 1877-0509. The 6th International Conference on Smart Computing and Communications. Available: <http://www.sciencedirect.com/science/article/pii/S1877050917328776>. Citation on page 42.

BIELIK, T.; FONIO, E.; FEINERMAN, O.; DUNCAN, R. G.; LEVY, S. T. Working together: Integrating computational modeling approaches to investigate complex phenomena. **Journal of Science Education and Technology**, v. 30, n. 1, p. 40–57, Feb 2021. ISSN 1573-1839. Available: <https://doi.org/10.1007/s10956-020-09869-x>. Citation on page 33.

BISHOP, C. **Pattern Recognition and Machine Learning**. Springer, 2006. Available: <https://www.microsoft.com/en-us/research/publication/pattern-recognition-machine-learning/>. Citations on pages 30, 33, and 35.

BOCCARA, N. **Modeling complex systems**. [S.l.]: Springer-Verlag, 2010. Citation on page 47.

BODINE, E. N.; PANOFF, R. M.; VOIT, E. O.; WEISSTEIN, A. E. Agent-based modeling and simulation in mathematics and biology education. **Bulletin of Mathematical Biology**, v. 82, n. 8, p. 101, Jul 2020. ISSN 1522-9602. Available: <https://doi.org/10.1007/s11538-020-00778-z>. Citation on page 47.

BOGDAN, P.; CAETANO-ANOLLéS, G.; JOLLES, A.; KIM, H.; MORRIS, J.; MURPHY, C. A.; ROYER, C.; SNELL, E. H.; STEINBRENNER, A.; STRAUSFELD, N. Biological Networks across Scales—The Theoretical and Empirical Foundations for Time-Varying Complex Networks that Connect Structure and Function across Levels of Biological Organization. **Integrative and Comparative Biology**, v. 61, n. 6, p. 1991–2010, 05 2021. ISSN 1540-7063. Available: <https://doi.org/10.1093/icb/icab069>. Citation on page 37.

BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B.; VAROQUAUX, G. API design for machine learning software: experiences from the scikit-learn project. In: **ECML PKDD Workshop: Languages for Data Mining and Machine Learning**. [S.l.: s.n.], 2013. p. 108–122. Citation on page 82.

BUSH, V. **Science, the Endless Frontier**. Princeton: Princeton University Press, 2020. ISBN 9780691201658. Available: <https://doi.org/10.1515/9780691201658>. Citation on page 29.

CAO, Y.; ZHANG, X.; FU, Y.; LU, Z.; SHEN, X. Urban spatial growth modeling using logistic regression and cellular automata: A case study of hangzhou. **Ecological Indicators**, v. 113, p. 106200, 2020. ISSN 1470-160X. Available: <http://www.sciencedirect.com/science/article/pii/S1470160X20301370>. Citation on page 42.

CASANOVA, D. **Redes complexas em visão computacional com aplicações em bioinformática**. Phd Thesis (PhD Thesis) — Universidade de São Paulo, 2013. Citation on page 37.

CERVI, E.; CAMMI, A.; ZIO, E. A new approach for nuclear reactor analysis based on complex network theory. **Progress in Nuclear Energy**, v. 112, p. 96–106, 2019. ISSN 0149-1970. Available: <http://www.sciencedirect.com/science/article/pii/S0149197018303020>. Citation on page 37.

COMER, K. W. **Who goes first? An examination of the impact of activation on outcome behavior in Agent-Based Models**. Phd Thesis (PhD Thesis) — George Mason University, 2014. Available: <https://hdl.handle.net/1920/9070>. Citations on pages 49 and 50.

COMER, K. W.; LOERCH, A. G. The impact of agent activation on population behavior in an agent-based model of civil revolt. **Procedia Computer Science**, v. 20, p. 183–188, 2013. ISSN 1877-0509. Complex Adaptive Systems. Available: <https://www.sciencedirect.com/science/article/pii/S1877050913010582>. Citation on page 50.

COOK, M. Universality in elementary cellular automata. **Complex systems**, [Champaign, IL, USA: Complex Systems Publications, Inc., c1987-, v. 15, n. 1, p. 1–40, 2004. Citation on page 43.

COSCIA, M. **The Atlas for the Aspiring Network Scientist**. 2021. Citation on page 34.

COSTA, L. da F.; BOAS, P. R. V.; SILVA, F. N.; RODRIGUES, F. A. A pattern recognition approach to complex networks. **Journal of Statistical Mechanics: Theory and Experiment**, v. 2010, n. 11, p. P11015, nov 2010. Available: <https://dx.doi.org/10.1088/1742-5468/2010/11/P11015>. Citation on page 54.

COSTA, L. da F.; RODRIGUES, F. A.; TRAVIESO, G.; BOAS, P. R. V. Characterization of complex networks: A survey of measurements. **Advances in Physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007. Available: <https://doi.org/10.1080/00018730601170527>. Citations on pages 30, 39, 54, and 61.

CURRIE, C. S.; FOWLER, J. W.; KOTIADIS, K.; MONKS, T.; ONGGO, B. S.; ROBERTSON, D. A.; TAKO, A. A. How simulation modelling can help reduce the impact of covid-19. **Journal of Simulation**, Taylor & Francis, v. 14, n. 2, p. 83–97, 2020. Available: <https://doi.org/10.1080/17477778.2020.1751570>. Citation on page 35.

DATSERIS, G.; VAHDATI, A. R.; DUBOIS, T. C. Agents.jl: a performant and feature-full agent-based modeling software of minimal code complexity. **Simulation**, SAGE Publications, v. 0, n. 0, p. 003754972110688, Jan. 2022. Available: <https://doi.org/10.1177/00375497211068820>. Citations on pages 49 and 51.

DEANGELIS, D. L.; GRIMM, V. Individual-based models in ecology after four decades. **F1000Prime Rep**, England, v. 6, p. 39, Jun. 2014. Citation on page 56.

DEEPTHI, B.; SIVAKUMAR, B. Towards assessing the importance of individual stations in hydrometric networks: application of complex networks. **Stochastic Environmental Research and Risk Assessment**, v. 37, n. 4, p. 1333–1352, Apr 2023. ISSN 1436-3259. Available: <https://doi.org/10.1007/s00477-022-02340-w>. Citation on page 37.

DEMONGEOT, J.; GOLÈS, E.; TCHUENTE, M. **Dynamical systems and cellular automata**. [S.l.]: Academic Press, 1985. Citation on page 45.

DIESTEL, R. The basics. In: ____. **Graph Theory**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017. p. 1–34. ISBN 978-3-662-53622-3. Available: <https://doi.org/10.1007/978-3-662-53622-3_1>. Citation on page 37.

D'MELLO, R. J.; WAAS, A. M.; MAIARU, M.; KOON, R. Integrated computational modeling for efficient material and process design for composite aerospace structures. In: ____. **AIAA Scitech 2020 Forum**. [s.n.], 2020. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2020-0655>. Citation on page 35.

DOĞANAKSOY, A.; GÖLOĞLU, F. On lempel-ziv complexity of sequences. In: SPRINGER. **Sequences and Their Applications–SETA 2006: 4th International Conference Beijing, China, September 24-28, 2006 Proceedings 4**. [S.l.], 2006. p. 180–189. Citation on page 79.

DOROGOVTSEV, S. N.; MENDES, J. F. F. Evolution of networks. **Advances in Physics**, Taylor & Francis, v. 51, n. 4, p. 1079–1187, 2002. Available: <https://doi.org/10.1080/00018730110112519>. Citation on page 62.

DOWNEY, A. **Think complexity: Complexity Science and Computational Modeling**. 2nd. ed. Sebastopol, CA: "O'Reilly Media, Inc.", 2018. Citation on page 34.

ERDÖS, P. On random graphs. **Publicationes Mathematicae Debrecen**, v. 6, p. 290, 1959. Citation on page 62.

ERDOS, P.; RÉNYI, A. On the evolution of random graphs. **Publ. Math. Inst. Hung. Acad. Sci**, v. 5, n. 1, p. 17–60, 1960. Citations on pages 37 and 41.

FAN, R.-E.; CHANG, K.-W.; HSIEH, C.-J.; WANG, X.-R.; LIN, C.-J. Liblinear: A library for large linear classification. **the Journal of machine Learning research**, JMLR. org, v. 9, p. 1871–1874, 2008. Citation on page 81.

FARFÁN, A. J. F.; SCABINI, L. F. S.; BRUNO, O. M. A web-based system to assess texture analysis methods and datasets. In: VENTO, M.; PERCANNELLA, G. (Ed.). **Computer Analysis of Images and Patterns**. Cham: Springer International Publishing, 2019. p. 425–437. ISBN 978-3-030-29891-3. Citations on pages 33 and 35.

FARZAM, A.; SAMAL, A.; JOST, J. Degree difference: a simple measure to characterize structural heterogeneity in complex networks. **Scientific Reports**, v. 10, n. 1, p. 21348, Dec 2020. ISSN 2045-2322. Available: <https://doi.org/10.1038/s41598-020-78336-9>. Citation on page 39.

FLAIG, J.; HOUY, N. Altruism and fairness in schelling's segregation model. **Physica A: Statistical Mechanics and its Applications**, v. 527, p. 121298, 2019. ISSN 0378-4371. Available: <http://www.sciencedirect.com/science/article/pii/S0378437119307563>. Citation on page 48.

FLORINDO, J. B.; METZE, K. A cellular automata approach to local patterns for texture recognition. **Expert Systems with Applications**, v. 179, p. 115027, 2021. ISSN 0957-4174. Available: <https://www.sciencedirect.com/science/article/pii/S0957417421004681>. Citation on page 55.

FUNK, C.; RAINIE, L.; PAGE, D. Public and scientists' views on science and society. **Pew Research Center**, v. 29, 2015. Citation on page 29.

GAO, J.; XU, B. Complex systems, emergence, and multiscale analysis: A tutorial and brief survey. **Applied Sciences**, v. 11, n. 12, 2021. ISSN 2076-3417. Available: <https://www.mdpi.com/2076-3417/11/12/5736>. Citation on page 35.

GARDNER, M. Mathematical games. **Scientific American**, Scientific American, a division of Nature America, Inc., v. 223, n. 4, p. 120–123, 1970. ISSN 00368733, 19467087. Available: <http://www.jstor.org/stable/24927642>. Citation on page 45.

GHARAKHANLOU, N. M.; MESGARI, M. S.; HOOSHANGI, N. Developing an agent-based model for simulating the dynamic spread of plasmodium vivax malaria: A case study of sarbaz, iran. **Ecological Informatics**, v. 54, p. 101006, 2019. ISSN 1574-9541. Available: <http://www.sciencedirect.com/science/article/pii/S1574954119303176>. Citation on page 48.

GIRVAN, M.; NEWMAN, M. E. Community structure in social and biological networks. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 99, n. 12, p. 7821–7826, 2002. Citation on page 37.

GONÇALVES, W. N.; MARTINEZ, A. S.; BRUNO, O. M. Complex network classification using partially self-avoiding deterministic walks. **Chaos**, AIP Publishing, v. 22, n. 3, p. 033139, Sep 2012. Citation on page 56.

GONZÁLEZ-MÉNDEZ, M.; OLAYA, C.; FASOLINO, I.; GRIMALDI, M.; OBREGóN, N. Agent-based modeling for urban development planning based on human needs. conceptual basis and model formulation. **Land Use Policy**, v. 101, p. 105110, 2021. ISSN 0264-8377. Available: <https://www.sciencedirect.com/science/article/pii/S026483771931169X>. Citation on page 48.

GONZALEZ, R. **Digital Image Processing**. 4th. ed. [S.l.]: Pearson Education, 2016. Citation on page 36.

GRIMM, V.; BERGER, U.; BASTIANSEN, F.; ELIASSEN, S.; GINOT, V.; GISKE, J.; GOSS-CUSTARD, J.; GRAND, T.; HEINZ, S. K.; HUSE, G.; HUTH, A.; JEPSEN, J. U.; JORGENSEN, C.; MOOIJ, W. M.; MÜLLER, B.; PE'ER, G.; PIOU, C.; RAILSBACK, S. F.; ROBBINS, A. M.; ROBBINS, M. M.; ROSSMANITH, E.; RÜGER, N.; STRAND, E.; SOUISSI, S.; STILLMAN, R. A.; VABO, R.; VISSER, U.; DEANGELIS, D. L. A standard protocol for describing individual-based and agent-based models. **Ecological Modelling**, v. 198, n. 1, p. 115–126, 2006. ISSN 0304-3800. Available: <http://www.sciencedirect.com/science/article/pii/S0304380006002043>. Citation on page 47.

GRINSZTAJN, L.; OYALLON, E.; VAROQUAUX, G. Why do tree-based models still outperform deep learning on typical tabular data? In: **Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track**. [s.n.], 2022. Available: <https://openreview.net/forum?id=Fp7%5F%5FphQszn>. Citation on page 80.

GUPTA, M.; MISHRA, R. Spreading the information in complex networks: Identifying a set of top-n influential nodes using network structure. **Decision Support Systems**, v. 149, p. 113608, 2021. ISSN 0167-9236. Available: <https://www.sciencedirect.com/science/article/pii/S0167923621001184>. Citation on page 37.

HAASE, K.; REINHARDT, O.; LEWIN, W.-C.; WELTERSBACH, M. S.; STREHLOW, H. V.; UHRMACHER, A. M. Agent-based simulation models in fisheries science. **Reviews in Fisheries Science & Aquaculture**, Taylor & Francis, v. 0, n. 0, p. 1–24, 2023. Available: <https://doi.org/10.1080/23308249.2023.2201635>. Citation on page 48.

HALL-BEYER, M. Glcm texture: A tutorial v. 3.0. University of Calgary, 2017. Available: <https://prism.ucalgary.ca/items/8833a1fc-5efb-4b9b-93a6-ac4ff268091c>. Citation on page 36.

HAN, Y.; YANG, S.; CHEN, Q. Recognition and segmentation of complex texture images based on superpixel algorithm and deep learning. **Computational Materials Science**, v. 209, p. 111398, 2022. ISSN 0927-0256. Available: <https://www.sciencedirect.com/science/article/pii/S0927025622001690>. Citation on page 36.

HARAKANNANAVAR, S. S.; RUDAGI, J. M.; PURANIKMATH, V. I.; SIDDIQUA, A.; PRAMODHINI, R. Plant leaf disease detection using computer vision and machine learning algorithms. **Global Transitions Proceedings**, v. 3, n. 1, p. 305–310, 2022. ISSN 2666-285X. International Conference on Intelligent Engineering Approach(ICIEA-2022). Available: <https://www.sciencedirect.com/science/article/pii/S2666285X22000218>. Citation on page 35.

HELFER, G. A.; Victória Barbosa, J. L.; SANTOS, R. dos; da Costa, A. B. A computational model for soil fertility prediction in ubiquitous agriculture. **Computers and Electronics in Agriculture**, v. 175, p. 105602, 2020. ISSN 0168-1699. Available: <https://www.sciencedirect.com/science/article/pii/S0168169920304853>. Citation on page 35.

HINSEN, K. The nature of computational models. **Computing in Science & Engineering**, v. 25, n. 1, p. 61–66, 2023. Citation on page 29.

HOEKSTRA, A. G.; KROC, J.; SLOOT, P. M. Introduction to modeling of complex systems using cellular automata. In: ____. **Simulating Complex Systems by Cellular Automata**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 1–16. ISBN 978-3-642-12203-3. Available: <https://doi.org/10.1007/978-3-642-12203-3%5F1>. Citation on page 42.

JANI, A. An extension of schelling's segregation model: Modeling the impact of individuals' intolerance in the presence of resource scarcity. **Communications in Nonlinear Science and Numerical Simulation**, v. 85, p. 105202, 2020. ISSN 1007-5704. Available: <http://www.sciencedirect.com/science/article/pii/S100757042030037X>. Citation on page 48.

JIN, D.; LIU, D.; YANG, B.; LIU, J. Fast complex network clustering algorithm using agents. In: **2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing**. [S.l.: s.n.], 2009. p. 615–619. Citation on page 57.

KANEHISA, M.; SATO, Y.; KAWASHIMA, M.; FURUMICHI, M.; TANABE, M. Kegg as a reference resource for gene and protein annotation. **Nucleic Acids Res**, England, v. 44, n. D1, p. D457–62, Oct. 2015. Citation on page 63.

KASEREKA, S.; KASORO, N.; KYAMAKYA, K.; Doungmo Goufo, E.-F.; CHOKKI, A. P.; YENGO, M. V. Agent-based modelling and simulation for evacuation of people from a building in case of fire. **Procedia Computer Science**, v. 130, p. 10–17, 2018. ISSN 1877-0509. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops. Available: <http://www.sciencedirect.com/science/article/pii/S1877050918303569>. Citation on page 48.

KAZIL, J.; MASAD, D.; CROOKS, A. Utilizing python for agent-based modeling: The mesa framework. In: THOMSON, R.; BISGIN, H.; DANCY, C.; HYDER, A.; HUSSAIN, M. (Ed.). **Social, Cultural, and Behavioral Modeling**. Cham: Springer International Publishing, 2020. p. 308–317. ISBN 978-3-030-61255-9. Citation on page 51.

KHALIL, M. A.; FATMI, M. R. How residential energy consumption has changed due to covid-19 pandemic? an agent-based model. **Sustainable Cities and Society**, v. 81, p. 103832, 2022. ISSN 2210-6707. Available: <https://www.sciencedirect.com/science/article/pii/S2210670722001597>. Citation on page 48.

KUHN, M.; JOHNSON, K. **Applied predictive modeling**. [S.l.]: Springer, 2013. Citation on page 81.

LEMAîTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. **Journal of Machine Learning Research**, v. 18, n. 17, p. 1–5, 2017. Available: <http://jmlr.org/papers/v18/16-365.html>. Citation on page 82.

LEMPEL, A.; ZIV, J. On the complexity of finite sequences. **IEEE Transactions on information theory**, IEEE, v. 22, n. 1, p. 75–81, 1976. Citation on page 79.

LESKOVEC, J.; KREVL, A. **SNAP Datasets: Stanford Large Network Dataset Collection**. 2014. <http://snap.stanford.edu/data>. Citation on page 64.

LI, T.; ZHU, S.; OGIHARA, M. Using discriminant analysis for multi-class classification: an experimental investigation. **Knowledge and information systems**, Springer, v. 10, n. 4, p. 453–472, 2006. Citation on page 81.

LIMA, G. F.; MARTINEZ, A. S.; KINOUCHI, O. Deterministic walks in random media. **Phys. Rev. Lett.**, American Physical Society, v. 87, p. 010603, Jun 2001. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.87.010603>. Citation on page 56.

LO, S.; WANG, W.; LIU, S.; MA, J. Using agent-based simulation model for studying fire escape process in metro stations. **Procedia Computer Science**, v. 32, p. 388–396, 2014. ISSN 1877-0509. The 5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014), the 4th International Conference on Sustainable Energy Information Technology (SEIT-2014). Available: <http://www.sciencedirect.com/science/article/pii/S1877050914006395>. Citation on page 48.

LUQUE-CHANG, A.; CUEVAS, E.; CHAVARIN, A.; PEREZ, M. Agent-based image contrast enhancement algorithm. **IEEE Access**, v. 11, p. 6060–6077, 2023. Citation on page 48.

MA, X.; HE, Z.; YANG, P.; LIAO, X.; LIU, W. Agent-based modelling and simulation for life-cycle airport flight planning and scheduling. **Journal of Simulation**, Taylor & Francis, v. 0,

n. 0, p. 1–14, 2023. Available: <https://doi.org/10.1080/17477778.2023.2169643>. Citation on page 48.

MARRINK, S. J.; CORRADI, V.; SOUZA, P. C.; INGÓLFSSON, H. I.; TIELEMAN, D. P.; SANSOM, M. S. Computational modeling of realistic cell membranes. **Chemical Reviews**, American Chemical Society, v. 119, n. 9, p. 6184–6226, May 2019. ISSN 0009-2665. Available: <https://doi.org/10.1021/acs.chemrev.8b00460>. Citation on page 35.

MARSLAND, S. **Machine learning: an algorithmic perspective**. 2nd. ed. [S.l.]: CRC press, 2015. Citation on page 80.

MARTÍNEZ, G. J.; ADAMATZKY, A.; SECK-TUOH-MORA, J. C. Some notes about the game of life cellular automaton. In: ____. **The Mathematical Artist: A Tribute To John Horton Conway**. Cham: Springer International Publishing, 2022. p. 93–104. ISBN 978-3-031-03986-7. Available: <https://doi.org/10.1007/978-3-031-03986-7%5F4>. Citation on page 42.

MATKOVIC, F.; IVASIC-KOS, M.; RIBARIC, S. A new approach to dominant motion pattern recognition at the macroscopic crowd level. **Engineering Applications of Artificial Intelligence**, v. 116, p. 105387, 2022. ISSN 0952-1976. Available: <https://www.sciencedirect.com/science/article/pii/S0952197622003918>. Citation on page 35.

MERENDA, J. V. B. de S. **Reconhecimento de padrões em redes complexas**. Master's Thesis (Master's Thesis) — Universidade de São Paulo, 2023. Unpublished Master's Thesis. Citations on pages 30, 36, and 56.

MIRANDA, G. H. B.; MACHICAO, J.; BRUNO, O. M. Exploring spatio-temporal dynamics of cellular automata for pattern recognition in networks. **Scientific Reports**, v. 6, n. 1, p. 37329, Nov 2016. ISSN 2045-2322. Available: <https://doi.org/10.1038/srep37329>. Citations on pages 36, 54, 61, 64, 81, and 132.

____. Network analysis using spatio-temporal patterns. **Journal of Physics: Conference Series**, IOP Publishing, v. 738, n. 1, p. 012011, aug 2016. Available: <https://dx.doi.org/10.1088/1742-6596/738/1/012011>. Citations on pages 30, 55, and 79.

MITCHELL, M. **Complexity: A guided tour**. [S.l.]: Oxford university press, 2009. Citation on page 42.

MORDVINTSEV, A.; RANDAZZO, E.; NIKLASSON, E.; LEVIN, M. Growing neural cellular automata. **Distill**, 2020. Https://distill.pub/2020/growing-ca. Citation on page 41.

MUDIGONDA, S. P.; NÚÑEZ-CORRALES, S.; VENKATACHALAPATHY, R.; GRAHAM, J. Scheduler dependencies in agent-based models: A case-study using a contagion model. In: YANG, Z.; BRIESEN, E. von (Ed.). **Proceedings of the 2021 Conference of The Computational Social Science Society of the Americas**. Cham: Springer International Publishing, 2022. p. 56–70. ISBN 978-3-030-96188-6. Citation on page 50.

MÜLLER, A. C.; GUIDO, S. **Introduction to machine learning with Python: a guide for data scientists**. Sebastopol, CA: "O'Reilly Media", 2017. Citation on page 81.

NARAYAN, V.; MALL, P. K.; AWASTHI, S.; SRIVASTAVA, S.; GUPTA, A. Fuzzynet: Medical image classification based on glcm texture feature. In: **2023 International Conference on Artificial Intelligence and Smart Communication (AISC)**. [S.l.: s.n.], 2023. p. 769–773. Citation on page 36.

NEWMAN, M. **Networks: an introduction**. [S.l.]: Oxford University Press, 2010. Citation on page 37.

NEWMAN, M. E. The structure and function of complex networks. **SIAM review**, SIAM, v. 45, n. 2, p. 167–256, 2003. Citation on page 37.

NGUYEN, P. C. H.; CHOI, J. B.; UDAYKUMAR, H. S.; BAEK, S. Challenges and opportunities for machine learning in multiscale computational modeling. **Journal of Computing and Information Science in Engineering**, v. 23, n. 6, p. 060808, 05 2023. ISSN 1530-9827. Available: <https://doi.org/10.1115/1.4062495>. Citation on page 35.

NIEDERER, S. A.; LUMENS, J.; TRAYANOVA, N. A. Computational models in cardiology. **Nature Reviews Cardiology**, v. 16, n. 2, p. 100–111, Feb 2019. ISSN 1759-5010. Available: <https://doi.org/10.1038/s41569-018-0104-y>. Citation on page 35.

NRC, N. R. C. **A Framework for K-12 Science Education: Practices, Crosscutting Concepts, and Core Ideas**. Washington, DC: The National Academies Press, 2012. ISBN 978-0-309-21742-2. Available: <https://nap.nationalacademies.org/catalog/13165/a-framework-for-k-12-science-education-practices-crosscutting-concepts>. Citation on page 29.

OMAR, Y. M.; PLAPPER, P. A survey of information entropy metrics for complex networks. **Entropy**, MDPI, v. 22, n. 12, p. 1417, 2020. Citation on page 79.

OU, X.; PAN, W.; ZHANG, X.; XIAO, P. Skin image retrieval using gabor wavelet texture feature. **International journal of cosmetic science**, Wiley Online Library, v. 38, n. 6, p. 607–614, 2016. Citation on page 36.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citation on page 80.

PE'ER, G.; SALTZ, D.; FRANK, K. Virtual corridors for conservation management. **Conservation Biology**, v. 19, n. 6, p. 1997–2003, 2005. Available: <https://conbio.onlinelibrary.wiley.com/doi/abs/10.1111/j.1523-1739.2005.00227.x>. Citation on page 68.

RAHIM, A.; HOSSAIN, N.; WAHID, T.; AZAM, S. Face recognition using local binary patterns (lbp). **Global Journal of Computer Science and Technology**, v. 13, n. 4, p. 1–8, 2013. Citation on page 36.

RIBAS, L. C.; MACHICAO, J.; BRUNO, O. M. Life-like network automata descriptor based on binary patterns for network classification. **Information Sciences**, v. 515, p. 156–168, 2020. ISSN 0020-0255. Available: <https://www.sciencedirect.com/science/article/pii/S0020025519309144>. Citations on pages 30, 36, 55, and 132.

ROPELEWSKA, E. The application of image processing for cultivar discrimination of apples based on texture features of the skin, longitudinal section and cross-section. **European Food Research and Technology**, v. 247, n. 5, p. 1319–1331, May 2021. ISSN 1438-2385. Available: <https://doi.org/10.1007/s00217-021-03711-3>. Citation on page 36.

SAYAMA, H. **Introduction to the modeling and analysis of complex systems**. Open SUNY Textbooks, 2015. Available: <https://milneopentextbooks.org/introduction-to-the-modeling-and-analysis-of-complex-systems/>. Citations on pages 29, 36, 41, and 49.

SCABINI, L. F.; RIBAS, L. C.; NEIVA, M. B.; JUNIOR, A. G.; FARFÁN, A. J.; BRUNO, O. M. Social interaction layers in complex networks for the dynamical epidemic modeling of COVID-19 in brazil. **Physica A: Statistical Mechanics and its Applications**, v. 564, p. 125498, 2021. ISSN 0378-4371. Available: <https://www.sciencedirect.com/science/article/pii/S0378437120307962>. Citation on page 35.

SCHELLING, T. C. Dynamic models of segregation. **The Journal of Mathematical Sociology**, Routledge, v. 1, n. 2, p. 143–186, 1971. Available: <https://doi.org/10.1080/0022250X.1971.9989794>. Citation on page 48.

SHALIZI, C. R. Methods and techniques of complex systems science: An overview. In: **Complex systems science in biomedicine**. [S.l.]: Springer, 2006. p. 33–114. Citation on page 42.

SHWARTZ-ZIV, R.; ARMON, A. Tabular data: Deep learning is not all you need. **Information Fusion**, v. 81, p. 84–90, 2022. ISSN 1566-2535. Available: <https://www.sciencedirect.com/science/article/pii/S1566253521002360>. Citation on page 80.

SIEGENFELD, A. F.; BAR-YAM, Y. An introduction to complex systems science and its applications. **Complexity**, Hindawi, v. 2020, p. 6105872, Jul 2020. ISSN 1076-2787. Available: <https://doi.org/10.1155/2020/6105872>. Citation on page 29.

SILVA, T. C.; ZHAO, L. Case study of network-based supervised learning: High-level data classification. In: ____. **Machine Learning in Complex Networks**. Cham: Springer International Publishing, 2016. p. 207–240. ISBN 978-3-319-17290-3. Available: <https://doi.org/10.1007/978-3-319-17290-3_8>. Citation on page 54.

SOROUSHMEHR, S. M. R.; NAJARIAN, K. Transforming big data into computational models for personalized medicine and health care. **Dialogues in Clinical Neuroscience**, Taylor & Francis, v. 18, n. 3, p. 339–343, 2022. Available: <https://doi.org/10.31887/DCNS.2016.18.3/ssoroushmehr>. Citation on page 35.

STANLEY, H. E.; BULDYREV, S. V. The salesman and the tourist. **Nature**, v. 413, n. 6854, p. 373–374, Sep 2001. ISSN 1476-4687. Available: <https://doi.org/10.1038/35096668>. Citation on page 55.

SUTTIDATE, N.; PIDGEON, A. M.; HOBI, M. L.; ROUND, P. D.; DUBININ, M.; RADELOFF, V. C. The effects of habitat heterogeneity, as measured by satellite image texture, on tropical forest bird distributions. **Biological Conservation**, v. 281, p. 110002, 2023. ISSN 0006-3207. Available: <https://www.sciencedirect.com/science/article/pii/S0006320723001027>. Citation on page 36.

TUESTA-MONTEZA, V. A.; MEJIA-CABRERA, H. I.; ARCILA-DIAZ, J. Coleaf-db: Peruvian coffee leaf images dataset for coffee leaf nutritional deficiencies detection and classification. **Data in Brief**, v. 48, p. 109226, 2023. ISSN 2352-3409. Available: <https://www.sciencedirect.com/science/article/pii/S2352340923003451>. Citation on page 36.

VANFOSSAN, S.; DAGLI, C. H.; KWASA, B. An agent-based approach to artificial stock market modeling. **Procedia Computer Science**, v. 168, p. 161–169, 2020. ISSN 1877-0509. “Complex Adaptive Systems”Malvern, PennsylvaniaNovember 13-15, 2019. Available: <http://www.sciencedirect.com/science/article/pii/S1877050920304191>. Citation on page 47.

WALI, A.; SAEED, M. Biologically inspired cellular automata learning and prediction model for handwritten pattern recognition. **Biologically Inspired Cognitive Architectures**, v. 24, p. 77–86, 2018. ISSN 2212-683X. Available: <http://www.sciencedirect.com/science/article/pii/S2212683X18300203>. Citation on page 41.

WANG, L.-N.; WANG, K.; SHEN, J.-L. Weighted complex networks in urban public transportation: Modeling and testing. **Physica A: Statistical Mechanics and its Applications**, v. 545, p. 123498, 2020. ISSN 0378-4371. Available: <http://www.sciencedirect.com/science/article/pii/S0378437119319521>. Citation on page 37.

WANG, Y.; JIANG, L.; WANG, X. yu; CHEN, W.; SHAO, Y.; CHEN, Q. kai; LV, J. lei. Evidence of altered brain network centrality in patients with diabetic nephropathy and retinopathy: an fmri study using a voxel-wise degree centrality approach. **Therapeutic Advances in Endocrinology and Metabolism**, v. 10, p. 2042018819865723, 2019. PMID: 31384421. Available: <https://doi.org/10.1177/2042018819865723>. Citation on page 61.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of ‘small-world’ networks. **Nature**, v. 393, n. 6684, p. 440–442, Jun 1998. ISSN 1476-4687. Available: <https://doi.org/10.1038/30918>. Citations on pages 37, 41, and 62.

WAXMAN, B. Routing of multipoint connections. **IEEE Journal on Selected Areas in Communications**, v. 6, n. 9, p. 1617–1622, 1988. Citation on page 62.

WILENSKY, U.; RAND, W. **An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo**. [S.l.]: Mit Press, 2015. Citations on pages 30 and 51.

WILLS, P.; MEYER, F. G. Metrics for graph comparison: A practitioner’s guide. **PLOS ONE**, Public Library of Science, v. 15, n. 2, p. 1–54, 02 2020. Available: <https://doi.org/10.1371/journal.pone.0228728>. Citation on page 61.

WOLFRAM, S. Cellular automata as models of complexity. **Nature**, Springer, v. 311, n. 5985, p. 419–424, 1984. Citations on pages 30 and 41.

____. **A new kind of science**. [S.l.]: Wolfram Media Champaign, IL, 2002. Citations on pages 41, 44, and 46.

ZHAI, J.; QI, J.; SHEN, C. Binary imbalanced data classification based on diversity oversampling by generative models. **Information Sciences**, v. 585, p. 313–343, 2022. ISSN 0020-0255. Available: <https://www.sciencedirect.com/science/article/pii/S0020025521011804>. Citation on page 35.

ZHANG, J.; LUO, Y. Degree centrality, betweenness centrality, and closeness centrality in social network. In: **Proceedings of the 2017 2nd International Conference on Modelling, Simulation and Applied Mathematics (MSAM2017)**. Atlantis Press, 2017. p. 300–303. ISBN 978-94-6252-324-1. ISSN 1951-6851. Available: <https://doi.org/10.2991/msam-17.2017.68>. Citation on page 61.

ZHAO, J.; YU, H.; LUO, J.; CAO, Z. W.; LI, Y. Complex networks theory for analyzing metabolic networks. **Chinese Science Bulletin**, v. 51, n. 13, p. 1529–1537, Jul 2006. ISSN 1861-9541. Available: <https://doi.org/10.1007/s11434-006-2015-2>. Citation on page 63.

ZHOU, Q.; WOMER, F. Y.; KONG, L.; WU, F.; JIANG, X.; ZHOU, Y.; WANG, D.; BAI, C.; CHANG, M.; FAN, G. *et al.* Trait-related cortical-subcortical dissociation in bipolar disorder: analysis of network degree centrality. **The Journal of clinical psychiatry**, Physicians Postgraduate Press, Inc., v. 78, n. 5, p. 3831, 2017. Citation on page 61.

ZIELINSKI, K.; RIBAS, L. C.; MACHICAO, J.; BRUNO, O. A network classification method based on density time evolution patterns extracted from network automata. Elsevier BV, 2022. Available: <https://dx.doi.org/10.2139/ssrn.4289236>. Citations on pages 30, 36, 55, 81, and 132.

GLOSSARY

Agent-Based Model: An agent-based model is a computational simulation approach that represents systems by modeling individual entities (agents) and their interactions, allowing for the exploration of emergent behaviors and patterns arising from decentralized decision-making at the level of individual agents.

Celullar Automata: Cellular automata are discrete, computational models characterized by a grid of cells, each with a state that evolves over discrete time steps based on predefined rules determined by the states of neighboring cells.

Class: Target variable, in machine learning is the variable which we want to predict.

Classification: Classification is a specific subtask within pattern recognition. It involves categorizing or labeling data points into predefined classes or categories based on their attributes or features.

Classifier: A *classifier* is an algorithm that categorizes data into a predefined set of classes.

Dataset: A *dataset* is typically a collection of structured or unstructured data that is used to train a machine learning model, whereas a *database* is a structured collection of data that is designed for efficient storage, retrieval, and management. While a *dataset* can be stored in a database, the term *dataset* specifically refers to the data that is used for machine learning purposes.

Imbalanced data: A dataset that contains different number of samples in its categories.

Pattern Recognition: Pattern recognition is a field that encompasses the process of identifying patterns or regularities in data. These patterns can take various forms, including shapes, structures, features, or statistical distributions.

Structured data: Data that is organized in a well-defined format and has a predetermined schema, like relational databases, CSV files, etc.

Tabular data: Structured data represented in tabular form, in which each column represents a specific attribute or feature, and each row represents an individual data instance.

Unstructured data: Data that does not have a predefined format or organization, like images, videos, networks, etc.

ICMC USP
SÃO CARLOS