Cláudia Raquel da Rocha Eirado

Essays on economics using networked data sets and machine learning

Ensaios sobre Economia usando conjuntos de dados em rede e aprendizado de máquina

Brasilia

2025

Cláudia Raquel da Rocha Eirado

Essays on economics using networked data sets and machine learning

Ensaios sobre Economia usando conjuntos de dados em rede e aprendizado de máquina

Doctoral Thesis presented to the Postgraduate Program in Economics of the Department of Economics at the University of Brasília, as a partial requirement for obtaining the degree of Doctor in Economics.

Universidade de Brasilia – UnB

Faculdade de Administração, Contabilidade e Economia - FACE

Departamento de Economia - ECO

Programa de Pós-Graduação em Economia - PPGECO

Supervisor Dr. Daniel Oliveira Cajueiro

Brasilia

2025

Cláudia Raquel da Rocha Eirado
Essays on economics using networked data sets and machine learning

Ensaios sobre Economia usando conjuntos de dados em rede e aprendizado de máquina/ Cláudia Raquel da Rocha Eirado. – Brasilia, 2025-
157p. : il. (some color.) ; 30 cm.

Supervisor Dr. Daniel Oliveira Cajueiro

Tesis (Doctoral) – Universidade de Brasilia – UnB
Faculdade de Administração, Contabilidade e Economia - FACE
Departamento de Economia - ECO
Programa de Pós-Graduação em Economia - PPGECO, 2025.

1. Machine Learning for Solving Problems in Complex Networks: A Network Scientist's Perspective. 2. Indirect Contagion and Systemic Risk: A News Similarity Network Approach. 3. Digital Twins and Network Resilience in the EU ETS: Analyzing Structural Shifts in Carbon Trading. I. Supervisor Dr. Daniel Oliveira Cajueiro. II. Universidade de Brasília. III. Programa de Pós-Graduação em Economia. IV. Essays on economics using networked data sets and machine learning

Ensaios sobre Economia usando conjuntos de dados em rede e aprendizado de máquina

To my family, my husband Leonardo, and my children Mateus and Isabela, the great loves of my life.

ACKNOWLEDGEMENTS

I would first like to thank the Lord, as all glory belongs to Him. I wish to express my deepest gratitude to my family. To my husband, the love of my life, Leonardo, for being my greatest encourager, for his unconditional support and companionship. For the countless days and nights when I was working and studying, and he took care of our family with so much love.

I also want to thank my children, Mateus and Isabela, who have been loving, understanding, and always by my side.

I extend my sincere gratefulness to my parents, Nonato and Fátima, for their constant support, the efforts they made for me, and their lifelong love. To my brothers, Marcos and Jorge, my sisters-in-law, Fernanda and Alinne, and my nieces Júlia and Pamela, and nephews Rodrigo and Guilherme, for their care, concern, and support.

I am immensely grateful to my academic advisor, Dr. Daniel O. Cajueiro, for being a patient mentor throughout my academic journey. Thank you for your guidance, insightful advice, high expectations that pushed my work to a higher level, and your constant availability. It has been a great honor to be your advisee. I also thank my co-advisor, Dr. Douglas Silveira, for his patience, partnership, and willingness to help.

I am also deeply grateful to all my professors at UnB for sharing their vast knowledge with me and to the Department of Economics for being my academic home over the past five years.

Finally, I would like to thank my friends and colleagues, especially Carlos Piccioni and Thiago Trafane, for their partnership, time, and the countless problem sets we tackled together.

“All models are wrong, but some are useful.”
(George E. P. Box)

RESUMO

Este trabalho compreende três artigos em Economia utilizando conjuntos de dados em rede e aprendizado de máquina. No primeiro artigo, conduzimos uma revisão de aplicações de aprendizado de máquina para resolver problemas complexos de rede. Cobrimos conceitos de aprendizado de máquina, incluindo aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço, juntamente com métodos como clustering, incorporação e PCA. Além disso, exploramos conceitos de construção de rede e centralidade, abrangendo previsão de nós e links. O artigo também discute abordagens de linguagem natural, incorporando teorias do Processamento de Linguagem Natural (PNL). O segundo artigo investiga o conceito de risco sistémico no domínio financeiro, investigando o seu potencial para desencadear contágio indireto. Um aspecto fundamental da pesquisa envolve a aplicação de um modelo utilizando uma rede de similaridade de notícias para prever probabilidades estacionárias como proxy da centralidade na rede e nas relações entre empresas, estabelecendo uma relação entre elas e identificando caminhos de contágio indireto. Ao examinar as interações e a propagação do contágio entre empresas com base em artigos de notícias, o estudo visa descobrir insights sobre a interconectividade e os efeitos em cascata no sistema financeiro e se existe impacto em outros setores. O artigo conclui com uma discussão sobre as aplicações potenciais da IA e do ML na compreensão e previsão do risco sistémico no cenário financeiro. O terceiro artigo é um exercício empírico sobre Modelagem de Gêmeos Digitais aplicada ao Mercado de Carbono Europeu (EU ETS). Utilizamos as transações de EU-ETS para discernir padrões de interconexão entre países. Para atingir isso, construímos redes complexas para delinear relacionamentos entre nações, representando os caminhos de contágio, simulamos com Gêmeos Digitais a entrada e saída de novos agentes e o estabelecimento de novas conexões baseadas em análise preditiva utilizando modelos de Aprendizado de Máquina.

Palavras-chave: redes complexas, aprendizado de máquina, inteligência artificial, gêmeos digitais, risco sistêmico, contágio indireto, centralidade, EU ETS.

ABSTRACT

This work comprises three articles in Economics that utilize network data sets and machine learning. In the first article, we conduct a review of machine learning applications to solve complex network problems. We cover concepts of machine learning, including supervised learning, unsupervised learning, and reinforcement learning, along with methods such as clustering, embedding, and PCA (Principal Component Analysis). Additionally, we explore network construction and centrality concepts, addressing node and link prediction. The article also discusses natural language approaches, incorporating theories from Natural Language Processing (NLP). The second article investigates the concept of systemic risk in the financial domain, exploring its potential to trigger indirect contagion. A fundamental part of the research involves applying a model that uses a news similarity network to predict stationary probabilities as a proxy for network centrality and relationships between companies. The study establishes connections among companies, identifying pathways of indirect contagion. By analyzing interactions and the spread of contagion between companies based on news articles, the study seeks to uncover insights into interconnectivity and cascading effects within the financial system, as well as potential impacts on other sectors. The article concludes with a discussion of the potential applications of AI and ML in understanding and predicting systemic risk in the financial landscape. The third article presents an empirical exercise on Digital Twin Modeling applied to the EU Emissions Trading System (EU ETS). We use EU ETS transaction data to identify patterns of interconnection between countries. To achieve this, we build complex networks to outline relationships among nations, representing contagion pathways. Using Digital Twins, we simulate the entry and exit of new agents and the formation of new connections based on predictive analysis through machine learning models.

Keywords: complex networks, machine learning, artificial intelligence, digital twins, systemic risk, indirect contagion, centrality, EU ETS. .

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUTION

This work comprises three articles in Economics that utilize network data sets and machine learning.

Chapter 2 presents the first article, "Machine Learning for Solving Problems in Complex Networks: A Network Scientist's Perspective." This study reviews the applications of machine learning in addressing challenges within complex networks. It explores fundamental machine learning concepts, including supervised, unsupervised, semi-supervised, and reinforcement learning, along with key methodologies such as clustering, embedding, and Principal Component Analysis (PCA). The discussion extends to network construction, centrality measures, and predictive tasks such as node and link prediction. Additionally, the article integrates natural language approaches by incorporating theories from Natural Language Processing (NLP) and Deep Learning. While machine learning enhances scalability and interpretability, challenges persist in areas such as fine-tuning models, handling sparsity, and ensuring robustness. Future research can focus on improving generalization, reducing computational complexity, and integrating multiple techniques for more effective network analysis. Applying machine learning to complex networks strengthens the ability to analyze real-world systems, enabling data-driven decision-making across diverse domains, including fraud detection, economics, recommendation systems, biological interactions, ecological networks, infrastructure, and security.

Chapter 3 introduces the second article, "Indirect Contagion and Systemic Risk: A News Similarity Network Approach." This study develops a method for measuring systemic risk by constructing a network of firms based on news similarity, following the model proposed by Cajueiro et al. (2021). Using financial news articles from major media sources such as The New York Times, Reuters, Fox News, Financial Times, The Guardian, and CNN, the study examines how firms connect through media coverage. The dataset includes S&P 500 firms from 2020 to 2022 and employs NLP techniques to assess textual similarity between companies. A key aspect of the analysis involves leveraging network structures to estimate stationary probabilities as a proxy for firm centrality, allowing for the identification of indirect contagion pathways. The findings indicate that firms in the Financials sector with high centrality in the news similarity network exhibit greater exposure to financial shocks, reinforcing the role of public perception in systemic risk transmission. Moreover, firms with strong media-based connections do not always belong to the same sector, suggesting that financial contagion extends beyond traditional industry classifications. These insights highlight the value of tracking media-driven firm relationships as a tool for regulators and investors to assess systemic risk.

Chapter 4 presents the third article, "Digital Twins and Network Resilience in the EU ETS: Analyzing Structural Shifts in Carbon Trading." This study examines the structural evolution of the European Union Emissions Trading System (EU ETS), the world's largest carbon market and a cornerstone of EU climate policy. Using transaction and account data from the European Union Transaction Log (EUTL), which records all emissions allowance transfers, the study applies Digital Twins, complex network analysis, and machine learning to model emissions trading as a dynamic system. The results suggest that ongoing market fragmentation could disrupt price formation and reduce market integration, potentially impacting liquidity and compliance costs. Predictive modeling indicates that emerging trading barriers may hinder market efficiency, emphasizing the need for policymakers to evaluate whether existing mechanisms effectively sustain competition and emissions reduction targets.

In Chapter 5 we discuss our findings e perspectives.

2 MACHINE LEARNING FOR SOLVING PROBLEMS OF COMPLEX NETWORKS: A NETWORK SCIENTIST PERSPECTIVE

This chapter explores machine learning techniques for network analysis, focusing on building network structures, measuring node importance, and predicting structural changes. It examines centrality approximation methods that reduce computational costs while preserving ranking accuracy, embedding techniques that capture network structure for improved clustering and classification, and clustering algorithms that allow flexible community detection. The study also discusses link prediction methods that combine network topology with past interactions, reinforcement learning approaches that adapt community detection to evolving networks, and visualization techniques that simplify complex structures using dimensionality reduction. These methods apply to fraud detection, economics, recommendation systems, biological interactions, infrastructure, and security. While machine learning improves scalability and interpretability, challenges remain in fine-tuning, handling sparsity, and ensuring robustness. Future research can enhance generalization, reduce computational complexity, and integrate multiple techniques for more effective network analysis.

2.1 Introduction

Our work reviews how machine learning (ML) can be used to solve problems in complex networks. Complex network theory is well established, however, traditional algorithms and techniques can be computationally costly and inefficient, especially when dealing with big data. In this work, we offer a guide of ML techniques that can optimize and improve network-based data analysis, which addresses the main problems, tasks, and applications in the area: network construction, centrality, influence, node classification, community detection with modularity, node and link prediction, and visualization. The application of ML and complex networks improves these solutions.

Solving problems with complex networks and machine learning enhances the ability to understand real-world systems, making analysis more robust and enabling data-driven decision making across various domains. This set of techniques facilitates the modeling, analysis, and prediction of phenomena involving dynamic interactions and non-trivial structures. This framework provides a powerful approach to representing interconnected systems, such as social networks, supply chains, financial systems, biological networks, and urban infrastructure, as we will explore next.

Zanin et al. (2016) conceptualizes a complex network as a system represented by graph theory, encompassing boundaries, constituent parts, and relationships. The structure created by these interactions is referred to as the network topology. Describing the system, i.e., its collective behavior, becomes impossible when examining its isolated components. For this reason, researchers refer to them as complex networks. The authors explores the combination of complex network theory and data mining. By integrating the analytical techniques of data mining with the concepts from complex network theory, their methodology seeks to explore and extract valuable insights from intricate and interconnected datasets.

In our paper, there appears to be a resemblance to the topic covered in mentioned book, but in reality, our focus is distinctly different. In that work, the study revolves around the node relationships in a graph with physics statistics, dynamic models, and data mining tools from computer sciences. In contrast, here, we establish a connection between complex networks and machine learning. Our objective is to delve into the concepts of machine learning and demonstrate the application of these tools in constructing and interpreting complex networks, providing a comprehensive panorama and guide from a network scientist's perspective , utilizing more recently developed tools. Data Mining focuses on pattern discovery, while Machine Learning focuses on prediction and automation based on those patterns.

Silva & Zhao (2016) present in their book a comprehensive description of network-based machine learning. It's a comparable topic but a very distinct presentation, they offer a panoramic and introductory vision of methods of supervised learning, semi-supervised learning and unsupervised learning. Nonetheless, the natural evolution after 2016 does not appear. We incorporate more recent advances in models, integrating graph neural networks (GNNs), deep learning, and graph-embedding learning, which were not widely explored, and also describe how to use ML in the tasks present in the first paragraph.

Over the years, Network Representation Learning (NRL) has evolved significantly, incorporating various approaches to capture the structure and properties of complex networks efficiently. The domain of NRL progresses from basic dimensionality reduction to sophisticated deep learning architectures that dynamically adapt to changing graph structures. Initial efforts focus on dimensionality reduction techniques, such as factorization methods, Laplacian eigenmaps, and locally linear embedding (LLE), which lack scalability (Hamilton, Ying & Leskovec, 2017b; Goyal & Ferrara, 2018). Scalable embedding methods emerge with random walks, including LINE and DeepWalk, introducing more efficient network encoding techniques (Zhang et al., 2018). Graph neural networks (GNNs) and attention-based models revolutionize representation learning by incorporating end-to-end learning mechanisms, making it possible to dynamically update embeddings in evolving biological networks (Muzio, O'Bray & Borgwardt, 2021). Deep learning models (GCN, GAT) achieve the highest classification accuracy due to their ability to leverage node features and graph structures (Chen et al., 2020). Approaches such as DeepWalk and Node2Vec generate embeddings through simulation-truncated random walks, enabling the representations to capture both structural and contextual information. Other methods build on deep learning to enhance these embeddings by learning more expressive features. (Luo et al., 2022).

In the wake of this development, network embeddings emerge to improve the representations of nodes in a network, proposing methods that automatically learn and preserve certain properties of the graph. Embeddings include incorporation of characteristics from original networks, such as orientation and dynamics in the network, local neighborhood, and walking network, node attributes, group labels, and supervision labels (Wang et al., 2018). Additionally, Dalmia & Gupta (2018) asserts that network embedding (or representation) models are useful for applications such as node classification, link prediction, and recommendation. The authors suggest interpreting these node representations with the aim of understanding why a particular embedding model works better for certain graph mining tasks. The unsupervised node representation learning models considered in this study are DeepWalk, LINE and Node2Vec.

The field of evolutionary network analysis gain attention since networks frequently evolve, altering their characteristics over time. Aggarwal & Subbian (2014) provides an overview of the vast literature on graph evolution analysis and the numerous applications that arise in the web, social networks, communication networks, road networks, recommendations, news networks, bibliographic networks and biological networks. Graphs evolve over time through the continuous addition of new edges and the removal of existing ones. Evolutionary network analysis comprises two main categories: maintenance methods and analytical evolution analysis. Maintenance methods aim to continuously and incrementally preserve the results of the data mining process over time. In contrast, analytical evolution methods focus on directly quantifying and understanding the changes that occur in the underlying network, emphasizing the modeling of such changes. Many networks are dynamic rather than static, evolving over time. A dynamic graph emerges from changes in vertices (or nodes) and edges (or links). Research in this area primarily focuses on evolution models, graph similarity measures, anomaly detection in large network-based data, and clustering similar graphs (Bilgin & Yener, 2006). Spiliopoulou (2011) examine volatility in social networks by observing how these networks evolve over time. They define evolution as the changes that occur within the network across temporal intervals. For example, tracking community formation and dissolution helps researchers understand and anticipate social, economic, and behavioral patterns. Studying network dynamics is crucial because it addresses key questions such as: "Why do communities emerge or vanish?", "Why do they merge or split?", "When do these events occur?", and "What are the initial movements that indicate this trend?".

Another important area of application is recommender systems, which play a crucial role in predicting the preferences of users from large pools of potential items, thereby helping to mitigate information overload. A common challenge in this domain is data sparsity, for which embedding techniques offer an effective solution. Several recommendation models address this issue by relying on graph embeddings, including bipartite graph embeddings, general graph embeddings, and knowledge graph embeddings. Bipartite graph embedding captures direct interactions between users and items by representing them in a user-item bipartite graph. Techniques such as matrix factorization, Bayesian analysis, and deep learning extract meaningful relationships within these graphs. General graph embedding expands beyond direct user-item interactions by incorporating additional information, such as social networks, item-item relationships, and other contextual data. These embedding techniques capture complex relationships and higher-order proximities, leading to more refined recommendations. Knowledge graph embedding integrates external knowledge bases into the recommendation process. By embedding entities and relations from knowledge graphs, the system leverages rich semantic relationships between users, items, and attributes, enhancing recommendation quality. Despite these advancements, conventional recommendation models remain widely adopted. Interestingly, traditional models demonstrate superior performance in predicting implicit user-item interactions, highlighting a comparative weakness in graph embedding-based recommendation models for such tasks (Deng, 2022).

In software development, NetworkX stands out as a well-known Python library that provides a powerful framework for Complex Network theory. It implements a wide range of classical algorithms and includes some routines that can be considered machine learning methods (Hagberg, Swart & Chult, 2008). A newer library, sknet (Toledo, 2021), is a Python package designed for machine learning tasks in complex networks. It is compatible with scikit-learn (Buitinck et al., 2013) and NetworkX, but unlike NetworkX, it focuses on integrating Complex Networks and Machine Learning. sknet provides structures for unsupervised, supervised, and semi-supervised learning, a constructor for transforming data into complex network representations, and a set of utility functions to support other packages.

In the following, we review the main machine learning methods that tackle the problems involving complex networks, as we outline in the first paragraph of this introduction.

2.2 Machine learning

Machine learning (ML) is a subfield of artificial intelligence that focuses on developing algorithms capable of identifying patterns and making predictions based on data. Instead of relying on explicit programming, these algorithms adjust their behavior through experience, allowing them to adapt to new information. In the context of complex networks, ML methods offer computationally efficient alternatives to traditional approaches, which often struggle with large-scale data. ML algorithms are commonly referred to as models, as they transform data, extract structure, and generate predictions. Their growing importance stems from advancements in computational power and the increasing availability of large datasets from various sources, including social networks, biological systems, and financial markets.

Machine learning techniques can be broadly categorized based on how they process and learn from data (Domingos, 2015; Mitchell, 2019). This section is organized into five parts, each covering a distinct learning paradigm: supervised learning (2.2.1), unsupervised learning (2.2.2), semi-supervised learning (2.2.3), reinforcement learning (2.2.4) and embedding aproaches (2.2.5).

2.2.1 Supervised Learning

Supervised learning relies on labeled data, where each example is associated with a known outcome. The objective is to find a function $g(\theta, X)$ that approximates $y = f(X)$, where y is the target variable, X represents the input data, and θ denotes the model parameters.

Supervised learning encompasses two primary tasks: regression and classification. Regression applies when y is continuous, while classification is used when y belongs to a finite set of categories. Classification models are further divided into binary classification, where there are only two possible classes, and multinomial classification, which involves multiple categories.

The goal of supervised learning models is to learn patterns from labeled data and use these patterns to make predictions on new, unseen data. This process typically involves solving an optimization problem to determine the best parameter set θ.

In regression models, the training process often minimizes a loss function such as the mean squared error (MSE), given by

$$\min_{\theta \in \Theta} \sum_{i=1}^{n} (y_i - g(x_i, \theta))^2 \,, \tag{2.1}$$

where y_i is the observed outcome, and $x_i = [x_{i1}, \ldots, x_{iK}]'$ is the feature vector for observation i in a dataset of size n. Alternative loss functions, such as the mean absolute error (MAE), can also be used depending on the problem.

For classification problems, the cross-entropy loss function is commonly employed, particularly when the model outputs probability estimates. In the binary case, if $g(x_i, \theta)$ represents the predicted probability of a positive outcome, the optimization objective is

$$\min_{\theta} - \sum_{i=1}^{n} \left[y_i \log g(x_i, \theta) + (1 - y_i) \log(1 - g(x_i, \theta)) \right] , \tag{2.2}$$

where $y_i \in \{0, 1\}$ is the observed class label for the i-th instance.

Several approaches exist for implementing supervised learning models. Linear models, such as linear regression and logistic regression, assume a direct relationship between input features and the target variable. Tree-based methods, including decision trees (Breiman et al., 1984; Quinlan, 1986), random forests (Breiman, 2001), and gradient boosting (Friedman, 2001; Friedman, 2002; Chen & Guestrin, 2016), allow for more flexible, non-linear decision boundaries. Support vector machines (SVMs) (Cortes & Vapnik, 1995) seek to identify an optimal separating hyperplane, particularly useful for classification tasks. Neural networks, ranging from simple feedforward architectures to deep learning models, capture complex patterns in high-dimensional data. The history of neural networks is rich and has evolved over the decades. The beginnings of neural networks introduced the Perceptron, the first computational model inspired by the human brain (Rosenblatt, 1958). Then came Backpropagation, making the training of multilayer neural networks feasible and popularizing their application (Rumelhart, Hinton & Williams, 1986). Convolutional Neural Networks (CNNs) emerge, laying the foundations for Deep Learning (LeCun et al., 1989). The Long Short-Term Memory (LSTM) architecture solves the gradient vanishing problem in Recurrent Neural Networks (RNNs). LSTMs are widely used in tasks such as NLP, time series analysis, and speech recognition (Hochreiter & Schmidhuber, 1997). Afterward, deep networks efficiently train using unsupervised learning in successive layers, leading to the explosion of interest in Deep Learning (Hinton, Osindero & Teh, 2006). Finally, the Transformer replace recurrent networks with attention mechanisms, revolutionizing Natural Language Processing (NLP) and leading to models like BERT, GPT, and T5 (Vaswani et al., 2017).

Each of these methods can be adapted for both regression and classification problems, depending on their formulation and intended application.

2.2.2 Unsupervised Learning

Unlike supervised learning, *unsupervised learning* operates without labeled data, meaning only the input features X are available without predefined outcomes. The objective is to uncover patterns, relationships, or structures within the dataset without relying on explicit supervision. Unsupervised learning is widely applied in various analytical tasks. One of its most common applications is *cluster analysis*, which organizes unlabeled observations into groups based on shared characteristics. These methods help identify natural divisions within data, facilitating tasks such as market segmentation, anomaly detection, and community detection in networks. Another significant approach is *dimensionality reduction*, which aims to represent the data using fewer variables while retaining essential information. This is achieved by eliminating redundant features, transforming the data into a lower-dimensional space, or selecting representative observations. Reducing dimensionality improves computational efficiency and enhances interpretability while preserving meaningful structures in the dataset. A third category within unsupervised learning involves *association rule learning*, which identifies meaningful relationships between variables. This technique is commonly used in transactional datasets to determine how the presence of certain items correlates with others. Applications include recommendation systems, inventory optimization, and behavior analysis, where understanding item co-occurrence patterns can inform decision-making. Each of these methods contributes to extracting valuable structure from unlabeled data, making unsupervised learning essential for tasks where predefined labels are unavailable or costly to obtain.

2.2.2.1 Clustering

2.2.2.1.1 K-means

K-means clustering partitions a dataset of n observations into K clusters by assigning each data point to the nearest cluster center (MacQueen, 1967; Lloyd, 1982; Gnanadesikan, 2011). The algorithm iteratively updates the cluster assignments and centroids, as outlined in Figure 1:

```
procedure Lloyds
    Choose $K$ points as initial cluster centers
    while C changes do
        for all $x \in X$ do
            Assign $x$ to the closest cluster center $c_k$ using a distance metric
        end for
        for all Clusters $C_k$ do
            Update $c_k$ as the mean of all points in $C_k$
        end for
    end while
end procedure
```

Figure 1 – Lloyd's algorithm for K-means clustering.

Although K-means does not explicitly assume any probabilistic structure, it tends to work best when clusters exhibit similar spherical covariance structures. Selecting the appropriate number of clusters, K, is a crucial step in applying K-means. The Silhouette score (Izenman, 2008) is a widely used metric to assess clustering quality. Given a clustering C_K, the Silhouette score for the i-th data point is computed as

$$s_{i\,K} = \frac{b_i - a_i}{\max\{a_i,\, b_i\}},$$

where a_i is the mean intra-cluster distance, and b_i is the mean nearest-cluster distance. The score ranges from -1 to 1, with values near 1 indicating well-separated clusters, and values approaching -1 suggesting poor assignment.

2.2.2.1.2 Gaussian Mixture Models

Given a dataset $X = \{x_1, \ldots, x_N\}$ with N observations of a D-dimensional variable, the Gaussian Mixture Model (GMM) assumes that each data point x_n is generated from a mixture of K Gaussian components. The distribution of each observation is expressed as:

$$p(x_n) = \sum_{k=1}^{K} \pi_k p(x_n|\mu_k, \Sigma_k), \tag{2.3}$$

where π_k represents the mixing proportions, and $p(x_n|\mu_k, \Sigma_k)$ is the Gaussian density function defined as:

$$p(x_n|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \exp -\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1}(x_n - \mu_k). \tag{2.4}$$

Here, μ_k denotes the mean vector, Σ_k the covariance matrix, and $|\Sigma_k|$ the determinant of Σ_k. The model is estimated using the Expectation-Maximization (EM) algorithm, which introduces a latent variable d_n indicating the cluster assignment for each observation (Dempster, Laird & Rubin, 1977).

The EM algorithm alternates between two steps: the E-step computes the expected values of d_n using Bayes' rule, while the M-step updates the parameters μ_k, Σ_k, and π_k by maximizing the likelihood function. An observation is assigned to the cluster with the highest posterior probability.

The covariance matrix Σ_k defines the shape and orientation of clusters. Several constraints can be imposed to adjust its complexity (Banfield & Raftery, 1993; Celeux & Govaert, 1995; Gan, Ma & Wu, 2020): full covariance allows distinct ellipsoidal clusters, tied covariance enforces a common shape across clusters, diagonal covariance restricts orientation to coordinate axes, and spherical covariance assumes uniform variance in all directions.

2.2.2.1.3 Fuzzy C-Means (FCM)

Fuzzy C-Means (FCM) is a clustering method where each data point has a membership degree to multiple clusters rather than belonging to a single group (Bezdek et al., 1982; Bezdek, Ehrlich & Full, 1984). The objective is to minimize the function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \|x_i - c_j\|^2, \tag{2.5}$$

where u_{ij} represents the membership degree of data point x_i in cluster j, c_j is the cluster center, and m is the fuzziness parameter controlling the degree of overlap between clusters. The algorithm iteratively updates memberships and cluster centers until convergence, allowing for more flexible cluster assignments compared to traditional hard clustering techniques.

2.2.2.1.4 Single-Linkage (SL) Clustering

Single-Linkage (SL) clustering iteratively merges the two closest groups based on a similarity metric (Sibson, 1973). The process starts with each data point as an individual cluster, represented as a vertex in a disconnected graph. At each iteration, the algorithm determines the most similar clusters, denoted as G_1 and G_2, and merges them if their distance satisfies the threshold:

$$d_{\text{thr}} = \gamma \cdot \max(d_1, d_2), \tag{2.6}$$

where d_1 and d_2 are the average dissimilarities within G_1 and G_2, and $\gamma \geq 0$ is a parameter controlling the threshold strictness. The algorithm continues this merging process until all points form a single cluster or another stopping condition is met. By using pairwise distances, SL clustering builds a hierarchy of nested clusters that can be visualized in a dendrogram, offering an alternative approach to grouping data compared to centroid-based methods.

2.2.2.1.5 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) reduces high-dimensional data by projecting it onto a lower-dimensional space while retaining as much variance as possible (Pearson, 1901; Hotelling, 1933; Shlens, 2014). This transformation relies on identifying orthogonal axes, known as principal components, that capture the most significant variations in the dataset.

The original data matrix $X \in \mathbb{R}^{m\times n}$ is decomposed into two matrices: the principal component matrix $W \in \mathbb{R}^{m\times k}$ and the projection matrix $H \in \mathbb{R}^{k\times n}$, where $k \ll \min(m, n)$:

$$X \approx WH. \tag{2.7}$$

PCA applies Singular Value Decomposition (SVD) to factorize X:

$$X = U\Sigma V^T, \tag{2.8}$$

where U contains the principal components, Σ is a diagonal matrix of singular values representing variance, and V^T is the projection of the data. The top k components approximate the original dataset, capturing the dominant patterns.

The process involves standardizing the data, computing the covariance matrix C, extracting eigenvectors, and projecting the data onto the principal components:

$$C = \frac{1}{n}XX^T, \quad H = W^T X. \tag{2.9}$$

2.2.2.1.6 Non-Negative Matrix Factorization (NMF)

Non-Negative Matrix Factorization (NMF) approximates a non-negative data matrix X as a product of two lower-rank non-negative matrices (Lee & Seung, 1999):

$$X \approx WH, \tag{2.10}$$

where W contains basis components, and H represents the weight of each component. NMF solves:

$$\min_{W,H} ||X - WH||_F^2, \quad \text{subject to } W, H \geq 0. \tag{2.11}$$

The non-negativity constraint improves interpretability, making NMF suitable for applications like text analysis, image decomposition, and bioinformatics.

2.2.2.2 Natural Language Approaches

2.2.2.2.1 Vector Space Models

The vector space model represents sentences as numerical vectors, allowing for the measurement of semantic similarity and relevance. Each document in a collection of N_S sentences is encoded as a vector of dimension N_V, where N_V corresponds to the number of unique terms in the vocabulary. The purpose of this model is to extract and rank the most relevant sentences within a document. To formally define this model, we introduce the sentence-term matrix $\mathbf{M}_{\text{tfisf}}$, an $N_S \times N_V$ matrix that establishes relationships between terms and sentences:

$$\mathbf{M}_{\text{tfisf}} = \begin{array}{c} \\ s_1 \\ s_2 \\ \vdots \\ s_{N_S} \end{array} \begin{array}{c} \begin{array}{cccc} w_1 & w_2 & & w_{N_V} \end{array} \\ \begin{bmatrix} \omega_{1,1} & \omega_{1,2} & \cdots & \omega_{1,N_V} \\ \omega_{2,1} & \omega_{2,2} & \cdots & \omega_{2,N_V} \\ \vdots & \vdots & \cdots & \vdots \\ \omega_{N_S,1} & \omega_{N_S,2} & \cdots & \omega_{N_S,N_V} \end{bmatrix} \end{array} \tag{2.12}$$

where each row represents a sentence, each column represents a term, and $\omega_{j,i}$ quantifies the relevance of term i in sentence j.

The weight $\omega_{j,i}$ depends on three components. The local factor captures the term frequency within a sentence. The global factor measures the term's importance across the entire document. The normalization component adjusts for sentence length, ensuring comparability. The weight is computed as:

$$\omega_{j,i} = \frac{\widetilde{\omega}_{j,i}}{\text{norm}_j}, \tag{2.13}$$

where

$$\widetilde{\omega}_{j,i} = \begin{cases} f_{\text{tf}}(\text{tf}_{i,j}) \times f_{\text{isf}}(\text{sf}_i) & \text{if } \text{tf}_{i,j} > 0 \\ 0 & \text{if } \text{tf}_{i,j} = 0 \end{cases} . \tag{2.14}$$

In Eq. (2.13), norm_j normalizes sentence length. In Eq. (2.14), $f_{\text{tf}}(\text{tf}_{i,j})$ represents term frequency weighting, and $f_{\text{isf}}(\text{sf}_i)$ captures sentence frequency weighting. Table 1 lists common choices for f_{tf}, f_{isf}, and norm_j, adapted from Baeza-Yates & Ribeiro-Neto (2008), Schütze, Manning & Raghavan (2008), and Dumais (1991).

Term frequency	$f_{\text{tf}}(\text{tf}_{i,j})$
Binary	$\min\{\text{tf}_{i,j}, 1\}$
Natural (raw frequency)	$\text{tf}_{i,j}$
Augmented	$0.5 + 0.5\frac{\text{tf}_{i,j}}{\max_{i'} \text{tf}_{i',j}}$
Logarithm	$1 + \log_2(\text{tf}_{i,j})$
Log average	$\frac{1 + \log_2(\text{tf}_{i,j})}{1 + \log_2(\text{avg}_{w_{i'} \in d_j} \text{tf}_{i',j})}$
Sentence frequency	$f_{\text{isf}}(\text{df}_i)$
None	1
Inverse frequency	$\log_2\left(\frac{N_S}{\text{sf}_i}\right)$
Entropy	$1 - \sum_j \frac{p_{i,j} \log(p_{i,j})}{\log(N_S)}$ $p_{i,j} = \frac{\text{tf}_{i,j}}{\sum_j \text{tf}_{i,j}}$
Normalization	norm_j
None	1
Cosine	$\sqrt{\sum_i^{N_V} \widetilde{\omega}_{i,j}^2}$
Word count	$\sum_i^{N_V} \text{tf}_{i,j}$

Table 1 – Common TF-ISF weighting variants.

2.2.2.2.2 LDA

Latent Dirichlet Allocation (LDA) (Blei, Ng & Jordan, 2003; Blei, 2012) is a generative probabilistic model designed to uncover hidden thematic structures within a corpus. It assumes that each document is generated from a mixture of topics, where each topic is defined by a probability distribution over words. The model takes the vocabulary size w and the number of topics z as hyperparameters, with documents modeled as Dirichlet distributions over topics and topics represented as Dirichlet distributions over words.

The generative process follows three steps. First, a Dirichlet distribution α defines the document's topic distribution. For each word in a document, a topic is chosen from a multinomial distribution defined by θ. Once the topic is selected, the word itself is drawn from the topic-specific word distribution β. This iterative process allows the model to infer the underlying structure of the text, grouping words into semantically coherent topics.

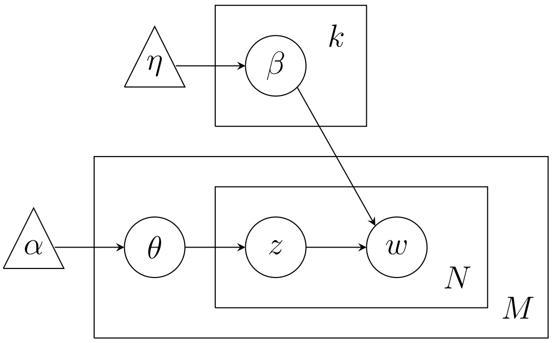


Figure 2 – LDA model: η represents the Dirichlet distribution of topics over the vocabulary, k denotes the topics, β corresponds to the multinomial distributions of words, α is the Dirichlet distribution of documents over topics, M is the corpus, N represents the documents, θ is the multinomial distribution of topics, z is the list of topics drawn from θ, and w consists of the words forming a generated document.

2.2.3 Semi-Supervised Learning

Semi-supervised learning (SSL) is a machine learning approach that integrates a small set of labeled data with a much larger set of unlabeled data. Initially, a model is trained on the labeled data, then used to predict labels for the unlabeled set. The model is retrained using only the high-confidence predictions, reinforcing stability against small input perturbations such as noise and transformations. Graph Neural Networks (GNNs) are commonly employed in SSL to propagate labels through a graph, where nodes represent data points and edges encode relationships. By leveraging both labeled and unlabeled data, deep learning models can enhance their predictive performance.

Deep learning and SSL are central to modern machine learning, particularly in cases where labeled data is scarce but a large volume of unlabeled data is available. Deep learning models, especially neural networks with multiple layers, learn hierarchical representations from raw data, automatically extracting relevant features. These models are widely used in applications such as image recognition, natural language processing (NLP), and graph-based learning. GNNs, introduced by Scarselli et al. (2008), Li et al. (2015), have led to more advanced architectures, including Graph Convolutional Networks (GCNs), Graph Sample and Aggregate (GraphSAGE), Graph Attention Networks (GAT), and Graph Isomorphism Networks (GIN).

A seminal study on semi-supervised classification using GCNs was presented by Kipf & Welling (2016). GCNs effectively capture both local and global graph structures, improving classification accuracy, particularly in scenarios with limited labeled data. These models encode graph structure and node features in a way that is well-suited for semi-supervised classification. Based on stochastic gradient descent (SGD), GCNs employ graph convolution operations to aggregate features from neighboring nodes, learning node embeddings through spectral graph convolutions. This smoothing process ensures that nodes in the same community share more similar representations.

The forward propagation rule for a GCN at layer l is defined as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{2.15}$$

where $H^{(l)}$ represents the node feature matrix at layer l, and $W^{(l)}$ is the trainable weight matrix for that layer. The matrix $\tilde{A} = A + I$ is the adjacency matrix with added self-loops, while $\tilde{D}$ is its corresponding diagonal degree matrix. The function σ applies a non-linearity, such as ReLU.

This formulation normalizes feature aggregation using the symmetric normalized Laplacian $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, ensuring stable training and improved convergence. By propagating information across multiple layers, the GCN model enables each node to incorporate multi-hop neighborhood features, capturing both local and higher-order structural information.

Graph Sample and Aggregate (GraphSAGE) (Hamilton, Ying & Leskovec, 2017a) is a graph neural network (GNN) model designed for inductive learning on large-scale graphs. Unlike traditional GNNs, such as Graph Convolutional Networks (GCNs), which require the entire graph for training, GraphSAGE generalizes to unseen nodes, making it efficient for dynamic and evolving graphs. At each layer l, GraphSAGE updates node embeddings using the following rule:

$$h_v^{(l)} = \sigma\left(W \cdot \text{AGGREGATE}(\{h_u^{(l-1)} : u \in \mathcal{N}(v)\})\right) \tag{2.16}$$

where $h_v^{(l)}$ represents the node embedding at layer l, and $\mathcal{N}(v)$ is the set of neighbors of node v. The AGGREGATE function can use different neighborhood aggregation strategies:

Aggregation Type	**Formula**
Mean	$\frac{1}{\lvert\mathcal{N}(v)\rvert}\sum_{u\in\mathcal{N}(v)} h_u^{(l-1)}$
LSTM	$\text{LSTM}(\{h_u^{(l-1)} : u \in \mathcal{N}(v)\})$
Max-Pooling	$\max\left(\{\sigma(W_{\text{pool}}h_u^{(l-1)}) : u \in \mathcal{N}(v)\}\right)$

Table 2 – GraphSAGE Aggregation Methods

The weight matrix W is trainable, and σ represents a non-linear activation function such as ReLU. GraphSAGE generates node embeddings by iteratively applying this aggregation process across multiple layers.

Graph Attention Networks (GAT) (Velickovic et al., 2017) introduce attention mechanisms into GNNs. Unlike GCNs, which treat all neighbors equally, GAT assigns different importance weights to neighbors through self-attention. This improves performance in heterogeneous graphs, where connections vary in significance. The update rule for the node representation at layer $l+1$ is:

$$h_v^{(l+1)} = \sigma\left(\sum_{u\in\mathcal{N}(v)} \alpha_{vu} W h_u^{(l)}\right) \tag{2.17}$$

where W is a learnable weight matrix, and α_{vu} is an attention coefficient computed as:

$$\alpha_{vu} = \frac{\exp\left(\text{LeakyReLU}(a^T[Wh_v \| Wh_u])\right)}{\sum_{j\in\mathcal{N}(v)} \exp\left(\text{LeakyReLU}(a^T[Wh_v \| Wh_j])\right)} \tag{2.18}$$

where a is a learnable attention vector, and $[\cdot\|\cdot]$ denotes concatenation. The softmax function normalizes attention scores. Multi-head attention extends this mechanism by averaging K independent attention mechanisms:

$$h_v^{(l+1)} = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{u\in\mathcal{N}(v)} \alpha_{vu}^{(k)} W^{(k)} h_u^{(l)}\right) \tag{2.19}$$

This enhances model stability and expressiveness, capturing multiple perspectives from neighboring nodes.

Graph Isomorphism Network (GIN) (Xu et al., 2018) is designed to match the expressiveness of the Weisfeiler-Lehman (WL) graph isomorphism test. Unlike GCNs and GraphSAGE, which rely on mean or max aggregation, GIN uses sum aggregation to distinguish different graph structures. The update rule for node representation at layer $l+1$ is:

$$h_v^{(l+1)} = \text{MLP}\left((1+\epsilon)\cdot h_v^{(l)} + \sum_{u\in\mathcal{N}(v)} h_u^{(l)}\right) \tag{2.20}$$

where ϵ is a learnable parameter that adjusts the influence of a node's own features. The function MLP$(\cdot)$ is a multi-layer perceptron applied to the aggregated representation. Sum aggregation ensures that GIN retains the same discriminative power as the WL test, making it robust for graph-level tasks such as molecular property prediction and social network analysis.

Training large-scale Graph Convolutional Networks (GCNs) poses challenges due to high computational costs and memory constraints. Cluster-GCN (Chiang et al., 2019) mitigates this by leveraging graph clustering. Instead of computing convolutions over the entire graph, Cluster-GCN partitions the graph into smaller clusters and trains on these subgraphs. Given a partitioned graph $\mathcal{G} = \{C_1, C_2, ..., C_K\}$, where each C_i is a cluster, the GCN update for a node v in cluster C_i at layer $l+1$ is:

$$H^{(l+1)} = \sigma\left(\tilde{D}_{C_i}^{-\frac{1}{2}}\tilde{A}_{C_i}\tilde{D}_{C_i}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right) \tag{2.21}$$

where $\tilde{A}_{C_i}$ represents the adjacency matrix of cluster C_i, $\tilde{D}_{C_i}$ is its diagonal degree matrix, and $W^{(l)}$ is the trainable weight matrix. The function σ applies a non-linearity such as ReLU.

By restricting computations to clustered subgraphs, Cluster-GCN reduces complexity and memory usage while preserving connectivity. This enables large-scale graph learning using stochastic gradient descent (SGD) while maintaining strong performance in deep graph networks.

2.2.4 Reinforcement Learning

Reinforcement learning (RL) focuses on mapping states or situations to actions to maximize cumulative rewards over time. This approach enables an agent to interact with an environment, take actions, and receive feedback in the form of rewards or penalties. According to Sutton & Barto (2018), a reinforcement learning system consists of four main elements beyond the agent and the environment: a policy, a reward signal, a value function, and, optionally, a model of the environment. A policy determines the agent's behavior by mapping observed states to actions. The reward signal provides immediate feedback on the desirability of an action, while the value function estimates the expected long-term reward from a given state. Rewards serve as the primary feedback, whereas value functions predict the cumulative reward to guide decision-making. A model of the environment, if available, enables the agent to simulate state transitions and plan future actions. Model-based methods use this information to optimize decisions, whereas model-free approaches rely on trial-and-error learning.

A trajectory in reinforcement learning is a sequence of states, actions, and rewards:

$$\tau = (S_0, A_0, R_0, S_1, A_1, R_1, \dots),$$

where S_t represents the state at time t, A_t is the action taken, and R_t is the reward obtained after executing A_t in state S_t. The initial state, S_0, follows a probability distribution ρ_0, defining the starting conditions:

$$S_0 \sim \rho_0(\cdot).$$

State transitions can be deterministic or stochastic. In a deterministic setting, the next state S_{t+1} is given by:

$$S_{t+1} = f(S_t, A_t),$$

whereas in a stochastic process, the next state follows a probability distribution:

$$S_{t+1} \sim P(S_{t+1} \mid S_t, A_t).$$

The reward function provides immediate feedback after an action A_t in state S_t, leading to S_{t+1}:

$$R_t = r(S_t, A_t, S_{t+1}).$$

Reinforcement learning problems are often modeled as Markov Decision Processes (MDPs), which facilitate dynamic programming techniques (Otterlo & Wiering, 2012). In an MDP, the probability of transitioning to the next state S_{t+1} depends only on the current state S_t:

$$P(S_{t+1} \mid S_t) = P(S_{t+1} \mid S_0, S_1, S_2, \dots, S_t),$$

and is defined by:

$$p(s' \mid s) = p(S_{t+1} = s' \mid S_t = s). \tag{2.22}$$

Several reinforcement learning algorithms have been developed over time. Monte Carlo methods estimate value functions and optimal policies by sampling entire episodes (Kalos & Whitlock, 2008). Temporal-Difference (TD) learning combines ideas from Monte Carlo methods and dynamic programming by adjusting estimates incrementally before an episode ends (Sutton, 1988). Q-learning is a widely used model-free algorithm that learns action-value functions without requiring a model of the environment (Watkins, 1989; Watkins & Dayan, 1992). SARSA (State–Action–Reward–State–Action) is another model-free approach, introduced by Rummery & Niranjan (1994), which refines Q-learning by incorporating policy updates based on the agent's current action selection.

Deep Q-Networks (DQN) extend Q-learning by using deep neural networks to approximate value functions (Mnih et al., 2015). Deep Deterministic Policy Gradient (DDPG) is an actor-critic method that extends Deterministic Policy Gradient (DPG) algorithms (Silver et al., 2014; Lillicrap et al., 2015). Actor-Critic methods, such as Advantage Actor-Critic (A2C) (Mnih et al., 2016), optimize policies by combining value-based and policy-based methods. The Asynchronous Advantage Actor-Critic (A3C) framework (Mnih et al., 2016) parallelizes policy updates, improving sample efficiency and stability.

Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) and Proximal Policy Optimization (PPO) (Schulman et al., 2017) refine policy gradient methods by improving training stability through constrained optimization. Twin Delayed Deep Deterministic Policy Gradient (TD3) enhances DDPG by mitigating overestimation bias using clipped double Q-learning (Fujimoto, Hoof & Meger, 2018). Soft Actor-Critic (SAC) further improves exploration-exploitation balance by maximizing expected rewards and policy entropy (Haarnoja et al., 2018). Distributional Soft Actor-Critic (DSAC) refines SAC by modeling reward distributions instead of expected values, improving robustness in complex environments (Duan et al., 2021; Ren et al., 2020; Duan et al., 2023).

Unlike supervised and unsupervised learning, reinforcement learning involves active decision-making where an agent interacts with an environment rather than relying on predefined training datasets. Instead of passively observing data, the agent continuously refines its strategy based on feedback, making reinforcement learning particularly well-suited for dynamic, uncertain environments where optimal actions must be discovered through experience.

2.2.5 Embedding Approaches

Embedding methods in machine learning map high-dimensional, categorical, or complex data into continuous, lower-dimensional vector spaces. This transformation enables algorithms to efficiently process and analyze data by converting sparse, high-dimensional representations into dense, compact forms. Such data may include images, text, sound, music, and other unstructured information.

In complex networks, graph embedding focuses on encoding structural information into a lower-dimensional space (Wandelt, Shi & Sun, 2020). Network embedding techniques vary based on their approach to preserving structural patterns, including random walks, traditional factorization, and neighborhood aggregation.

2.2.5.1 Random Walks

Random walks are not considered machine learning methods. However, a discussion of random walks is provided by Sarkar & Moore (2011), highlighting their applications in fields such as social network analysis, computer vision, personalized graph search, database keyword search, and spam detection. Random walks provide a versatile framework for integrating information from multiple paths between nodes, making them particularly useful for graph-based learning. However, developments in Machine Learning emerge from random walks such as Deepwalk and node2vec.

2.2.5.1.1 DeepWalk

DeepWalk (Perozzi, Al-Rfou & Skiena, 2014) is a method for learning latent representations of nodes in a network. These representations encode social relationships in a continuous vector space, which can be utilized by statistical models. The algorithm consists of two main components: a random walk generator and an update procedure. First, the generator samples a random vertex from the graph as the root of a random walk. Then, for each vertex, a random walk is generated, and the obtained sequences are used to update the node embeddings. The SkipGram algorithm (Mikolov et al., 2013) optimizes these embeddings by maximizing the likelihood of neighboring nodes appearing in the same sequence. Experimental results demonstrated that DeepWalk outperformed methods such as Spectral Clustering, Modularity, EdgeCluster, Weighted-Vote Relational

Neighbor, and Majority across datasets from social networks, including YouTube, Flickr, and BlogCatalog.

2.2.5.1.2 node2vec

The node2vec framework (Grover & Leskovec, 2016) provides a flexible and scalable approach to learning continuous feature representations for network nodes. By adjusting the parameters of biased random walks, it balances local (BFS-like) and global (DFS-like) graph structural properties. Inspired by techniques from natural language processing, node2vec optimizes embeddings while preserving both community structure and structural equivalence. The method follows three main steps: computing transition probabilities, simulating random walks, and optimizing embeddings using Stochastic Gradient Descent (SGD). The authors validated the effectiveness of node2vec on datasets from Facebook, Protein-Protein Interactions, and arXiv ASTRO-PH, showing improvements in capturing complex graph structures.

2.2.5.1.3 TemporalNode2Vec

TemporalNode2vec (Haddad et al., 2020) extends node2vec to dynamic graphs, addressing the limitation of traditional embeddings that ignore temporal evolution. Many real-world networks evolve over time, with nodes and edges appearing or disappearing. TemporalNode2vec incorporates time-dependent features to capture evolving network structures, ensuring that embeddings reflect dynamic proximities between nodes. The method accounts for short-term interactions and long-term relationships, improving tasks such as node classification, link prediction, and community detection.

The framework follows a structured process in steps. First, it preprocesses temporal graph data by tracking nodes and edges that change over time, representing the dynamic network as a sequence of snapshots or continuous updates. Second, it converts the graph into a format suitable for embedding, using adjacency lists, edge lists, or matrix representations while applying windowing techniques for time-series analysis. Third, it extracts temporal features such as timestamps and edge weights, applies temporal random walks to capture evolving structures, and defines transition probabilities that incorporate time dependencies. Fourth, node embeddings are generated using adapted algorithms, such as TemporalNode2vec, trained with optimization techniques like SGD or negative sampling to ensure they capture both structural and temporal dynamics. Finally, the learned embeddings are evaluated on tasks such as link prediction, node classification, or community detection, validating their ability to represent temporal changes in network structures.

2.2.5.2 Traditional Factorization

Traditional factorization methods are discussed in Section 2.2.2. In the context of advanced embedding techniques, several approaches have been developed to capture network structures effectively.

2.2.5.2.1 Locally-Linear Embedding (LLE)

Unlike traditional factorization, LLE preserves local neighborhood structures by assuming that each data point and its neighbors lie on a locally linear patch of the manifold. It constructs a weighted graph representing these local neighborhoods and computes low-dimensional embeddings that best maintain these relationships (Roweis & Saul, 2000).

2.2.5.2.2 Laplacian Eigenmaps

Laplacian Eigenmaps use the graph Laplacian matrix to obtain low-dimensional representations. The method constructs a graph where nodes represent data points, and edges capture similarities. The embedding is derived by solving the eigenvalue problem of the graph Laplacian (Belkin & Niyogi, 2001).

2.2.5.2.3 Graph Factorization

This approach decomposes a graph's adjacency or Laplacian matrix into lower-dimensional representations, preserving the inner-product structure of the graph. It is similar to traditional matrix factorization but specifically designed for graph structures (Ahmed et al., 2013).

2.2.5.2.4 Large-scale Information Network Embedding (LINE)

LINE preserves both first-order (local) and second-order (global) proximities in the network by optimizing separate objective functions for each and combining them. Unlike traditional factorization, LINE directly optimizes embeddings to retain specific types of proximities (Tang et al., 2015).

2.2.5.2.5 HOPE

HOPE captures high-order proximities in directed and undirected graphs by decomposing similarity matrices, such as Katz similarity or Rooted PageRank. It extends beyond first and second-order proximities, capturing more complex relationships (Ou et al., 2016).

2.2.5.2.6 NetSMF

NetSMF factorizes a matrix that approximates the random walk transition matrix, preserving the global network structure while scaling efficiently for large networks. It leverages sparse matrix factorization to handle large-scale networks effectively, unlike traditional factorization methods, which may struggle with scalability (Qiu et al., 2019).

Not all embedding methods, such as SVD, node2vec, and traditional factorization techniques, effectively capture the structural properties of real-world complex networks. Many low-dimensional representations fail to retain critical features such as node degree and link structure, missing essential network characteristics (Seshadhri et al., 2020). The primary finding of authors is that low-rank representations (specifically those based on dot products) cannot effectively model graphs with strong clustering and triangle structures, particularly for low-degree nodes.

An alternative approach is proposed by Gao et al. (2019), introducing the edge2vec model, which incorporates edge semantics into graph representations. Unlike traditional node embedding models that primarily focus on nodes, edge2vec explicitly integrates edge types into the embedding process, making it suitable for heterogeneous graphs where edges represent different types of relationships. The model constructs an edge-type transition matrix that captures transition probabilities between nodes via specific edge types. This matrix is optimized using an Expectation-Maximization (EM) approach to refine transition probabilities. Stochastic Gradient Descent (SGD) is then applied to learn node embeddings, ensuring that the resulting representations capture both node properties and edge semantics. Edge2vec has been validated on biomedical datasets, demonstrating superior performance in tasks such as biomedical entity classification, compound-gene bioactivity prediction, and information retrieval.

2.2.5.3 Neighborhood Aggregation

Neighborhood aggregation aggregates features from a node's neighbors to learn a new representation for the node. This method updates node features based on neighboring node attributes. Core techniques include Graph Convolutional Networks (GCN), GraphSAGE, and Graph Attention Networks (GAT), discussed in Section 2.2.3.

In graph embedding and semi-supervised learning, neighborhood aggregation is the fundamental operation in GCNs. Applications include user classification in social networks based on interactions and profile attributes, research paper classification using citation links and content features, and time-aware recommendation systems in e-commerce and social media.

GCN applies convolutions on graph data, aggregating information from neighboring nodes and updating feature representations. This is achieved by transforming neighboring feature vectors and applying non-linear activations. GraphSAGE is an inductive model that learns node embeddings by aggregating features from sampled subsets of neighbors, making it efficient for large-scale graphs and adaptable to unseen nodes. GAT introduces attention mechanisms to weigh the importance of each neighbor when aggregating features, allowing the model to prioritize more relevant neighbors. This enables GAT to capture heterogeneous relationships more effectively compared to uniform aggregation methods.

2.3 Network construction

To construct a network, it is essential to address two fundamental questions (Newman, 2018). First, who are the nodes? These represent the key actors within the network. Second, what are the edges? These are the connections and relationships between these actors. We typically align the selection of nodes and edges with the research problem under investigation. When the system naturally lends itself to a network representation, such as airline routes, the choices are straightforward (Sallan & Lordan, 2019). In this case, the nodes are airports, and the edges are the paths connecting them through at least one airline. However, when working with vector-based datasets, we may choose to represent the system as a network to model local relationships among data points and uncover global structures derived from these local interactions. By representing vector-based datasets as networks, we can also use network-based learning methods to extract deeper insights from the data.

Vector-based datasets usually consist of a set of features associated with each individual entity. From these features, we can build a network that connects the entities in a meaningful way. ML techniques, in particular, provide powerful tools for constructing complex networks, enabling us to model intricate relationships within the data. In general, the basic idea is to collect the similar items or items that belong to the same group given a given metric or model and to connect these entities in order to build a network.

In the remainder of this section, we describe techniques for measuring similarity from data. We may categorize these techniques into three main groups: *similarity-based techniques*, *clustering-based techniques* and *dimension reduction techniques*.

2.3.1 Similarity

We may define the similarity between two entities in a dataset as a scalar value that indicates how closely these entities are based on a specific criterion (Comin et al., 2020).

To use similarity to represent a network, we must first identify the actors (nodes) and establish a similarity relationship that will form the edges. However, a key question is to determine how similar two nodes (data points) are within a network. For continuous data, it is relatively straightforward to find similarity measures based on correlation, node distance, and their variations. However, for categorical data, similarity may not be as obvious. Textual data, for instance, are categorical and are widely available today in social networks, magazines, newspapers, and documents in general. The choice of similarity measure, which is formed by relationships between words, tokens, or even similar phrases - basically, content - can create different networks depending on the criteria and features used in the construction of the network. In practice, selecting the right attributes to characterize the content is far from trivial.

It is also important to note that the selection of the appropriate similarity measure depends on three key factors: the *type of data*, the *dimensionality of the feature space*, and the *significance of magnitude versus direction* in the data. The type of data, whether continuous or categorical, influences the choice of similarity measure. For example, distance measures are better suited for continuous data, while overlap measures are more appropriate for categorical data[1]. Another important factor is the dimensionality of the data. In high-dimensional spaces, data points tend to become more dispersed, and the concept of "distance" between points loses its relevance due to the "curse of dimensionality". In such cases, cosine-based measures may be more effective. Cosine measures are particularly useful in several important situations. They work well when the relative composition of features is more significant than their absolute values, or when we have preprocessed the data with scaling or normalization techniques, as they focus on direction rather than magnitude. Additionally, cosine measures are advantageous in sparse datasets, where most entries in the feature vectors are zeros. In such cases, cosine similarity is effective because the large number of zero-valued features does not affect its measure. The characteristics common in text data make cosine measures particularly appropriate. It's important to remember that in many text data models, the features of a text are the words that compose it, resulting in high-dimensional data. The focus is on the relative proportion of the words used rather than their absolute counts. Additionally, text data is usually sparse because, in a collection of texts, many words that appear in one group may not appear in another.

We typically classify similarity measures into three categories: *distance measures*, *cosine measures*, and *overlap measures*. It is important to note that although distance measures are not inherently measures of similarity but rather the inverse, we may determine how similar two items are based on the distance between them. To do this, we may set a threshold and consider two items similar if the distance between them is smaller than this threshold.

For example, using similarity (distance) measures, we can convert our vector-based dataset into a network. By setting a threshold, we may construct a network where a

[1] Although overlap measures are originally more suitable for dealing with categorical data, it is worth noting that there are now extensions of these measures for the continuous case (Costa, 2021b).

connection exists between two entities if the similarity (distance) measure is greater (smaller) than the threshold (Chi, Liu & Lau, 2010). A critical decision in this process is choosing the threshold that will determine whether to accept or reject connections between entities. The chosen threshold value can lead to either denser or sparser networks.

Although the concept of similarity spans multiple disciplines, it plays a central role in machine learning (ML), where models assess and quantify the degree of similarity or dissimilarity between pairs of data instances. This capability supports a wide range of applications, including recommendation systems (Singh et al., 2020; Liu et al., 2014), image recognition (Rahman, Bhattacharya & Desai, 2007; Deepak & Ameer, 2020), and anomaly detection (Schneider, Ertel & Ramos, 2016), where understanding inter-instance relationships is critical. Similarity-based network models convert non-relational datasets into relational structures by linking data points based on similarity, thus enabling the construction of networks that reflect the strength of these relationships. Whereas supervised learning predicts labels from individual data instances and unsupervised learning reveals latent structures in unlabeled data, similarity-based learning focuses on modeling relationships between pairs of instances. This approach always requires a reference pair to define similarity or difference. It offers distinct advantages by enabling tasks such as metric learning, ranking, and identity verification, which extend beyond the scope of standard supervised or unsupervised methods. By emphasizing pairwise relationships, similarity-based models capture local patterns and subtle distinctions that traditional methods often overlook—particularly those that ignore network topology or structural dependencies in data. In this paper, we present a comprehensive overview of similarity-based learning and its contributions to relational modeling and network-based representation. For further details on similarity measures and methodologies, we refer readers to Zadeh & Goel (2013), Aggarwal et al. (2015), Silva & Zhao (2016), Vijaymeena & Kavitha (2016), Shvydun (2023), Boriah, Chandola & Kumar (2008).

2.3.1.1 Distance measures

The most common way to define the distance between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is to use the L_p-norm as in

$$d_p(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{K} |x_i - y_i|^p \right)^{1/p}, \tag{2.23}$$

where K is the number of features of the set of vectors (Kolmogorov & Fomin, 1975). There are two special cases. When $p = 2$, we get the Euclidean distance, and when $p = 1$ we get the the Manhattan distance.

Another popular measure of distance is the Mahalanobis distance. Supposing, as before, that we use K features to characterize an entity. We may define Σ as the $K \times K$

covariance matrix of the data set. In this case, the (i, j)th entry of the covariance matrix is equal to the covariance between the dimensions i and j. Then, we may evaluate the Mahalanobis distance as

$$d_M(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y}).\Sigma^{-1}.(\boldsymbol{x} - \boldsymbol{y})^T}. \tag{2.24}$$

The Mahalanobis distance is similar to the Euclidean distance, except that it normalizes the data on the basis of the inter-attribute covariances.

2.3.1.2 Cosine measures

The Cosine similarity computes the cosine of the angle between two vectors, capturing their orientation rather than their magnitude (Salton, Wong & Yang, 1975; Schütze, Manning & Raghavan, 2008). We may define it as

$$\cos(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|}.$$

As previously mentioned at the start of this section, cosine similarity is a commonly used measure in text analysis. A typical application of this measure is in texts represented by space vector models, as discussed in Section 2.2.2.2. For example, Cajueiro et al. (2021) introduce a network based on news similarities to model indirect contagion. In their study, they use news articles about companies to measure similarities between them, employing cosine similarity as the measure.

Another important measure of similarity in this context is the correlation coefficient (Pearson, 1896), which is equivalent to cosine similarity for mean-centered vectors. To evaluate the correlation between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, we calculate the cosine similarity between the vectors after subtracting their means. We may find a well-known application of the correlation measure to build networks in Mantegna (1999). In this study, the authors derive a graph from the matrix of correlation coefficients computed between all pairs of stocks in the portfolio by considering the synchronous time evolution of the differences in the logarithm of daily stock prices.

2.3.1.3 Overlap measures

Focusing on vectors with qualitative information, the most simple overlap measure is to count the number of times two features arise in two vectors. Thus, we may evaluate the Overall similarity between two vectors as

$$S(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{K} b(x_i, y_i),$$

where K is the number of features and

$$b(x_i, y_i) = \begin{cases} 1, \ if \ x_i = y_i \\ 0, \ otherwise. \end{cases}$$

We may generalize the overall similarity by normalizing it in different ways.

The Inverse occurrence frequency generalizes it by replacing b by introducing a weighting mechanism that makes similarity more informative when dealing with categorical data of varying frequency distributions (Boriah, Chandola & Kumar, 2008). The logarithm function helps to penalize common categories more, reducing their contribution to similarity, with k atributtes.

$$f_I(x_i, y_i) = \begin{cases} 1, & \text{if } x_i = y_i \\ \frac{1}{1+\log f_k(x_i) \times \log f_k(y_i)}, & \text{otherwise} \end{cases}$$

The Goodall measure assigns a high similarity if the corresponding values are rare, regardless of the frequencies of other values. Let $p_k(x)$ be the fraction of records in which the kth attribute takes on the value of x in the data set. This measure replaces b by

$$f_G(x_i, y_i) = \begin{cases} 1 - p_k^2(x_i), \ if \ x_i = y_i \\ 0, \ otherwise. \end{cases}$$

The Jaccard Index (Jaccard similarity coefficient) measures the similarity between two vectors by comparing the number of their common features to the total number of features that arise in the two vectors (Jaccard, 1901). We may define it for two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ as

$$J(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{K} b(x_i, y_i)}{n_T},$$

where n_T is the total unique coordinate-category pairs in both vectors. The Jaccard Index ranges from 0 to 1, where 0 means no overlap and 1 means complete overlap. A higher value indicates more similarity.

The Sorensen-Dice Coefficient is similar to the Jaccard Index but gives more weight to the elements that overlap between the two vectors (Sorensen, 1948). We may define it for two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ as

$$D(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{K} b(x_i, y_i)}{d}.$$

Like the Jaccard Index, the Sørensen-Dice Coefficient ranges from 0 to 1. Thus, a higher value indicates more similarity.

As mentioned earlier, to maintain consistency with our goal of creating networks based on the similarity of vectors representing entities, we have explicitly defined the Jaccard Index and the Sorensen-Dice Coefficient by comparing the categorical coordinates of the vectors. Typically, these similarity measures are defined differently, usually based on the sizes of sets of features and their unions and intersections. For more details, see Aggarwal et al. (2015). It is also worth noting that we can use the Jaccard Index in its more conventional form to generate networks. For example, Wachs & Kertész (2019) applies the Jaccard Index to weight the connections of firms based on the similarity of their co-bidding behavior, which is part of a method to detect cartels in public auction markets.

2.3.2 Clustering

We may also employ clustering approaches to create networks from vector-based datasets. The construction involves using these techniques to transform a vector-based dataset into a network where each node represents an item, and edges constitutes relationships or similarities between these items. In order to avoid that the resulting network become disconnected, consisting of isolated clusters with no interconnections, the clustering technique should allow items to belong to multiple clusters simultaneously. The process begins with data preparation, where you start with a vector-based dataset in which each item is represented by a feature vector. This numerical representation captures the attributes of each item, making it suitable for clustering algorithms. The next step is to apply overlapping clustering. Choose an appropriate overlapping clustering method based on the specific characteristics of your data and the objectives of your analysis. By applying the selected method, you obtain cluster membership degrees or probabilities for each item, indicating the extent to which each item belongs to different clusters. After clustering, you define the network nodes by designating each item in the dataset as a node in the network. This establishes a one-to-one correspondence between data items and network nodes, forming the foundation of your network structure. To establish edges between the nodes, you have two options. The first option is to create an edge between two nodes if they share any common cluster. This means that if two items are both members of at least one cluster, they are directly connected in the network. The second option is to weight the edges based on the similarity of their cluster membership degrees, such as using cosine similarity measures. This approach not only connects nodes that share clusters but also quantifies the strength of their connection based on how similar their cluster memberships are. The resulting network is a more connected and intricate structure where items are linked based on shared characteristics.

There are several techniques for clustering that allow the items to belong to more than a group. The most common are the Gaussian Mixture Model and the Fuzzy C-Means discussed in Section 2.2.2.1. If we may characterize the items only based on textual data, another option is to use the LDA discussed in Section 2.2.2.2.2. Other clustering methods that we can use are hierarchical clustering and spectral clustering. Furthermore, Cupertino, Huertas & Zhao (2013) propose the use of an aglomerative clustering algorithm known as the Single-Linkage (SL) clustering heuristic (Sibson, 1973), discussed in Section 2.2.2.1, defending that the formed network is sparse and connected.

2.3.3 Dimension reduction techniques

In an approach similar to the construction of networks using similarity, dimension reduction techniques have a previous step: instead of applying the similarity directly in vector, we proceed with dimension reduction, and then apply the techniques in Section 2.3.1. Dimension reduction approaches are also useful to create networks from vector-based datasets. The idea is very similar to the one considered in the clustering approach to construct networks discussed in the Section 2.3.2.

Implementing dimension reduction for network construction involves several key steps that transform high-dimensional data into meaningful network representations. The process begins with data preparation, where you start with a high-dimensional vector-based dataset in which each item is represented by a feature vector. This numerical representation captures the essential attributes of each item, making it suitable for analysis and subsequent dimension reduction. The next step is to apply dimension reduction. We may choose an appropriate dimension reduction technique based on the nature of the data and the analysis objectives. We present some of these methods such as PCA and NMF in Section 2.2.2.1. Other methods are t-Distributed Stochastic Neighbor Embedding (t-SNE) (Maaten & Hinton, 2008), Uniform Manifold Approximation and Projection (UMAP) (McInnes, Healy & Melville, 2018), Isomap (Balasubramanian & Schwartz, 2002), and Diffusion Maps (Coifman et al., 2005).

After reducing the dimensionality, we proceed to construct the network. Each item in the dataset becomes a node in the network, establishing the fundamental elements of your network structure. To establish edges between the nodes, a common approach is to calculate the correlation or similarity between items' reduced representations, using measures such as cosine similarity or Euclidean distance, and connect nodes that exceed a certain similarity threshold. This approach allows to quantify the strength of the relationship between items and connect those that are sufficiently similar. The resulting network is one where items are connected based on their proximity or similarity in the reduced-dimensional space, effectively capturing complex relationships in the data.

2.4 Centrality

One of the key attributes of a node within a complex network is its centrality, which allows us to identify some special nodes. For instance, in banking networks, our focus is on identifying highly interconnected nodes, often referred to as "too connected to fail" (Gabrieli, 2012; Yun, Jeong & Park, 2019). In transportation networks, our interest lies in pinpointing central locations, critical for efficient network functioning (Sabzekar, Malakshah & Amini, 2023; Stamos, 2023). In electrical power networks, we aim to identify the critical stations to prevent widespread blackouts (Shuvro et al., 2019; Sami & Naeini, 2024). In ecological food chains, our concern is identifying the species whose extinction could have cascading effects on others, exploring strategies to minimize species extinction within these chains (Tang, Wang & Zhou, 2024; McDonald-Madden et al., 2016). In biological networks, such as molecular systems, the interconnectedness of molecular components typically involve the joint interaction of multiple molecules rather than individual factors (Wang, Wang & Zheng, 2022). The central or key node within a network holds significant importance as it influences other nodes and offers insights into the behavior, communication patterns and dynamics of the system structure (Rodrigues, 2018).

The centrality measures determine more important and relevant elements. By identifying central nodes, one can gain a better understanding of the dynamics of the network, its flow of information, and the pathways through which interactions occur. These central nodes serve as crucial hubs, have the role of connecting communities, and play a pivotal role to understand the network's connectivity.

In network science, various traditional methods are available for evaluating centrality, including local centrality measures like degree centrality (Shaw, 1954),Nieminen (1974) and Freeman et al. (2002) , and global centrality measures such as eigenvector centrality (Negre et al., 2018), pagerank centrality (Page et al., 1999), Katz centrality (Katz, 1953) , distance-based measures (betweenness (Shaw, 1954; Freeman et al., 2002) and closeness centralities (Sabidussi, 1966)), path-based and walk-based measures (Freeman, 1977; Newman, 2005), vitality (Restrepo, Ott & Hunt, 2006), and general feedback centrality (Koschützki et al., 2005), but often rely on predefined mathematical formulations and algorithms with high and complex computation, with large time to processing. With the rise of machine learning (ML), new data-driven approaches have emerged to enhance and extend centrality analysis in various networked systems. This section provides an overview of key ML techniques for centrality prediction and optimization, discusses their benefits over traditional methods. Alternatively, are employed approaches include supervised methods, unsupervised methods, reinforcement learning techniques and embedding aproaches.

2.4.1 Supervised methods

Supervised methods for calculating centrality often rely on approximation techniques, especially for massive networks containing millions or billions of vertices and edges.

In general, traditional centrality algorithms of weighted graphs operate in polynomial time. However, as networks grow to encompass millions or even billions of vertices, the processing time for these measures increases dramatically. Research indicates that computing the exact betweenness and closeness centrality in massive networks could take days, months, or even years (Bader et al., 2007; Wang, 2006; Cohen et al., 2014).

The supervised methods has four steps: the collection of the necessary training data, the application of a training algorithm such as one of those presented in Section 2.2.1, the analysis using an accuracy model, and the evaluation of the centralities using these best-fit regression model. A significant challenge when using supervised methods is the need for labeled data— specifically, training data that includes network centrality information. In these models as discussed in Section 2.2.1, we have the input features linked to the target variable, which must incorporate centrality labels from historical data to estimate the centrality of news vertices. However, obtaining such labels is often difficult, which is why applications of this approach are relatively rare. Proxy variables can capture the centrality in network and serve as target variables in the model.

In particular, Kumar, Mehrotra & Mohan (2015) propose the use of a feedforward neural network with error backpropagation training to estimate vertex centrality in social networks. The main objective is to predict the relative order of centrality between nodes, and not necessarily the absolute values. To this purpose, they build a simple architecture with four inputs, a hidden layer with two neurons and a single output. For input, they use features such as the number of nodes, number of edges, node degree, and the count of nodes within a two-hop reach. These attributes are chosen because they are easily computable and because they capture both the global size of the network and the local structure of the vertex. The choice of a 4-2-1 architecture aims to keep the model simple and avoid overfitting. During training, the authors use traditional algorithms like Power Iteration to calculate the exact eigenvector centrality (or PageRank) as supervised output, and evaluate the network's performance using the Pearson correlation between the predicted and actual values. The neural network can achieve correlations greater than 0.9 with the actual centrality values, while drastically reducing the calculation time compared to conventional methods. The Power Iteration method can be very slow, as the number of iterations required to converge depends on the network's structure. For large-scale networks, traditional centrality algorithms can become computationally infeasible due to their high time complexity. By using neural networks, the authors find these models are much faster in estimating the ordering of nodes based on centrality values, offer significant advantages in terms of speed, scalability, and computational efficiency, especially for large-scale social networks, making it a more practical solution for real-world applications.

Grando & Lamb (2015) propose neural networks and decision trees models to approximate complex centrality values efficiently training with input features of low complexity centrality measures (such as Betweenness, Closeness, Degree, and Eccentricity) to estimate target variable from high complexity centrality measures (Eigenvector Centrality, Information Centrality, Subgraph Centrality and Walk-based Betweenness). The data comes from the simulation of several complex networks modeled to reflect the characteristics of real-world social networks and other types of complex networks. The final dataset contains 5,765 networks, totaling 1,446,643 vertices and 39,655,102 edges. Similarly, Grando & Lamb (2016) use, in the training phase, degree and eigenvector centralities as input features, as they are easier to compute and provide relevant structural information about the network. The more complex centrality measures, Betweenness and Closeness, are target variables. The objective of the machine learning models was to approximate these time-consuming centralities based on the simpler input features. The authors generated 2,700 synthetic networks using the Block Two-Level Erdős and Rényi (BTER) model, which produces networks that closely resemble real-world networks. Also, regression models like neural networks (GNN) for approximating huge networks using the Block Two-Level Erdős and Rényi (BTER) model demonstrate superior performance and reduce computational costs compared to traditional approaches such as Linear Regression, Regression Trees, Support Vector Machines (SVM) (Grando, Granville & Lamb, 2018).

Hajarathaiah et al. (2024) also apply supervised learning models to estimate node importance in complex networks based on centrality measures. They use several real-world datasets, including U.S. airport connectivity, neural connections in a nematode worm, co-authorship networks in the field of network science, and a political blog network from the 2004 U.S. presidential election, which includes 643 blogs and 2,280 hyperlinks. The input features for the nodes in the networks include traditional centrality measures like Degree, Clustering Coefficient, Katz Centrality, and novel centrality measures like Local Relative change in Average Degree, Local Relative change in Average Clustering Coefficient and Local Relative change in Average Katz. Additionally, the model includes infection rate as an important feature to capture the node's ability to propagate information or infections across the network.This study defines the target variable as node significance, using simulation from epidemic models to determine it, specifically the SIR (Susceptible-Infected-Recovered) and Independent Cascade (IC). These models simulate the infection process and evaluate the true spreadability of a node. The labels for the nodes reflect the extent of their contribution to the epidemic spread. They use final scale of an outbreak associated with each node to assign labels and employ them to train the machine learning models. The paper concludes that machine learning techniques could effectively improve node significance identification, especially in propagation-based scenarios, by considering both local and global structural information and infection rates.

2.4.2 Unsupervised methods

Unsupervised methods in network centrality do not rely on labeled centrality scores, unlike supervised learning. This offers an advantage in cases where computational constraints, as discussed at the beginning of the Section 2.4.1, prevent the calculation of traditional measures in large-scale networks. Instead, it discovers patterns, structures, and hidden representations of nodes that can be used to infer centrality measures. The modeling process involves feature extraction, dimensionality reduction, clustering, and graph embedding.

Unsupervised methods encompass a broad range of models and tools, as discussed in Section 2.2.2. These approaches reduce computational costs and remove the need for labeled data, which makes them especially suitable for large and dynamic networks. To build an unsupervised model, one typically follows these steps: define the centrality type to estimate; prepare the graph data, including the adjacency matrix and, if available, node features; select an embedding method to project nodes into a low-dimensional space; train the model using the techniques outlined in Section 2.2.2; evaluate the resulting centrality estimates; optionally fine-tune the model to improve performance; and finally, apply the model to new graphs or unseen nodes for scalable centrality prediction.

For instance, Rakaraddi & Pratama (2021) propose Centrality using Unsupervised Learning (CUL), a method that ranks nodes by centrality without relying on labeled data—an advantage in settings where such information is unavailable or expensive to obtain. They design an encoder-decoder architecture that classifies nodes based on their structural relevance. The encoder, implemented as a Graph Neural Network (GNN), takes as input the graph structure (through the adjacency matrix) and node-level features (such as degree) to produce low-dimensional embeddings that reflect each node's position and structural role within the network. The decoder, a Multi-layer Perceptron (MLP), maps these embeddings to centrality scores. The authors compare CUL to a supervised baseline—Centrality with Supervised Learning (CSL)—which uses synthetic graphs labeled with Eigenvector Centrality (EC) values. Results show that CUL outperforms CSL in identifying high-centrality nodes and computes faster than conventional EC estimation methods.

Coppola & Elgazzar (2020) use a YouTube dataset of user interactions to detect communities, identify central users (influencers), and find maximal cliques. They first apply unsupervised clustering methods (Spectral Clustering and Louvain Modularity) to partition the network into communities. Then, they compute degree centrality, clique centrality, and a combined measure called "average rank" centrality, which balances a node's direct connectivity with its participation in densely connected groups. By comparing centrality scores within and across communities, they identify the most influential users both locally and globally. This combined use of community detection and centrality enables a more nuanced understanding of user roles in the network and proves useful in applications such as pinpointing niche influencers for marketing strategies.

2.4.3 Reinforcement learning

The idea of applying reinforcement learning to assess the centrality of nodes in complex networks comes from methods that model network characteristics using walkers within these networks. A walker is a conceptual entity that moves through the nodes of a network according to specific rules. Specifically, we may identify three different types of walkers: Random Walkers (RWs) (Noh & Rieger, 2004; Costa & Travieso, 2007; Xia et al., 2019), Directed Walkers (DWs) and Travelling Walkers (TWs). A RW is a walker that moves from node to node in a network by selecting one of the neighboring nodes at random in each step. They are useful in scenarios where we want to understand the overall structure of the network without a specific target. But, they are inefficient for reaching specific targets or optimizing travel paths due to the lack of direction[2]. A DW, on the other hand, uses additional information to navigate through the network more efficiently compared to random walkers (Tadić, Thurner & Rodgers, 2004; Liu et al., 2007). This additional information can come from local information (like node degree or edge weight) or global information (like shortest path or centrality measures). It is more efficient for targeted navigation, optimization, and scenarios where reaching a specific node or optimizing a path is important. But, DW requires additional information and computational resources to determine the best direction to move. A TW travels with an optimization goal, such as minimizing costs, distance, or time (Danila et al., 2007). This approach considers the overall network structure and uses sophisticated strategies to achieve efficient navigation.

The use of reinforcement learning to explore node centrality in complex networks typically focuses on variations of TW that minimize costs. In particular, Cajueiro (2009) extends traditional network centrality concepts, such as minimal access information, which evaluates the ease of accessing all other nodes from a specific node, and hide, which assesses how difficult we can locate a node from another random node within the network (Sneppen, Trusina & Rosvall, 2005; Rosvall et al., 2005). This adaptation depends on how he defines the costs for traveling a specific path. Moreover, Cajueiro (2010) applies the methods from Cajueiro (2009) to evaluate the centrality of two complex networks: the Boston subway network and the London rapid transit rail system. He also compares the centrality derived from the reinforcement learning paradigm with traditional centrality measures such as degree, closeness centrality, graph centrality, and betweenness centrality. Additionally, Cajueiro & Andrade (2009) presents a comprehensive framework using a first-visit Monte Carlo algorithm to identify and quantify progress in the learning paths process by the walker, exploring the difficulty of learning paths in complex networks. They show that this difficulty relates closely to the network's topology. He tests the method on random networks, scale-free networks, Apollonian networks, and four real-world networks.

[2] It is worth mentioning that there are more efficient variations of random walkers such as the ones that do not return to the node it situated at the previous step, that try to avoid walking in loops or tries to avoid revisiting the node that it has ever visited in a run of search (Yang, 2005).

2.4.4 Embedding approaches

Embedding methods for centrality tasks map nodes or edges into a low-dimensional vector space that preserves the network's structural properties. These representations simplify tasks such as centrality estimation, node ranking, and influence detection, especially in large and complex networks. Traditional centrality measures—such as degree, betweenness, or closeness—often rely on manual feature design and specific assumptions about network structure. In contrast, embedding approaches capture structural patterns automatically, offering scalable and more flexible alternatives.

Puzis et al. (2018) introduce Embedding Centrality (EmbC), a fully unsupervised method that estimates centrality based on node embeddings. Using a Word2Vec-style model (CBOW or skip-gram), they apply random walks to learn vector representations of nodes. The centrality score of a node corresponds to the dot product between its embedding and the center of mass of all embeddings in the network. This score reflects a node's overall affinity with the graph. The method produces results that fall between traditional measures like betweenness, closeness, and eigenvector centrality, while remaining adaptable across different types of networks.

Most embedding models only preserve first- or second-order proximities, which may fail to capture broader notions of importance tied to centrality. To address this, Chen et al. (2019) propose GraphCSC, a model that incorporates centrality information directly into a graph convolutional network. They modify neighbor sampling by prioritizing nodes with higher centrality (based on measures such as degree, betweenness, closeness, or PageRank). For each type of centrality, they construct separate embeddings and combine them using an attention mechanism that weights each view according to its relevance. This approach produces embeddings that integrate both local proximity and global importance.

Wandelt, Shi & Sun (2020) also explore how deep learning can approximate complex centrality measures. They apply a neighborhood aggregation model with Gated Recurrent Units (GRUs) to generate node embeddings, then estimate centrality scores using a neural network trained with a pairwise ranking loss. Rather than predicting exact centrality values, the model learns to preserve the relative ranking of nodes. This unsupervised approach allows fast approximation of centrality metrics—such as betweenness, closeness, eigenvector, and Katz—without computing them directly, which makes it suitable for large-scale networks.

Zou, Li & Luo (2024) propose the CNCA-IGE model (Complex Network Centrality Approximation using Inductive Graph Embedding) to efficiently approximate closeness and betweenness centrality rankings in large-scale networks. Rather than computing these metrics directly—which remains computationally expensive even with traditional approximation techniques—the authors reframe the problem as a learning task. CNCA-IGE combines inductive graph neural networks with an encoder-decoder architecture to predict centrality rankings. The encoder learns node embeddings from structural information, while the decoder maps these embeddings to centrality scores. For betweenness centrality, the model integrates an MLP-Mixer decoder to improve robustness and predictive capacity. The model trains on diverse synthetic and real-world networks using known centrality values and achieves strong performance while drastically reducing computation time. By learning to approximate complex rankings from network structure alone, CNCA-IGE provides a scalable alternative for centrality estimation in large or dynamic graphs.

2.5 Influence

In complex networks, influence refers to a node's ability to trigger and sustain diffusion processes—such as the spread of information, behaviors, or contagion—throughout the system. While centrality measures often identify prominent nodes, they fail to capture the influence of less central ones that can nonetheless initiate broad cascades (Lawyer, 2015). At the local level, influence reflects the directional effect from one node to another and depends on edge strength. Globally, network structure can assign disproportionate influence to certain nodes, regardless of their degree or centrality (Sun & Tang, 2011).

Influential nodes are not always central. In many networks, especially covert or strategic ones like terrorist organizations, centrality can mislead. Leaders may avoid central positions to conceal their role, maintaining influence through indirect control or coordination. Influence, in these cases, depends on access to resources, ability to mobilize others, or initiate actions, not on structural visibility (Xuan, Yu & Wang, 2014). Centrality-based methods often fail in such settings, as they prioritize highly connected nodes and overlook others with disproportionate impact (Zhu, Zhan & Li, 2023).

Traditional influence models—such as Information Diffusion (Matsubara et al., 2012), Influence Maximization (Kempe, Kleinberg & Tardos, 2003), and propagation-based models like Linear Threshold (LT) and Independent Cascade (IC) (Granovetter, 1978)—rely on manual feature design. These features vary by domain and often lack generalizability. A metric that performs well for retweet prediction on Twitter, for instance, may fail when applied to citation networks, where behaviors and structures differ. As a result, traditional approaches require costly redesign when applied to new contexts.

Machine learning offers an alternative by automatically learning influence patterns from data. These models adapt to varying network structures without manual intervention. Qiu et al. (2018) introduce DeepInf, a deep learning framework that predicts whether a user will adopt a behavior—such as retweeting or citing—based on the behavior of nearby nodes and the structure of their local network. The model receives a fixed-size subgraph centered on the user, including neighbor activity status and node-level features.

A pre-trained embedding layer converts each node into a vector. DeepInf then applies a Graph Neural Network (GNN) — either a Graph Convolutional Network (GCN) or a Graph Attention Network (GAT)— to capture influence patterns. GAT improves upon GCN by assigning attention weights, emphasizing more influential neighbors. The model outputs a binary prediction: whether the user will perform the action within a time window. Trained on real-world data from Twitter, Weibo, and Digg, DeepInf learns directly from network dynamics and outperforms traditional models like Logistic Regression (LR), Support Vector Machine (SVM), and PSCN. DeepInf-GAT, in particular, achieves the best results.

While DeepInf exploits local structure and attention mechanisms to improve prediction, it still assumes that the influence exerted by each neighbor is independent. This simplification overlooks important correlations in behavioral adoption across networks. Many existing models assume that influence probabilities across neighbors are independent. Luceri, Braun & Giordano (2018) challenge this assumption and develop a model that explicitly captures interdependencies among peers. Their framework recognizes that a user's behavior depends not only on which neighbors have adopted an action but also on the connections between those active neighbors. They formulate the problem as a supervised classification task and implement a deep neural network (DNN) to model these influence dynamics. The input comprises a one-hot vector representing the target user and a binary vector indicating which of their friends have adopted the behavior. These vectors pass through fully connected layers: lower layers identify local influence patterns, while upper layers learn global structural relationships. The model is trained on data from Plancast, an event-based social network that records both online interest and offline participation. Each observation corresponds to a user-event pair labeled according to actual attendance. By incorporating both positive and negative examples and learning from the underlying dependency structure, the model outperforms classical approaches such as Independent Cascade (IC) and Linear Threshold (LT), which rely on independence assumptions.

2.6 Node classification

In node classification, each node in the network belongs to one (or more) categories or classes. Given some nodes with known labels and the structure of the network, the goal is to predict the labels of the remaining nodes. This includes leveraging the network topology and node attributes to predict and understand the roles, behaviors, or characteristics of nodes within the network. Some applications are: to identify influential users or detect communities in social networks, to classify proteins based on interaction networks in biological networks, and to identify topic-specific authorities or detect anomalies in information networks. The classification involves methods algorithms such as Supervised

Learning, and Transfer Learning.

2.6.1 Supervised learning

In supervised learning, the goal is to train a classifier based on the examples of nodes that are labeled so we can apply it to the unlabeled nodes to predict labels for them. These methods use features and structures from the network, such as node attributes, graph topology, and centrality measures, to train classifiers that can predict node labels. However, a major challenge lies in the complexity of the graph structure itself, which encodes rich relational data that traditional classifiers often struggle to process. Furthermore, supervised models typically require a substantial amount of labeled data, which can be costly and time-consuming to obtain. These models may also fail to generalize to new graphs or to unobserved regions of the network in the absence of structurally similar instances. There is a rich literature on node classification, available in Zhao, Zhang & Wang (2021), Tang, Aggarwal & Liu (2016), Maurya, Liu & Murata (2022), Rong et al. (2019), Wang et al. (2020), Luan et al. (2021).

A foundational survey by Bhagat, Cormode & Muthukrishnan (2011) offers a comprehensive overview to existing approaches to node classification in social networks. The authors elucidate that traditional machine learning classifiers often rely on node features such as profile variables (e.g., age, location) to train models for label prediction. However, the authors discuss an approach that actively incorporates structural features from the graph to improve the accuracy of node classification. These structural features, such as proximity, degree, similarity, and paths between nodes, neighborhood label distributions, and connectivity patterns provide a richer context for classification tasks. The paper emphasizes the utility of incorporating properties of nearby nodes, suggesting that the labels of a node's neighbors can form a canonical structure that is predictive of the node's own label. This idea exploits the homophily principle, where nodes that are close or similar tend to share similar labels, as well as co-citation regularity, which holds when similar individuals tend to refer to or connect with the same entities. The study compares several methods for node classification. Feature-based methods rely on straightforward, interpretable features such as degree, neighborhood size, and shortest path distances to train classifiers like Naive Bayes or Decision Trees, though these may not capture the complex patterns inherent in network data. Alternatively, random walk-based methods propagate labels across the graph by simulating random walks; the model infers a node's label by iteratively propagating existing labels through the network and computing the resulting label distribution among its neighbors. It uses a transition matrix (like a Markov chain), where each entry p_{ij} represents the probability of going from node i to node j in one step. Over several iterations (or in the limit, when the process converges), we obtain a probability distribution over which labels are most likely for each node. For instance, in Label Propagation, at each step, each unlabeled node takes the set of distributions of its neighbors from step $t-1$ and takes their mean as its label distribution for step t. Furthermore, the authors design an iterative classification method that constructs node features by combining a node's own attributes with the labels of its neighbors and, if needed, those of more distant nodes in the graph. The iterative classification algorithm repeatedly updates feature vectors and applies a classifier to predict labels for unlabeled nodes, allowing the classifier to dynamically propagate and refine information through the network. Similarly, random walks for label propagation adjust labels over multiple iterations, enabling the labels to spread from labeled nodes to unlabeled nodes based on their proximity, thereby capturing indirect relationships and influence flows within the network. Overall, the work provides a comprehensive comparison of these approaches, demonstrating that incorporating structural and contextual information from the network can significantly enhance node classification performance compared to traditional feature-based methods.

The previous paper supports a hybrid approach that integrates node attributes and structural features, highlighting the value of both local information and network topology in node classification. In contrast, in a more recent development, Li & Pi (2019) introduce DNNNC (Deep Neural Network for Node Classification), a supervised learning model for node classification in complex networks. This model is attractive because it relies solely on the network structure and does not incorporate additional node features. This is, in fact, one of the central contributions of the article: demonstrating that it is possible to achieve superior performance in supervised classification tasks even without access to node attributes, using only the network topology. Unlike traditional approaches that separate the representation learning and classification stages, typically using network embeddings followed by classifiers such as SVM, DNNNC unifies these stages into a single end-to-end trainable architecture, thus avoiding suboptimal solutions. The methodology begins by constructing the Positive Pointwise Mutual Information (PPMI) matrix from the network adjacency matrix. This matrix captures co-occurrence patterns among nodes using a random surfing strategy inspired by the PageRank model, and it serves as input to a deep neural network composed of two stacked sparse autoencoders and a softmax layer. The model first employs unsupervised pre-training of the autoencoders to learn compressed, nonlinear structural representations of nodes. It then uses the softmax layer to train on the available node labels, completing a supervised learning process. Finally, it fine-tunes the entire architecture via backpropagation to jointly optimize feature extraction and classification. The authors evaluate DNNNC on three widely real-world datasets: BlogCatalog, Flickr, and Cora. Across all settings and metrics, including Macro-F1, Micro-F1, and accuracy, DNNNC consistently outperforms benchmark methods such as DeepWalk, Node2Vec, LINE, SDNE, DNGR, and several graph neural network models including GCN, GNNCheby, and SemiGCN. The model demonstrates stable convergence behavior, robustness to changes in key parameters, and competitive efficiency, especially on larger networks. The new model achieves its superior performance by leveraging the expressive power of deep neural networks to capture high-level nonlinear structural patterns. It also differentiates itself by not requiring node features, which is advantageous in scenarios where only topological information is available.

Also, in review, Bhagat, Cormode & Muthukrishnan comment that in some contexts, homophily does not hold, such as when there is intentional dissimilarity (for example, oppositely positioned videos being co-viewed on YouTube). Advances in the architectural design of neural networks have led to the maturation of new tools by Wu et al. (2022), such as NodeFormer, a Transformer-based scalable model for graph node classification, designed to overcome key limitations of traditional Graph Neural Networks (GNNs), like heterophily, where connected nodes often belong to different classes, long-range dependencies, incomplete or missing graph structures. The model operates in a supervised setting, taking node attribute matrices as input and optionally the adjacency matrix. It learns layer-wise latent graph structures with linear computational complexity, enabling all-pairs message passing even in large-scale graphs. The core innovation lies in the kernelized Gumbel-Softmax operator, which enables differentiable sampling of discrete graph structures while avoiding the quadratic cost of standard Transformers. The architecture combines message passing, relational bias, and edge regularization. Training minimizes a supervised classification loss in conjunction with a structure-aware regularization term. The paper evaluates the NodeFormer model using datasets from various domains. For node classification on real-world graphs, it uses Cora, Citeseer, Deezer, and Actor datasets. To test scalability on large-scale graphs, it employs OGB-Proteins and Amazon2M, which contain over 100K and 2 million nodes, respectively. In scenarios without explicit graph structures, the model uses Mini-ImageNet and 20News-Groups datasets to image and text classification tasks, where graphs are artificially constructed via k-NN based on node attributes. Empirical results demonstrate that NodeFormer outperforms both classical GNNs and state-of-the-art structure learning methods in accuracy, memory, and runtime efficiency and remains effective even without explicit input graphs. A Bayesian interpretation supports the model's ability to approximate optimal latent structures for the downstream task. Overall, NodeFormer represents a significant advance in graph learning, offering a robust, scalable, and accurate solution for node-level prediction.

2.6.2 Transfer learning

Transfer learning leverages knowledge from pre-trained models to boost performance on new tasks, particularly when data and computational resources are limited. In the realm of node classification, transfer learning and deep learning offer powerful strategies that exploit the inherent structure of graphs. Recent breakthroughs in graph-based machine learning and graph neural networks (GNNs) have transformed node classification by directly using the graph structure to learn meaningful representations. A key motivation for employing transfer learning in GNNs is to reuse knowledge acquired from a source task to improve performance on a target task—especially when labeled data for the target task are scarce. This concept already thrives in areas like image classification, where researchers fine-tune pre-trained models (such as those trained on ImageNet) for new challenges. Researchers explore how to extend these ideas to graph-based learning. In node classification tasks, transfer learning allows us to transfer knowledge from one graph (the source) to another (the target) with similar domains. An interesting application is on a social network like Facebook, and you can train a model to classify users based on their interests (sports, politics, music, etc.) using information like their profile, connections, and interactions. This model, by capturing general patterns of social behavior and community formation, can use the learning to apply it to another network like X, where connections and interactions have similar characteristics. With a small amount of labeled data from X, the previously model from Facebook can be fine-tuned to identify interests of users in the new graph. The other example refers to the context of citation networks, such as the Cora and PubMed datasets, each node represents an article, and the edges represent citations. Even though the topics in the articles may differ (e.g., computer science vs. medicine), the network structure and the way articles cluster into topic communities can be similar. A model can train to classify articles in Cora can classify articles in PubMed by transferring knowledge about how citation communities form. This approach adapts to new graphs with limited labeled data, which proves particularly valuable in real-world applications where obtaining labels is expensive or impractical. Successful transfer depends on proper fine-tuning and alignment of embeddings, especially when source and target graphs differ in structure or feature distribution. In summary, if you have plenty of labeled data, supervised learning may be sufficient. If not, Transfer Learning could be the solution.

Kooverjee, James & Zyl (2022) present a study on node classification using graph neural networks (GNNs) and a methodology to evaluate the performance of the model. Furthermore, they provide a procedure for generating synthetic datasets using a new synthetic graph classification method, called DANcer, with controlled community structure and attributes. Also, they use real data from Open Graph Benchmark (OGB) datasets, such as Arxiv and MAG (Microsoft Academic Graph) to apply the methodology. Furthermore, the authors introduced 'damaged' versions of the data, replacing the node attributes with Gaussian noise, to investigate whether the models could perform a transfer based solely on the graph structure, regardless of the attributes. GNNs can learn node embeddings that capture both node features and their relational context in the network, making them effective for node classification tasks in complex networks. The transferability of knowledge in GNNs depends on the similarity between the source and target tasks, particularly in terms of the community structure of the graphs. They test three GNNs: Graph Convolution

Networks (GCN), GraphSAGE, and Graph Isomorphism Networks (GIN), all of which demonstrated the ability to transfer knowledge effectively across training on the target task. The transfer learning framework using GNNs proves effective across both synthetic and real-world datasets in the context of node classification. On real-world data, GCN, GraphSAGE, and GIN successfully transfer knowledge between graphs with different structures and label distributions. Notably, even when node attributes are damaged, models like GIN and GCN maintain positive transfer, indicating that structural information alone can drive generalization. On synthetic data, the results show that transfer is particularly successful when the source graph exhibits strong modularity. GraphSAGE demonstrates the ability to exploit both structural and attribute-based properties, while GIN predominantly benefits from structural modularity. These findings confirm that the model generalizes well to both synthetic and real-world graph scenarios, and the interplay between network topology and node features influence its effectiveness.

2.7 Community detection

Community detection focuses on identifying groups of nodes that are more densely connected to each other than to the rest of the network. These groups, or communities, can provide insights into the network's structure and the functions of its components. Examples include social network analysis with discovery of communities into friends, recommendations for new friendships, identifying groups around topics, hashtags, biology with protein-protein interaction networks, recommendation systems in e-commerce, analysis to crisis management, etc.

The conventional ways to detect communities in complex networks are described on Fortunato (2010). Graph Partitioning divides the network into a predefined number of communities by optimizing a global criterion like minimizing the number of edges between communities using methods as Kernigham-Lin algorithm (Kernighan & Lin, 1970), spectral bisection (Barnes, 1982), and a range of algorithms as level-structure partitioning, geometric algorithm, and multilevel algorithms (Pothen, 1997). Hierarchical Clustering builds a hierarchy of clusters either by agglomerative (bottom-up) or divisive (top-down) approaches (Hastie et al., 2009). Modularity Optimization maximizes the modularity score, which measures the density of links inside communities compared to links between communities (Girvan & Newman, 2002; Newman & Girvan, 2004). Girvan-Newman have high complexity and do not scale well to networks with millions of nodes and edges. Many classical algorithms assume specific topologies (e.g., sparse or scale-free graphs) and may not generalize well to dynamic or heterogeneous networks.

An interesting concept associated with community detection is modularity. Modularity quantifies the strength of the community structure in a network by comparing the density of links inside communities versus between them. It serves as an optimization criterion for community detection algorithms, where higher modularity values indicate well-defined community structures. We define traditional modularity measures in Table 3. A higher modularity score indicates well-defined trading clusters. In modularity measure, Q represents the modularity score, which measures the difference between the actual density of links inside communities and the expected density in a random network. Higher values of Q indicate stronger community structure, meaning that nodes within the same group are more interconnected than expected by chance. The Louvain and Leiden methods iteratively merge nodes into communities to maximize modularity Q. The Leiden algorithm improves upon Louvain by ensuring better partition stability and faster convergence. Spectral clustering uses the eigenvectors of the Laplacian matrix to identify communities. The eigenvectors corresponding to the smallest nonzero eigenvalues capture key structural properties of the network, enabling clustering based on node proximity in the spectral space. This method effectively partitions graphs by projecting nodes into a lower-dimensional space and applying traditional clustering algorithms such as k-means. While spectral clustering is highly effective for small to medium-sized networks, it becomes computationally expensive for large-scale graphs due to the eigen decomposition step.

Measure	**Formula**
Modularity Q	$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$ [3]
Greedy Optimization (Louvain & Leiden)	$\Delta Q = \frac{1}{2m} \left[\sum_{in} \left(A_{ij} - \frac{k_i k_j}{2m} \right) - \sum_{tot} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \right]$ [4]
Spectral Clustering	$L = D - A, \quad L_{\text{norm}} = D^{-1/2} L D^{-1/2}$ [5]

Table 3 – Mathematical Formulas for Traditional Modularity Methods

Machine Learning significantly improves modularity-based analysis, especially in large, heterogeneous, or dynamic networks. Traditional methods like Louvain and

[3] In Modularity: Q is modularity score, $A_{i,j}$ is the adjacency matrix between nodes i and j, k_i and k_j are the degree of these nodes, m represents the total number of edges in the network, and $\delta(c_i, c_j)$ equals 1 if nodes i and j belong to the same community and 0 otherwise.

[4] In Greedy Optimization (Louvain & Leiden): ΔQ is the change in modularity when merging two communities, m is the otal number of edges in the network. A_{ij} is the adjacency matrix between nodes i and j. k_i, k_j are the degree of node i and j , representing the number of edges connected to it. $\sum_{\text{in}}$ is summation over edges that are inside a community and $\sum_{\text{tot}}$ is summation over all edges connected to nodes in the community.

[5] In Spectral Clustering: where L represents the unnormalized graph Laplacian and L_{norm} is the normalized Laplacian. The degree matrix D is a diagonal matrix where each element D_{ii} corresponds to the degree k_i of node i, representing the total number of edges connected to it. The adjacency matrix A is a square matrix where $A_{ij} = 1$ if there is an edge between nodes i and j, otherwise 0. The normalized Laplacian L_{norm} is computed by rescaling L with the inverse square root of the degree matrix, which ensures that node importance is balanced across different graph structures.

Spectral Clustering remain useful for small-scale applications, but ML models like GNNs, Node2Vec, and Reinforcement Learning improve scalability, accuracy, and adaptability. Advances and evolution in community detection are present in Lancichinetti & Fortunato (2009), Fortunato & Hric (2016), Zhang, Cui & Zhu (2020). Choosing between traditional and ML approaches depends on network size, structure, and the need for real-time adaptability. ML can contribute with models that help in understanding the structure and function in complex networks, and can incorporate multiply attributes like topology, centrality, textdata, metadata, evolution in time into community detection with Clustering, Embeddings techniques and Reinforcement Learning.

2.7.1 Clustering

Clustering plays a central role in community detection by identifying groups of nodes that interact more intensely with each other than with the rest of the network (Agrawal & Patel, 2020). This approach helps uncover the modular structure underlying many complex systems, where communities often correspond to functional units, interest groups, or regions of coordinated behavior. Clustering relies on principles of similarity or structural connectivity to group nodes, offering a way to simplify and interpret large-scale networks. While some clustering techniques originate from general unsupervised learning frameworks, others are specifically designed to detect communities in networked data, where structural dependencies and topological features are essential.

Several taxonomies have been proposed to classify clustering methods for community detection. One useful distinction lies between the direct application of general-purpose clustering algorithms, such as K-means or spectral clustering, and those explicitly designed to address structural characteristics of networks. Another important dimension separates methods that rely solely on topological information from those that incorporate additional attributes, such as node content, interaction metadata, or temporal patterns.

Classical clustering methods, though not originally designed for networks, have been widely adapted for community detection tasks. The Kernighan–Lin algorithm minimizes edge cuts while preserving balance across partitions, whereas hierarchical methods, such as agglomerative clustering and the Girvan–Newman algorithm, iteratively divide the graph based on edge betweenness. Spectral clustering embeds nodes using eigenvectors of graph Laplacians and applies traditional clustering in the embedded space. To improve scalability, multi-level approaches like Metis and Graclus perform graph coarsening and refinement, while MLR-MCL (Multi-level Regularized Markov Clustering) incorporates regularization to capture hierarchical structures more effectively. As network data have evolved to include rich node attributes, several models integrate structural and content-based information. The Group-Topic model combines link structure with textual features in a Bayesian framework to identify topic-coherent communities. The Community-User-Topic (CUT) model, in two variants, models either topics conditioned on users and communities (CUT1) or assumes that communities generate topics, which then shape user behavior (CUT2). The Community-Author-Recipient-Topic (CART) model applies this logic to email networks, jointly modeling senders, recipients, topics, and latent communities. In heterogeneous or multi-relational settings, methods such as NetClus and RankClus extend this generative approach. NetClus estimates posterior probabilities across multiple entity types, such as authors, conferences, and topics, in star-schema networks, while RankClus alternates between ranking inference and clustering in bi-typed networks. These methods, along with their extensions, represent a broad spectrum of clustering strategies that address the structural, semantic, and relational complexity of real-world networks (Parthasarathy, Ruan & Satuluri, 2011).

Temporal dynamics introduce further complexity into community detection. FacetNet addresses this challenge by allowing soft community membership and tracking how nodes shift between communities over time. The model balances two objectives: snapshot quality (the accuracy of community assignments at each time point) and temporal smoothness (consistency across consecutive snapshots). It achieves this by minimizing a Kullback–Leibler divergence-based objective function, enabling it to capture evolving community structures in dynamic networks.

Other approaches adapt traditional clustering techniques to networked contexts. Silva et al. (2016) apply the K-means algorithm to analyze the public transportation system of Curitiba, Brazil. They cluster bus stations based on geographic location and construct a complex network to examine regional accessibility. This integration of spatial clustering and network analysis highlights the potential for hybrid methods to reveal patterns of connectivity and mobility in urban systems.

Building on the limitations of K-means, particularly its sensitivity to initial center selection, Cai et al. (2019) introduce the DDJKM algorithm (Density-Degree centrality–Jaccard–K-means). This method selects initial cluster centers using a combined score of node degree and local density, promoting balanced and well-distributed seed points. To further improve separation, it applies the Jaccard similarity to prevent cluster centers from clustering too closely in the network. Empirical results show that DDJKM improves upon standard K-means by producing more accurate and stable community assignments in large-scale network datasets.

Oliveira et al. (2008) propose an angle-based clustering algorithm that defines similarity in terms of angular alignment between nodes. The algorithm groups nodes by minimizing angles within clusters and maximizing angular differences between clusters, particularly for nodes with few shared neighbors. This method identifies communities of varying density and shape, and supports multi-resolution exploration through flexible refinement. Comparative evaluations demonstrate its robustness and adaptability relative to baseline methods such as Single Linkage, Average Linkage, and K-means.

In the context of multiplex networks, where nodes interact through multiple types of relations, Amelio & Tagarelli (2018) adapt the silhouette coefficient to evaluate community structure. They introduce a multiplex version of the metric that considers both geodesic distance and homophily-based affinity. To address the computational cost of traditional silhouette calculations, their method selects a representative node for each community and computes distances only between this representative and other nodes. This adaptation reduces complexity while preserving the ability to assess intra- and inter-community cohesion effectively.

Chameleon, introduced by Karypis, Han & Kumar (1999), represents an early effort to combine inter-cluster connectivity and intra-cluster closeness in a hierarchical clustering framework. Unlike methods that rely exclusively on either density or distance, Chameleon uses a dynamic modeling strategy to evaluate the relative closeness and connectivity of candidate clusters. The algorithm builds a k-nearest neighbor graph and recursively merges clusters based on adaptive criteria, yielding flexible and structure-aware community assignments.

These clustering approaches reflect the diversity of techniques available for community detection, each grounded in different assumptions about network structure, node similarity, and data availability. General-purpose algorithms such as K-means and spectral clustering offer scalability and interpretability but often struggle to capture the complex topological dependencies inherent in networked systems. In contrast, network-specific methods, particularly those incorporating probabilistic modeling, node attributes, or temporal dynamics, provide more flexible and accurate representations of community structure, especially in heterogeneous or evolving networks. Choosing the appropriate clustering strategy depends on the network's characteristics, the availability of auxiliary data, and the specific goals of the analysis, whether descriptive, predictive, or explanatory.

2.7.2 Embedding-based community detection

Figure 3 illustrates a common pipeline for embedding-based community detection. The process begins with a graph, which is then transformed into a low-dimensional vector space through an embedding technique. In this space, node similarity reflects network structure, enabling the application of conventional clustering algorithms to identify communities. Embedding simplifies the original graph by preserving key properties such as proximity, homophily, and structural equivalence, while reducing dimensionality and allowing for efficient computation.

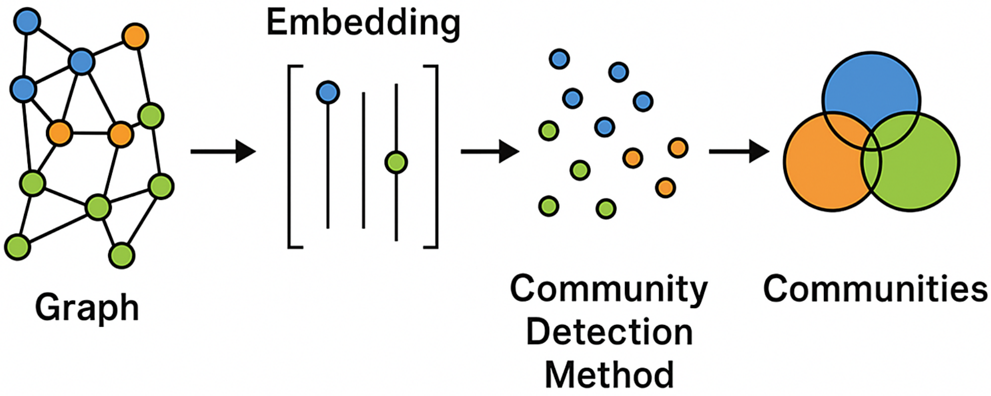


Figure 3 – Community Detection with Embedding Pipeline

The main advantage of embedding is that it captures network structure in a dense and continuous form, allowing machine learning models to operate directly on vectorized node representations. This improves the scalability of clustering, classification, and pattern recognition tasks by avoiding direct computations over sparse adjacency matrices.

Rozemberczki & Sarkar (2018) propose Diff2Vec (D2V), a sequence-based embedding method that constructs node embeddings from diffusion graphs. The algorithm generates node sequences via Euler walks, which preserve all adjacency relations in the subgraph in an efficient linear sequence. These sequences serve as input to a neural network that learns low-dimensional node representations. To detect communities, the authors apply k-means clustering in the embedding space and evaluate the results using modularity. D2V outperforms Node2Vec (N2V) in both efficiency and quality, particularly in preserving local proximity features. This work demonstrates that sequence-based embeddings rooted in diffusion processes yield better community detection performance than random walk–based methods.

Murata & Afzal (2018) propose a structurally supervised embedding method that combines graph convolutional networks (GCNs) with a modularity optimization objective. The model uses either standard GCNs or Chebyshev polynomials (ChebNet) to encode node representations from the adjacency and attribute matrices, aggregating neighborhood information at varying depths. To promote community-awareness in the learned embeddings, the authors incorporate modularity in two ways: as a regularization term in the output layer and as an auxiliary loss function in a separate layer. The network is trained using gradient descent to jointly optimize the classification loss (via cross-entropy on labeled nodes) and the modularity objective. This combination improves the quality of embeddings in semi-supervised settings, especially when labeled data are sparse, by encouraging the network to align with high-level community structure.

Zhu et al. (2021) present SENMF (Structural Equivalence Non-Negative Matrix Factorization), a similarity factorization approach that integrates node homophily and structural equivalence into a unified embedding framework. The method first constructs a similarity matrix that combines first- and second-order proximities, Dice coefficients, and role similarity measures. It then applies non-negative matrix factorization (NMF) to obtain low-dimensional node embeddings and soft community assignments. A modularity maximization term is included in the objective function to ensure that the resulting embeddings reflect cohesive community structure. The model uses alternating optimization to refine the embeddings iteratively, and final community labels are assigned using k-means. SENMF captures both local and global network patterns and outperforms several baseline models, including DeepWalk, Node2Vec, Walklets, GEMSEC, and M-NMF.

These models represent distinct families of embedding-based community detection techniques. Diff2Vec belongs to the category of sequence-based embeddings using diffusion processes. The GCN-based approach exemplifies structurally supervised embeddings that integrate modularity into the learning objective. SENMF is a similarity factorization method that combines homophily and structural equivalence into a low-dimensional representation. In each case, embeddings enable standard clustering algorithms to detect communities from geometrically meaningful patterns embedded in continuous space.

2.7.3 Reinforcement Learning

Reinforcement Learning (RL) offers a promising framework for optimizing community detection in dynamic networks. Unlike static approaches that apply a fixed algorithm to the entire network, RL allows an agent to interact with the network over time, learning to select the most effective detection strategies based on feedback from previous decisions. In each time step, the agent observes the current state of the network (e.g., its structure or changes in connectivity), chooses an action (e.g., a community detection method or parameter setting), receives a reward based on the quality of the resulting partition, and updates its policy accordingly. This feedback loop enables the agent to adapt its behavior as the network evolves.

In dynamic environments, RL provides two key advantages. First, it avoids recalculating the entire community structure from scratch by updating only the affected substructures, which improves scalability. Second, it allows the use of reward functions—such as density-aware modularity—that mitigate known issues in traditional modularity measures, including the resolution limit. While some earlier studies explored RL in static settings by combining modularity-based objectives with traditional community detection algorithms

(Paim, Bazzan & Chira, 2020; Martins & Zhao, 2020), their adaptability remains limited when applied to evolving networks.

Costa (2021a) propose a reinforcement learning approach that dynamically selects and combines community detection algorithms to maximize modularity over time. Their method formulates the problem as a Markov Decision Process and applies Q-Learning with SARSA (State-Action-Reward-State-Action) to guide the learning process. At each time step, the agent observes a state s, selects an action (i.e., a candidate community partition), and evaluates it based on a modularity-derived reward. The agent then updates its Q-function to favor community structures that improve the quality of the partition.

To manage changes in the network, the method updates only those parts of the community structure that are affected by new nodes or edges. It maintains an ensemble of candidate partitions and applies an extremal update mechanism: if a newly generated partition achieves higher modularity than the worst in the ensemble, it replaces the latter. This iterative process continues until no further modularity gains are observed. The final community structure is selected from the ensemble based on the highest modularity score.

Overall, reinforcement learning improves community detection by introducing adaptiveness, strategic decision-making, and modularity-aware optimization into the process. This makes RL-based approaches particularly well-suited for large-scale, dynamic networks where traditional static algorithms are insufficient.

The section below started with an interesting style of presenting methods for link prediction. How can we organize the methods and use them as examples? Do the ideas overlap much from one method to another? I think this can be a good guide for the discussion of the following section:

2.8 Node and link prediction

Node and link prediction play central roles in the analysis of networked systems. Link prediction estimates the likelihood that a connection between two nodes will form or has been omitted from the observed data (Getoor & Diehl, 2005; Lü & Zhou, 2011; Wang et al., 2014; Martínez, Berzal & Cubero, 2016; Daud et al., 2020). Researchers have applied this technique in several domains, including friend recommendation in social networks (Zhao & Zhao, 2024), personalized suggestions in e-commerce (Su et al., 2020), scientific collaboration forecasting (Resce, Zinilli & Cerulli, 2022), biological interaction mapping (Musawi, Roy & Ghosh, 2023), genetic risk prediction (Breit et al., 2020), web hyperlink creation (Adafre & Rijke, 2005; Han, Sun & Zhao, 2011), and record linkage in data integration tasks (Hasan & Zaki, 2011).

We can distinguish two broad categories of link prediction: static and temporal.

Static link prediction focuses on identifying missing links in a single network snapshot. Temporal link prediction, by contrast, uses time-stamped interaction data to forecast future connections. This distinction proves especially useful in dynamic environments, such as bipartite graphs in recommendation systems, where interactions between users and products evolve over time.

Node prediction refers to the task of identifying new entities that may join the network in the future. For instance, researchers may predict which users are likely to join a social platform or which proteins will appear in a biological interaction network (Sharan, Ulitsky & Shamir, 2007; Haslbeck & Waldorp, 2018; Rezaei et al., 2023). Although less frequently explored, node prediction often relies on the same methodological foundations as link prediction, especially when using graph-based learning approaches.

Earlier research often relied on heuristic scores such as common neighbors, Jaccard similarity, or the Katz index. These metrics evaluate topological proximity between node pairs but struggle to incorporate complex patterns or heterogeneous data. Supervised machine learning approaches offer a more flexible alternative. These methods frame the prediction task as a binary classification problem: given a pair of nodes, predict whether a link exists or will form. This setup requires researchers to label node pairs, extract relevant features, train a classifier, and evaluate its performance using metrics such as accuracy, precision, recall, F-measure, or ROC-AUC.

Several studies have proposed frameworks that improve different stages of this pipeline. Pecli, Cavalcanti & Goldschmidt (2018) investigate how automated feature selection can strengthen classification performance. They compare forward selection, backward elimination, and evolutionary strategies across six classifiers, including support vector machines (SVM), k-nearest neighbors (KNN), naïve Bayes, random forests, and multilayer perceptrons. Their results show that forward and evolutionary strategies lead to more accurate and compact models by removing redundant or irrelevant variables, which also reduces computational costs.

Building on the role of feature design, Hasan et al. (2006) construct a detailed framework for predicting co-authorship links. They develop three categories of features: proximity-based (e.g., keyword overlap), author-level (e.g., publication counts and number of coauthors), and structural (e.g., shortest path distance and clustering coefficient). By combining textual, statistical, and topological information, they improve prediction accuracy. Their experiments show that SVM with an RBF kernel performs best, and they use feature ranking to identify which variables contribute most to predictive success.

Lichtenwalter, Lussier & Chawla (2010) address a different challenge: class imbalance in large, sparse networks. They construct training datasets by labeling node pairs based on temporal windows and extract a rich set of degree- and path-based features. To reduce the bias toward negative samples, they implement neighborhood-based stratification and undersampling, training separate classifiers for different distance intervals. Their ensemble classifiers, especially random forests, consistently outperform traditional heuristics. They also introduce PropFlow, a localized random walk metric used as an unsupervised benchmark, and demonstrate that supervised models improve AUC scores by over 30%.

Ahmed, ElKorany & Bahgat (2016) extend this supervised learning framework to social media by incorporating behavioral and content-based features. Using Twitter data, they define a link as positive if a follow relationship appears within a future window and negative otherwise. They extract four categories of features: structural proximity (e.g., common neighbors), community similarity (based on modularity-based clustering), interaction intensity (mentions and replies), and trust (retweet patterns). They train several classifiers, including decision trees, SVM, logistic regression, and ensemble methods, and use random undersampling to balance the dataset. By focusing on user pairs within two-hop neighborhoods, they capture the most likely link formation areas. Their ensemble methods, especially RotationForest and Bagging, yield the best results.

These studies illustrate how supervised learning enables accurate and generalizable link prediction. Rather than relying on hand-crafted heuristics, these methods learn from labeled data, systematically combine multiple types of features, and adjust to structural and behavioral variation across domains. Feature selection, class rebalancing, and ensemble learning all contribute to their success. Together, these strategies provide a robust foundation for both link and node prediction tasks in complex networks.

2.8.1 Embedding-based link prediction

Node embeddings offer a compact and information-rich representation of network structure, enabling significant advances in link and node prediction tasks. By encoding both local and global topological patterns, such as community structure, structural equivalence, and role similarity, embeddings allow machine learning models to predict links more accurately than traditional heuristics like Common Neighbors or the Adamic-Adar index. Embedding methods avoid the need for manually designed proximity scores, instead transforming nodes into vectors that capture relational information. These low-dimensional vectors make classification more scalable, robust, and adaptable to sparsity and noise.

The typical embedding-based link prediction pipeline includes five steps. First, researchers prepare network snapshots, particularly in dynamic settings. Second, an embedding model generates vector representations of nodes. Third, a supervised classifier, often drawn from the techniques discussed in Sections 2.2.1, 2.2.2, or 2.2.3, learns to distinguish linked from unlinked node pairs. Fourth, the model undergoes evaluation using metrics such as ROC-AUC, precision, recall, or F1-score. Finally, the trained classifier predicts the likelihood of future or missing links based on the learned embeddings.

Hisano (2018) propose a semi-supervised embedding model that improves link prediction in dynamic networks by jointly modeling past link dynamics and current graph structure. Their framework combines a supervised component, which encodes past link formation and dissolution, with an unsupervised component based on random walks over the current network. The process begins by representing the dynamic network as a sequence of adjacency matrices across discrete time steps. The model constructs two past graphs: one for newly formed links and another for dissolved links. A complex-valued bilinear model maps these link dynamics to a latent space, assigning similar vectors to nodes with similar link histories. This supervised component uses a Hermitian inner product to capture both symmetric and asymmetric relationships.

To incorporate current network structure, the authors apply an unsupervised skip-gram model using DeepWalk. This component generates node sequences through random walks and learns embeddings that preserve contextual proximity. The final model optimizes a joint loss function that balances supervised prediction accuracy and unsupervised structural coherence. During training, stochastic gradient descent (SGD) simultaneously updates both components. The resulting embeddings predict future links using a sigmoid-transformed inner product. The model handles link formation and dissolution separately but also introduces a hybrid formulation to capture rewiring patterns, where nodes that frequently gain links also tend to dissolve older ones. Experimental results on four real-world datasets show that this semi-supervised approach outperforms purely supervised and unsupervised models, particularly in predicting link dissolution.

While Hisano (2018) focus on modeling historical dynamics and combining unsupervised embeddings with supervised training, Mallick et al. (2019) introduce a novel embedding method — Topo2Vec — designed specifically for link prediction in scale-free networks. Traditional embedding models often rely on random walk–based sampling, which may fail to capture key structural dependencies in social networks. Topo2Vec replaces random walks with a goal-oriented greedy sampling strategy, which more effectively explores edge relationships. This sampling process improves the training of neural embedding models, leading to more informative vector representations.

To perform link prediction, the authors develop an efficient pairwise feature generation technique that avoids computationally expensive pairwise kernels and supports scalable classification using Random Forests. They also explore clustering methods for embedding post-processing. Their evaluation across multiple datasets, including PPI networks, YouTube, Homo Sapiens, and BlogCatalog, shows that Topo2Vec consistently outperforms established methods such as LINE, node2vec, and GraphRep. Their findings highlight the benefits of guided sampling and efficient pairwise modeling for link prediction tasks in complex, heterogeneous networks.

Together, these studies demonstrate the versatility of embedding-based methods for link prediction. Whether through hybrid supervision or topological innovation, embeddings serve as a powerful abstraction of graph structure. They reduce the prediction task to a well-posed learning problem in vector space, allowing researchers to integrate temporal dynamics, structural roles, and domain-specific signals into scalable models.

2.9 Visualization

While visualizing small networks is relatively straightforward, larger and more complex networks present significant challenges. A high number of nodes and edges often leads to cluttered plots with substantial overlap, making interpretation difficult. To address this, network visualization techniques aim to uncover hidden structures, relationships, and clusters within complex datasets. These techniques are especially valuable in domains such as social networks, text analysis, and biological systems. The general approach involves reducing the dimensionality of the data. Common dimensionality reduction methods include Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF), and t-Distributed Stochastic Neighbor Embedding (t-SNE).

PCA, NMF, and t-SNE each offer distinct approaches to managing the complexity of large networks. PCA is a linear technique that projects data onto orthogonal axes capturing the greatest variance, thereby enabling the visualization of global network structure. However, it struggles with non-linear patterns. NMF, also a linear method, decomposes a matrix into non-negative factors, providing an additive and interpretable representation of network data. This makes it particularly useful for identifying hierarchical clusters and semantically meaningful latent relationships. In contrast, t-SNE is a non-linear technique that excels at preserving local relationships and revealing subtle structural patterns. However, it may distort global distances and is sensitive to hyperparameter configurations. The choice of technique depends on the type of structure one aims to highlight.

PCA simplifies network visualization by reducing data to its most significant components. In practice, network nodes and edges exist in high-dimensional space, which PCA reduces to two or three dimensions for visualization. This method is computationally efficient, making it suitable for large-scale networks (Borgatti & Halgin, 2011; Newman, 2003; Witten & Tibshirani, 2009; Albert & Barabási, 2002). Brandes & Wagner (2004) introduce Visione, a tool for visualizing social networks. Visione represents nodes as actors and edges as relationships, applying PCA to project high-dimensional graphs into two dimensions. For large graphs, Visione uses sampling and a heuristic for the k-center problem to reduce computational load. It adapts this method for directed edges by reserving embedding dimensions to represent directionality and factoring edge lengths into distance computations. These refinements increase the visual clarity of substructures, enabling more accurate and interpretable network representations. Although PCA offers computational advantages, it may yield components that lack intuitive interpretation and fails to capture complex non-linear structures.

NMF decomposes high-dimensional network data into two lower-dimensional, non-negative matrices W and H such that $V \approx WH$. This ensures interpretability and sparsity, facilitating effective network embedding and clustering. NMF-derived embeddings can be visualized in two or three dimensions and are also suitable for other network analysis tasks. Dias et al. (2017) combine NMF with graph matching to generate a hierarchical network representation. The method groups similar nodes using the topics derived from the NMF decomposition and defines a similarity metric to cluster nodes accordingly. The hierarchical clustering generated by this method outperforms traditional NMF and other classical clustering approaches. One limitation of NMF is its sensitivity to initialization, which may cause optimization to converge to local, suboptimal solutions.

The method t-SNE (Maaten & Hinton, 2008) maps high-dimensional similarities into joint probabilities and minimizes the divergence between these in both high and low dimensions. This preserves local neighborhood structures and captures non-linear relationships. Although computationally intensive and parameter-sensitive, t-SNE remains popular for generating interpretable network visualizations. Perplexity, a key parameter, balances local versus global structure: lower perplexity values emphasize local relationships, while higher values highlight broader patterns. However, t-SNE does not inherently incorporate graph structure. To address this, Kruiger et al. (2017) propose tsNET, a variant of t-SNE tailored to networks. tsNET integrates network-specific features such as edge weights into the dimensionality reduction process. It also improves computational efficiency, enabling visualization of larger networks without fine-tuning. By modifying the objective function, tsNET improves the representation of both global and local network structures.

Xiao, Hong & Huang (2023) examine perplexity optimization in t-SNE. They extend the algorithm to accept underestimated perplexity values and assess the impact on layout quality. They propose an estimation method for selecting optimal perplexity values and adapt their approach for use with the Barnes-Hut implementation (Maaten, 2014), achieving scalable visualization of large graph datasets. Leow, Laurent & Bresson (2019) further strengthen t-SNE through GraphTSNE, which incorporates both graph structure and node features. Using a Graph Convolutional Network (GCN), GraphTSNE learns a non-linear mapping from high-dimensional data to low-dimensional space. It employs a modified t-SNE loss that combines a graph clustering component and a feature clustering component. This allows for better balance between preserving local detail and global structure. Although computationally intensive, GraphTSNE provides a flexible solution for complex network visualization where both topological and feature-based similarities are important.

In summary, PCA offers a fast linear approach ideal for identifying broad trends; NMF emphasizes interpretability and latent semantic structure; and t-SNE and its variants, including tsNET and GraphTSNE, excel at revealing fine-grained patterns and network communities, especially in the presence of non-linearities and high dimensionality.

2.10 Final Remarks

This chapter has explored key methods for constructing networks, assessing node importance, and predicting structural changes, emphasizing the role of machine learning in advancing these tasks. Learning-based approximations of centrality measures, for instance, allow researchers to analyze large-scale networks by significantly reducing computational costs while preserving ranking accuracy. Embedding techniques, such as *node2vec* and *GraphSAGE*, encode local and global structural information into compact representations. These embeddings improve performance in clustering, node classification, and link prediction tasks by providing feature-rich inputs to downstream models.

Clustering algorithms — including Gaussian Mixture Models, hierarchical clustering, and community detection methods — group structurally or functionally similar nodes, often enabling overlapping memberships that better reflect the complexity of real-world networks. In link prediction, models that combine network topology with historical interactions successfully anticipate future connections, proving useful in diverse applications such as fraud detection, recommendation systems, and social influence modeling.

Reinforcement learning extends these approaches to dynamic settings, where algorithms adapt to changes in network structure and optimize partitioning strategies over time. Similarly, advances in visualization, through techniques like PCA, NMF, and t-SNE, enable interpretable representations of complex graphs by projecting high-dimensional structures into low-dimensional spaces. Extensions such as tsNET and GraphTSNE further improve layout quality by incorporating edge structure and node attributes, balancing local and global preservation.

These methods provide widespread application in economics, infrastructure systems, biology, cybersecurity, and digital platforms. While machine learning contributes to scalability, adaptability, and interpretability, challenges remain. Fine-tuning model parameters, addressing sparsity in real-world networks, and ensuring robustness across domains are persistent obstacles. Future research should may benefit from hybrid frameworks that integrate multiple techniques to reduce computational complexity and improve generalization to support more reliable and flexible network analysis.

3 INDIRECT CONTAGION AND SYSTEMIC RISK: A NEWS SIMILARITY NETWORK APPROACH

This chapter presents a novel approach to systemic risk analysis by constructing a network of firms based on news similarity. Using financial news articles from major media sources, including The New York Times, Reuters, Fox News, Financial Times, The Guardian, CNN, and S&P 500 reports, we examine firm connections and risk transmission pathways from 2020 to 2022. Applying natural language processing techniques, we assess how media coverage influences firm relationships and financial contagion. Firms with high centrality in the news similarity network show greater exposure to financial shocks, reinforcing the role of public perception in risk propagation. Community detection reveals clusters that do not always align with traditional sector classifications, highlighting cross-industry dependencies. Regression analysis further suggests that firm size and stock price volatility influence network centrality, indicating an interaction between financial characteristics and media-driven contagion. By incorporating textual data into systemic risk assessments, this study complements traditional models and offers a new perspective for regulators and investors monitoring financial stability.

3.1 Introduction

The stability of the global financial system depends not only on the strength of individual firms but also on their interconnections (Acharya, 2009; Crockett, 2000). The increasing complexity of financial markets has made systemic risk a pressing concern for regulators and investors (Acharya et al., 2017). Systemic risk arises when disruptions affecting a single institution or a small group propagate through financial linkages, triggering widespread instability. While banks and other financial entities manage risk by diversifying investments, the widespread adoption of similar strategies can create hidden vulnerabilities (Beale et al., 2011). A crisis in one part of the system can quickly spread, overwhelming safeguards designed for individual institutions.

This paper introduces a method for measuring systemic risk by constructing a network of firms based on news similarity. Using financial news articles from major media sources, including The New York Times, Reuters, Fox News, Financial Times, The Guardian, and CNN, we examine how firms are connected through media coverage. The dataset, which covers S&P 500 firms from 2020 to 2022, is processed using natural language processing (NLP) techniques to assess textual similarity between companies. In this framework, firms serve as nodes, while links represent the degree of similarity in news coverage. To ensure statistical robustness, we apply a token permutation algorithm and an entropy-based filtering model (Cajueiro et al., 2021). Unlike traditional financial indicators such as stock returns or balance sheet data, this approach captures how firms are associated in public discourse, providing a media-driven perspective on interconnectedness. A key component of this analysis involves using network structures to estimate stationary probabilities as a proxy for firm centrality, allowing us to map relationships and identify indirect contagion pathways.

Our study contributes to the literature on systemic risk by incorporating textual data into contagion analysis, extending previous research on the role of media in financial uncertainty. Baker, Bloom & Davis (2016) construct an economic policy uncertainty index by tracking the frequency of newspaper articles containing specific keywords related to economic and political uncertainty. Using a similar approach, Ma et al. (2024) develop an uncertainty measure for China. Other studies explore the relationship between textual data and economic conditions (Bybee et al., 2020). Instead of building an index, we employ network analysis to assess news similarity and identify firm relationships. This approach builds on the idea that network structures shape how financial market participants communicate and react to information (Tedeschi, Iori & Gallegati, 2009; Tedeschi, Iori & Gallegati, 2012).

This research also connects to the literature on indirect contagion. The 2007–2009 global financial crisis demonstrated how financial instability spreads beyond direct exposures such as interbank lending, extending to market perceptions, asset price movements, and investor sentiment (Haldane & May, 2011). Recognizing this, researchers have applied network science to financial systems to examine firm-level interactions and measure contagion pathways (Summer, 2013; Petrone & Latora, 2018). Previous studies construct financial networks using principal component analysis (PCA) and Granger-causality models applied to return data (Billio et al., 2012), balance sheets of European banks (Cont & Schaanning, 2019), and interbank loan structures (Roncoroni et al., 2021). These approaches primarily focus on financial correlations and risk transmission through market data.

However, financial contagion extends beyond balance sheets and asset prices. Public perception and information flows also shape systemic risk. The increasing availability of textual data has enabled new approaches to financial risk analysis. Media coverage influences investor expectations and contributes to market reactions (Tetlock, 2007; Kaplanski & Levy, 2010). Advances in NLP have allowed researchers to extract information from financial news, social media, and corporate reports (Nadkarni, Ohno-Machado & Chapman, 2011; Gentzkow, Kelly & Taddy, 2019; Liu et al., 2023; Cajueiro et al., 2023). These techniques provide an alternative perspective on firm relationships, capturing risk dynamics that traditional financial metrics may not fully reflect.

Our results show that firms with strong media-based connections do not always belong to the same sector, suggesting that financial contagion can extend beyond conventional industry classifications. In the Financials sector, firms with high centrality in the news similarity network exhibit greater exposure to financial shocks, reinforcing the role of public perception in systemic risk transmission. Community detection identifies clusters of firms that reflect patterns of shared media coverage rather than strictly financial relationships. For instance, Citigroup appears more interconnected with firms outside the financial sector, while other major banks form a distinct group. Regression analysis further indicates that company size and stock price volatility influence network centrality, highlighting the role of market perception in shaping systemic vulnerabilities. These findings suggest that tracking media-driven firm relationships can provide regulators and investors with an additional tool for assessing systemic risk.

Our paper proceeds as follows. Section 3.2 outlines the methodological framework for analyzing news-based similarities between firms and their role in systemic risk. Section 3.3 describes the dataset and preprocessing techniques. Section 3.4 presents and interprets the findings. Section 3.5 discusses their broader implications. Finally, Section 3.6 summarizes the study and suggests directions for future research.

3.2 Methodology

In this section we outline the methodological framework we explore to analyze news-based similarities between companies and evaluate their implications within a networked structure. Section 3.2.1 and Section 3.2.2 apply Cajueiro et al. (2021). In Section 3.2.1, we evaluate the similarities between companies by analyzing their associated news stories using NLP techniques. This step constructs a similarity network, where nodes represent companies, and edges reflect the strength of relationships based on textual content. The approach incorporates a token permutation algorithm to filter out random word overlaps and an entropy-based similarity measure to quantify meaningful connections. Section 3.2.2 introduces the concept of infection probabilities within the network. Using a nonlinear system of equations, this framework models the likelihood of a company influencing or being influenced by its neighbors. The perception matrix, derived from the similarity network, underpins this contagion analysis, enabling us to identify how associations propagate across the network.

In Section 3.2.3, we apply community detection to identify clusters of companies based on shared news-based similarities, offering insights into indirect contagion within the network. Using the Louvain method based on Blondel et al. (2008), we partition companies into groups that either align with predefined sectors or span multiple industries. The method maximizes modularity, ensuring strong intra-community connections while keeping inter-community links sparse. Examining these clustering patterns clarifies how information propagates among companies and across industries. In Section 3.2.4, we examine the relationship between company centrality within the network and financial attributes. Centrality reflects a company's importance or influence, and regression analysis helps uncover the underlying factors that drive these dynamics. By integrating robust estimation techniques, we ensure the reliability and interpretability of the results.

Together, these components provide a comprehensive framework for understanding the relationships and interactions between companies through the lens of their associated news stories. This methodology not only quantifies similarities but also uncovers the mechanisms of influence and the factors contributing to network dynamics.

3.2.1 Evaluating News-Based Similarities Between Companies

This framework evaluates similarities between companies by analyzing news associated with them. The methodology from Cajueiro et al. (2021) employs NLP techniques to construct a similarity network.

Define w_i as a word referred to as a *term*, uniquely indexed by i. The set of all distinct terms from the stories forms the *vocabulary*, denoted as $\mathcal{V} = \{w_1, w_2, \dots, w_{N_\mathcal{V}}\}$, where $N_\mathcal{V}$ is the total number of terms. The index set for all terms is $I_\mathcal{V} = \{1, 2, \dots, N_\mathcal{V}\}$.

Each story s_j contains a sequence of L_j non-unique terms:

$$s_j = \left[w_{i_1}, w_{i_2}, \ldots, w_{i_{L_j}}\right], \quad 1 \leq L_j \leq N_{\mathcal{V}}, \quad i_k \in I_{\mathcal{V}}, \tag{3.1}$$

where L_j is the number of terms in story s_j, and $I_{\mathcal{V}}$ denotes the set of term indices. The vocabulary of story s_j, $\mathcal{V}^{s_j}$, includes all unique terms in s_j. We represent the set of all stories as $\mathcal{S} = \{s_1, s_2, \ldots, s_{N_{\mathcal{S}}}\}$, where $N_{\mathcal{S}}$ indicates the total number of stories.

Let $\mathcal{C}$ denote the set of all companies, $\mathcal{C} = \{1, 2, \ldots, N_{\mathcal{C}}\}$, where $N_{\mathcal{C}}$ is the total number of companies. Each story $s_j \in \mathcal{S}$ associates with a specific company $k \in \mathcal{C}$. For each company k, define s^k as the concatenation of all its stories. The concatenated stories form the set $\mathcal{S}^{\mathcal{C}} = \{s^1, s^2, \ldots, s^{N_{\mathcal{C}}}\}$.

The *term-story matrix*, $\mathbf{M}$, quantifies the relationship between terms and companies. The matrix dimensions are $N_{\mathcal{V}} \times N_{\mathcal{C}}$, where rows correspond to terms, and columns correspond to concatenated stories of each company. The matrix is structured as:

$$\mathbf{M} = \begin{bmatrix} n_{11} & n_{12} & \ldots & n_{1N_{\mathcal{C}}} \\ n_{21} & n_{22} & \ldots & n_{2N_{\mathcal{C}}} \\ \vdots & \vdots & \ddots & \vdots \\ n_{N_{\mathcal{V}}1} & n_{N_{\mathcal{V}}2} & \ldots & n_{N_{\mathcal{V}}N_{\mathcal{C}}} \end{bmatrix}, \tag{3.2}$$

where n_{ik} counts the frequency of term w_i in the concatenated stories of company k.

We model the interactions between companies as a network. Nodes represent companies, and edges connect companies k and l based on the probability of associating a story about k with l. Define the set of neighbors of k as $\mathcal{N}_k$, where $l \in \mathcal{N}_k$ if a link exists between k and l.

The similarity measure $q_{k,l}$ quantifies the relationship between companies k and l:

$$q_{k,l} = \cos(\theta_{k,l}) = \frac{\sum_{i=1}^{N_{\mathcal{V}}} f^k(w_i) f^l(w_i)}{\|f^k\| \|f^l\|}, \tag{3.3}$$

where $\theta_{k,l}$ is the angle between the frequency vectors f^k and f^l. The function $f^k(w_i)$ calculates the importance of term w_i for company k using the Entropy Model:

$$f^k(w_i) = \omega_{\text{local}}(i, k) \cdot \omega_{\text{global}}(i), \tag{3.4}$$

with:

$$\omega_{\text{local}}(i, k) = \log_2(n_{ik} + 1), \tag{3.5}$$

$$\omega_{\text{global}}(i) = 1 + \frac{\sum_{k=1}^{N_{\mathcal{C}}} p_{ik} \log_2 p_{ik}}{1 + \log_2 N_{\mathcal{C}}}, \tag{3.6}$$

and:

$$p_{ik} = \frac{n_{ik}}{\sum_{l=1}^{N_{\mathcal{C}}} n_{il}}. \tag{3.7}$$

The local weight $\omega_{\text{local}}(i, k)$ evaluates the importance of term w_i in company k's stories, while the global weight $\omega_{\text{global}}(i)$ adjusts for the term's relevance across all companies. Words that frequently appear in one company but rarely in others receive higher importance. The probability m_{kl} measures the perception of association between companies k and l:

$$m_{kl} = \alpha q_{kl}^{\beta}, \tag{3.8}$$

where $\alpha \in (0, 1]$ scales all connections, and $\beta \in [1, \infty)$ emphasizes stronger connections. Larger β values reinforce similarities with q_{kl} close to 1, strengthening the connection between highly similar companies.

3.2.2 Evaluating Infection Probabilities

This section presents the stationary probabilities of infection within the network of companies. Using the perception matrix $M = [m_{kl}]$ we may define a dynamical nonlinear system of equations with stationary probabilities π_k for each company $k \in \mathcal{C}$ given by

$$\pi_k = 1 - \prod_{k \neq l} (1 - m_{kl}\pi_l), \tag{3.9}$$

where π_k represents the probability that company k becomes infected. [1] [2]

The contagion process here uses π_k as proxies for measuring the likelihood of a positive or negative association in the stories of neighboring companies. The term $m_{kl}\pi_l$ quantifies the probability that company l infects company k.[3]

Equation (3.9) provides a mathematical foundation for evaluating the spread of contagion between companies. By interpreting the perception matrix M and its influence on π_k, we can quantify how neighboring companies impact each other through shared associations in their stories.

3.2.3 Detecting Communities

Detecting how companies form clusters in the network provides valuable insights into indirect contagion. Community detection allows us to identify whether companies

[1] The product term $\prod_{k \neq l}(1 - m_{kl}\pi_l)$ calculates the joint probability that no company l in the neighborhood of k infects k. Consequently, Eq. (3.9) gives the complement, which represents the probability of at least one neighbor l infecting k.

[2] This system relates to the stationary solution of a Susceptible-Infected-Susceptible (SIS) model on networks, often used in epidemiological models, under a mean-field approximation (Pastor-Satorras & Vespignani, 2001; Meloni, Arenas & Moreno, 2009).

[3] To solve Eq. (3.9), we apply fixed-point iteration. This method iteratively updates π_k until the system converges to a stable solution. For convergence to hold, the Jacobian matrix of the system must satisfy a max-norm condition:

$$\|J\|_\infty < 1, \tag{3.10}$$

where $\|J\|_\infty$ is the max-norm of the Jacobian evaluated near the solution π. Condition (3.10) ensures that the iterative process converges to a unique solution for the stationary probabilities.See Heath (1998) for details.

cluster within their sectors, form cross-sectorial groups, or act as bridges between different communities (Wan et al., 2021).

We use the Louvain method (Blondel et al., 2008) to detect communities in our news-based network, which we construct using the similarity measure defined in Eq. (3.3). The Louvain method optimizes the modularity score Q, a measure of how well the network divides into communities. Modularity evaluates the density of connections within communities compared to what a random network model would predict. We calculate Q using:

$$Q = \frac{1}{2m} \sum_{k,l} \left(A_{k,l} - \frac{K_k K_l}{2m} \right) \delta(c_k, c_l), \tag{3.11}$$

where $A_{k,l}$ represents the weight of the edge between companies k and l, derived from the similarity measure. The terms K_k and K_l sum the edge weights connected to nodes k and l, m represents the total edge weight in the network, c_k and c_l are the communities of k and l, and $\delta(c_k, c_l)$ equals 1 if k and l belong to the same community and 0 otherwise. A high modularity score indicates that nodes within the same community are densely connected, while links between different communities remain sparse.

The Louvain method is well-suited for undirected, weighted networks, making it an appropriate choice for our news-based similarity network. This method efficiently partitions companies into communities by maximizing modularity, ensuring stronger internal connections while keeping inter-community links weaker. Although widely used, its application here stems from the fact that our network satisfies the conditions for which it was designed. The algorithm follows three steps. First, each node starts in its own community. Then, nodes iteratively move between communities to maximize modularity. Finally, the method aggregates communities into single nodes and repeats the process until modularity stabilizes. This approach reveals clustering patterns based on shared news stories, showing whether companies remain within their predefined sectors or form cross-sector communities. Identifying firms that connect different sectors highlights their role in indirect contagion, offering insights into how news-based associations shape inter-company relationships.

3.2.4 Centrality and Regression Analysis

Understanding the factors that influence a company's centrality within a network is critical for interpreting contagion dynamics. Centrality captures how essential a company is to the network, and analyzing its relationship with financial attributes provides insight into the underlying drivers of influence. To investigate these relationships, we use linear regression models. The response variable y, representing centrality, is a linear function of the explanatory variables in X:

$$y = X\beta + \varepsilon, \tag{3.12}$$

where y is an $n \times 1$ vector of centrality measures, X is an $n \times p$ matrix of financial attributes, β is a $p \times 1$ vector of coefficients, and ε represents errors. Ordinary Least Squares (OLS) estimation identifies the coefficients $\hat{\beta}$ that minimize the sum of squared residuals (SSR):

$$b = \arg\min_{\hat{\beta}} \mathrm{SSR}(\hat{\beta}) = \sum_{i=1}^{n} (y_i - X_i' \hat{\beta})^2, \tag{3.13}$$

where SSR quantifies the discrepancy between observed and predicted values.[4]

3.3 Data

The dataset for this study focuses on news articles about American banks listed in the *S&P 500* index.[5] We collect these articles from six major media outlets: the *New York Times*, *Reuters*, *Fox News*, *Financial Times*, *The Guardian*, and *CNN*. This database spans from 1st January 2020 to 31st December 2022 and contains 14,057 articles.

To ensure representativeness, we excluded companies with fewer than 15 articles over the entire period. After filtering, the dataset includes stories about 48 companies. This refinement step ensures that each company has sufficient coverage for meaningful analysis. We pre-process the news data in four essential steps: (i) tokenization, (ii) removal of stopwords, (iii) stemming, and (iv) filtering of rare words. In step (i), we break the text into individual components, or tokens, by removing punctuation (e.g., commas, periods, and hyphens), converting all words to lowercase, and splitting the text into words. Tokenization organizes the data into manageable units for analysis, resulting in 3,618,322 tokens. In step (ii), we eliminate all gramatical words such as prepositions and conjunctions, which do not provide meaningful information about the companies and and keep all content words. Since no universal stopword list exists, we carefully select terms to exclude irrelevant content while retaining relevance. Step (iii) reduces words to their root forms, consolidating variations (e.g., “running” and “ran”) into a single representation, minimizing redundancy, and improving analytical clarity. Finally, step (iv) removes words that appear fewer than five times in a single article or in fewer than ten articles. This step focuses the analysis on relevant terms and reduces noise, such as uncommon proper nouns.

In addition to news articles, we integrate financial data for the 48 companies. This financial information complements the textual data, allowing us to analyze the relationship between company characteristics and network centrality measures. From this point forward, we define the vocabulary $\mathcal{V}_w$ as the set of tokens that remain after pre-processing. Each

[4] OLS remains a popular choice for its simplicity, but violations of homoscedasticity can lead to inefficient estimates and unreliable hypothesis tests. To address this, we use the HC3 estimator, which adjusts for heteroscedasticity and performs well in small samples. We also apply the White test to detect heteroscedasticity and validate the assumption of constant error variance, ensuring reliable and interpretable regression results.

[5] S&P 500 index. Available at url: https://www.spglobal.com

token $w_i \in \mathcal{V}_w$ represents a stemmed term that excludes stopwords and rare words. The full dictionary is available in Table 9 of the Appendix A.1.

3.4 Results

This section presents the main findings of our analysis, examining how news similarity networks capture firm interconnections and systemic risk propagation. Section 3.4.1 describes the structural properties of the network, focusing on connectivity patterns and sectorial clustering. Section 3.4.2 applies community detection techniques to identify clusters of firms and determine whether news-based linkages correspond to traditional sector classifications. Section 3.4.3 examines the relationship between firms' centrality in the network and their financial characteristics, explaining the factors that influence media-driven contagion. Section 3.4.4 builds on these results, discussing their implications for systemic risk assessment and market stability.

3.4.1 Evaluation of Similarities

Using Eq. (3.3), we calculate the strength of connections between pairs of companies based on the text of their stories. A major challenge in this process involves distinguishing meaningful links from those formed by uninformative words (e.g., generic terms common across multiple companies). To address this, we run a token permutation algorithm that generates a randomized baseline for comparison. This approach identifies genuine similarities while filtering out noise caused by random overlaps in word usage.

The algorithm simulates "random stories" by permuting tokens between companies, preserving token frequency distributions. Some companies rely on a small, focused vocabulary to describe their core business, while others use broader language. By maintaining these distributions, the algorithm prevents misinterpretation of random overlaps as meaningful connections. This procedure follows Cajueiro et al. (2021).

The steps of the algorithm are as follows:

1. *Random Selection of Frequency:* Select a token frequency $\bar{n}$ at random.
2. *Company Pair Selection:* Randomly choose two companies, k and l.
3. *Token Identification:* Identify tokens with frequency $\bar{n}$ in the stories of both companies. Then, randomly choose one token with frequency n from each company, namely w_k and w_l;
4. *Token Exchange:* Swap w_k and w_l if w_k does not appear in company l's stories, and w_l does not appear in company k's stories.

This procedure preserves the frequency distribution of tokens as defined in Eq. (3.2), ensuring that the randomized stories retain the statistical properties of the original ones, except for token assignments.

Figure 4 compares the link strength distributions in the original network (blue line) and the randomized network (red dashed line). The measure q, calculated using Eq. (3.3), evaluates the similarity between companies. The randomized network establishes a baseline for understanding whether observed similarities are meaningful. Under the null hypothesis, similarities result from random overlaps in word usage. The red curve represents this null distribution, illustrating the expected range of similarities by chance. The blue curve represents the original network, capturing the actual observed similarities. Furthermore, the vertical black line shows the 5% significance level for the null hypothesis, that sits around q = 0.7095. This level is used as threshold for separating links in the original network that can be attributed to random similarities, located left from the significance threshold, from those presenting evidence for strong relation between companies news.

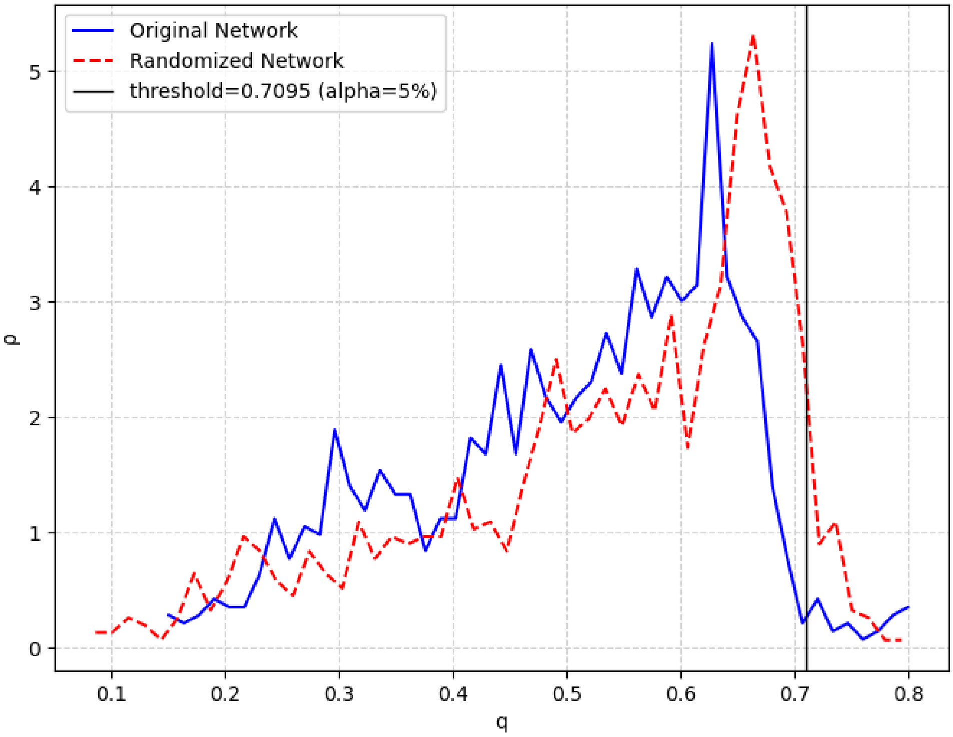


Figure 4 – Distributions of link strengths for the original network (blue line) , the randomized network (red dashed line) and vertical black dashed line for 5% significant threshold.

To determine whether the distributions of link strengths differ significantly, we use the Kolmogorov-Smirnov (KS) test. The test statistic is 0.19889, with a p-value $<$ 0.0001. These results strongly reject the null hypothesis that the two distributions arise from the same process. The significant difference between the two distributions indicates that the original network reflects meaningful connections. Companies with higher q values likely share genuine characteristics, such as similar business models, markets, or strategies. In other words, the KS test confirms that differences in Figure 4 are not due to random variation. Links on the extreme right of the original distribution represent the strongest evidence for meaningful connections. These connections suggest shared market dynamics, overlapping supply chains, or similar strategic approaches.[6] This analysis provides a systematic way to identify meaningful patterns within the network. By isolating these connections, we gain insights into the relationships and interactions between companies.

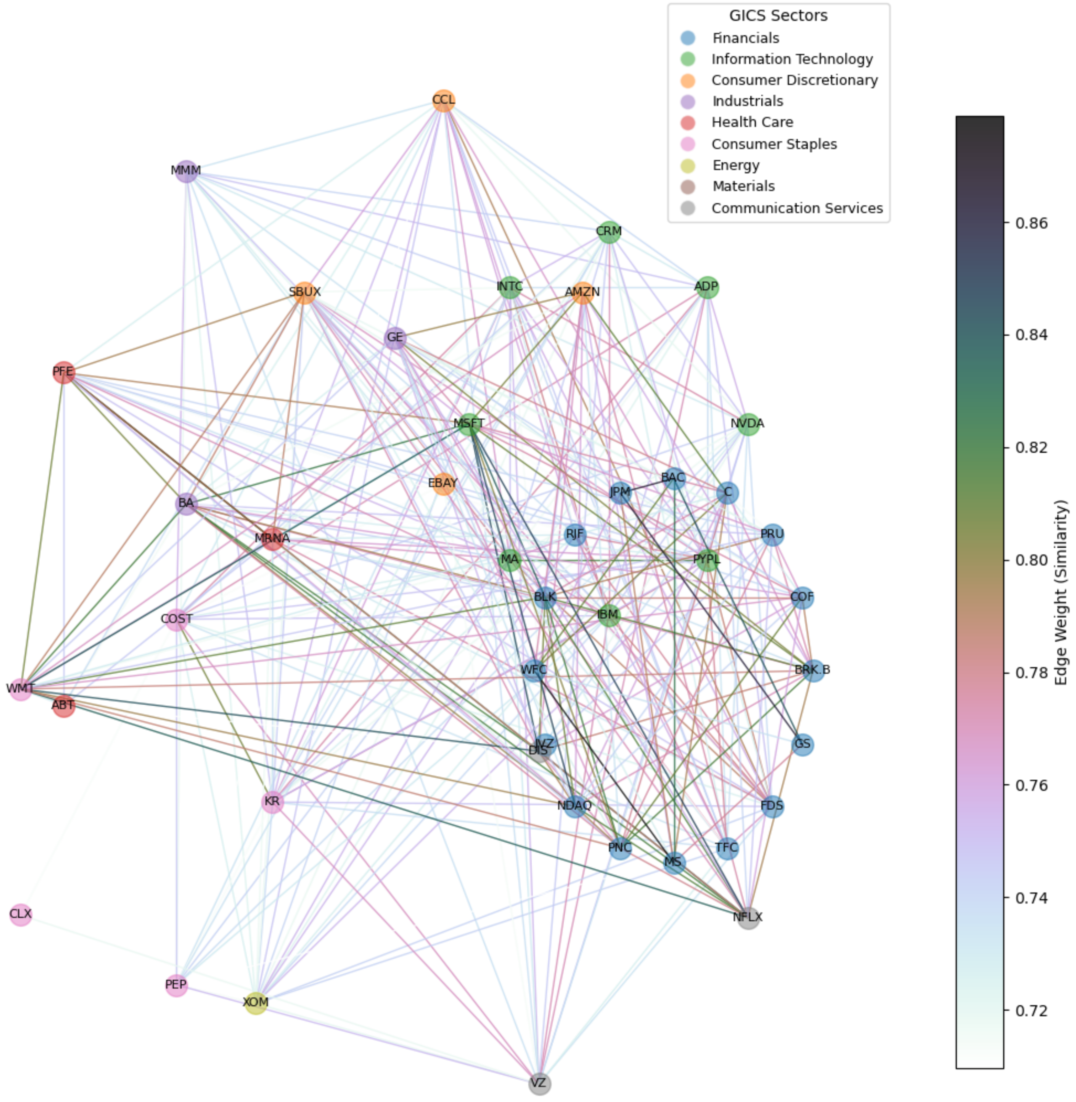
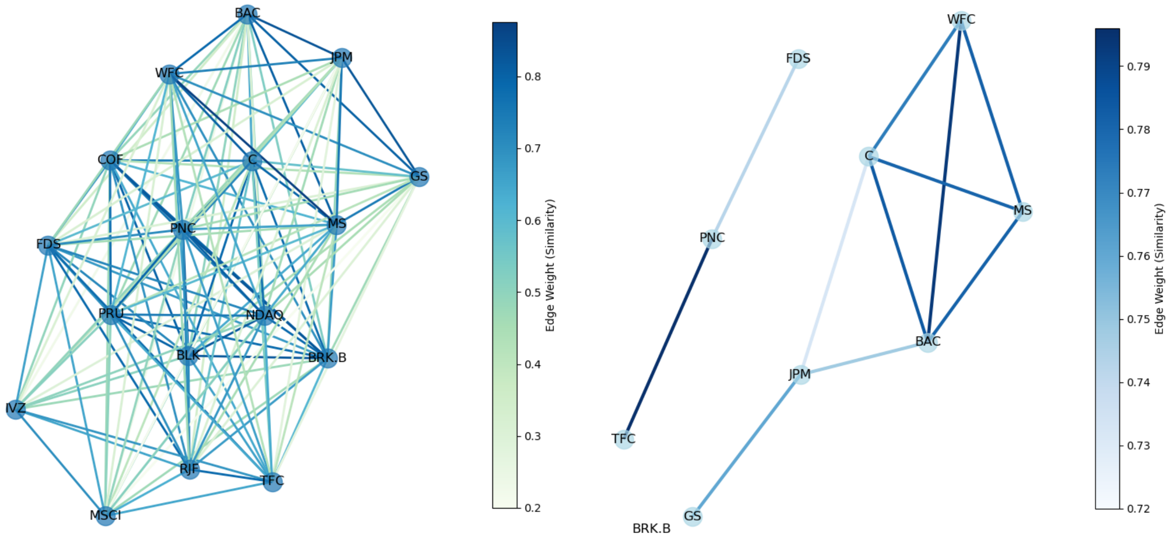


Figure 5 – The Fully Connected Network.

Figure 5 illustrates the fully connected network, highlighting densely connected clusters within specific sectors after filtering for the random connections. For example, the Financials sector exhibits strong intra-sector connections, suggesting that companies in

[6] In Appendix A.1 we illustrate the shapeless patterns of the randomized network in Figure 26.

this group share significant similarities, possibly driven by shared market dynamics or collaborative activities. Health Care and Consumer Discretionary sectors display notable interconnections with other sectors, reflecting their broad influence and dependencies across industries. Within Information Technology, Microsoft (MSFT) emerges as a hub, linking companies within the sector and bridging connections to other industries. IT companies demonstrate dense connectivity, underscoring their pervasive role across sectors due to the widespread adoption of digital transformation and technology-driven services. In the Financials sector, key players like JPMorgan (JPM) and Bank of America (BAC) act as pivotal nodes, reinforcing the sector's cohesion. Some sectors feature prominent hubs that bridge multiple industries. For instance, Information Technology frequently interacts with Consumer Discretionary, highlighting the growing importance of digital platforms in retail and entertainment.



(a) Complete Financials network

(b) Strongest Financial Network filtered

Figure 6 – The Financials Network.

Examining patterns across sectors reveals distinct structural differences. Financials and Information Technology form larger, cohesive subgroups, that can reflect strong intra-sector relationships. In contrast, sectors such as Consumer Discretionary and Energy display more fragmented structures, with companies often acting as intermediaries or connectors. This fragmentation apparently arises from the diverse nature of these industries and their reliance on inputs and partnerships across various sectors.[7]

[7] Table 10 in Appendix A.1 provides an overview of the networks, companies, and their respective labels. Panel (a) lists the companies that form the largest network, which includes all nodes. However, as shown in Figure 27, the companies Cisco, Comcast, MSCI, and Qualcomm have irrelevant links. We consider a link as relevant if the similarity between two companies exceeds the threshold derived from the null distribution of similarities. Hence, we exclude these companies from the fully connected

Figure 6 provides a detailed view of the Financials sector, showing how firms are connected. Panel (a) includes all relationships, forming a dense web of links. Panel (b) filters out weaker connections using the algoritm presented in Section 3.4.1, keeping only the strongest ties. In this refined network, Wells Fargo (WFC) and Bank of America (BAC) have one of the closest relationships, while JPMorgan (JPM) acts as a hub, linking multiple institutions. Some firms, such as BlackRock (BLK) and PNC Financial Services (PNC), remain connected but have fewer direct links, reflecting their specialized roles in the financial market.

Applying a threshold to remove weaker links reveals differences in how sectors are structured. The Financials sector remains the most connected, with strong internal relationships between major firms. Outside of Financials, the only significant intra-sector link is between Disney and Netflix, highlighting their competitive but interdependent roles.

This network analysis shows that while some industries maintain dense internal connections, others rely on external partnerships. Financial institutions form a closely connected system, most likely due to shared market dependencies. In contrast, companies in sectors such as Industrials and Consumer Discretionary tend to have fewer direct links, reflecting the more diverse nature of their business activities.

3.4.2 Community Detection

We apply the Louvain method to detect communities in the network, uncovering clusters of companies with stronger internal connections based on shared relationships or similarities. This analysis reveals distinct patterns of intra- and inter-community interactions, shedding light on how companies relate within and across sectors.

Figure 7 shows the communities in the network. Node colors indicate community affiliation, with blue representing Community 1, beige representing Community 2, and cyan representing Community 3. Intra-community edges use colors matching their respective communities, while light gray highlights inter-community edges.

Community 1 features a concentrated in banks, JPMorgan Chase (JPM), Bank of America (BAC), Wells Fargo (WFC), Goldman Sachs (GS) and Morgan Stanley (MS), which dominate the fully connected financial network in Figure 6(b), the exception is Citigroup (C) that stays in Community 2. This isolated cluster suggests a high level of intra-sector dependency, with these firms primarily engaging with each other rather than external communities. The lighter-weight connections extending toward other groups indicate limited but existing inter-sector interactions, reinforcing the idea that these financial institutions operate within a more self-contained framework. Community 2 is a mix of companies from various sectors, including discretionary consumer, communication

network depicted in Figure 5. Panel (b) of Table 10 lists the companies and their respective labels included in this fully connected network.

services, financials, health care, and industrials. Instead, it features a dense web of internal connections, indicating strong intra-community relationships. Companies such as Amazon (AMZN), Microsoft (MSFT), and Netflix (NFLX) show significant integration within this cluster, suggesting their strategic roles in maintaining sector-wide interactions. The presence of financial firms such as Citigroup (C) , Capital One (COF), and Berkshire Hataway (BRK.B) within this community highlights a potential overlap between finance and consumer-driven businesses, likely due to market dependencies. Community 3 forms a well-integrated and highly interconnected structure, primarily consisting of companies from technology, consumer staples, and industrials. Firms such as Nvidia (NVDA), Intel(INTC), Mastercard (MA), and PepsiCo (PEP) exhibit dense intra-community linkages, highlighting strong sectoral relationships. The circular layout of this community suggests equal-weighted connectivity among its members, indicative of balanced collaboration or shared market influences. Although it includes several IT companies, they do not exhibit strong intra-sector connections, as noted in Section 3.4.1.

Cross-sector dynamics are also evident in the Consumer Discretionary and Communication Services sectors, which display dispersed connectivity. Companies like Amazon (AMZN) and Netflix (NFLX), maintain links within Community 2 but also connect to sectors such as Financials and Technology. This dispersion indicates that these companies engage in broader inter-sector interactions, reflecting their multi-industry business models and market reach. This community analysis provides insights into how companies form tightly knit groups while maintaining cross-sector relationships. Table 11 in the Appendix compares companies by sector and community, further illustrating these dynamics. The clear clustering of Financials, specifically in banks, in Community 1 and the diversified composition of Community 2 and Community 3 underscore the varying degrees of connectivity and interdependence across industries.

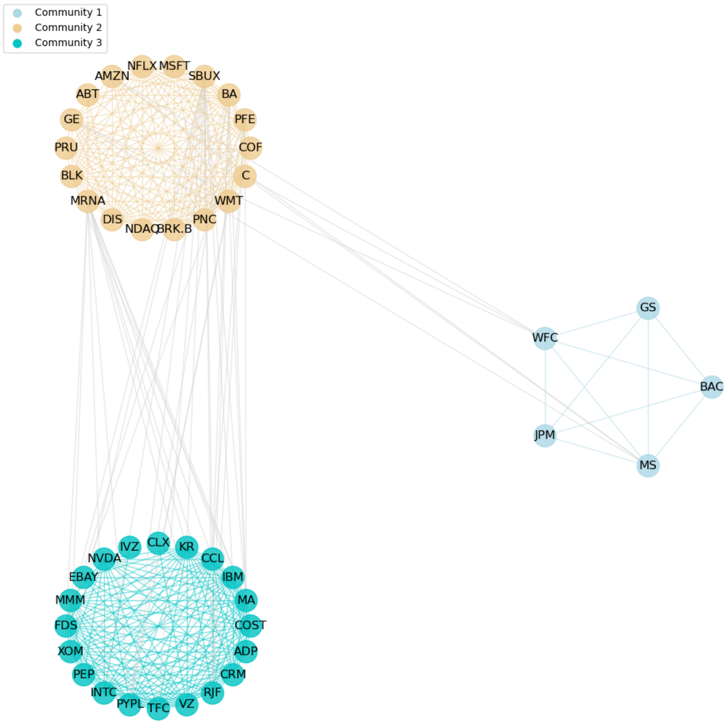


Figure 7 – Communities identified in the network using the Louvain method.

3.4.3 Centrality and Contagion Pathways

Centrality within a network represents the influence of nodes and the potential pathways of contagion among them. In the context of financial networks, centrality identifies companies that have a disproportionate impact on others, embodying the concept of "too big to fail." These firms play a pivotal role in propagating shocks across the network.

To measure centrality, we solve Eq. (3.9), which evaluates each company's stationary probability π associated with the dynamical system where π serves as a proxy for centrality, capturing the likelihood that a company becomes "infected" through its connections in the network. As in works like Cajueiro et al. (2019), Cajueiro et al. (2021), this approach enables a structured evaluation of contagion dynamics and the identification of influential companies. [8] The function $f(x) = \alpha x^{\beta}$, which determines the perception parameter m_{ij}, is strictly increasing. Consequently, the values of α and β influence the magnitude of m_{ij} but do not alter the relative ranking of the probabilities π.

[8] The computation of π requires verifying that the Jacobian matrix associated with the system of equations satisfies the convergence condition—specifically, that its norm is smaller than 1 in a neighborhood of the solution. This ensures the stability and uniqueness of the probabilities derived from Eq. (3.9).

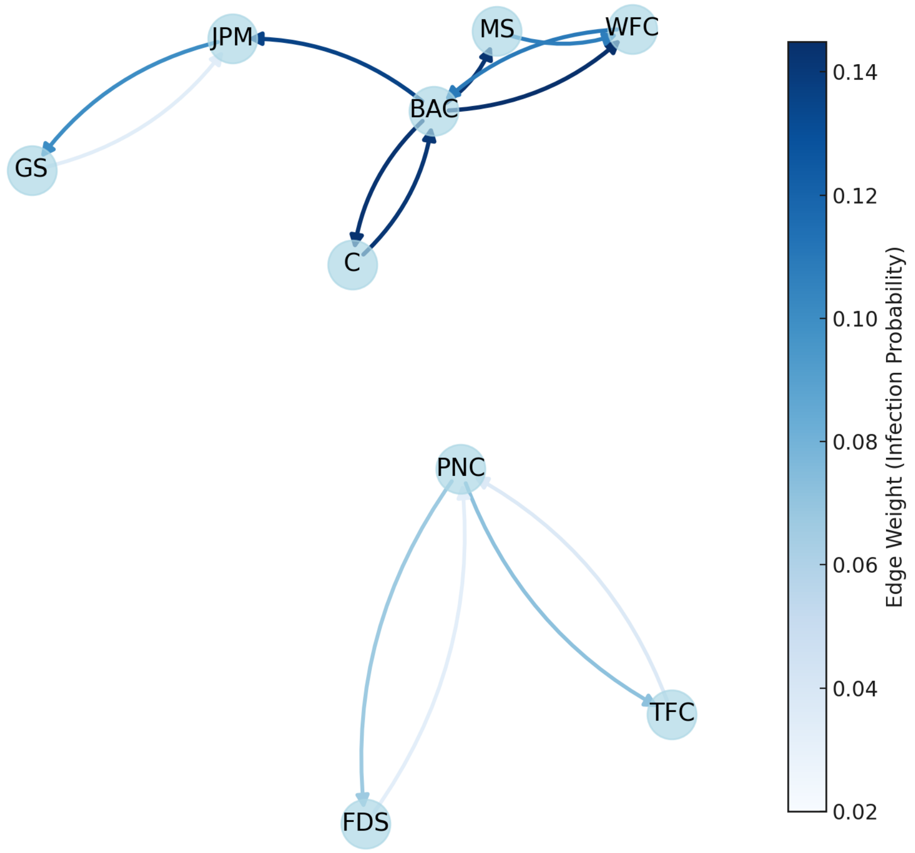


Figure 8 – The Financials Industry network.

To ensure consistent and interpretable results, we adopt parameter values of $\alpha = 1$ and $\beta = 4.68$.[9] This configuration allows us to analyze the pathways of contagion effectively, shedding light on how shocks propagate through the network and which firms are most central to its structure.

Figure 8, in conjunction with the dagger (†) superscripts in Panel (C) of Table 10, represents the financial industry network, which clearly divides into two distinct sub-networks. The first sub-network, marked by asterisks ($*$) in Panel (C) of Table 10, primarily comprises large, diversified banks. At its center is Bank of America (BAC), which forms strong connections with major financial institutions such as Citigroup (C), Wells Fargo (WFC), and Morgan Stanley (MS). Additional links to JPMorgan Chase (JPM) and Goldman Sachs (GS) underscore Bank of America's pivotal role in this tightly connected cluster of influential banking entities.

The second sub-network, identified by double-dagger (‡) superscripts in Panel (C) of Table 10, highlights PNC Financial Services (PNC) as its central node. This network

[9] This satisfy the convergence criteria and align with the findings of Cajueiro et al. (2021).

includes regional banks like Truist (TFC) and data providers such as FactSet (FDS), emphasizing PNC's role in bridging regional banking activities with financial information services. This configuration highlights the specialized nature of this sub-network, with PNC Financial Services serving as a critical intermediary connecting diverse elements of the regional financial landscape.

3.4.4 Explaining Network Centrality

We analyze the centrality of banking companies within the network using financial data. For this purpose, we employ an Ordinary Least Squares (OLS) regression model, where the dependent variable (y) represents the centrality of companies, and the explanatory variables to help us explain a company's role in the network include `logemployees`, `changepercent`, and `dividendYield`. As indicated by the correlation matrix in Table 4, these variables exhibit minimal correlation.

	centrality	`changepercent`	`logemployees`	`dividendYield`
centrality	1.00			
`changepercent`	-0.01	1.00		
`logemployeess`	0.7	-0.4	1.00	
`dividendYield`	0.3	-0.2	-0.1	1.00

Table 4 – Correlation Matrix

The OLS model, adjusted with the HC3 robust covariance estimator to account for heteroscedasticity, takes the following form:

$$\begin{multline} y = -0.4833 + 0.0412 \times \texttt{logemployees} + 2.8766 \times \texttt{ChangePercent} \\ + 1.6116 \times \texttt{dividendYield}, \tag{3.14} \end{multline}$$

where `logemployees` is the logarithm of the number of employees, a proxy for firm size. Normalizing employee counts in this way ensures comparability across firms of varying sizes. `Changepercent` measures the percentage change in stock prices from the previous trading day, capturing market perceptions and stock price dynamics. `DividendYield` represents the income-generating potential of a stock relative to its price, calculated as:

$$\texttt{dividendYield} = \left(\frac{\texttt{ttmDividendRate}}{\texttt{Previous Day Close Price}} \right) \times 100. \tag{3.15}$$

The regression results summarized in Table 5 provide valuable insights into the factors influencing centrality within the network. The model achieves a high explanatory power, with an R^2 value of 0.700, indicating that 70% of the variation in centrality is explained by the selected variables: `logemployees`, `ChangePercent`, and `dividendYield`.

The variable **`logemployees`** is highly statistically significant ($p < 0.0001$) with a positive coefficient ($\beta = 0.0412$). The result suggests that larger firms, as measured by the logarithm of the number of employees, tend to hold more central positions in the network. The positive association emphasizes the critical role of firm size as a driver of influence and operational capacity within the network.

Reflecting the percentage change in stock prices, `ChangePercent` demonstrates strong significance ($p = 0.002$) and a large positive coefficient ($\beta = 2.8767$). This indicates that firms with higher stock price volatility or positive price trends are perceived as central players in the network. Such firms likely attract more attention and connections, reflecting their dynamic roles in financial markets.

Although less statistically significant ($p = 0.086$), `dividendYield` has a positive coefficient ($\beta = 1.6117$). The result suggests that companies offering higher dividend yields may occupy slightly more central positions. Dividend-paying firms are often seen as stable and reliable, which could enhance their perceived importance.[10]

Dependent Variable:	centrality	**R-squared:**	0.7000
Method:	OLS	**Adj. R-squared:**	0.6310
No. Observations:	17	**F-statistics:**	13.1300
Df. Residuals:	13	**Log-Likelihood:**	0.0003
Df. Model:	3	**AIC:**	-55.94
Covariance Type:	HC3	**BIC:**	-52.61

	Coef	**Std. Err.**	**z**	**P > |z|**	**[0.025**	**0.975]**
constant	-0.4833	0.090	-5.382	<0.0001	-0.659	-0.307
changePercent	2.8767	0.936	3.073	0.002	1.042	4.711
logemployees	0.0412	0.009	4.717	<0.0001	0.024	0.058
dividendYield	1.6117	0.939	1.716	0.086	-0.229	3.452

Omnibus	2.308	**Durbin-Watson**	2.332
Prob(Omnibus)	0.315	**Jarque-Bera (JB)**	1.840
Skew	-0.717	**Prob (JB)**	0.398
Kurtosis	2.262		

Table 5 – OLS results with robust covariance estimator HC3.

The fit between OLS predictions and actual values is illustrated in Figure 9. The scatterplot reveals a strong alignment along the 45-degree line, confirming the predictive accuracy of the model. Firms with lower predicted centrality values align closely with the diagonal, demonstrating that the model performs particularly well for less central firms. A few data points deviate from the diagonal, suggesting that certain companies' centrality may be influenced by non-financial factors, such as media coverage, reputation, or

[10] The HC3 robust covariance estimator accounts for potential heteroscedasticity in the data, ensuring the reliability of the coefficients. White's test results (Panel (a) of Table 12) confirm the presence of heteroscedasticity, justifying the use of heteroscedasticity-consistent standard errors.

strategic alliances. The linear trend reinforces the robustness of the model, while deviations highlight opportunities for incorporating additional variables, such as sentiment analysis or sector-specific metrics.[11]

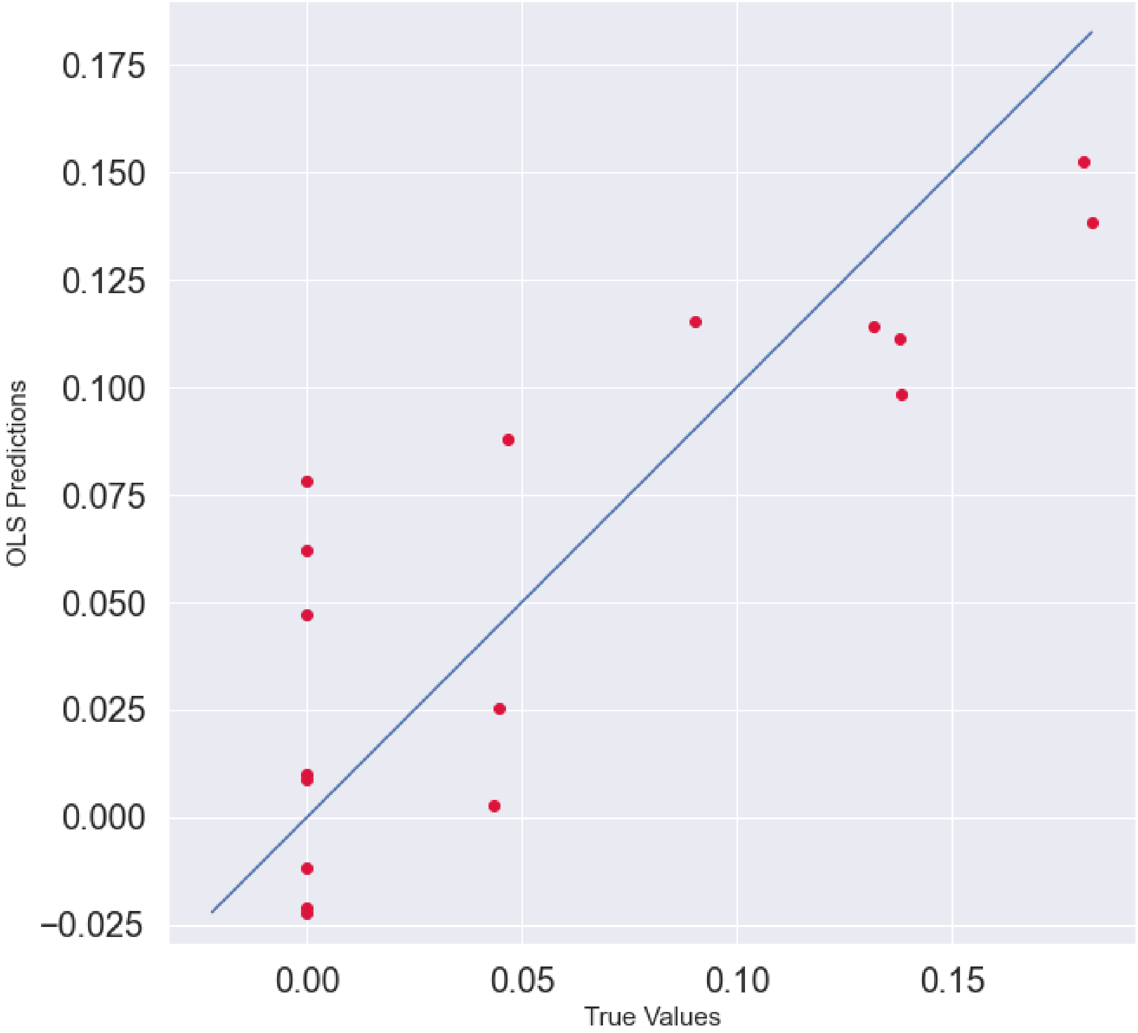


Figure 9 – OLS Predictions with HC3 Robust Covariance.

The low error values indicate that the model predictions are closely aligned with the observed centrality values. These metrics reinforce the suitability of the selected financial variables (`logemployees`, `ChangePercent`, and `dividendYield`) in explaining network centrality.

While financial variables explain a significant portion of centrality, they do not capture the full picture. Other factors, such as market sentiment and socio-political dynamics, likely influence centrality. These unobserved components, embedded in textual patterns and qualitative aspects of news data, remain areas for future exploration. This

[11] The error metrics we present in Panel (b) of Table 12 quantify the performance of the regression model in predicting centrality. The (MAE) of 0.0319 is relatively low and suggests strong predictive accuracy. The MSE of 0.0013, further reflects the model's ability to minimize large errors. The RMSE of 0.0369, aligns with the low MSE and confirming the robustness of the model.

emphasizes the multidimensional nature of centrality, where financial, textual, and social indicators interact to shape the structure of the network.

3.5 Discussion

This study demonstrates how a news similarity network can reveal relationships between firms beyond traditional sector classifications. By applying natural language processing (NLP) techniques to financial news, we extract structured information from unstructured textual data. Constructing a network based on news similarity enables the identification of firms that are perceived as interconnected, even when they do not share direct financial ties. Companies across sectors can use this approach to assess indirect competition and understand how they are positioned relative to others in the market. Additionally, tracking media coverage patterns may help anticipate price movements or sector-wide instability.

The news similarity network also allows for estimating stationary probabilities as a proxy for network centrality, which helps identify firms most exposed to indirect contagion. Within the Financials sector, firms with high centrality in the network experience greater exposure to financial shocks, suggesting that public perception influences systemic risk transmission. These findings underscore the importance of managing reputational risk and maintaining a diversified public image. The relationship between centrality and volatility in the network could also assist regulators in prioritizing firms for audits or financial stability assessments. In our Financials network, the most connected firms include Bank of America, Citigroup, Goldman Sachs, JPMorgan Chase, Morgan Stanley, and Wells Fargo. Additionally, regression results confirm that company size and reputation contribute to systemic risk, reinforcing the role of transparency and the diversity of information sources in preventing risk clusters driven by concentrated narratives.

Community detection using the Louvain method further illustrates how firm relationships extend beyond standard industry classifications. A single community includes five of the most central banks—Bank of America, Goldman Sachs, JPMorgan Chase, Morgan Stanley, and Wells Fargo—while Citigroup appears more integrated with firms in other sectors. Identifying such communities suggests that regulators could refine their supervisory frameworks by incorporating information-based linkages and public perception into risk assessments. This network-driven approach offers regulators a tool to detect systemic vulnerabilities that may not be apparent from financial data alone, capturing market sentiment, corporate reputation, and socio-political influences.

The results support policies aimed at maintaining balanced information dissemination and mitigating the feedback loops between media coverage, market volatility, and systemic risk. Monitoring how opinion leaders shape market narratives is essential, as these narratives can amplify financial instability. Additionally, networks formed by interlocking corporate boards influence strategic decision-making, leading firms to coordinate actions that can have sector-wide consequences (Davis & Greve, 1997; Battiston, Weisbuch & Bonabeau, 2003). Regulators could monitor these networks through news analysis and financial data to reduce systemic risks. This study expands the analytical tools available for financial risk assessment by integrating market perception through news, complementing traditional approaches based on financial statements and quantitative models.

Beyond financial stability applications, this framework may assist in crisis prediction, mapping inter-firm connections based on media narratives, and informing corporate strategy. Understanding how firms are linked through public perception provides regulators and market participants with additional insights into risk transmission and market behavior.

3.6 Conclusion

This chapter applies an approach to systemic risk analysis by constructing a network of American firms based on news similarity of six big media news outlets: The New York Times, Reuters, Fox News, Financial Times, The Guardian, CNN. Unlike traditional methods that rely on financial statements and asset prices, this framework captures indirect contagion by analyzing how firms are connected through media coverage. By applying natural language processing techniques, we identify firm relationships that influence risk transmission but may not be evident in balance-sheet data.

The results demonstrate that firms with high centrality in the news similarity network are more exposed to the propagation of financial shocks, reinforcing the role of public perception in systemic risk. Community detection reveals clusters of firms that extend beyond conventional sector classifications, highlighting the influence of cross-sector information flows. Additionally, regression analysis shows that firm size and stock price volatility contribute to network centrality, suggesting that financial characteristics interact with media-driven contagion.

These findings offer practical insights for regulators and investors seeking to monitor financial risks from a broader perspective. By incorporating textual data into systemic risk assessment, this approach complements traditional quantitative models and enhances the ability to detect emerging vulnerabilities. Future research could refine this methodology by integrating machine learning techniques to predict contagion events based on the temporal evolution of news networks. Expanding the analysis to different markets and time periods would further validate the effectiveness of news-based networks in assessing systemic risk. Another approach could explore how news-reported events influence strategic decisions in companies interconnected through shared board memberships, providing valuable insights into the dynamics of information flow and corporate governance.

4 DIGITAL TWINS AND NETWORK RESILIENCE IN THE EU ETS: ANALYZING STRUCTURAL SHIFTS IN CARBON TRADING

The European Union Emissions Trading System (EU ETS) has transitioned from a centralized, hub-dominated network to a more fragmented structure, raising concerns about market efficiency, price stability, and allowance distribution. Regulatory adjustments and shifting trade relationships have altered market connectivity, with some countries forming stable trading clusters while others face increasing isolation. This study examines these structural changes using Digital Twins, complex network analysis, and machine learning to model emissions trading as a dynamic system. The findings suggest that continued fragmentation may disrupt price formation and reduce market integration, potentially affecting liquidity and compliance costs. Predictive modeling reveals that emerging trading barriers could hinder market efficiency, underscoring the need for policymakers to assess whether existing mechanisms sustain competition and emissions reduction targets.

4.1 Introduction

The European Union Emissions Trading System (EU ETS) is the world's largest carbon market and a central component of the EU's climate strategy. As the EU pursues its carbon neutrality goal for 2050, the EU ETS must allocate allowances efficiently and cost-effectively to support this transition (Bouckaert et al., 2021). While most analyses emphasize carbon prices, the structure of trading relationships also plays a critical role in market performance. Firms rely on the trading network to reallocate allowances across sectors, and this structure affects liquidity, access, and the market's ability to respond to shocks. A robust and resilient trading network helps avoid circulation bottlenecks, supports consistent market functioning, and sustains progress toward broader sustainability targets.

In this paper, we use EU ETS transaction data to analyze how the network of trading relationships evolves over time. To address this question, we integrate Digital Twin modeling with machine learning to simulate the entry of new participants and predict future trading links. This forward-looking approach allows us to assess the system's structural resilience under changing regulatory and market conditions. We implement Graph Neural Networks (GNNs) (Zhang & Chen, 2018; Chen & Chen, 2021) and Logistic Regression (Jr, Lemeshow & Sturdivant, 2013; He et al., 2019) to predict the formation of new links in the trading network. To capture shifts in influence and network topology, we compute centrality measures and apply modularity-based community detection (Girvan & Newman, 2002; Blondel et al., 2008). We also use preferential attachment models (Gracious et al., 2021) to understand how new entrants establish relationships within the market. Together, these methods allow us to evaluate whether the structure of trading connections continues to support the core objectives of the cap-and-trade system – namely, efficient and flexible allowance reallocation.

This analysis may help policymakers assess whether the current market structure maintains liquidity, supports efficiency, and sustains emissions reductions as trade volumes grow and regulations change. Structural features of the trading network shape not only economic outcomes but also the system's capacity to achieve climate goals. Hierarchical and asymmetric trading networks can reduce market liquidity and increase transaction costs by limiting access to counterparties and distorting price discovery — for example, by making it harder to match buyers and sellers at competitive prices (Karpf, Mandel & Battiston, 2018). These features not only raise the cost of compliance but may also impair the flexibility needed for effective allowance reallocation. Understanding the structure of trading relationships is therefore essential for anticipating frictions that could undermine both economic and environmental performance within the cap-and-trade framework.

Our study contributes to a growing literature that applies network-based methods to understand the structure and dynamics of the EU Emissions Trading System. Some studies have analyzed the architecture of the carbon market by modeling trading relationships as complex networks. For example, Borghesi & Flori (2018) use centrality measures to identify which countries played key roles in Phases I and II of the EU ETS. They show that a few national registries (e.g., France, Germany, the UK) emerged as structural hubs and that person holding accounts (PHAs) significantly influenced the network's configuration, particularly through strategic account placement. Similarly, Liu, Gao & Guo (2018) examine the growth, structure, and scale-free properties of the EU ETS trading network. They document how the network has evolved over time and demonstrate that firm-level trading activity follows a broken power law, indicating persistent heterogeneity in network connectivity.

Our work also builds on research that links network structure to market efficiency and price dynamics. Karpf, Mandel & Battiston (2018) show that hierarchical trading patterns and asymmetries in the EU ETS transaction network contributed to inefficiencies such as inflated bid-ask spreads and informational frictions. These patterns allowed central actors to extract advantages at the expense of peripheral participants, raising concerns about equity and system-wide effectiveness. While they emphasize static inefficiencies, we extend this line of inquiry by studying the evolution of such structural features over time.

Our work is also related to another strand of the literature that uses network representations to improve carbon price forecasting. Xu et al. (2020) propose a hybrid method that combines complex network features with an extreme learning machine (ELM), showing that the inclusion of network topology improves both level and directional prediction accuracy. Building on this, Xu & Wang (2021) apply visibility graph algorithms to extract topological structures from carbon price time series, further improving the predictive power of various benchmark models. These studies highlight the importance of incorporating network characteristics into empirical models but focus primarily on price trends, not on the underlying trading relationships.

Our methodology extends previous work by integrating Graph Neural Networks and Digital Twin modeling to simulate structural transitions in the market and predict the entry of new connections and participants. This adds a forward-looking and structural dimension to the existing literature, connecting network theory not only to historical analysis but also to the design and resilience of future carbon markets.

Our findings reveal that the EU ETS has shifted from a hub-dominated structure to a more decentralized and fragmented network. While some clusters of trading partners remain stable, others dissolve or reconfigure in response to regulatory shifts and firm-level adjustments. These changes reshape how allowances circulate through the system and directly affect market structure and connectivity. Increased fragmentation raises the risk of localized shortages, higher transaction costs, and reduced access to counterparties — factors that can impair the efficient reallocation of permits. While our analysis focuses on structural dynamics rather than direct environmental or economic outcomes, the patterns we observe may have broader implications. In a cap-and-trade system with price bounds, a fragmented network can reduce the system's ability to self-correct through trade, raising the likelihood of hitting price floors or ceilings. In turn, this may reduce cost-effectiveness and weaken the flexibility needed to support decarbonization. Economic analysis suggests that inefficient allowance allocation can hinder compliance and delay investment in low-carbon technologies (Mattauch et al., 2022; Flori, Borghesi & Marin, 2024). These factors are critical for long-term sustainability. Therefore, our results underscore the importance of monitoring not only price signals but also the underlying structure of trading relationships to preserve the EU ETS's effectiveness and long-term resilience.

The rest of this paper is organized as follows. Section 4.2 provides an overview of the EU ETS. Section 4.3 describes the methodological framework, including network analysis, machine learning techniques, and Digital Twin modeling. Section 4.4 presents the dataset. Section 4.5 discusses the results in three subsections: Statistical Analysis, Complex Network Analysis, and Digital Twin Applications. Section 4.6 outlines the policy implications of our findings.

4.2 European Union Emissions Trading System

Covering around 40% of the EU's total greenhouse gas emissions, the European Union Emissions Trading System (EU ETS) stands as the largest and first multinational carbon market (Jenkins, 2014). It applies to around 10,000 power stations, industrial facilities, and intra-EU flights, making it the primary policy tool for reducing emissions in the region (Ellerman, Convery & Perthuis, 2010). The EU ETS operates under a cap-and-trade mechanism, which places a strict upper limit (cap) on total emissions from covered sectors. Within this framework, entities receive or purchase emission allowances, each granting the right to emit one tonne of carbon dioxide equivalent (Juhász & Lane, 2024). Since the total number of allowances decreases over time, the system creates a financial incentive for firms to reduce emissions while enabling flexibility in compliance through market-based trading (Baudry, Faure & Quemin, 2021).

The EU ETS originates from the 1997 Kyoto Protocol, the first international treaty to establish legally binding emission reduction targets for industrialized nations. To meet these obligations, the European Commission initiated discussions on emissions trading as a cost-effective policy tool. In March 2000, the Green Paper on Greenhouse Gas Emissions Trading laid the groundwork for the system, identifying fundamental design elements and inviting stakeholder input (Convery, 2009). This consultation process shaped the final regulatory framework, balancing economic efficiency with environmental effectiveness (Hepburn et al., 2016). The EU formally launched the ETS in 2005, structuring it into distinct compliance periods, or phases, to allow for progressive refinement of the market.[1]

The first phase (2005–2007) served as a pilot period, establishing an initial carbon price, facilitating emissions trading across the EU, and developing the necessary monitoring, reporting, and verification (MRV) infrastructure. Due to limited historical emissions data, allocation caps were based on estimates, leading to an oversupply of allowances. This surplus caused carbon prices to collapse to zero by 2007, as unused allowances could not be carried over to the second phase (Ellerman & Buchner, 2007). Despite these shortcomings, Phase I provided essential lessons for improving future phases.

Phase II (2008–2012) introduced stricter emissions limits, reducing the cap by approximately 6.5% below 2005 levels. Three additional countries – Iceland, Liechtenstein, and Norway – joined the system, and nitrous oxide (N_2O) emissions from nitric acid production were included. Free allocation of allowances declined slightly to around 90%, with some countries introducing auctions. Regulators set a penalty of €100 per excess tonne of CO_2 emissions to enforce compliance. Firms could also offset emissions by purchasing international credits, totaling approximately 1.4 billion tonnes of CO_2 emissions. A major structural change was the transition to a centralized Union Registry, replacing national registries for allowance tracking, alongside the introduction of the European Union Transaction Log (EUTL) to monitor compliance (Borghesi, 2011). The aviation sector joined the ETS in 2012, though authorities temporarily suspended enforcement for flights to and from non-European countries. While Phase II benefited from more accurate emissions data, the 2008 financial crisis unexpectedly reduced industrial activity, leading to a surplus of allowances and a prolonged period of low carbon prices (Koch et al., 2016).

Recognizing the inefficiencies of previous phases, Phase III (2013–2020) introduced significant reforms to improve market stability and effectiveness. National emission caps were replaced with a single EU-wide cap, ensuring uniformity across member states. The allocation of allowances shifted from predominantly free allocation to auctioning as the default method, reducing market distortions. The ETS adopted harmonized rules for the remaining free allocations, giving priority to sectors vulnerable to carbon leakage. Additional sectors and greenhouse gases were brought under regulation, while a reserve of 300 million allowances (NER 300) was set aside to finance renewable energy innovation and carbon capture and storage (CCS) projects. These adjustments increased price stability and encouraged investment in low-carbon technologies (Kollenberg & Taschini, 2019). Studies suggest that these reforms reduced emissions without negatively impacting economic competitiveness (Dechezleprêtre, Nachtigall & Venmans, 2023).

Phase IV (2021–2035) aligns with the EU's broader climate strategy under the European Green Deal and the Fit for 55 legislative package. The overarching objective is to achieve climate neutrality by 2050, with an intermediate goal of reducing net greenhouse gas

[1] See Sato et al. (2022) for a critical review of the EU ETS evolution during Phases I–IV.

emissions by at least 55% by 2030. This phase implements several structural adjustments, including a steeper annual reduction of the emissions cap to accelerate decarbonization, a revised allocation system ensuring a gradual transition from free allocation to full auctioning, and an expansion of the Market Stability Reserve (MSR) to address allowance surplus and improve price resilience (Dubois, Sahuc & Vermandel, 2025). Additionally, Phase IV introduces carbon pricing mechanisms for previously uncovered sectors, such as shipping and road transport. Transaction costs remain an ongoing concern, as they may impact liquidity and market efficiency in future phases (Baudry, Faure & Quemin, 2021). By integrating these reforms, the EU ETS aims to improve market efficiency while providing a robust framework for achieving long-term emissions reductions.

4.3 Methodology

This section outlines the methodological framework we use to analyze the structure, stability, and evolution of the EU ETS trading network. In Section 4.3.1, we describe how we intend to represent the EU ETS as a complex network. Section 4.3.2 introduces the Digital Twin framework, which allows us to simulate the system's evolution by incorporating historical trading data and modeling structural changes under different regulatory scenarios. In Section 4.3.3, we extend this approach by integrating machine learning techniques into the Digital Twin framework to improve the prediction of future trading relationships.

4.3.1 Complex Network Representation of the EU ETS

We intend to model the European Union Emissions Trading System (EU ETS) as a complex network to analyze its structural characteristics and trading relationships. In this framework, grounded in graph theory, nodes represent countries, and directed edges capture the flow of emissions allowances between them (Zanin et al., 2016). We assign weights to the edges based on the volume of allowances transferred, enabling a quantitative evaluation of trading intensity and patterns. This network-based approach goes beyond the analysis of isolated transactions by capturing the broader structure of allowance exchanges. Countries with high connectivity occupy central positions in the network, shaping liquidity and potentially influencing price dynamics. By examining the network across different phases of the ETS, we track structural shifts driven by regulatory reforms or external shocks.

To identify which countries are more central and how these roles change over time, we introduce centrality and network density measures indicators in Section 4.3.1.1 and 4.3.1.2, respectively. These metrics also help assess the degree of market integration. In Section 4.3.1.3, we present the Louvain method for community detection, a widely used algorithm to identify trading clusters. This approach allows us to evaluate how trading relationships evolve and whether cohesive subgroups emerge or dissolve throughout the system's development.

4.3.1.1 Centrality Measures

Centrality measures quantify the relative importance of nodes, revealing influential players and their roles in the emissions market. Degree centrality evaluates how well-connected a country is within the trading network, distinguishing between sources and recipients of allowances.

The degree centrality $C_D(v)$ of a node v is defined as:

$$C_D(v) = \frac{k_v}{n-1}, \tag{4.1}$$

where k_v represents the number of edges connected to node v, and n is the total number of nodes. A country's degree centrality indicates its participation level in allowance trading. The in-degree measures the number of distinct counterparties transferring allowances to a country, while the out-degree represents destinations for outgoing transactions.

The degree distribution $P(k)$ describes the probability that a randomly selected node has degree k:

$$P(k) = \frac{\text{Number of nodes with degree } k}{\text{Total number of nodes}}. \tag{4.2}$$

Analyzing this distribution determines whether a few countries dominate trading activity or if participation is more evenly spread. A skewed degree distribution suggests market concentration, while a flatter distribution implies broader participation.

4.3.1.2 Network Density

Network density measures how interconnected the system is by comparing the observed number of edges to the total possible connections. For a directed network, density D is calculated as:

$$D = \frac{m}{n(n-1)}, \tag{4.3}$$

where m is the number of edges and n is the number of nodes. A higher density indicates a more active trading environment with greater market integration. Tracking density changes across EU ETS phases identifies whether trading relationships have become more concentrated or diversified over time.

4.3.1.3 Community Detection via the Louvain Method

Identifying groups of countries that frequently trade allowances provides insights into market segmentation and trading clusters. We apply the Louvain method (Blondel et al., 2008) for community detection, which partitions the network into groups of nodes with stronger internal connections than external links. The modularity score Q measures the effectiveness of this partitioning:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{i,j} - \frac{K_i K_j}{2m} \right) \delta(c_i, c_j), \tag{4.4}$$

where $A_{i,j}$ is the weight of the edge between nodes i and j, K_i and K_j are the total edge weights of these nodes, m represents the total edge weight in the network, and $\delta(c_i, c_j)$ equals 1 if nodes i and j belong to the same community and 0 otherwise. A higher modularity score indicates well-defined trading clusters.

The Louvain method follows an iterative process: it first assigns each node to its own community, then reassigns nodes to maximize modularity gains, and finally aggregates communities into meta-nodes before repeating the process. This method efficiently detects trading blocs within the EU ETS network. Applying community detection across multiple ETS phases determines whether trading clusters remain stable or shift due to regulatory or economic factors. Persistent clusters suggest long-term trading alliances, while frequent reconfigurations indicate market adjustments to policy interventions or external shocks.

4.3.2 Digital Twins

We employ Digital Twins to simulate the evolution of the EU ETS network during Phase IV, which began in July 2021. Digital Twins are virtual representations of physical systems that integrate real-world data with simulation models to track changes, predict future states, and support decision-making (Grieves, 2014; Batty, 2018). Originally developed for engineering and industrial applications, they have gained prominence in network science for analyzing dynamic systems, including communication networks, social structures, and power grids. In the context of the EU ETS, Digital Twins offer a framework for evaluating structural evolution, stability, and resilience by incorporating historical trading data and modeling network dynamics (Gupta, Iyer & Kumar, 2025).

We use three dynamic mechanisms to simulate the structural evolution of the EU ETS network within the Digital Twin framework. These mechanisms are designed to capture how the trading network adapts as new participants enter, existing relationships shift, and communities reorganize. The first mechanism models node entry and link formation based on preferential attachment: new participants are more likely to connect with well-established, highly connected nodes, reflecting real-world trading behavior. The second mechanism introduces edge rewiring, selectively replacing a portion of existing links to account for changing partnerships and evolving trade preferences. The third mechanism enables community reorganization, whereby closely interconnected clusters merge if the density of inter-community links exceeds a predefined threshold — simulating market integration or consolidation.

We initialize the Digital Twin using transaction data from Phases I–IV, assigning node attributes such as centrality measures and community memberships, and encoding edge properties including weight, direction, and transaction frequency. These mechanisms together allow the model to simulate the transition into Phase IV and capture anticipated changes in trading relationships (Topirceanu, Udrescu & Marculescu, 2018; Papachristou & Yuan, 2024).

The preferential attachment model, introduced by Barabási & Albert (1999), describes how new nodes tend to connect to those that already have high connectivity, following a "rich-get-richer" mechanism. In our model, the probability of a new edge forming between a new node and an existing node i is given by:

$$P(i) = \frac{k_i}{\sum_j k_j}, \tag{4.5}$$

where k_i represents the degree centrality of node i, and the denominator sums the degree centralities of all nodes in the system. This mechanism reflects real-world market behavior, where well-established participants are more likely to attract new trading partners. In addition to new node connections, we model edge formation based on combined degree centrality, ensuring that high-degree nodes continue to shape network structure. The probability of forming an edge between two nodes i and j is:

$$P(i \to j) = \frac{\text{Degree}(i) + \text{Degree}(j)}{\sum_{k,l}(\text{Degree}(k) + \text{Degree}(l))}. \tag{4.6}$$

By incorporating preferential attachment into the Digital Twin framework, we generate a more realistic evolution of trading relationships, where highly connected nodes remain influential while still allowing new links to emerge dynamically (Albert & Barabási, 2002; Jeong, Néda & Barabási, 2003).

The edge rewiring mechanism modifies existing trade relationships by selectively replacing connections. The algorithm evaluates each edge and, with a 20% probability, removes and replaces it with a new connection. The reassignment process ensures that new edges do not create duplicate links or self-loops. This mechanism captures shifts in trading relationships, reflecting how some participants disengage from prior connections and establish new ones.

The community reorganization mechanism adjusts the network structure based on internal edge density. First, the Louvain method detects initial communities. The algorithm then measures the density of inter-community connections and identifies clusters with strong cross-links. When the density exceeds a predefined threshold, the model merges the communities, simulating commercial integration as previously distinct groups consolidate through intensified trading relationships.

Beyond structural modeling, Digital Twins enable scenario analysis by simulating network responses to policy changes, economic shifts, and regulatory adjustments. Policymakers can test alternative allocation schemes, assess the impact of adding new industrial sectors, or evaluate how interventions affect trading relationships. For example, restricting transactions involving major hubs could expose vulnerabilities in network connectivity and help identify potential systemic risks (Holt & Shobe, 2013; Horn, 2015; Coalition, 2020).

Market participants can also use Digital Twins to refine trading strategies, anticipate price fluctuations, and detect anomalies in transaction patterns. By synchronizing with real-time data, the Digital Twin enhances transparency, supports risk mitigation, and strengthens informed decision-making (Kaewunruen et al., 2021; Hezam et al., 2024).

Ultimately, integrating Digital Twins with complex network analysis improves our ability to forecast developments in the EU ETS. This framework provides a structured tool for anticipating new trading relationships, ensuring regulatory compliance, and designing more adaptive and effective climate policies (Abayadeera & Ganegoda, 2024).

4.3.3 Combining Digital Twins with Machine Learning

Machine Learning (ML) extends the predictive capabilities of Digital Twins, allowing for more precise simulations of network evolution and improving the ability to anticipate structural changes. By learning patterns from past EU ETS phases, ML models identify essential factors influencing edge formation and evolving connectivity dynamics. This integration strengthens the capacity to forecast new trading relationships and detect shifts in the emissions trading system.

The modeling process consists of multiple stages. First, historical data from Phases I–IV serve as both training and validation sets, incorporating key features such as node degree centrality, edge weights, directional flows, and community properties, including modularity and node density. These features capture structural and behavioral aspects of the EU ETS network, providing a foundation for predicting future link formations. The core objective is to model and predict new edge formations using supervised learning, reflecting how economic, regulatory, and network-driven forces shape the evolution of trading relationships.

To achieve this, we implement two predictive approaches: Graph Neural Networks (GNNs) and Logistic Regression. GNNs capture complex dependencies between nodes through iterative message passing, enabling a more refined representation of network structure and link probabilities. Logistic Regression, in contrast, offers a simpler yet effective approach, using node-pair embeddings to estimate the probability of edge formation based on predefined structural features. The combination of these methods ensures a balance between interpretability and predictive performance.

Integrating Digital Twins with ML-based predictive modeling allows us to identify emerging trading relationships and anticipate structural adjustments in the EU ETS network. This approach applies network science principles to empirical data, providing a clearer view of market evolution in Phase IV.

4.3.3.1 Graph Neural Network

Graph Neural Networks (GNNs) model graph-structured data by learning node representations through iterative message passing. This approach allows each node to incorporate information from its neighbors, gradually expanding its receptive field across multiple layers (Gkarmpounis et al., 2024; Wu et al., 2020). The node embedding at layer l is updated as:

$$h_v^{(l+1)} = \sigma \left(W^{(l)} \cdot h_v^{(l)} + \sum_{u \in \mathcal{N}(v)} \frac{1}{c_{vu}} W^{(l)} \cdot h_u^{(l)} \right), \tag{4.7}$$

where $h_v^{(l)}$ represents the embedding of node (country) v at layer l, and $\mathcal{N}(v)$ denotes its set of neighboring countries. The normalization term c_{vu} is typically set as the number of neighbors or adjusted based on edge weights. The trainable weight matrix $W^{(l)}$ is updated during training, and σ is the activation function. ReLU is used within the GCN layers, while a Sigmoid function generates link existence probabilities between nodes.

GNNs operate through three main steps: message aggregation, node state updates, and final prediction. The aggregation step collects information from neighboring nodes:

$$m_v^{(l)} = \sum_{u \in \mathcal{N}(v)} \text{MSG} \left(h_u^{(l)}, h_v^{(l)} \right), \tag{4.8}$$

where MSG$(\cdot)$ is a function that processes neighbor information. The node state update follows:

$$h_v^{(l+1)} = \text{UPD} \left(h_v^{(l)}, m_v^{(l)} \right), \tag{4.9}$$

where UPD$(\cdot)$ incorporates new information. After multiple layers, final embeddings serve as inputs for classification tasks such as edge prediction:

$$P(y = 1|X) = \text{Sigmoid} \left(W^{(L)} h_v^{(L)} \right). \tag{4.10}$$

Variants such as Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs) (Veličković et al., 2017), and GraphSAGE (Hamilton, Ying & Leskovec, 2017a) provide alternative approaches for capturing different network properties.

4.3.3.2 Logistic Regression

Logistic Regression serves as a complementary predictive method, offering a simpler yet effective approach for classifying new link formations. The model estimates the probability that an edge exists between two nodes based on structural features such as degree centrality, common neighbors, and preferential attachment. The probability of edge formation follows:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{3} \beta_i x_i)}}, \tag{4.11}$$

where x_1 represents node degree, x_2 captures the number of shared neighbors, and x_3 models the preferential attachment mechanism. Positive examples are derived from existing edges, while negative examples are randomly sampled non-edges. Negative sampling ensures the model learns to distinguish real connections from randomly generated node pairs. To optimize model performance, we split the dataset into 80% training and 20% testing, using cross-validation to prevent overfitting. The final model classifies node pairs based on a probability threshold, typically set at 0.5:

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|X) \geq 0.5, \\ 0 & \text{if } P(y = 1|X) < 0.5. \end{cases} \tag{4.12}$$

4.3.3.3 Evaluation Metrics

To assess the predictive accuracy of our models, we employ several widely used classification metrics (James et al., 2013). Given that our objective is to predict the formation of new trading relationships in the EU ETS, it is essential to evaluate how well the model distinguishes between actual and predicted edges. One of the primary tools for model evaluation is the Receiver Operating Characteristic (ROC) curve, which illustrates the tradeoff between the true positive rate (sensitivity) and the false positive rate at different classification thresholds. The Area Under the Curve (AUC) quantifies the model's ability to differentiate between edges (existing or future links) and non-edges (absence of a trading relationship). A higher AUC indicates a stronger predictive capability, as the model effectively ranks true edges higher than false ones.

Since the dataset is inherently imbalanced – new edges are relatively rare compared to non-edges – standard accuracy alone is not a reliable measure of performance. Instead, we employ precision, recall, and F1-score to provide a more nuanced evaluation. Precision measures the proportion of predicted edges that are actual edges. A high precision value indicates that the model makes fewer false positive predictions, meaning it does not mistakenly classify non-existent links as valid trading relationships. This is particularly important in regulatory and market analyses, where incorrectly predicting a link could lead to misleading conclusions about emerging trading structures. Recall (or sensitivity) measures the proportion of actual edges that the model correctly identifies. A high recall ensures that most existing or emerging trading relationships are captured, even if it means allowing some false positives. This metric is indispensable when the priority is to identify as many potential link formations as possible, even at the risk of occasional misclassification. F1-score is the harmonic mean of precision and recall, balancing both concerns. It provides a single measure of model performance when there is a tradeoff between precision and recall. This is particularly relevant in our context, as overlooking a potential trading relationship (false negative) and incorrectly predicting an edge (false positive) have different implications for market analysis.

Given the complexity of emissions trading networks, an ideal model should achieve a balance between high precision and high recall, ensuring that predicted trading relationships are both accurate and comprehensive. We report these metrics to evaluate the robustness of our predictions.

4.4 Data

We use transaction and account data from the European Union Transaction Log (EUTL), which records all transfers of emissions allowances within the EU ETS.[2] The dataset contains 1,997,165 records, including details on transaction IDs, transaction dates, transferring and acquiring account IDs, and the volume of allowances exchanged. Additionally, account-level data provide information on the account holder, account type, and the country where the account is registered. We present a detailed dictionary of these variables in Tables 13, and 14 (Appendix B.1).

The EUTL dataset distinguishes between the registry administering an account and the country where it operates. An entity can register an account in a different country due to factors such as regulatory requirements, fiscal advantages, or the need to access specific exchange platforms that mandate registry compliance (Borghesi & Flori, 2018). However, according to Annex XIV(4) of Regulation 389/2013, transaction records in the EUTL are only made public on May 1 of the third year following the transaction date. [3]This delayed disclosure affects the timeliness of market analysis.

A fundamental limitation of this dataset is its focus on transfers rather than direct market trades, leaving out critical transaction details such as trade execution prices. This issue occurs because allowance transactions often take place in futures markets, where agreements settle at a later date, and the final transferred amount may not reflect intra-day price variations. Without observed trading prices, assessing supply and demand dynamics

[2] The routines to extract the data sources are available at https://github.com/jabrell/eutl_scraper

[3] See the Commission Regulation (EU) No389/2013 for further details.

at specific points in time becomes difficult, limiting the ability to evaluate market efficiency and price formation mechanisms.

Another challenge is the loss of direct counterparty information. Because intermediaries, such as clearinghouses, clear many transactions, the dataset does not reveal the original buyer-seller relationships. This complicates network analysis, as the actual trading structure cannot be fully reconstructed. The settlement process effectively obscures trading relationships, preventing the identification of major market participants, their influence, and the emergence of trading clusters.

The dataset also reveals a recurring spike in transaction volume every December, reflecting regulatory deadlines and the settlement of futures contracts. These annual surges underscore the discrepancy between observed transfers and actual market trades, reinforcing the need for more granular data, such as real-time trade records and transaction prices, to improve market analysis.

In the absence of complete transaction-level data, network modeling offers an alternative approach to infer interactions between market participants. Historical patterns can help reconstruct missing relationships, allowing for a more detailed examination of market structure and dynamics. However, the dataset's limitations underscore the need for access to more detailed records to fully capture trading behavior within the EU ETS.

4.5 Results

In this section, we examine the evolution of emissions trading in the EU ETS. Section 4.5.1 summarizes transaction patterns, showing a concentrated trading structure where a few installations dominate activity. Trading is unevenly distributed, with industrialized regions serving as key hubs. Section 4.5.2 analyzes the EU ETS network across Phases I-IV, revealing a shift from centralized to fragmented structures. Community detection captures trading cluster dynamics, while centrality measures track the changing influence of participants. Section 4.3.3 employs Digital Twins to simulate network evolution, modeling node additions, edge rewiring, and structural shifts. These simulations identify emerging trading relationships and market participants. Finally, Section 4.5.4 integrates machine learning models – Graph Neural Networks (GNNs) and Logistic Regression – within the Digital Twin framework to predict future network configurations. While Logistic Regression reinforces established hubs, GNNs capture emerging connections, enhancing our understanding of market structure and regulatory impacts.

4.5.1 Descriptive Statistics

This subsection explores transaction patterns and installation activity within the EU ETS. By summarizing the data, we identify relevant trends in trading frequency, installation participation, and transaction volumes. Figure 10 shows the spatial distribution of installations and transaction densities across Europe. Trading activity is highly concentrated in industrialized regions, with notable clusters in the United Kingdom, Germany, France, and the Benelux area. Major urban and economic centers – such as London, Paris, and Berlin – exhibit significant transaction volumes.

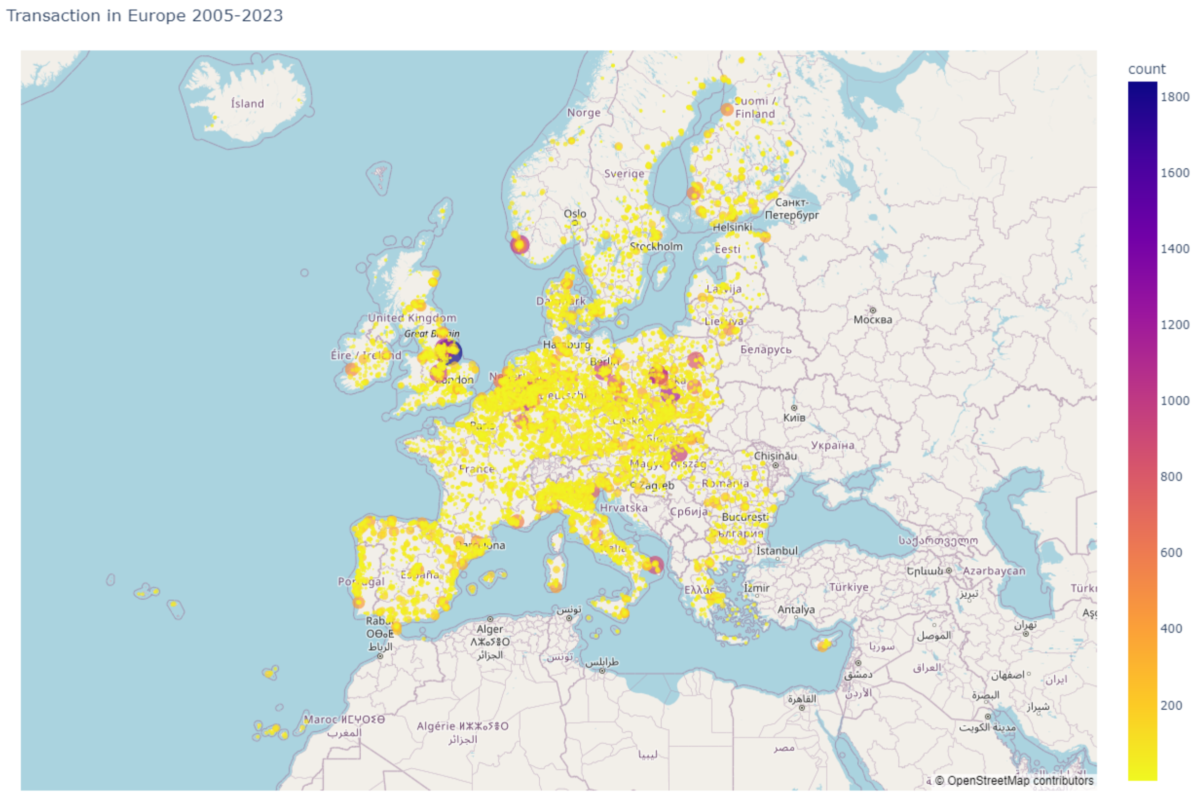


Figure 10 – Map of installations with the number of transactions between 2005 and 2023.

In contrast, trading density diminishes in Northern and Eastern Europe, with sparser activity in rural regions and along the system's geographic periphery. The observed distribution shed lights on the central role of industrial hubs in emissions trading. While the EU ETS covers a broad geographic range, transactions are unevenly distributed, reflecting economic activity and regulatory engagement. Installations appear more frequently in regions with established industrial bases, reinforcing the network's core trading relationships.

	Obs	**Mean**	**Std. Dev.**	**Min**	**25%**	**50%**	**75%**	**Max**
Transactions per installation	16,898	235.79	218.51	1	92.5	185	305.5	1840

Table 6 – Descriptive statistics of the number of transactions per installation.

Table 6 summarizes the descriptive statistics of installation participation. Across 16,898 installations, the average installation engaged in 236 transactions, with a median of

185. The interquartile range (IQR) spans from 92 to 305 transactions, indicating moderate variability. However, extreme outliers – visible in the middle panel – illustrate installations that participate disproportionately. Any installation with over 625 transactions qualifies as an outlier, underscoring the imbalance between highly active participants and those engaging sporadically.

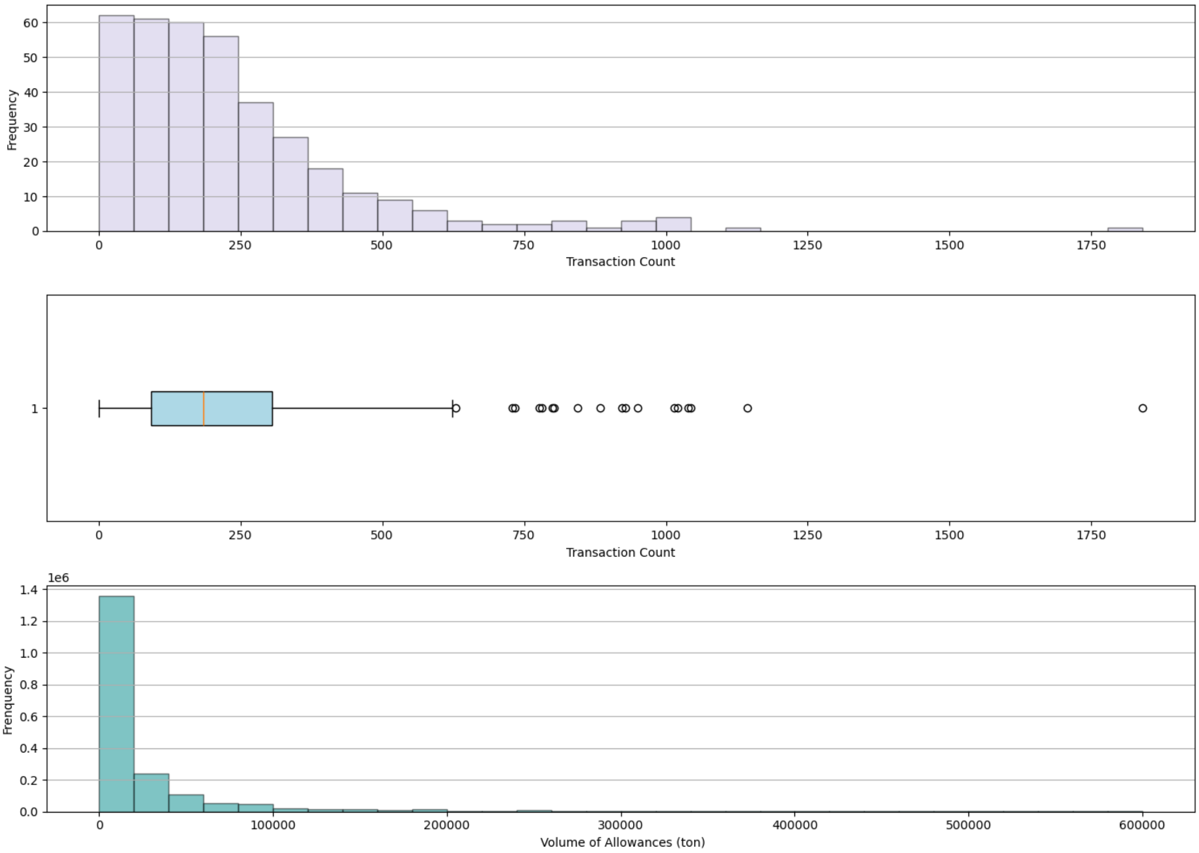


Figure 11 – **Top**: distribution of transactions per installation. **Middle**: boxplot of transaction count per installation. **Bottom**: volume of Allowances (ton).

Figure 11 provides a closer look at transaction frequency across installations. The top panel reveals that most installations engage in relatively few transactions, with a long right-skewed tail indicating the presence of highly active participants. While some installations appear only once, the most active installation recorded 1,840 transactions. The panel at the bottom of Figure 11 illustrates the distribution of transaction volumes in tonnes of allowances. The histogram confirms a highly skewed distribution, where most transactions involve relatively small quantities, while a small number of trades account for exceptionally large volumes. This pattern suggests that a few installations play a dominant role in market activity, either due to their regulatory obligations or strategic trading behavior.

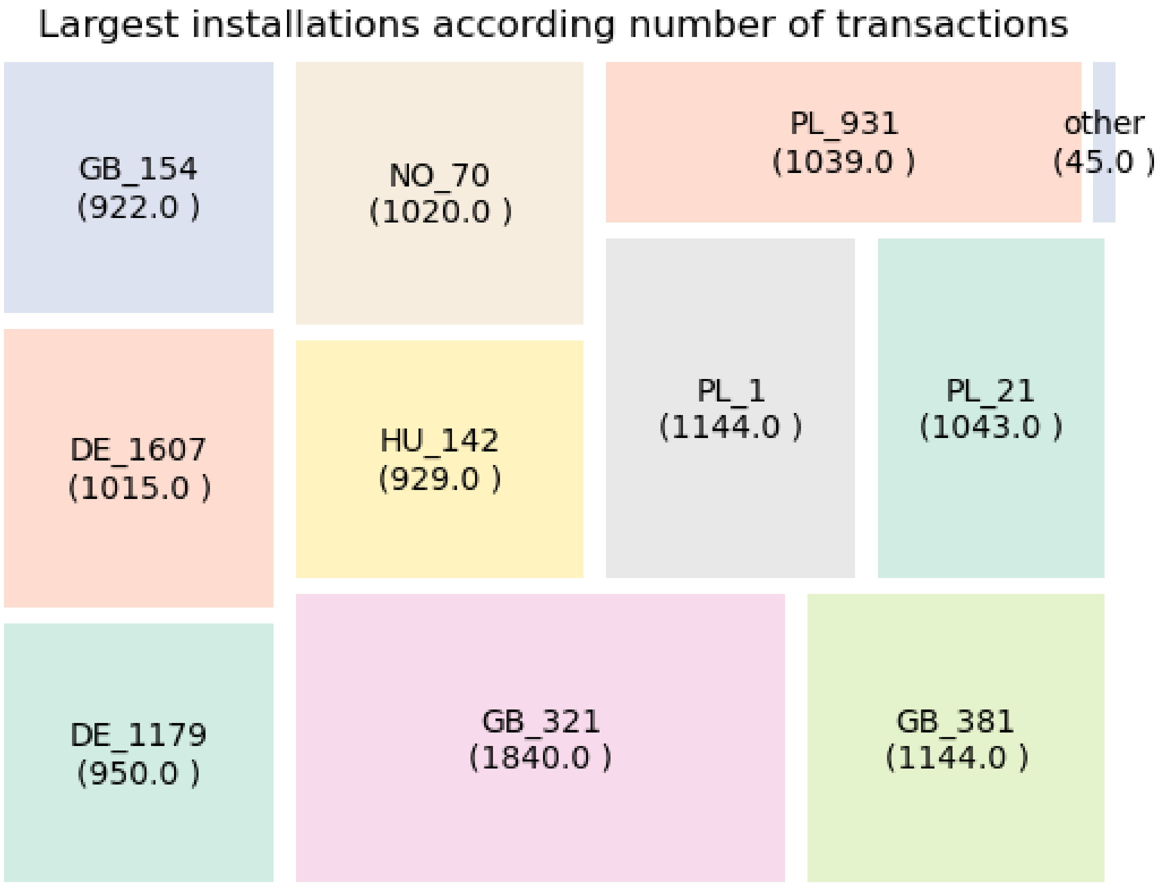


Figure 12 – Largest installations by number of transactions in the EU ETS (2005–2023). GB refers to Great Britain, NO to Norway, PL to Poland, DE to Germany, HU to Hungary.

Figure 12 further emphasizes market concentration, displaying the installations with the highest transaction counts. Three installations in Great Britain (GB_321, GB_381, and GB_154) top the list, reinforcing the country's prominent role in emissions trading. Poland (PL) follows closely, with multiple installations ranking among the most active. Germany (DE), Norway (NO), and Hungary (HU) also feature prominently. The treemap[4] reveals a concentrated trading structure, where a small number of installations drive a significant portion of the market's activity. A minor segment labeled "other" represents installations with much lower transaction counts, illustrating a stark contrast between frequent and infrequent participants.

[4] Each rectangle in the treemap represents an installation, with size proportional to its number of transactions in the EU ETS from 2005 to 2023. Labels follow the format `CC_ID`, where `CC` is the ISO country code and `ID` is an anonymized identifier. The "other" category aggregates all remaining installations with low transaction volume.

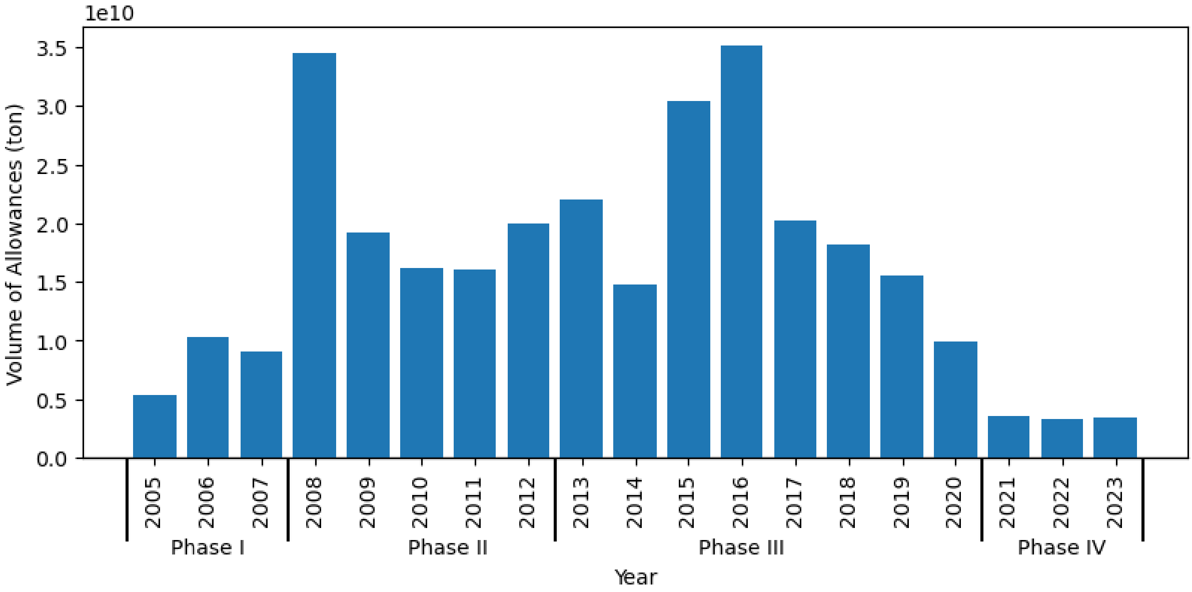


Figure 13 – Trading volume of allowances by phase in the EU ETS (2005–2023).

Figure 13 presents trading volumes across the four phases of the EU ETS. Phase I (2005–2007) exhibits a steady but modest increase in trading activity, serving as a test period for market mechanisms. Phase II (2008–2012) sees a significant jump in volume, peaking in 2008 as regulatory frameworks stabilize. Volumes remain high but fluctuate, reflecting economic conditions and market adjustments. Phase III (2013–2020) records the highest trading volumes, particularly in 2015 and 2016, when regulatory changes and policy shifts likely influenced trading behavior. After 2016, a gradual decline occurs, possibly due to market saturation, regulatory constraints, or allowance supply adjustments. Phase IV (2021–2023) introduces structural reforms that lead to a sharp reduction in trading volumes. Stricter emissions caps and adjustments in allocation mechanisms contribute to this decline, reflecting the evolving nature of the EU ETS.

The descriptive analysis reveals clear patterns in transaction frequency, volume, and market structure. While the EU ETS spans thousands of installations, a subset of highly active participants drives the bulk of transactions. The skewed distribution of transaction counts and allowance volumes underscores the presence of dominant market players. A complex network approach offers deeper insights beyond these summary statistics. By modeling the EU ETS as a network of trading relationships, we can capture how installations interact, detect structural shifts, and anticipate evolving trading patterns. The next section applies network science techniques to map trading relationships and assess market dynamics.

4.5.2 Structural Evolution of the EU ETS Network

We construct networks for Phases I-IV to represent emissions allowance transactions over each period. Figures 14, 15, 16, and 17 visualize the evolving network structure, highlighting communities and centrality. Communities are detected using the Louvain method, while node sizes reflect degree centrality, indicating the relative importance of each country. Different colors distinguish distinct trading communities, allowing for a clearer understanding of network segmentation.

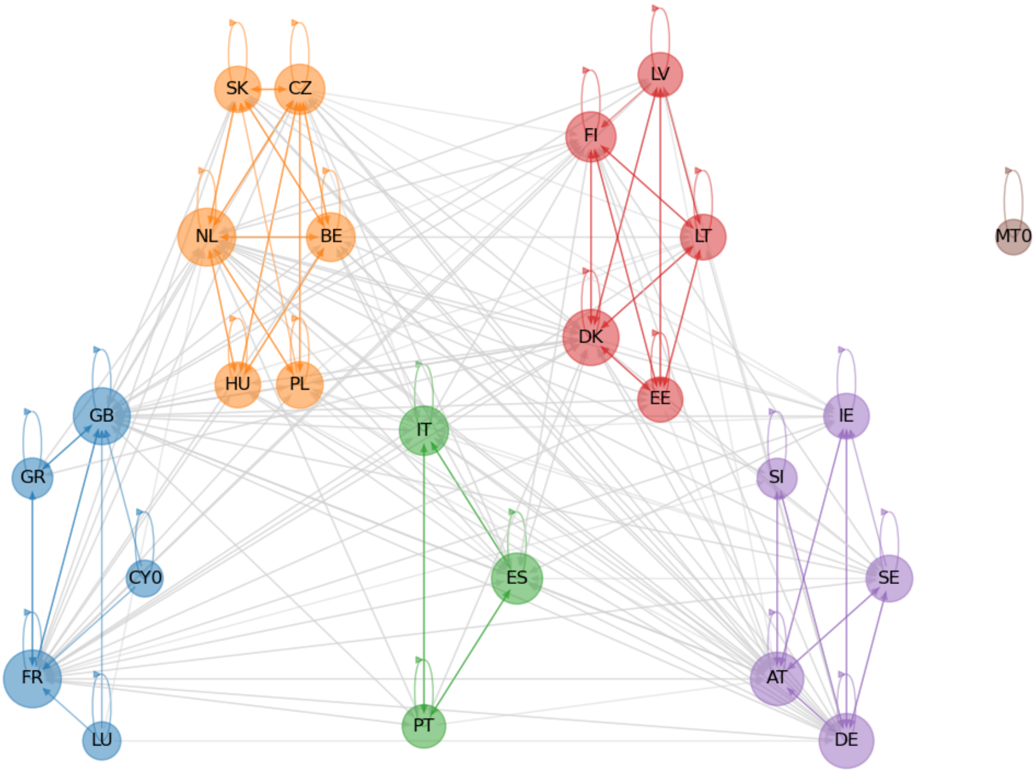


Figure 14 – Phase I - 2005-2007 - Network: Separated Communities and Degree Centrality

Self-links in network representations indicate transactions where the source and target accounts belong to the same country. This occurs because emissions trading within the EU ETS involves allowance transfers between different accounts, which may be administered under the same national registry. For example, installations within the same country frequently engage in internal trades, either due to corporate ownership structures, compliance adjustments, or strategic trading decisions. These self-loops capture domestic trading activity, distinguishing it from cross-border transactions and providing insights into how emissions allowances circulate within national markets.

In Phase I (Figure 14), the network is relatively dense, consisting of 25 nodes and 292 edges, yielding a network density of 0.487. The average degree is 23.36, meaning each country is, on average, linked to over 23 others. A few central nodes dominate connectivity, creating a moderately centralized structure. The six detected communities vary in size, with the largest including five countries. The United Kingdom (GB) and France (FR) are central within their communities, while other important hubs include the Netherlands (NL), Spain (ES), and Denmark (DK). Germany (DE) and Austria (AT) compete for centrality in their cluster. Notably, Malta (MT0) appears as a completely isolated entity, indicating its detachment from broader market interactions.

Phase II (Figure 15) shows substantial expansion, with the network growing to 42 nodes and 727 edges. Despite this, network density declines slightly to 0.422, indicating a broader but somewhat less interconnected market. The average degree increases to 34.61, suggesting higher interaction frequency. Degree centrality becomes more distributed, indicating the emergence of additional hubs. The number of communities rises to eight, reflecting structural diversification. Some previously distinct groups merge, reducing modularity and increasing inter-community connections. New entrants such as Bulgaria (BG), Switzerland (CH), Iceland (IS), Liechtenstein (LI), Norway (NO), Romania (RO), and Ukraine (UA) expand the network, further increasing complexity. Germany (DE) and the Netherlands (NL) solidify their bridging roles. Additional isolated nodes emerge, such as Cyprus (CY0) and Croatia (HR), suggesting that certain countries remained outside the primary trading clusters.

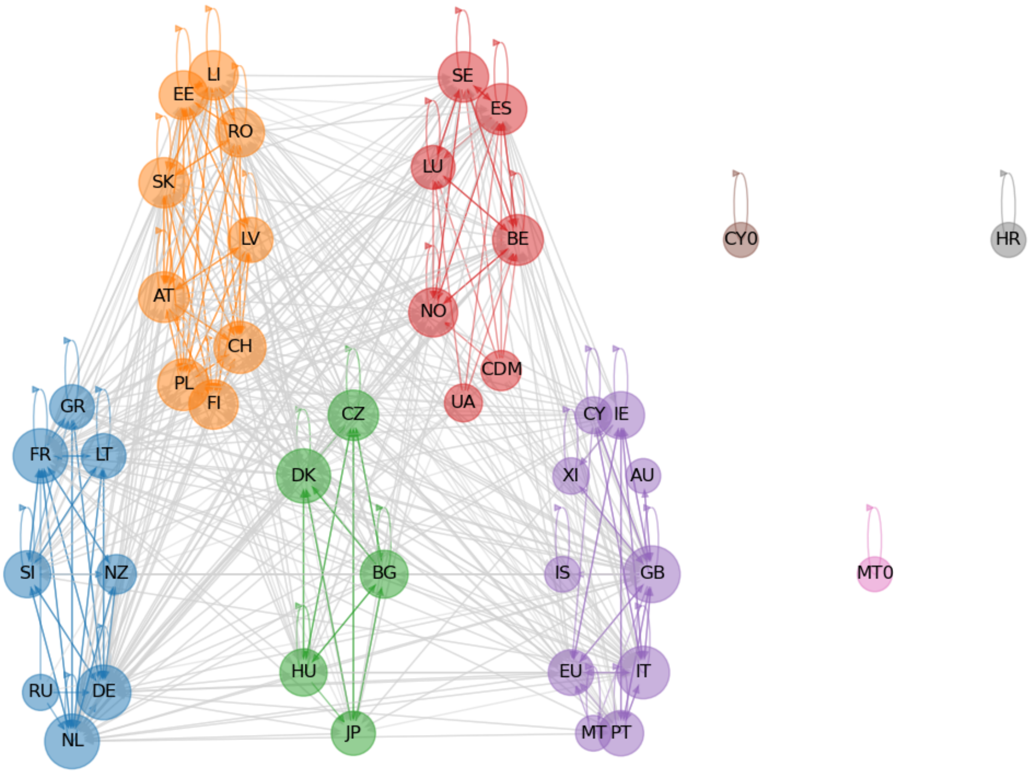


Figure 15 – Phase II - 2008-2012 - Network: Separated Communities and Degree Centrality

In Phase III (Figure 16), the network becomes denser, comprising 40 nodes and 810 edges. The network density increases to 0.519, with the average degree rising to 40.5. The number of communities reduces to five, indicating a trend toward consolidation. The largest community consists of 11 nodes, reflecting tighter integration among certain groups. The overall structure becomes more interconnected, with prominent subgroups forming within communities. Intra-community connections strengthen, suggesting that emissions trading relationships solidify over time. While inter-community edges remain selective, preferential attachment mechanisms drive new links toward already well-connected nodes. Countries like France (FR), the United Kingdom (GB), and Germany (DE) maintain their centrality, reinforcing their influence. Smaller clusters persist, reflecting localized trading patterns, potentially influenced by regional regulations or strategic agreements.

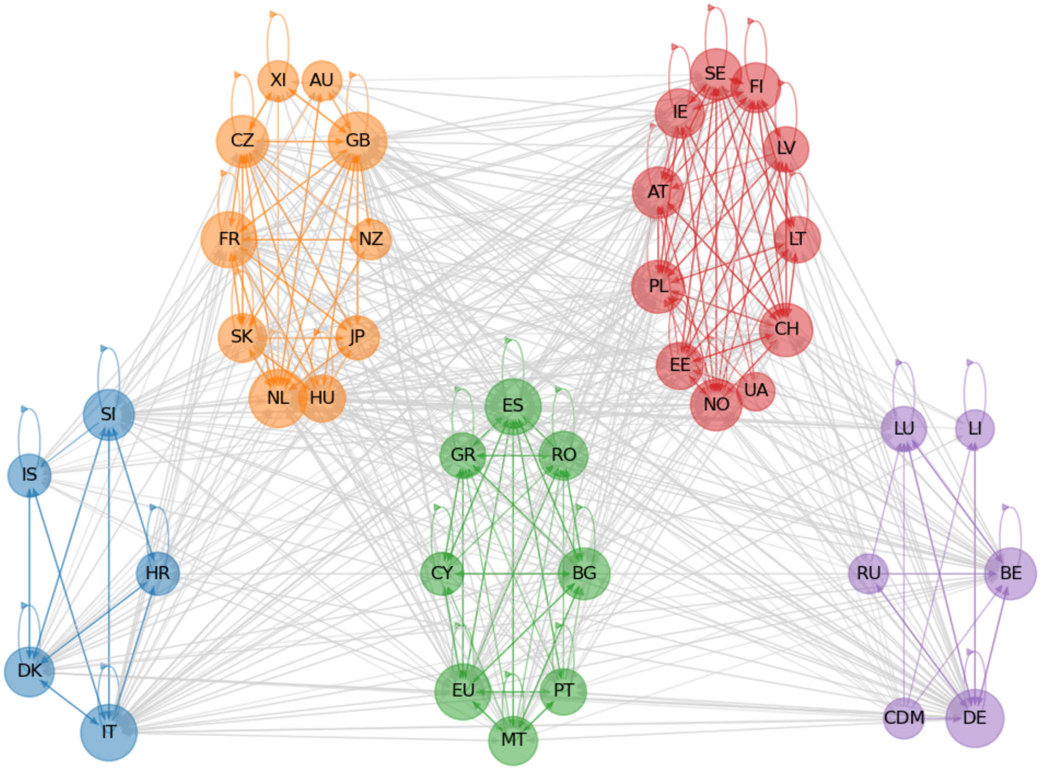


Figure 16 – Phase III - 2013-2020 - Network: Separated Communities and Degree Centrality

Phase IV (Figure 17) presents a stark structural shift. The number of nodes declines to 29, while the number of edges drops dramatically to 58, resulting in a sparse network density of 0.071. The average degree plummets to 4.0, signaling a fragmented system. The network disintegrates into 21 communities, with most containing only one or two nodes, reflecting a significant loss of interconnectivity. Notably, the “EU” node emerges as a dominant entity, indicating that many participants now consolidate transactions under a single European registry. This shift likely stems from regulatory adjustments and centralized compliance mechanisms. The drastic fragmentation suggests reduced cohesion, possibly driven by structural changes in allowance allocation and the impact of the COVID-19 pandemic on emissions trading.

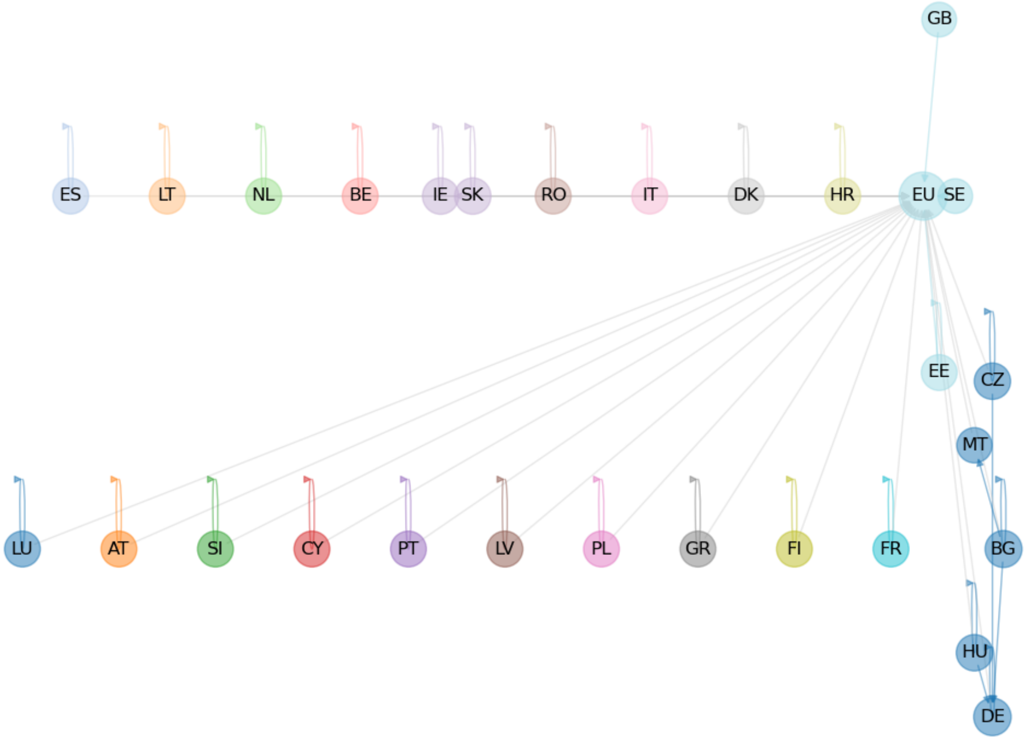


Figure 17 – Phase IV - 2021-2023 - Network: Separated Communities and Degree Centrality

Overall, network analysis reveals a progression from a fragmented system in Phase I to a more interconnected market in Phases II and III, followed by substantial disintegration in Phase IV. This evolution underscores the changing dynamics of the EU ETS, emphasizing the effects of regulatory reforms and external disruptions on market structure.

Table 15 (Appendix B.1) presents the community transitions of countries across Phases I to IV. Each country is assigned a community ID for each phase, allowing us to track structural changes in the network over time. Some countries, such as Lithuania (LI) and Slovenia (SI), remained within the same communities across multiple phases, indicating stable roles in the emissions trading network. Conversely, countries like France (FR), the United Kingdom (GB), and Germany (DE) shifted across different communities, reflecting dynamic participation and evolving market interactions. Some countries, such as Cyprus (CY0) and Malta (MT0), appeared in earlier phases but became disconnected in later ones. The entry of new participants, such as Croatia (HR) in Phase II, points out the network's expansion. Community evolution also reveals the role of influential participants. Larger communities, such as those including Germany (DE), the United

Kingdom (GB), and France (FR), suggest their bridging function in the market. Smaller, isolated communities, such as Malta (MT0), indicate limited interaction with the broader network. Over time, the network exhibits both fragmentation and consolidation, with some communities merging into larger structures, such as Poland (PL) and Bulgaria (BG) in Phase IV, while others split into new clusters. These shifts likely result from regulatory changes, economic factors, and market adaptations within the EU Emissions Trading System.

Table 16 (Appendix B.1) ranks the top influencers based on degree centrality across all phases. This ranking helps identify countries that maintained strong connectivity or lost influence over time. Early phases reveal a centralized network, where a few nodes dominate connectivity. As the system evolves, decentralization and fragmentation emerge. In Phases I and II, countries such as France (FR), the Netherlands (NL), and the United Kingdom (GB) exhibit high degree centrality, acting as dominant hubs. However, in later phases, their influence declines, and new hubs emerge. Observing these trends, early phases exhibit concentrated influence, whereas later phases show more distributed connectivity. The merging and splitting of communities reflect shifting relationships and market priorities. Established hubs retain influence, but their dominance diminishes as decentralization takes hold.

Figure 18 illustrates the evolution of degree centrality trends across phases. In this figure, we track the degree centrality of five top influencers over time to reveal structural shifts in the network. A degree centrality close to 2 suggests that a country maintains direct trading relationships with nearly every other country in the network, highlighting its role as a highly connected participant. This level of centrality reflects a position of strong market integration, where the country either facilitates transactions between others or dominates trading activity. The early phases (I-II) show a centralized system, while later phases (III-IV) indicate increasing fragmentation and decentralization. Countries such as the United Kingdom (GB) display a steady rise in centrality, peaking in Phase III before a sharp decline in Phase IV. The Netherlands (NL) and Germany (DE) follow similar patterns, maintaining stable centrality in early phases before experiencing a drop in Phase IV. France (FR) starts with high degree centrality in Phase I but declines steadily across all phases, with a pronounced drop in Phase IV. Denmark (DK) consistently ranks lower than the other top nodes, with a gradual decline culminating in Phase IV's sharp drop. The decline in degree centrality in Phase IV suggests a structural transformation in the network. The redistribution of influence, reduced connectivity, and increased fragmentation all indicate systemic changes. The emergence of the EU-wide account ("EU") in Phase IV further contributes to the network's decentralization, as many transactions become consolidated under a single entity rather than distributed among individual countries.

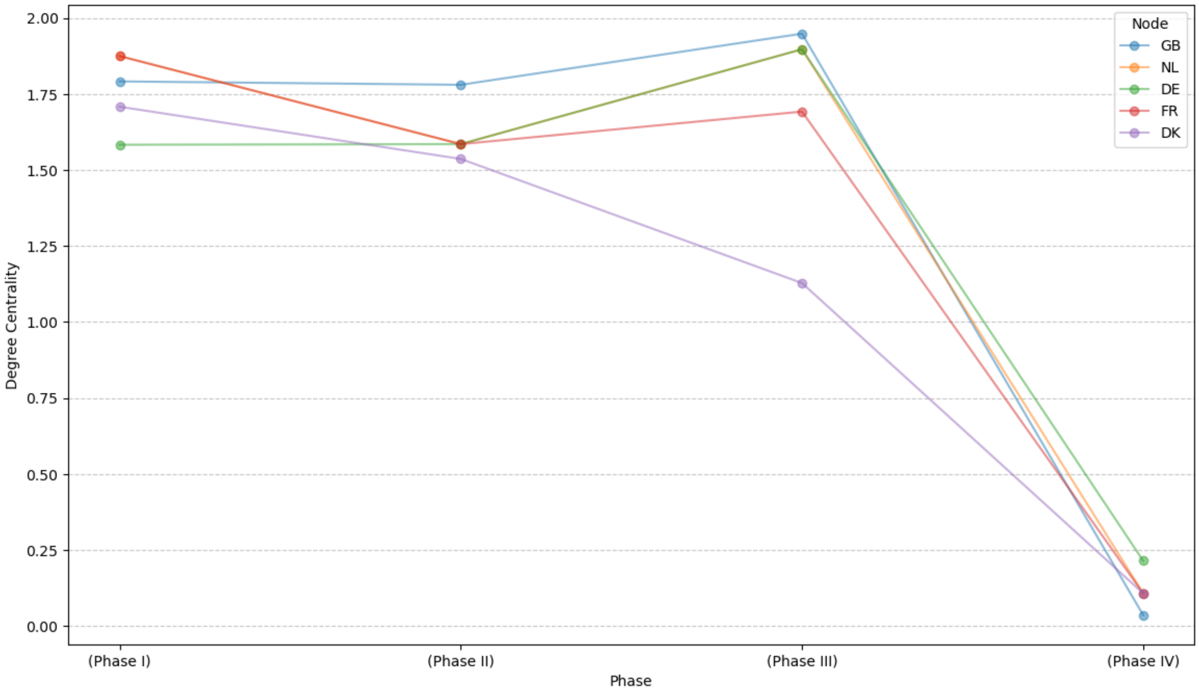


Figure 18 – Top Influencers' Degree Centrality Trends Across Phases - 2005-2023

The observed trends provide valuable insights into the network's evolution. Early phases exhibit high connectivity among a few dominant nodes, forming a relatively centralized structure. As the system matures, decentralization occurs, with more balanced connectivity and new participants emerging. By Phase IV, fragmentation becomes evident, likely influenced by external shocks such as regulatory changes or economic disruptions. The decline of previously dominant hubs suggests a redistribution of influence, reinforcing the shifting dynamics of the EU ETS trading system. Overall, these findings demonstrate the structural transformation of the emissions trading network, moving from concentrated power centers toward a more fragmented and decentralized system.

4.5.3 Simulating Structural Change with Digital Twins

To analyze the potential structural evolution of the EU ETS network, we construct a Digital Twin for Phase IV and simulate dynamic changes in node and edge configurations. Figure 19 presents the initial state of the Digital Twin, replicating the existing Phase IV network while introducing five new nodes, labeled $Twin.Node\_i$, where $i \in \{1, ..., s\}$. These nodes represent hypothetical new participants entering the emissions trading system, which could correspond to new countries joining the market or installations from a specific sector being integrated. The network consists of 34 nodes, including the five additions, and 68 edges, resulting in an average degree of 4.0. This metric indicates that each node, on average, maintains the same level of connectivity observed in Phase IV.

The next step simulates network evolution by applying the three dynamic mechanisms introduced in Section 4.3.2: (i) preferential attachment, where new nodes are more likely to connect to highly connected participants; (ii) edge rewiring, which randomly reassigns a subset of existing connections to reflect changes in trading relationships; and (iii) community reorganization, where the algorithm merges clusters with high inter-community density to simulate commercial integration. These mechanisms collectively drive the structural transformation of the network, capturing both incremental adjustments and more substantial shifts in trading behavior.

Following these modifications, the updated Digital Twin (Figure 19) displays increased network complexity. The simulation introduces new dynamic nodes, labeled $Dyn.Node\_j\_\{i+1\}$, which represent temporary states of nodes that evolve over multiple iterations. These nodes adjust based on trading activity, regulatory influences, or network constraints, mirroring the gradual adaptation of real-world market participants. The refined network expands to 40 nodes, including newly incorporated participants, and 80 edges, reflecting the evolution of trade relationships. Community structures also undergo significant changes, increasing to eight distinct trading clusters, with an average of 5.71 nodes per community. This suggests a tendency toward fragmentation or specialization within the emissions trading system.

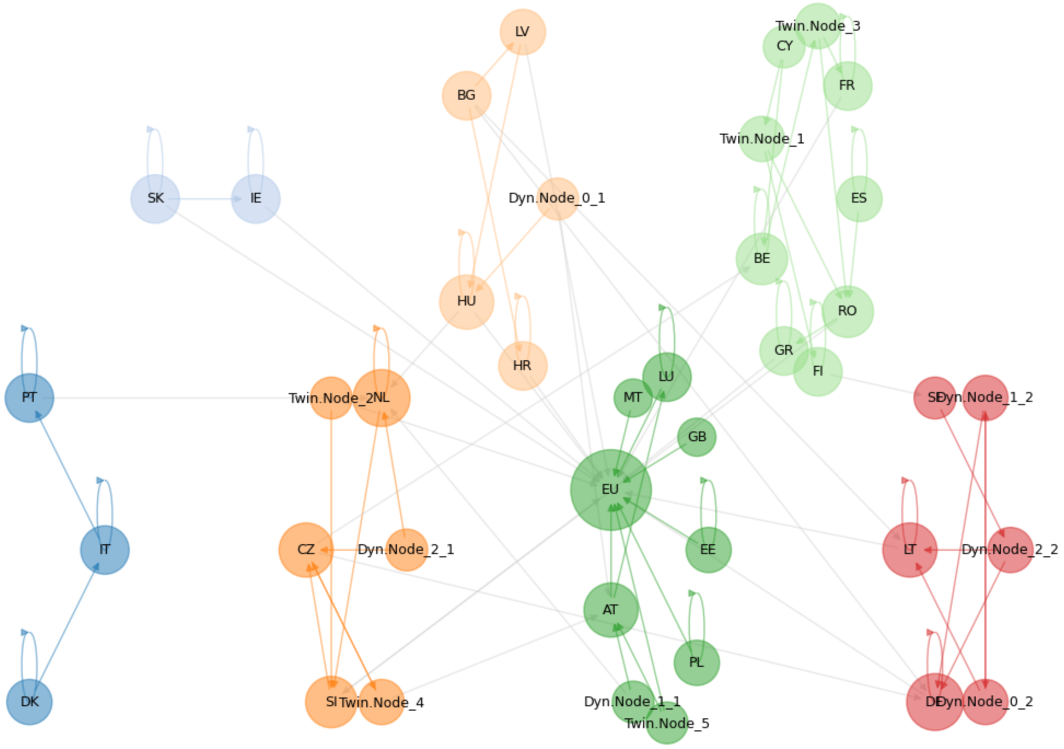


Figure 19 – Dynamic Digital Twin after Node and Edge Evolution

Figure 20 illustrates the final stage of network evolution, highlighting long-term transformations in node roles and structural configurations. In addition to $Twin.Node\_i$ and $Dyn.Node\_j\_\{i+1\}$, the model introduces $Evol.Node\_\{i+1\}$, representing nodes that have undergone cumulative adaptations over successive iterations. These nodes track gradual shifts in connectivity and influence, reflecting how certain participants transition from peripheral to central positions in the trading network. For instance, an installation or country with increasing trade volume may emerge as a dominant hub over time. The final network state comprises 43 nodes, incorporating three additional participants projected for Phase V. The number of edges increases to 86 due to newly formed and rewired connections, reflecting the dynamic nature of emissions trading. Despite minor shifts in community structure, the network stabilizes into eight communities, with an average of 7.17 nodes per cluster. This suggests a tendency toward consolidation, where previously fragmented trading groups merge into larger, more integrated clusters.

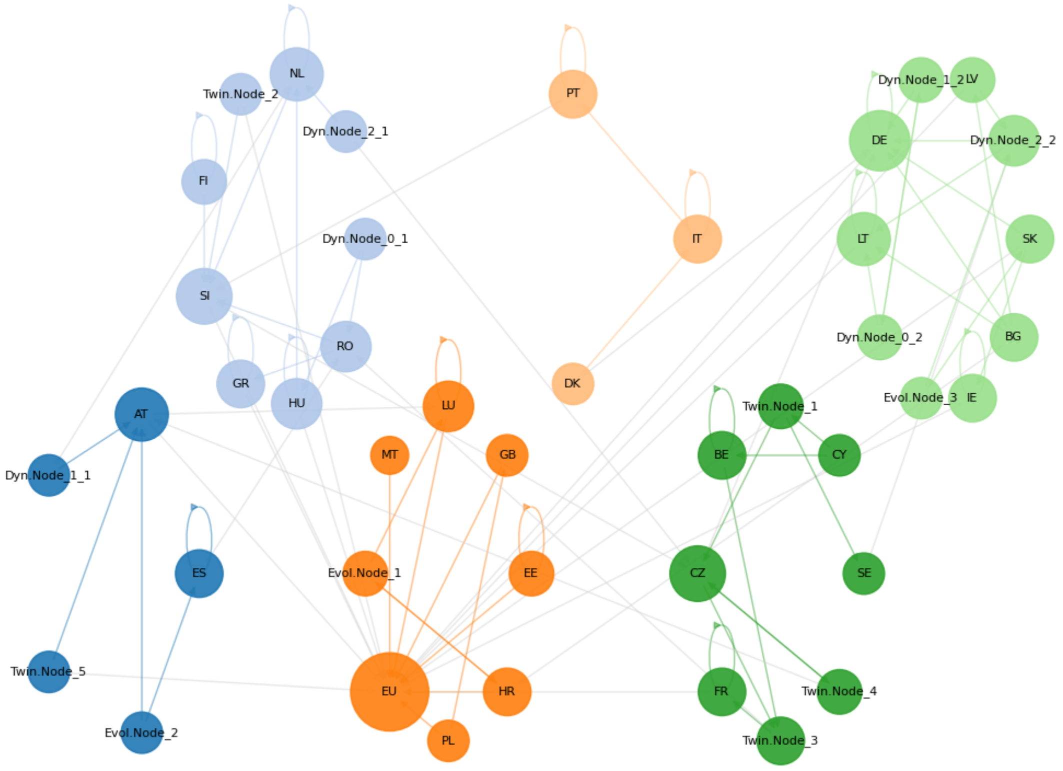


Figure 20 – Dynamic Digital Twin after Node and Edge Evolution

These results show how the EU ETS trading network can change as the market grows and regulations evolve. By using a Digital Twin to simulate these dynamics, we are able to explore how new connections form, how trading relationships shift, and how communities reorganize over time. This approach helps us identify patterns that may shape the future of the carbon market — an issue we take up in the next section, where we integrate machine learning to improve predictions about these structural changes.

4.5.4 The Future of the European Carbon Market

In this section, we use machine learning models integrated with Digital Twins to generate predictions about the future structure and behavior of the European carbon market. This integration increases the model's ability to simulate structural changes, track network evolution, and optimize connectivity. By leveraging machine learning techniques, we improve the accuracy of predictions related to edge formation and trading dynamics within the EU ETS. We explore Graph Neural Networks (GNNs) for link prediction, alongside classical machine learning models such as Logistic Regression. These models perform well when edge prediction features are carefully engineered to capture both structural and transactional characteristics.

We construct the input graph using data from Phases I–IV, incorporating nodes, edges, and relevant attributes, such as the volume of traded allowances between countries. The GNN model is trained on this dataset to predict new connections expected to form in Phase IV. The process involves defining node embeddings, training the model, and evaluating its predictive performance. To train and evaluate the machine learning model, we split the dataset into training and testing subsets.[5] The training set comprises 80% of the data, while the remaining 20% is reserved for testing. Specifically, we apply this to the edge index, ensuring a balanced distribution of edges across training and test sets. This approach allows the model to learn structural patterns from the training data while evaluating its generalization on unseen edges. By maintaining an 80/20 split, we strike a balance between providing sufficient data for training and preserving enough examples for robust performance assessment.

Epoch	**Train Loss**	**Test Loss**	**Test AUC**
0	0.6982	0.7146	0.5357
10	0.6164	0.6599	0.6214
20	0.5017	0.6749	0.7357
30	0.4178	0.6940	0.7071
40	0.3888	0.7094	0.7286
50	0.3687	0.6787	0.7357

Table 7 – Evaluation of GNN model performance for edge prediction.

Table 7 presents the evaluation results of the GNN model across multiple training epochs. Initially, both train and test losses are high, and the test AUC is close to 0.5, indicating performance close to random guessing. As training progresses, the test AUC stabilizes between 0.71 and 0.74, showing that the model effectively generalizes to unseen data. The decreasing training loss suggests improved learning of structural and transactional patterns within the network.

[5] We use the `train_test_split()` function from Scikit-Learn.

Figure 21 visualizes the predicted edges in Phase IV, with highlighted new connections identified by the GNN model. The EU node remains the dominant hub, with several newly predicted links directed toward or emerging from it. The model also forecasts connections involving peripheral nodes, such as Spain (ES), Portugal (PT), and Denmark (DK), suggesting an expected expansion in trading activity beyond the core participants. The ability of GNNs to leverage relational and structural properties allows them to detect edges that might not be immediately apparent with simpler models. For instance, connections between less central nodes, such as Croatia (HR) and Lithuania (LT), indicate that the model captures deeper network dependencies beyond direct trading relationships. This predictive capability offers insights into the future evolution of the network, pointing out potential shifts in trading behavior.

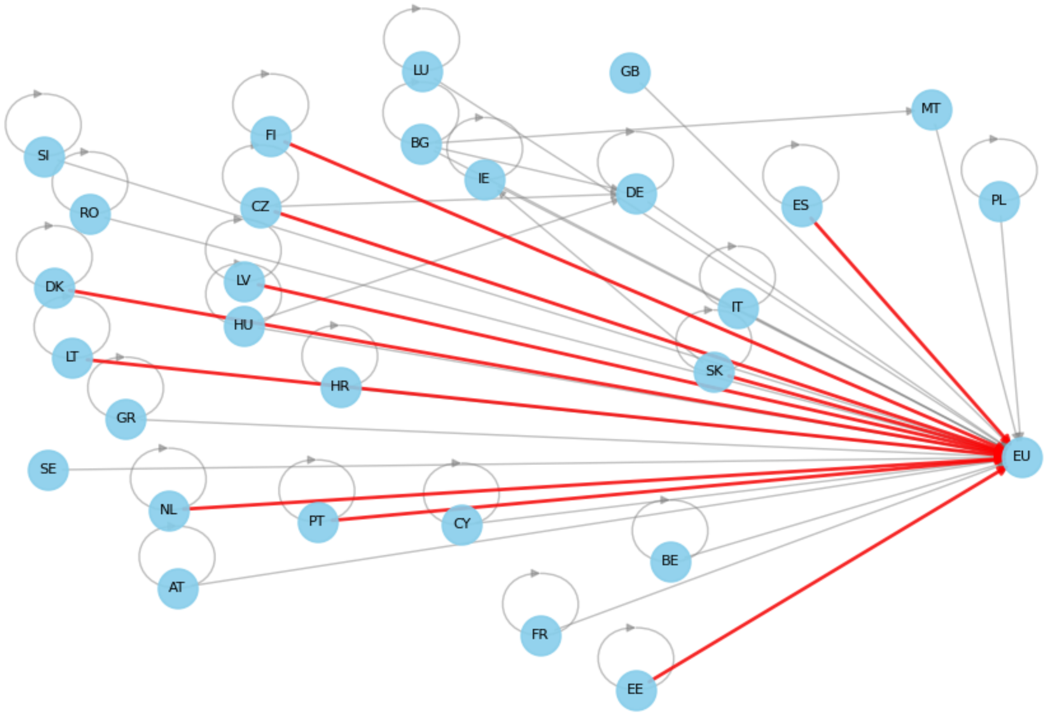


Figure 21 – Phase IV Network with highlighted predicted edges by GNN Model.

Figure 22 extends the analysis by incorporating the Digital Twin framework, where predicted edges are evaluated within an evolving network structure. The simulation introduces new entities, such as Twin.Nodes and Dyn.Nodes, representing potential participants in the system. The model predicts increased connectivity involving the EU node, reinforcing its role as the central trading hub. Additionally, the simulation reveals that Slovakia (SK) and Poland (PL) are gaining new connections, possibly reflecting their growing influence in the emissions trading market. The inclusion of dynamically generated nodes suggests that the GNN model can generalize beyond existing structures, forecasting interactions even for newly introduced participants. By integrating Digital Twins with ML-based link prediction, we obtain a more dynamic and adaptive representation of the EU ETS. The model not only predicts structural changes but also emphasizes emerging patterns that may influence regulatory decisions and market behavior in future trading phases.

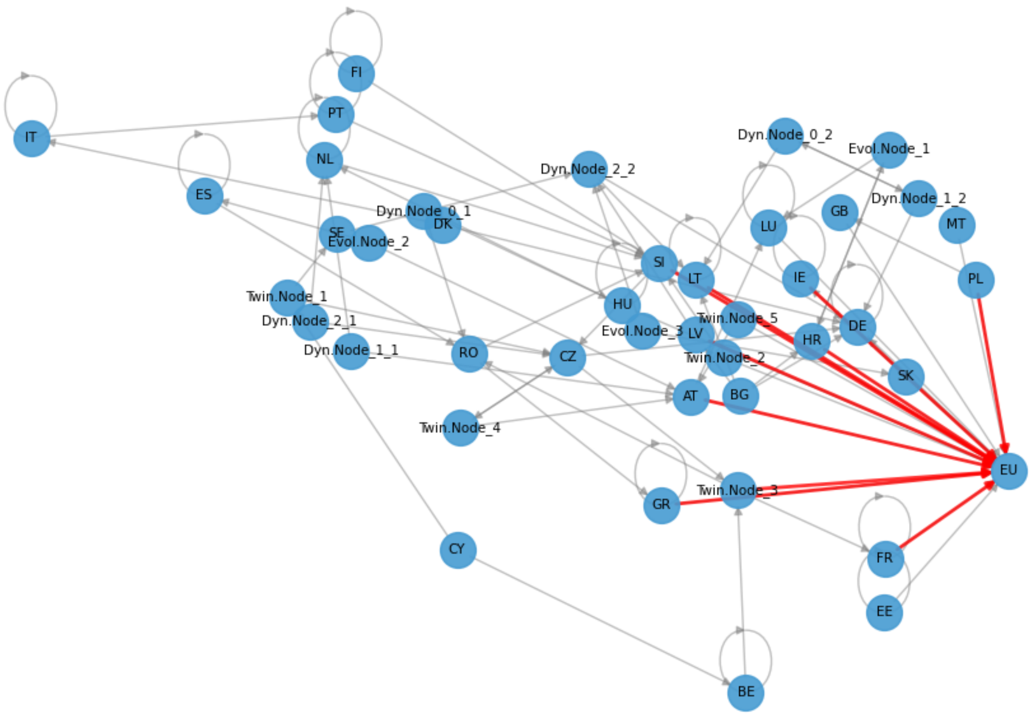


Figure 22 – Phase IV Digital Twin network with highlighted predicted edges by GNN Model.

Table 8 presents the evaluation metrics for the Logistic Regression model in predicting new edges. The model achieves an AUC score of 0.95, indicating strong classification performance. The precision of 0.9333 means that 93.33% of predicted edges correspond to actual edges, demonstrating a high level of reliability. With a recall of 1.0, the model successfully identifies all relevant edges, ensuring no potential connections are overlooked. The F1-score of 0.9655 reflects a well-balanced trade-off between precision and recall.

AUC Score	**Precision**	**Recall**	**F1-Score**
0.9500	0.9333	1.0000	0.9655

Table 8 – Evaluation metrics for edge prediction using Logistic Regression.

The ROC curve in Figure 23 visualizes the model's classification performance. The AUC of 0.95 confirms that the Logistic Regression model effectively distinguishes between existing and non-existing edges. The curve remains close to the top-left corner, reinforcing the model's high sensitivity and specificity.

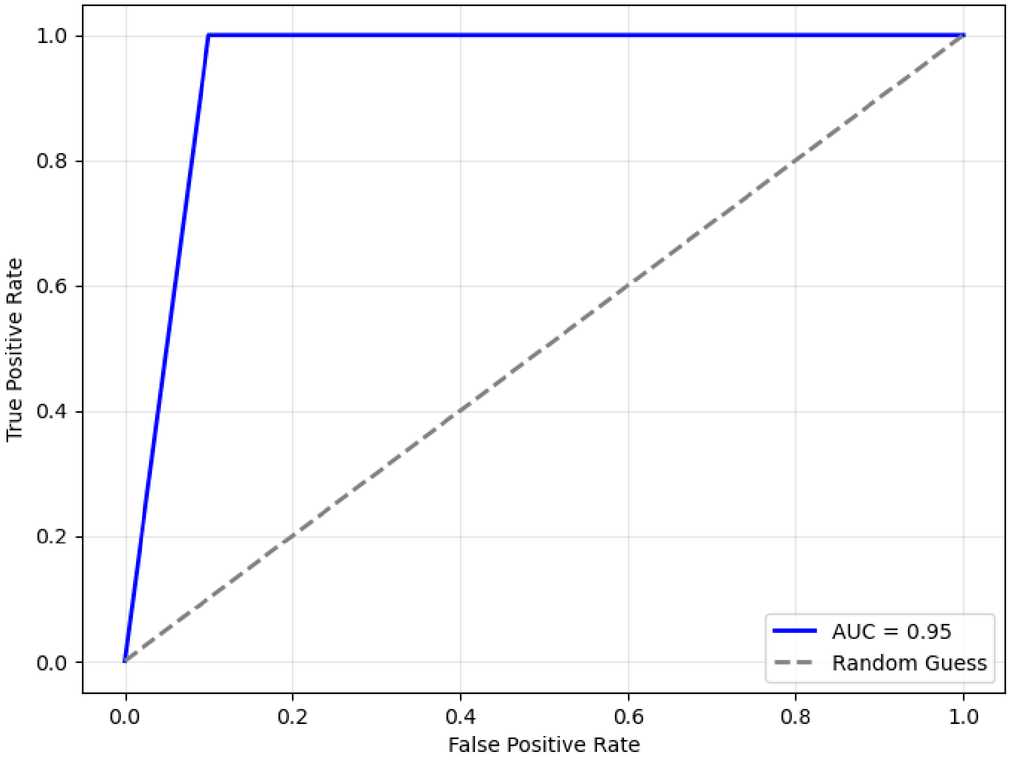


Figure 23 – ROC curve for Logistic Regression.

Figure 24 illustrates the predicted edges in Phase IV. The red lines highlight new connections forecasted by the model. Most predicted edges cluster around Bulgaria (BG), Czech Republic (CZ), Hungary (HU), and Germany (DE), indicating emerging trading links between these regions. Additional connections, such as those between Ireland (IE) and Slovakia (SK), suggest evolving market relationships. The EU node remains central, reinforcing its dominant position in the network.

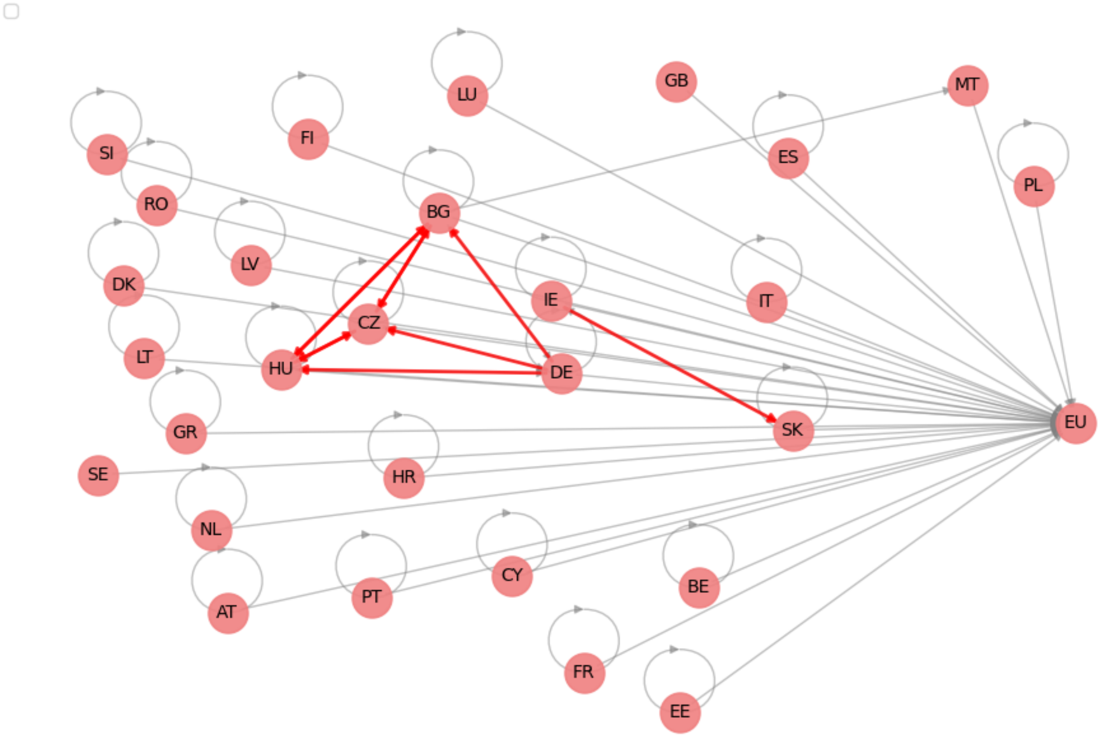


Figure 24 – Phase IV network with predicted edges highlighted by the Logistic Regression model.

Figure 25 extends this analysis to the Digital Twin framework, incorporating dynamic entities such as Twin.Nodes, Dyn.Nodes, and Evol.Nodes. These additional nodes represent forecasted participants and structural changes in the market. The red edges indicate predicted connections, with a notable concentration around the EU node. This suggests the EU's continued role as a trading hub while new nodes integrate into the network. The predicted connections reveal expanding trading relationships, particularly involving countries like Cyprus (CY), Sweden (SE), Denmark (DK), and Finland (FI). The Logistic Regression model emphasizes connections between already well-established nodes, reinforcing existing trading hubs. In contrast, the GNN model predicts a broader distribution of edges, identifying connections between peripheral nodes. This difference suggests that while Logistic Regression effectively captures strong trading relationships, GNNs may offer insights into emerging market structures. Overall, the Logistic Regression model delivers high accuracy and recall, making it reliable for predicting structural changes within the EU ETS network. However, its tendency to reinforce central hubs rather than explore new link formations highlights the complementary role of GNNs in network evolution analysis.

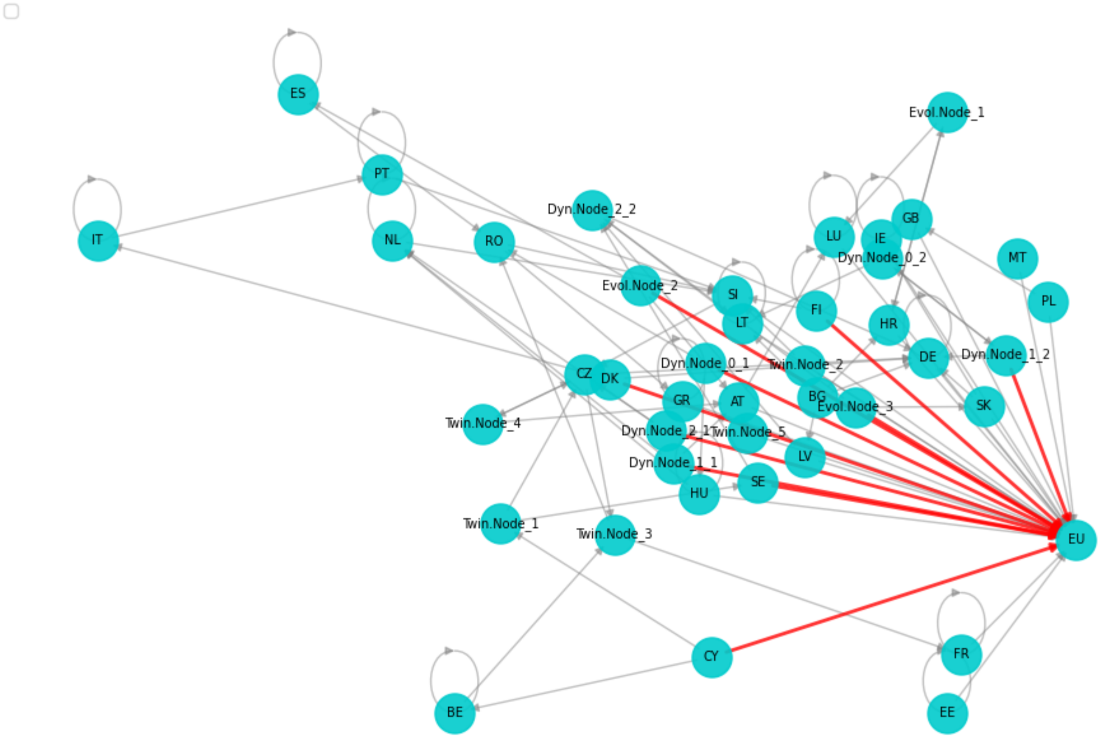


Figure 25 – Phase IV with Digital Twin network and highlighted predicted edges by the Logistic Regression model.

4.6 Conclusion

This paper applies Digital Twin modeling, complex network analysis, and machine learning to study the structural evolution of trading relationships in the European Union Emissions Trading System (EU ETS). Using detailed transaction data, we document a shift from a hub-dominated network to a more decentralized and fragmented structure. Our results show that while some trading clusters persist, others dissolve or reconfigure in response to regulatory adjustments and firm behavior.

By integrating predictive models such as Graph Neural Networks and logistic regression with Digital Twin simulations, we forecast future link formation and evaluate whether the network can absorb increased market activity. Our findings reveal growing fragmentation in the trading network, which may create structural bottlenecks, reduce liquidity, and limit the ability of firms to reallocate allowances efficiently. These patterns have direct implications for market efficiency: when firms face reduced access to counterparties or incur higher transaction costs, the allowance allocation mechanism may become less cost-effective. Over time, this can increase the likelihood of triggering price floors or ceilings, reducing the flexibility that cap-and-trade systems are designed to provide. The structural weaknesses we identify may indirectly affect sustainability. A fragmented or inefficient network could undermine firms' ability to meet emissions targets or adapt to policy changes, potentially reducing the overall effectiveness of the EU ETS in supporting decarbonization (Mattauch et al., 2022; Flori, Borghesi & Marin, 2024). Therefore, addressing structural vulnerabilities in the trading network is crucial for sustaining environmental performance and ensuring the market remains aligned with long-term sustainability objectives.

Incorporating sustainability metrics and explicitly modeling environmental outcomes alongside economic performance would strengthen the predictive power of these analyses, supporting policymakers in designing robust market structures that reliably promote environmental sustainability. Building on this framework, future research could extend Digital Twin simulations by incorporating firm-level behavioral assumptions, such as compliance strategies or risk preferences, to better capture the microfoundations of trading dynamics. Additional work could also integrate environmental and economic performance metrics more explicitly into the modeling environment. Finally, applying this approach to other emissions trading systems or to emerging cross-border carbon markets could offer comparative insights into the structural resilience and efficiency of different market designs. These results suggest that regulatory oversight should extend beyond price monitoring to include the structure of trading relationships. Incorporating network diagnostics into market surveillance could help identify early signs of fragmentation and inform targeted interventions to sustain a well-functioning and environmentally effective carbon market. Ultimately, reinforcing market structure resilience contributes directly to the broader goal of achieving sustainable emissions reductions and maintaining momentum towards Europe's ambitious climate neutrality targets.

5 CONCLUSION

We explore methods for constructing networks, measuring node importance, and predicting structural changes using Machine Learning. Machine learning enhances centrality approximations, reducing computational costs while preserving ranking accuracy. Embedding techniques like node2vec and GraphSAGE encode structural properties into compact representations, improving clustering, classification, and link prediction. Clustering methods, such as Gaussian Mixture Models and hierarchical approaches, group similar nodes while allowing overlapping memberships, making them effective for community detection.

Link prediction leverages network structure and past interactions to anticipate future connections, benefiting applications like fraud detection and recommendation systems. Reinforcement learning optimizes community detection in dynamic networks by refining partitioning strategies and enhancing modularity. Visualization techniques like PCA, NMF, and t-SNE simplify complex structures, while extensions such as tsNET and GraphTSNE improve layout representation by incorporating connectivity patterns.

These techniques apply to fields such as fraud detection, economics, recommendation systems, biological interactions, infrastructure, and security. Although machine learning improves scalability and interpretability, challenges persist in model fine-tuning, handling sparsity, and ensuring robustness. Future research should focus on reducing computational complexity, improving generalization, and integrating multiple methods for more precise network analysis.

The second paper apply a approach to systemic risk analysis by constructing a network of U.S. firms based on news similarity from major media outlets, including The New York Times, Reuters, Fox News, Financial Times, The Guardian, and CNN. Unlike traditional methods relying on financial statements and asset prices, this framework captures indirect contagion by examining how firms connect through media coverage. By applying NLP techniques, the study uncovers firm relationships that influence risk transmission but remain invisible in balance-sheet data.

The results indicate that firms with high centrality in the news similarity network face greater exposure to financial shocks, highlighting the role of public perception in systemic risk propagation. Community detection reveals that firms form clusters extending beyond conventional sector classifications, emphasizing the importance of cross-sector information flows. Regression analysis shows that firm size and stock price volatility contribute to network centrality, suggesting an interaction between financial characteristics and media-driven contagion.

These insights provide regulators and investors with a broader perspective on financial risk monitoring. Incorporating textual data into systemic risk assessments complements traditional quantitative models and enhances the ability to detect emerging vulnerabilities. Future research could refine this method by integrating machine learning to predict contagion events based on the temporal evolution of news networks. Expanding the analysis across different markets and time periods would further validate news-based networks in systemic risk assessment. Another approach could explore how news-reported events influence strategic decisions in firms interconnected through shared board memberships, offering insights into corporate governance and information flow dynamics.

The third paper examines the structural evolution of the European Union Emissions Trading System (EU ETS) and its implications for market efficiency, price stability, and regulatory effectiveness. As the trading network shifts from a centralized, hub-dominated structure to a more fragmented system, new challenges emerge. Decentralization and declining interconnectivity may weaken market integration, disrupt price formation, and reduce liquidity. Community detection reveals that some countries form persistent trading clusters, while others face growing isolation, raising concerns about trade barriers and uneven allowance distribution. Policymakers must determine whether existing mechanisms adequately sustain competition and emissions reduction targets or require adjustments to prevent inefficiencies in market structure.

Regulatory interventions impact not only price stability but also trading relationships, altering network resilience over time. The findings suggest that further fragmentation may reduce emissions trading flexibility, limiting firms' ability to adjust efficiently to carbon pricing signals. Predictive modeling indicates that emerging trading barriers could lead to long-term inefficiencies, emphasizing the need for regulatory strategies that preserve connectivity while balancing market stability and emissions reduction goals.

This study introduces a framework that integrates Digital Twins, complex network analysis, and machine learning to model emissions trading as a dynamic system. By treating the EU ETS as an evolving network, the study identifies structural shifts, monitors changing market power, and predicts future trading relationships. Graph Neural Networks (GNNs) and Logistic Regression forecast link formation, with GNNs detecting emerging relationships beyond direct neighbors and Logistic Regression reinforcing existing hubs. Combining classical network metrics with Digital Twin simulations improves forecasting and interpretation of emissions trading network evolution.

Beyond empirical findings, this study demonstrates how network science and predictive modeling can inform environmental policy. Future research could extend this approach by incorporating stress-testing scenarios, simulating policy adjustments before implementation, or developing Digital Twin frameworks with real-time transaction data. A deeper understanding of emissions trading network evolution will help regulators anticipate market shifts, ensuring that cap-and-trade systems remain effective in balancing environmental goals with economic efficiency.

We use a variety of methods and techniques to model network data-driven, observing patterns in both numerical and textual data. We demonstrated how to use various techniques to model complex networks and predict possible future behaviors like NLP, ML, and Digital Twins. Thus, we conclude that Complex Networks combined with machine learning and digital twins greatly assist in analyzing economic data.

BIBLIOGRAPHY

ABAYADEERA, M. R.; GANEGODA, G. Digital twin technology: A comprehensive review. 2024. 100

ACHARYA, V. V. A theory of systemic risk and design of prudential bank regulation. *Journal of financial stability*, Elsevier, v. 5, n. 3, p. 224–255, 2009. 71

ACHARYA, V. V. et al. Measuring systemic risk. *The review of financial studies*, Oxford University Press, v. 30, n. 1, p. 2–47, 2017. 71

ADAFRE, S. F.; RIJKE, M. de. Discovering missing links in wikipedia. In: *Proceedings of the 3rd international workshop on Link discovery*. [S.l.: s.n.], 2005. p. 90–97. 63

AGGARWAL, C.; SUBBIAN, K. Evolutionary network analysis: A survey. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 47, n. 1, p. 1–36, 2014. 18

AGGARWAL, C. C. et al. *Data mining: the textbook*. [S.l.]: Springer, 2015. v. 1. 39, 42

AGRAWAL, S.; PATEL, A. Clustering algorithm for community detection in complex network: a comprehensive review. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*, Bentham Science Publishers, v. 13, n. 4, p. 542–549, 2020. 58

AHMED, A. et al. Distributed large-scale natural graph factorization. In: *Proceedings of the 22nd international conference on World Wide Web*. [S.l.: s.n.], 2013. p. 37–48. 35

AHMED, C.; ELKORANY, A.; BAHGAT, R. A supervised learning approach to link prediction in twitter. *Social Network Analysis and Mining*, Springer, v. 6, p. 1–11, 2016. 65

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. *Reviews of modern physics*, APS, v. 74, n. 1, p. 47, 2002. 67, 99

AMELIO, A.; TAGARELLI, A. Silhouette for the evaluation of community structures in multiplex networks. In: SPRINGER. *Complex Networks IX: Proceedings of the 9th Conference on Complex Networks CompleNet 2018 9*. [S.l.], 2018. p. 41–49. 60

BADER, D. A. et al. Approximating betweenness centrality. In: SPRINGER. *Algorithms and Models for the Web-Graph: 5th International Workshop, WAW 2007, San Diego, CA, USA, December 11-12, 2007. Proceedings 5*. [S.l.], 2007. p. 124–137. 45

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern Information Retrieval*. 2nd. ed. USA: Addison-Wesley Publishing Company, 2008. ISBN 9780321416919. 26

BAKER, S. R.; BLOOM, N.; DAVIS, S. J. Measuring economic policy uncertainty. *The quarterly journal of economics*, Oxford University Press, v. 131, n. 4, p. 1593–1636, 2016. 71

BALASUBRAMANIAN, M.; SCHWARTZ, E. L. The isomap algorithm and topological stability. *Science*, American Association for the Advancement of Science, v. 295, n. 5552, p. 7–7, 2002. 43

BANFIELD, J. D.; RAFTERY, A. E. Model-based gaussian and non-gaussian clustering. *Biometrics*, JSTOR, p. 803–821, 1993. 24

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. *science*, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, 1999. 99

BARNES, E. R. An algorithm for partitioning the nodes of a graph. *SIAM Journal on Algebraic Discrete Methods*, SIAM, v. 3, n. 4, p. 541–550, 1982. 56

BATTISTON, S.; WEISBUCH, G.; BONABEAU, E. Decision spread in the corporate board network. *Advances in Complex Systems*, World Scientific, v. 6, n. 04, p. 631–644, 2003. 90

BATTY, M. Digital twins. *Environment and Planning B: Urban Analytics and City Science*, SAGE Publications Sage UK: London, England, v. 45, n. 5, p. 817–820, 2018. 98

BAUDRY, M.; FAURE, A.; QUEMIN, S. Emissions trading with transaction costs. *Journal of Environmental Economics and Management*, Elsevier, v. 108, p. 102468, 2021. 94, 96

BEALE, N. et al. Individual versus systemic risk and the regulator's dilemma. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 108, n. 31, p. 12647–12652, 2011. 71

BELKIN, M.; NIYOGI, P. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in neural information processing systems*, v. 14, 2001. 35

BEZDEK, J. et al. Fuzzy clustering; a new approach for geostatistical analysis: Int. *Jour. Sys., Meas., and Decisions*, 1982. 24

BEZDEK, J. C.; EHRLICH, R.; FULL, W. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, Elsevier, v. 10, n. 2-3, p. 191–203, 1984. 24

BHAGAT, S.; CORMODE, G.; MUTHUKRISHNAN, S. Social network data analytics. In: ______. [S.l.]: Springer, 2011. cap. Node classification in social networks, p. 115–148. 52, 54

BILGIN, C. C.; YENER, B. Dynamic network evolution: Models, clustering, anomaly detection. *IEEE Networks*, v. 1, 2006. 19

BILLIO, M. et al. Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of Financial Economics*, v. 104, n. 3, p. 535–559, 2012. 72

BLEI, D. M. Probabilistic topic models. *Communications of the ACM*, ACM New York, NY, USA, v. 55, n. 4, p. 77–84, 2012. 27

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *Journal of machine Learning research*, v. 3, n. Jan, p. 993–1022, 2003. 27

BLONDEL, V. D. et al. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, IOP Publishing, v. 2008, n. 10, p. P10008, 2008. 73, 76, 92, 97

BORGATTI, S. P.; HALGIN, D. S. Analyzing affiliation networks. *The Sage handbook of social network analysis*, v. 1, p. 417–433, 2011. 67

BORGHESI, S. The european emission trading scheme and renewable energy policies: credible targets for incredible results? *International Journal of Sustainable Economy*, Inderscience Publishers, v. 3, n. 3, p. 312–327, 2011. 95

BORGHESI, S.; FLORI, A. Eu ets facets in the net: Structure and evolution of the eu ets network. *Energy economics*, Elsevier, v. 75, p. 602–635, 2018. 93, 103

BORIAH, S.; CHANDOLA, V.; KUMAR, V. Similarity measures for categorical data: A comparative evaluation. In: SIAM. *Proceedings of the 2008 SIAM international conference on data mining*. [S.l.], 2008. p. 243–254. 39, 41

BOUCKAERT, S. et al. Net zero by 2050: A roadmap for the global energy sector. *International Energy Agency, available at https://www.iea.org/reports/net-zero-by-2050*, OECD Publishing, 2021. 92

BRANDES, U.; WAGNER, D. Analysis and visualization of social networks. In: *Graph drawing software*. [S.l.]: Springer, 2004. p. 321–340. 67

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, p. 5–32, 2001. 21

BREIMAN, L. et al. Classification and regression trees (cart). *Biometrics*, v. 40, n. 3, p. 358, 1984. 21

BREIT, A. et al. Openbiolink: a benchmarking framework for large-scale biomedical link prediction. *Bioinformatics*, Oxford University Press, v. 36, n. 13, p. 4097–4098, 2020. 63

BUITINCK, L. et al. Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013. 20

BYBEE, L. et al. *The structure of economic news*. [S.l.], 2020. 71

CAI, B. et al. Community detection method based on node density, degree centrality, and k-means clustering in complex network. *Entropy*, MDPI, v. 21, n. 12, p. 1145, 2019. 59

CAJUEIRO, D. O. Optimal navigation in complex networks. *Physical Review E*, APS, v. 79, n. 4, p. 046103, 2009. 48

CAJUEIRO, D. O. Optimal navigation for characterizing the role of the nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 389, n. 9, p. 1945–1954, 2010. 48

CAJUEIRO, D. O.; ANDRADE, R. F. S. Learning paths in complex networks. *Europhysics letters*, IOP Publishing, v. 87, n. 5, p. 58004, 2009. 48

CAJUEIRO, D. O. et al. A model of indirect contagion based on a news similarity network. *Journal of Complex Networks*, Oxford University Press, v. 9, n. 5, p. cnab035, 2021. 14, 40, 71, 73, 78, 84, 85

CAJUEIRO, D. O. et al. Markov chain approach to model intertemporal choices and coverages in air transport markets. *Physical Review E*, APS, v. 100, n. 6, p. 062303, 2019. 84

CAJUEIRO, D. O. et al. A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding. *arXiv preprint arXiv:2301.03403*, 2023. 72

CELEUX, G.; GOVAERT, G. Gaussian parsimonious clustering models. *Pattern Recognition*, Elsevier, v. 28, n. 5, p. 781–793, 1995. 24

CHEN, F. et al. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, Cambridge University Press, v. 9, p. e15, 2020. 18

CHEN, H. et al. Exploiting centrality information with graph convolutions for network representation learning. In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. [S.l.: s.n.], 2019. p. 590–601. 49

CHEN, J.; CHEN, H. Edge-featured graph attention network. *arXiv preprint arXiv:2101.07671*, 2021. 92

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 785–794. 21

CHI, K. T.; LIU, J.; LAU, F. C. A network perspective of the stock market. *Journal of Empirical Finance*, Elsevier, v. 17, n. 4, p. 659–667, 2010. 39

CHIANG, W.-L. et al. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. [S.l.: s.n.], 2019. p. 257–266. 30

COALITION, W. B. G. C. P. L. Simulating carbon markets. *World Bank, available at https://hdl.handle.net/10986/33687*, 2020. 100

COHEN, E. et al. Computing classic closeness centrality, at scale. In: *Proceedings of the second ACM conference on Online social networks*. [S.l.: s.n.], 2014. p. 37–50. 45

COIFMAN, R. R. et al. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the national academy of sciences*, National Acad Sciences, v. 102, n. 21, p. 7426–7431, 2005. 43

COMIN, C. H. et al. Complex systems: Features, similarity and connectivity. *Physics Reports*, Elsevier, v. 861, p. 1–41, 2020. 37

CONT, R.; SCHAANNING, E. Monitoring indirect contagion. *Journal of Banking & Finance*, v. 104, p. 85–102, 2019. 72

CONVERY, F. J. Origins and development of the eu ets. *Environmental and Resource Economics*, Springer, v. 43, n. 3, p. 391–412, 2009. 94

COPPOLA, C.; ELGAZZAR, H. Novel machine learning algorithms for centrality and cliques detection in youtube social networks. *arXiv preprint arXiv:2002.03893*, 2020. 47

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, p. 273–297, 1995. 21

COSTA, A. R. Towards modularity optimization using reinforcement learning to community detection in dynamic social networks. *arXiv e-prints*, p. arXiv–2111, 2021. 63

COSTA, L. d. F. Further generalizations of the jaccard index. *arXiv preprint arXiv:2110.09619*, 2021. 38

COSTA, L. d. F.; TRAVIESO, G. Exploring complex networks through random walks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, APS, v. 75, n. 1, p. 016102, 2007. 48

CROCKETT, A. Marrying the micro-and macro-prudential dimensions of financial stability. *BIS speeches*, v. 21, 2000. 71

CUPERTINO, T. H.; HUERTAS, J.; ZHAO, L. Data clustering using controlled consensus in complex networks. *Neurocomputing*, Elsevier, v. 118, p. 132–140, 2013. 43

DALMIA, A.; GUPTA, M. Towards interpretation of node embeddings. In: *Companion Proceedings of the The Web Conference 2018*. [S.l.: s.n.], 2018. p. 945–952. 18

DANILA, B. et al. Transport optimization on complex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, AIP Publishing, v. 17, n. 2, 2007. 48

DAUD, N. N. et al. Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications*, Elsevier, v. 166, p. 102716, 2020. 63

DAVIS, G. F.; GREVE, H. R. Corporate elite networks and governance changes in the 1980s. *American journal of sociology*, The University of Chicago Press, v. 103, n. 1, p. 1–37, 1997. 90

DECHEZLEPRÊTRE, A.; NACHTIGALL, D.; VENMANS, F. The joint impact of the european union emissions trading system on carbon emissions and economic performance. *Journal of Environmental Economics and Management*, Elsevier, v. 118, p. 102758, 2023. 95

DEEPAK, S.; AMEER, P. Retrieval of brain mri with tumor using contrastive loss based similarity on googlenet encodings. *Computers in Biology and Medicine*, Elsevier, v. 125, p. 103993, 2020. 39

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, v. 39, n. 1, p. 1–38, 1977. 24

DENG, Y. Recommender systems based on graph embedding techniques: A review. *IEEE Access*, IEEE, v. 10, p. 51587–51633, 2022. 19

DIAS, M. D. et al. A hierarchical network simplification via non-negative matrix factorization. In: IEEE. *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.], 2017. p. 119–126. 68

DOMINGOS, P. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. [S.l.]: Basic Books, 2015. 20

DUAN, J. et al. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE transactions on neural networks and learning systems*, IEEE, v. 33, n. 11, p. 6584–6598, 2021. 32

DUAN, J. et al. Dsac-t: Distributional soft actor-critic with three refinements. *arXiv preprint arXiv:2310.05858*, 2023. 32

DUBOIS, L.; SAHUC, J.-G.; VERMANDEL, G. A general equilibrium approach to carbon permit banking. *Journal of Environmental Economics and Management*, Elsevier, v. 129, p. 103076, 2025. 96

DUMAIS, S. T. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments, & Computers*, v. 23, n. 2, p. 229–236, Jun 1991. 26

ELLERMAN, A. D.; BUCHNER, B. K. The european union emissions trading scheme: Origins, allocation, and early results. *Review of Environmental Economics and Policy*, Oxford University Press, v. 1, n. 1, p. 66–87, 2007. 95

ELLERMAN, A. D.; CONVERY, F. J.; PERTHUIS, C. D. *Pricing Carbon: The European Union Emissions Trading Scheme*. [S.l.]: Cambridge University Press, 2010. 94

FLORI, A.; BORGHESI, S.; MARIN, G. The environmental-financial performance nexus of eu ets firms: A quantile regression approach. *Energy Economics*, Elsevier, v. 131, p. 107328, 2024. 94, 123

FORTUNATO, S. Community detection in graphs. *Physics reports*, Elsevier, v. 486, n. 3-5, p. 75–174, 2010. 56

FORTUNATO, S.; HRIC, D. Community detection in networks: A user guide. *Physics reports*, Elsevier, v. 659, p. 1–44, 2016. 58

FREEMAN, L. C. A set of measures of centrality based on betweenness. *Sociometry*, JSTOR, p. 35–41, 1977. 44

FREEMAN, L. C. et al. Centrality in social networks: Conceptual clarification. *Social network: critical concepts in sociology. Londres: Routledge*, v. 1, p. 238–263, 2002. 44

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, JSTOR, p. 1189–1232, 2001. 21

FRIEDMAN, J. H. Stochastic gradient boosting. *Computational statistics & data analysis*, Elsevier, v. 38, n. 4, p. 367–378, 2002. 21

FUJIMOTO, S.; HOOF, H.; MEGER, D. Addressing function approximation error in actor-critic methods. In: PMLR. *International conference on machine learning*. [S.l.], 2018. p. 1587–1596. 32

GABRIELI, S. Too-connected versus too-big-to-fail: Banks' network centrality and overnight interest rates. Banque de France Working Paper, 2012. 44

GAN, G.; MA, C.; WU, J. *Data Clustering: Theory, Algorithms, and Applications*. [S.l.]: SIAM, 2020. 24

GAO, Z. et al. edge2vec: Learning node representation using edge semantics. 2019. 36

GENTZKOW, M.; KELLY, B.; TADDY, M. Text as data. *Journal of Economic Literature*, American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203-2425, v. 57, n. 3, p. 535–574, 2019. 72

GETOOR, L.; DIEHL, C. P. Link mining: a survey. *Acm Sigkdd Explorations Newsletter*, ACM New York, NY, USA, v. 7, n. 2, p. 3–12, 2005. 63

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. 56, 92

GKARMPOUNIS, G. et al. Survey on graph neural networks. *IEEE Access*, IEEE, 2024. 101

GNANADESIKAN, R. *Methods for Statistical Data Analysis of Multivariate Observations*. [S.l.]: John Wiley & Sons, 2011. 22

GOYAL, P.; FERRARA, E. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, Elsevier, v. 151, p. 78–94, 2018. 18

GRACIOUS, T. et al. Neural latent space model for dynamic networks and temporal knowledge graphs. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2021. v. 35, n. 5, p. 4054–4062. 92

GRANDO, F.; GRANVILLE, L. Z.; LAMB, L. C. Machine learning in network centrality measures: Tutorial and outlook. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 51, n. 5, p. 1–32, 2018. 46

GRANDO, F.; LAMB, L. C. Estimating complex networks centrality via neural networks and machine learning. In: IEEE. *2015 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2015. p. 1–8. 45

GRANDO, F.; LAMB, L. C. On approximating networks centrality measures via neural learning algorithms. In: IEEE. *2016 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2016. p. 551–557. 46

GRANOVETTER, M. Threshold models of collective behavior. *American journal of sociology*, University of Chicago Press, v. 83, n. 6, p. 1420–1443, 1978. 50

GRIEVES, M. Digital twin: manufacturing excellence through virtual factory replication. *White paper*, v. 1, n. 2014, p. 1–7, 2014. 98

GROVER, A.; LESKOVEC, J. node2vec: Scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 855–864. 34

GUPTA, S.; IYER, R. S.; KUMAR, S. *Digital Twins: Advancements in Theory, Implementation, and Applications*. [S.l.]: Springer Nature, 2025. 98

HAARNOJA, T. et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018. 32

HADDAD, M. et al. Temporalnode2vec: Temporal node embedding in temporal networks. In: SPRINGER. *Complex Networks and Their Applications VIII: Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019 8*. [S.l.], 2020. p. 891–902. 34

HAGBERG, A.; SWART, P.; CHULT, D. S. *Exploring network structure, dynamics, and function using NetworkX*. [S.l.], 2008. 20

HAJARATHAIAH, K. et al. Node significance analysis in complex networks using machine learning and centrality measures. *IEEE Access*, IEEE, 2024. 46

HALDANE, A.; MAY, R. Systemic risk in banking ecosystems. *Nature*, v. 469, p. 351–355, 2011. 72

HAMILTON, W.; YING, Z.; LESKOVEC, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, v. 30, 2017. 29, 101

HAMILTON, W. L.; YING, R.; LESKOVEC, J. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017. 18

HAN, X.; SUN, L.; ZHAO, J. Collective entity linking in web text: a graph-based method. In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. [S.l.: s.n.], 2011. p. 765–774. 63

HASAN, M. A. et al. Link prediction using supervised learning. In: *SDM06: workshop on link analysis, counter-terrorism and security*. [S.l.: s.n.], 2006. v. 30, p. 798–805. 64

HASAN, M. A.; ZAKI, M. J. A survey of link prediction in social networks. *Social network data analytics*, Springer, p. 243–275, 2011. 63

HASLBECK, J. M.; WALDORP, L. J. How well do network models predict observations? on the importance of predictability in network models. *Behavior research methods*, Springer, v. 50, p. 853–861, 2018. 64

HASTIE, T. et al. *The elements of statistical learning: data mining, inference, and prediction*. [S.l.]: Springer, 2009. v. 2. 56

HE, H. et al. Risk factor identification of sustainable guarantee network based on logistic regression algorithm. *Sustainability*, MDPI, v. 11, n. 13, p. 3525, 2019. 92

HEATH, M. *Computing: An Introductory Survey*. [S.l.]: McGraw-Hill, 1998. 75

HEPBURN, C. et al. The economics of the eu ets market stability reserve. *Journal of Environmental Economics and Management*, Elsevier, v. 80, p. 1–5, 2016. 94

HEZAM, I. M. et al. Digital twin and fuzzy framework for supply chain sustainability risk assessment and management in supplier selection. *Scientific Reports*, Nature Publishing Group UK London, v. 14, n. 1, p. 17718, 2024. 100

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 18, n. 7, p. 1527–1554, 2006. 22

HISANO, R. Semi-supervised graph embedding approach to dynamic link prediction. In: SPRINGER. *Complex Networks IX: Proceedings of the 9th Conference on Complex Networks CompleNet 2018 9*. [S.l.], 2018. p. 109–121. 66

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT press, v. 9, n. 8, p. 1735–1780, 1997. 21

HOLT, C. A.; SHOBE, W. M. *Investigation of the Effects of Emission Market Design on the Market-Based Compliance Mechanism of the California Cap on Greenhouse Gas Emissions*. [S.l.], 2013. 100

HORN, A. J. V. California's cap-and-trade market for greenhouse gas allowances, holding limits and changes needed to ab 32 rules. 2015. 100

HOTELLING, H. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, Warwick & York, v. 24, n. 6, p. 417, 1933. 25

IZENMAN, A. J. Modern multivariate statistical techniques. *Regression, Classification and Manifold Learning*, Springer, v. 10, p. 978–0, 2008. 23

JACCARD, P. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, v. 37, p. 547–579, 1901. 41

JAMES, G. et al. *An introduction to statistical learning*. [S.l.]: Springer, 2013. v. 112. 102

JENKINS, J. D. Political economy constraints on carbon pricing policies: What are the implications for economic efficiency, environmental efficacy, and climate policy design? *Energy Policy*, Elsevier, v. 69, p. 467–477, 2014. 94

JEONG, H.; NÉDA, Z.; BARABÁSI, A.-L. Measuring preferential attachment in evolving networks. *Europhysics letters*, IOP Publishing, v. 61, n. 4, p. 567, 2003. 99

JR, D. W. H.; LEMESHOW, S.; STURDIVANT, R. X. *Applied logistic regression*. [S.l.]: John Wiley & Sons, 2013. 92

JUHÁSZ, R.; LANE, N. The political economy of industrial policy. *Journal of Economic Perspectives*, American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203-2418, v. 38, n. 4, p. 27–54, 2024. 94

KAEWUNRUEN, S. et al. Digital twin aided vulnerability assessment and risk-based maintenance planning of bridge infrastructures exposed to extreme conditions. *Sustainability*, MDPI, v. 13, n. 4, p. 2051, 2021. 100

KALOS, M.; WHITLOCK, P. *Monte Carlo Method, ISBN: 978-3-527-40760-6*. [S.l.]: John Wiley & Sons, 2008. 32

KAPLANSKI, G.; LEVY, H. Sentiment and stock prices: The case of aviation disasters. *Journal of Financial Economics*, v. 95, n. 2, p. 174–201, 2010. 72

KARPF, A.; MANDEL, A.; BATTISTON, S. Price and network dynamics in the european carbon market. *Journal of Economic Behavior & Organization*, Elsevier, v. 153, p. 103–122, 2018. 92, 93

KARYPIS, G.; HAN, E.-H.; KUMAR, V. Chameleon: Hierarchical clustering using dynamic modeling. *computer*, IEEE, v. 32, n. 8, p. 68–75, 1999. 60

KATZ, L. A new status index derived from sociometric analysis. *Psychometrika*, Springer, v. 18, n. 1, p. 39–43, 1953. 44

KEMPE, D.; KLEINBERG, J.; TARDOS, É. Maximizing the spread of influence through a social network. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2003. p. 137–146. 50

KERNIGHAN, B. W.; LIN, S. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, Nokia Bell Labs, v. 49, n. 2, p. 291–307, 1970. 56

KIPF, T. N.; WELLING, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 28

KOCH, N. et al. Politics matters: Regulatory events as catalysts for price formation under cap-and-trade. *Journal of Environmental Economics and Management*, Elsevier, v. 78, p. 121–139, 2016. 95

KOLLENBERG, S.; TASCHINI, L. Stability of the european carbon market. *Nature Climate Change*, Springer, v. 9, p. 682–687, 2019. 95

KOLMOGOROV, A. N.; FOMIN, S. V. *Introductory real analysis.* [S.l.]: Courier Corporation, 1975. 39

KOOVERJEE, N.; JAMES, S.; ZYL, T. V. Investigating transfer learning in graph neural networks. *Electronics*, MDPI, v. 11, n. 8, p. 1202, 2022. 55

KOSCHÜTZKI, D. et al. Centrality indices. *Network analysis: methodological foundations*, Springer, p. 16–61, 2005. 44

KRUIGER, J. F. et al. Graph layouts by t-sne. In: WILEY ONLINE LIBRARY. *Computer graphics forum.* [S.l.], 2017. v. 36, n. 3, p. 283–294. 68

KUMAR, A.; MEHROTRA, K. G.; MOHAN, C. K. Neural networks for fast estimation of social network centrality measures. In: SPRINGER. *Proceedings of the Fifth International Conference on Fuzzy and Neuro Computing (FANCCO-2015).* [S.l.], 2015. p. 175–184. 45

LANCICHINETTI, A.; FORTUNATO, S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, APS, v. 80, n. 1, p. 016118, 2009. 58

LAWYER, G. Understanding the influence of all nodes in a network. *Scientific reports*, Nature Publishing Group UK London, v. 5, n. 1, p. 8665, 2015. 50

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural computation*, MIT Press, v. 1, n. 4, p. 541–551, 1989. 21

LEE, D. D.; SEUNG, H. S. Learning the parts of objects by non-negative matrix factorization. *nature*, Nature Publishing Group UK London, v. 401, n. 6755, p. 788–791, 1999. 25

LEOW, Y. Y.; LAURENT, T.; BRESSON, X. Graphtsne: a visualization technique for graph-structured data. *arXiv preprint arXiv:1904.06915*, 2019. 68

LI, B.; PI, D. Learning deep neural networks for node classification. *Expert Systems with Applications*, Elsevier, v. 137, p. 324–334, 2019. 53

LI, Y. et al. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015. 28

LICHTENWALTER, R. N.; LUSSIER, J. T.; CHAWLA, N. V. New perspectives and methods in link prediction. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining.* [S.l.: s.n.], 2010. p. 243–252. 64

LILLICRAP, T. P. et al. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. 32

LIU, H. et al. A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-based systems*, Elsevier, v. 56, p. 156–166, 2014. 39

LIU, P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, ACM New York, NY, v. 55, n. 9, p. 1–35, 2023. 72

LIU, Y.; GAO, X.; GUO, J. Network features of the eu carbon trade system: An evolutionary perspective. *Energies*, MDPI, v. 11, n. 6, p. 1501, 2018. 93

LIU, Z. et al. Method to enhance traffic capacity for scale-free networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, APS, v. 76, n. 3, p. 037101, 2007. 48

LLOYD, S. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, IEEE, v. 28, n. 2, p. 129–137, 1982. 22

LÜ, L.; ZHOU, T. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, Elsevier, v. 390, n. 6, p. 1150–1170, 2011. 63

LUAN, S. et al. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021. 52

LUCERI, L.; BRAUN, T.; GIORDANO, S. Social influence (deep) learning for human behavior prediction. In: SPRINGER. *Complex Networks IX: Proceedings of the 9th Conference on Complex Networks CompleNet 2018 9*. [S.l.], 2018. p. 261–269. 51

LUO, Q. et al. A survey of structural representation learning for social networks. *Neurocomputing*, Elsevier, v. 496, p. 56–71, 2022. 18

MA, W. et al. Measuring systemic risk in china: a textual analysis. *China Finance Review International*, Emerald Publishing Limited, 2024. 71

MAATEN, L. V. D. Accelerating t-sne using tree-based algorithms. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 3221–3245, 2014. 68

MAATEN, L. Van der; HINTON, G. Visualizing data using t-sne. *Journal of machine learning research*, v. 9, n. 11, 2008. 43, 68

MACQUEEN, J. Classification and analysis of multivariate observations. In: *In 5th Berkeley Symposium on Mathematical Statistics and Probability*. [S.l.: s.n.], 1967. v. 5, n. 1, p. 281–297. 22

MALLICK, K. et al. Topo2vec: A novel node embedding generation based on network topology for link prediction. *IEEE Transactions on Computational Social Systems*, IEEE, v. 6, n. 6, p. 1306–1317, 2019. 66

MANTEGNA, R. N. Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems*, Springer, v. 11, p. 193–197, 1999. 40

MARTÍNEZ, V.; BERZAL, F.; CUBERO, J.-C. A survey of link prediction in complex networks. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 49, n. 4, p. 1–33, 2016. 63

MARTINS, L. V.; ZHAO, L. Particle competition for unbalanced community detection in complex networks. In: SPRINGER. *Brazilian Conference on Intelligent Systems*. [S.l.], 2020. p. 322–336. 63

MATSUBARA, Y. et al. Rise and fall patterns of information diffusion: model and implications. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2012. p. 6–14. 50

MATTAUCH, L. et al. The economics of climate change with endogenous preferences. *Resource and Energy Economics*, Elsevier, v. 69, p. 101312, 2022. 94, 123

MAURYA, S. K.; LIU, X.; MURATA, T. Simplifying approach to node classification in graph neural networks. *Journal of Computational Science*, Elsevier, v. 62, p. 101695, 2022. 52

MCDONALD-MADDEN, E. et al. Using food-web theory to conserve ecosystems. *Nature communications*, Nature Publishing Group UK London, v. 7, n. 1, p. 10245, 2016. 44

MCINNES, L.; HEALY, J.; MELVILLE, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 43

MELONI, S.; ARENAS, A.; MORENO, Y. Traffic-driven epidemic spreading in finite-size scale-free networks. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 106, n. 40, p. 16897–16902, 2009. 75

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 33

MITCHELL, M. *Artificial intelligence: A guide for thinking humans*. [S.l.]: Penguin UK, 2019. 20

MNIH, V. et al. Asynchronous methods for deep reinforcement learning. In: PMLR. *International conference on machine learning*. [S.l.], 2016. p. 1928–1937. 32

MNIH, V. et al. Human-level control through deep reinforcement learning. *Nature*, Nature Publishing Group, v. 518, n. 7540, p. 529–533, 2015. 32

MURATA, T.; AFZAL, N. Modularity optimization as a training criterion for graph neural networks. In: SPRINGER. *Complex Networks IX: Proceedings of the 9th Conference on Complex Networks CompleNet 2018 9*. [S.l.], 2018. p. 123–135. 61

MUSAWI, A. F. A.; ROY, S.; GHOSH, P. A review of link prediction applications in network biology. *arXiv preprint arXiv:2312.01275*, 2023. 63

MUZIO, G.; O'BRAY, L.; BORGWARDT, K. Biological network analysis with deep learning. *Briefings in bioinformatics*, Oxford University Press, v. 22, n. 2, p. 1515–1530, 2021. 18

NADKARNI, P. M.; OHNO-MACHADO, L.; CHAPMAN, W. W. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, BMJ Group BMA House, Tavistock Square, London, WC1H 9JR, v. 18, n. 5, p. 544–551, 2011. 72

NEGRE, C. F. et al. Eigenvector centrality for characterization of protein allosteric pathways. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 115, n. 52, p. E12201–E12208, 2018. 44

NEWMAN, M. *Networks*. [S.l.]: Oxford university press, 2018. 37

NEWMAN, M. E. The structure and function of complex networks. *SIAM review*, SIAM, v. 45, n. 2, p. 167–256, 2003. 67

NEWMAN, M. E. A measure of betweenness centrality based on random walks. *Social networks*, Elsevier, v. 27, n. 1, p. 39–54, 2005. 44

NEWMAN, M. E.; GIRVAN, M. Finding and evaluating community structure in networks. *Physical review E*, APS, v. 69, n. 2, p. 026113, 2004. 56

NIEMINEN, J. On the centrality in a graph. *Scandinavian journal of psychology*, Wiley Online Library, v. 15, n. 1, p. 332–336, 1974. 44

NOH, J. D.; RIEGER, H. Random walks on complex networks. *Physical review letters*, APS, v. 92, n. 11, p. 118701, 2004. 48

OLIVEIRA, T. B. de et al. Data clustering based on complex network community detection. In: IEEE. *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. [S.l.], 2008. p. 2121–2126. 59

OTTERLO, M. V.; WIERING, M. Reinforcement learning and markov decision processes. In: *Reinforcement learning: State-of-the-art*. [S.l.]: Springer, 2012. p. 3–42. 32

OU, M. et al. Asymmetric transitivity preserving graph embedding. In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2016. p. 1105–1114. 35

PAGE, L. et al. The pagerank citation ranking: Bringing order to the web. Citeseer, 1999. 44

PAIM, E. C.; BAZZAN, A. L.; CHIRA, C. Detecting communities in networks: a decentralized approach based on multiagent reinforcement learning. In: IEEE. *2020 IEEE symposium series on computational intelligence (SSCI)*. [S.l.], 2020. p. 2225–2232. 63

PAPACHRISTOU, M.; YUAN, Y. Network formation and dynamics among multi-llms. *arXiv preprint arXiv:2402.10659*, 2024. 99

PARTHASARATHY, S.; RUAN, Y.; SATULURI, V. Social network data analytics. In: ______. [S.l.]: Springer, 2011. cap. Community discovery in social networks: Applications, methods and emerging trends, p. 79–113. 59

PASTOR-SATORRAS, R.; VESPIGNANI, A. Epidemic spreading in scale-free networks. *Physical Review Letters*, APS, v. 86, n. 14, p. 3200, 2001. 75

PEARSON, K. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, The Royal Society London, n. 187, p. 253–318, 1896. 40

PEARSON, K. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, Taylor & Francis, v. 2, n. 11, p. 559–572, 1901. 25

PECLI, A.; CAVALCANTI, M. C.; GOLDSCHMIDT, R. Automatic feature selection for supervised learning in link prediction applications: a comparative study. *Knowledge and Information Systems*, Springer, v. 56, p. 85–121, 2018. 64

PEROZZI, B.; AL-RFOU, R.; SKIENA, S. Deepwalk: Online learning of social representations. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.: s.n.], 2014. p. 701–710. 33

PETRONE, D.; LATORA, V. A dynamic approach merging network theory and credit risk techniques to assess systemic risk in financial networks. *Scientific Reports*, Nature Publishing Group UK London, v. 8, n. 1, p. 5561, 2018. 72

POTHEN, A. Graph partitioning algorithms with applications to scientific computing. In: *Parallel Numerical Algorithms*. [S.l.]: Springer, 1997. p. 323–368. 56

PUZIS, R. et al. Embedding-centrality: Generic centrality computation using neural networks. In: SPRINGER. *Complex Networks IX: Proceedings of the 9th Conference on Complex Networks CompleNet 2018 9*. [S.l.], 2018. p. 87–97. 49

QIU, J. et al. Netsmf: Large-scale network embedding as sparse matrix factorization. In: *The World Wide Web Conference*. [S.l.: s.n.], 2019. p. 1509–1520. 36

QIU, J. et al. Deepinf: Social influence prediction with deep learning. In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. [S.l.: s.n.], 2018. p. 2110–2119. 50

QUINLAN, J. R. Induction of decision trees. *Machine learning*, Springer, v. 1, p. 81–106, 1986. 21

RAHMAN, M. M.; BHATTACHARYA, P.; DESAI, B. C. A framework for medical image retrieval using machine learning and statistical similarity matching techniques with relevance feedback. *IEEE transactions on Information Technology in Biomedicine*, IEEE, v. 11, n. 1, p. 58–69, 2007. 39

RAKARADDI, A.; PRATAMA, M. Unsupervised learning for identifying high eigenvector centrality nodes: A graph neural network approach. In: IEEE. *2021 IEEE International Conference on Big Data (Big Data)*. [S.l.], 2021. p. 4945–4954. 47

REN, Y. et al. Improving generalization of reinforcement learning with minimax distributional soft actor-critic. In: IEEE. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.], 2020. p. 1–6. 32

RESCE, G.; ZINILLI, A.; CERULLI, G. Machine learning prediction of academic collaboration networks. *Scientific Reports*, Nature Publishing Group UK London, v. 12, n. 1, p. 21993, 2022. 63

RESTREPO, J. G.; OTT, E.; HUNT, B. R. Characterizing the dynamical importance of network nodes and links. *Physical review letters*, APS, v. 97, n. 9, p. 094102, 2006. 44

REZAEI, A. A. et al. A machine learning-based approach for vital node identification in complex networks. *Expert Systems with Applications*, Elsevier, v. 214, p. 119086, 2023. 64

RODRIGUES, F. A. Network centrality: an introduction. In: *A mathematical modeling approach from nonlinear dynamics to complex systems*. [S.l.]: Springer, 2018. p. 177–196. 44

RONCORONI, A. et al. Interconnected banks and systemically important exposures. *Journal of Economic Dynamics and Control*, Elsevier, v. 133, p. 104266, 2021. 72

RONG, Y. et al. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019. 52

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958. 21

ROSVALL, M. et al. Searchability of networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, APS, v. 72, n. 4, p. 046117, 2005. 48

ROWEIS, S. T.; SAUL, L. K. Nonlinear dimensionality reduction by locally linear embedding. *science*, American Association for the Advancement of Science, v. 290, n. 5500, p. 2323–2326, 2000. 35

ROZEMBERCZKI, B.; SARKAR, R. Fast sequence-based embedding with diffusion graphs. In: SPRINGER. *Complex Networks IX: Proceedings of the 9th Conference on Complex Networks CompleNet 2018 9*. [S.l.], 2018. p. 99–107. 61

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *nature*, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986. 21

RUMMERY, G. A.; NIRANJAN, M. *On-line Q-learning using connectionist systems*. [S.l.]: University of Cambridge, Department of Engineering Cambridge, UK, 1994. v. 37. 32

SABIDUSSI, G. The centrality index of a graph. *Psychometrika*, Springer, v. 31, n. 4, p. 581–603, 1966. 44

SABZEKAR, S.; MALAKSHAH, M. R. V.; AMINI, Z. Unsupervised learning for topological classification of transportation networks. *arXiv preprint arXiv:2311.13887*, 2023. 44

SALLAN, J. M.; LORDAN, O. *Air route networks through complex networks theory*. [S.l.]: Elsevier, 2019. 37

SALTON, G.; WONG, A.; YANG, C.-S. A vector space model for automatic indexing. *Communications of the ACM*, ACM New York, NY, USA, v. 18, n. 11, p. 613–620, 1975. 40

SAMI, N. M.; NAEINI, M. Machine learning applications in cascading failure analysis in power systems: A review. *Electric Power Systems Research*, Elsevier, v. 232, p. 110415, 2024. 44

SARKAR, P.; MOORE, A. W. Social network data analytics. In: ______. [S.l.]: Springer, 2011. cap. Random walks in social networks and their applications: a survey, p. 43–77. 33

SATO, M. et al. Allocation, allocation, allocation! the political economy of the development of the european union emissions trading system. *Wiley Interdisciplinary Reviews: Climate Change*, Wiley Online Library, v. 13, n. 5, p. e796, 2022. 95

SCARSELLI, F. et al. The graph neural network model. *IEEE transactions on neural networks*, IEEE, v. 20, n. 1, p. 61–80, 2008. 28

SCHNEIDER, M.; ERTEL, W.; RAMOS, F. Expected similarity estimation for large-scale batch and streaming anomaly detection. *Machine Learning*, Springer, v. 105, p. 305–333, 2016. 39

SCHULMAN, J. et al. Trust region policy optimization. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 1889–1897. 32

SCHULMAN, J. et al. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 32

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. *Introduction to information retrieval*. [S.l.]: Cambridge University Press Cambridge, 2008. v. 39. 26, 40

SESHADHRI, C. et al. The impossibility of low-rank representations for triangle-rich complex networks. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 117, n. 11, p. 5631–5637, 2020. 36

SHARAN, R.; ULITSKY, I.; SHAMIR, R. Network-based prediction of protein function. *Molecular systems biology*, John Wiley & Sons, Ltd Chichester, UK, v. 3, n. 1, p. 88, 2007. 64

SHAW, M. E. Group structure and the behavior of individuals in small groups. *The Journal of psychology*, Taylor & Francis, v. 38, n. 1, p. 139–149, 1954. 44

SHLENS, J. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014. 25

SHUVRO, R. A. et al. Predicting cascading failures in power grids using machine learning algorithms. In: IEEE. *2019 North American Power Symposium (NAPS)*. [S.l.], 2019. p. 1–6. 44

SHVYDUN, S. Models of similarity in complex networks. *PeerJ Computer Science*, PeerJ Inc., v. 9, p. e1371, 2023. 39

SIBSON, R. Slink: an optimally efficient algorithm for the single-link cluster method. *The computer journal*, Oxford University Press, v. 16, n. 1, p. 30–34, 1973. 24, 43

SILVA, E. L. C. da et al. Combining k-means method and complex network analysis to evaluate city mobility. In: IEEE. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.], 2016. p. 1666–1671. 59

SILVA, T. C.; ZHAO, L. *Machine learning in complex networks*. [S.l.]: Springer, 2016. 17, 39

SILVER, D. et al. Deterministic policy gradient algorithms. In: PMLR. *International conference on machine learning*. [S.l.], 2014. p. 387–395. 32

SINGH, R. H. et al. Movie recommendation system using cosine similarity and knn. *International Journal of Engineering and Advanced Technology*, v. 9, n. 5, p. 556–559, 2020. 39

SNEPPEN, K.; TRUSINA, A.; ROSVALL, M. Hide-and-seek on complex networks. *Europhysics Letters*, IOP Publishing, v. 69, n. 5, p. 853, 2005. 48

SORENSEN, T. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biologiske skrifter*, v. 5, p. 1–34, 1948. 41

SPILIOPOULOU, M. Evolution in social networks: A survey. *Social network data analytics*, Springer, p. 149–175, 2011. 19

STAMOS, I. Transportation networks in the face of climate change adaptation: A review of centrality measures. *Future Transportation*, Multidisciplinary Digital Publishing Institute, v. 3, n. 3, p. 878–900, 2023. 44

SU, Z. et al. Link prediction in recommender systems based on vector similarity. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 560, p. 125154, 2020. 63

SUMMER, M. Financial contagion and network analysis. *Annu. Rev. Financ. Econ.*, Annual Reviews, v. 5, n. 1, p. 277–297, 2013. 72

SUN, J.; TANG, J. A survey of models and algorithms for social influence analysis. *Social network data analytics*, Springer, p. 177–214, 2011. 50

SUTTON, R. S. Learning to predict by the methods of temporal differences. *Machine learning*, Springer, v. 3, p. 9–44, 1988. 32

SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: An introduction*. [S.l.]: MIT press, 2018. 31

TADIĆ, B.; THURNER, S.; RODGERS, G. J. Traffic on complex networks: Towards understanding global statistical properties from microscopic density fluctuations. *Physical Review E*, APS, v. 69, n. 3, p. 036102, 2004. 48

TANG, J.; AGGARWAL, C.; LIU, H. Node classification in signed social networks. In: SIAM. *Proceedings of the 2016 SIAM international conference on data mining*. [S.l.], 2016. p. 54–62. 52

TANG, J. et al. Line: Large-scale information network embedding. In: *Proceedings of the 24th international conference on world wide web*. [S.l.: s.n.], 2015. p. 1067–1077. 35

TANG, Y.; WANG, F.; ZHOU, W. Network structure indicators predict ecological robustness in food webs. *Ecological Research*, Wiley Online Library, v. 39, n. 5, p. 766–774, 2024. 44

TEDESCHI, G.; IORI, G.; GALLEGATI, M. The role of communication and imitation in limit order markets. *The European Physical Journal B*, Springer, v. 71, p. 489–497, 2009. 71

TEDESCHI, G.; IORI, G.; GALLEGATI, M. Herding effects in order driven markets: The rise and fall of gurus. *Journal of Economic Behavior & Organization*, Elsevier, v. 81, n. 1, p. 82–96, 2012. 71

TETLOCK, P. C. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, v. 62, n. 3, p. 1139–1168, 2007. 72

TOLEDO, T. sknet: A python framework for machine learning in complex networks. *Journal of Open Source Software*, v. 6, n. 68, p. 3864, 2021. 20

TOPIRCEANU, A.; UDRESCU, M.; MARCULESCU, R. Weighted betweenness preferential attachment: A new mechanism explaining social network formation and evolution. *Scientific reports*, Nature Publishing Group UK London, v. 8, n. 1, p. 10871, 2018. 99

VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017. 22

VELICKOVIC, P. et al. Graph attention networks. *stat*, v. 1050, n. 20, p. 10–48550, 2017. 29

VELIČKOVIĆ, P. et al. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 101

VIJAYMEENA, M.; KAVITHA, K. A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, v. 3, n. 2, p. 19–28, 2016. 39

WACHS, J.; KERTÉSZ, J. A network approach to cartel detection in public auction markets. *Scientific reports*, Nature Publishing Group UK London, v. 9, n. 1, p. 10818, 2019. 42

WAN, X. et al. Sentiment correlation in financial news networks and associated market movements. *Scientific reports*, Nature Publishing Group UK London, v. 11, n. 1, p. 3062, 2021. 76

WANDELT, S.; SHI, X.; SUN, X. Complex network metrics: Can deep learning keep up with tailor-made reference algorithms? *IEEE Access*, IEEE, v. 8, p. 68114–68123, 2020. 33, 49

WANG, D. E. J. Fast approximation of centrality. *Graph algorithms and applications*, v. 5, n. 5, p. 39, 2006. 45

WANG, M.; WANG, H.; ZHENG, H. A mini review of node centrality metrics in biological networks. *International Journal of Network Dynamics and Intelligence*, v. 1, n. 1, p. 99–110, 2022. 44

WANG, P. et al. Link prediction in social networks: the state-of-the-art. *arXiv preprint arXiv:1411.5118*, 2014. 63

WANG, Y. et al. Nodeaug: Semi-supervised node classification with data augmentation. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. [S.l.: s.n.], 2020. p. 207–217. 52

WANG, Y. et al. A brief review of network embedding. *Big Data Mining and Analytics*, TUP, v. 2, n. 1, p. 35–47, 2018. 18

WATKINS, C. J.; DAYAN, P. Q-learning. *Machine learning*, Springer, v. 8, p. 279–292, 1992. 32

WATKINS, C. J. C. H. Learning from delayed rewards. King's College, Cambridge United Kingdom, 1989. 32

WITTEN, D. M.; TIBSHIRANI, R. J. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical applications in genetics and molecular biology*, De Gruyter, v. 8, n. 1, 2009. 67

WU, Q. et al. Nodeformer: A scalable graph structure learning transformer for node classification. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2022. 54

WU, Z. et al. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, IEEE, v. 32, n. 1, p. 4–24, 2020. 101

XIA, F. et al. Random walks: A review of algorithms and applications. *IEEE Transactions on Emerging Topics in Computational Intelligence*, IEEE, v. 4, n. 2, p. 95–107, 2019. 48

XIAO, C.; HONG, S.; HUANG, W. Optimizing graph layout by t-sne perplexity estimation. *International Journal of Data Science and Analytics*, Springer, v. 15, n. 2, p. 159–171, 2023. 68

XU, H.; WANG, M. A novel carbon price fluctuation trend prediction method based on complex network and classification algorithm. *Complexity*, Wiley Online Library, v. 2021, n. 1, p. 3052041, 2021. 93

XU, H. et al. Carbon price forecasting with complex network and extreme learning machine. *Physica A: Statistical Mechanics and its Applications*, Elsevier, v. 545, p. 122830, 2020. 93

XU, K. et al. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018. 30

XUAN, D.; YU, H.; WANG, J. A novel method of centrality in terrorist network. In: IEEE. *2014 Seventh International Symposium on Computational Intelligence and Design*. [S.l.], 2014. v. 2, p. 144–149. 50

YANG, S.-J. Exploring complex networks by walking on them. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, APS, v. 71, n. 1, p. 016107, 2005. 48

YUN, T.-S.; JEONG, D.; PARK, S. "too central to fail" systemic risk measure using pagerank algorithm. *Journal of Economic Behavior & Organization*, Elsevier, v. 162, p. 251–272, 2019. 44

ZADEH, R. B.; GOEL, A. Dimension independent similarity computation. *Journal of Machine Learning Research*, v. 14, n. 6, 2013. 39

ZANIN, M. et al. Combining complex networks and data mining: why and how. *Physics Reports*, Elsevier, v. 635, p. 1–44, 2016. 17, 96

ZHANG, D. et al. Network representation learning: A survey. *IEEE transactions on Big Data*, IEEE, v. 6, n. 1, p. 3–28, 2018. 18

ZHANG, M.; CHEN, Y. Link prediction based on graph neural networks. *Advances in neural information processing systems*, v. 31, 2018. 92

ZHANG, Z.; CUI, P.; ZHU, W. Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 34, n. 1, p. 249–270, 2020. 58

ZHAO, K.; ZHAO, D. Link prediction in dynamic real-life friendship networks: A case study. In: IEEE. *2024 4th International Conference on Computer Communication and Artificial Intelligence (CCAI)*. [S.l.], 2024. p. 137–141. 63

ZHAO, T.; ZHANG, X.; WANG, S. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In: *Proceedings of the 14th ACM international conference on web search and data mining*. [S.l.: s.n.], 2021. p. 833–841. 52

ZHU, J. et al. Community detection in graph: an embedding method. *IEEE Transactions on Network Science and Engineering*, IEEE, v. 9, n. 2, p. 689–702, 2021. 62

ZHU, S.; ZHAN, J.; LI, X. Identifying influential nodes in complex networks using a gravity model based on the h-index method. *Scientific Reports*, Nature Publishing Group UK London, v. 13, n. 1, p. 16404, 2023. 50

ZOU, Y.; LI, T.; LUO, Z.-f. Node centrality approximation for large networks based on inductive graph neural networks. *arXiv preprint arXiv:2403.04977*, 2024. 49

Appendix

APPENDIX A – INDIRECT CONTAGION AND SYSTEMIC RISK: A NEWS SIMILARITY NETWORK APPROACH

A.1 Additional Figures and Tables

This appendix presents supplementary figures and tables that support the main analysis. Figure 26 visualizes the randomized network of firms, providing a baseline comparison against the actual news similarity network. Figure 27 displays the largest connected component of the network, showing how firms are linked based on media coverage. These additional materials complement the main text by offering further evidence on the network structure, firm interconnections, and the statistical validity of the findings. Table 9 provides a dictionary of variables, detailing the financial and market indicators used in the study. Table 10 categorizes companies based on their sector and network membership, distinguishing between the largest connected network, the fully connected network, and the Financials sector network. Table 11 compares firm groupings by GICS sector and the detected communities, illustrating how news similarity networks capture relationships beyond conventional classifications. Table 12 reports White's test results and error metrics to assess the robustness of our regression models.

Variable	Description
EBITDA*	Earnings before interest, taxes, depreciation, and amortization from the most recent quarterly financial report.
avg10Volume*	Average trading volume over the last 10 calendar days.
avg30Volume*	Average trading volume over the last 30 calendar days.
beta*	A measure of the asset or portfolio's volatility relative to the overall market. Levered beta calculated using 1 year of historical data compared to SPY.
companyName*	Name of the company associated with the security.
currentDebt*	Total current debt reported by the company.
day200MovingAvg*	200-day moving average based on calendar days.
day30ChangePercent*	Percentage change over the last 30 calendar days.
day50MovingAvg*	50-day moving average based on calendar days.
day5ChangePercent*	Percentage change over the last 5 calendar days.
debtToEquity*	Debt-to-equity ratio, calculated as total liabilities divided by shareholder equity.
dividendYield*	Trailing twelve-month dividend yield, calculated as the ratio of dividend rate to the previous day's closing price, expressed as a percentage.
employees*	Total number of employees in the company.
enterpriseValue*	Enterprise value (EV), representing the total value of the company, often viewed as an alternative to equity market capitalization.
enterpriseValueToRevenue*	Enterprise value-to-revenue (EV/R), comparing the company's enterprise value to its revenue.
exDividendDate*	Date of the last ex-dividend.
float	Value is 'null' as of December 1, 2020.
forwardPERatio*	Forward price-to-earnings ratio, using forecasted earnings for the calculation.
grossProfit*	Gross profit, defined as revenue minus the cost of goods or services sold.
marketcap*	Market capitalization, calculated as shares outstanding multiplied by the previous day's close.
maxChangePercent*	Maximum percentage change over calendar days.
month1ChangePercent*	Percentage change over the last 1 month (calendar days).
month3ChangePercent*	Percentage change over the last 3 months (calendar days).
month6ChangePercent*	Percentage change over the last 6 months (calendar days).
nextDividendDate*	Expected ex-dividend date of the next dividend payment.
nextEarningsDate*	Announced date of the next earnings report.
peHigh*	52-week high of the price-to-earnings ratio.
peLow*	52-week low of the price-to-earnings ratio.
peRatio*	Price-to-earnings (P/E) ratio.
pegRatio*	Price-to-earnings-to-growth (PEG) ratio, calculated as the P/E ratio divided by the trailing twelve-month earnings growth rate.
priceToBook*	Price-to-book ratio (P/B), comparing the current market price to the book value of the company.
priceToSales*	Price-to-sales ratio (P/S), calculated as market capitalization divided by annual revenue.
profitMargin*	Net profit margin, expressed as a percentage of revenue.
putCallRatio*	Total put option volume divided by total call option volume for all available option contracts.
revenue*	Total revenue generated from the company's primary operations over the last twelve months.
revenuePerEmployee*	Revenue per employee, calculated as total revenue divided by the number of employees.
revenuePerShare*	Revenue per share, calculated as total revenue divided by outstanding shares.

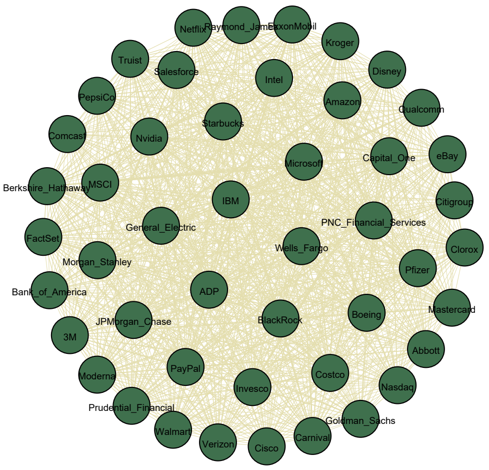


Figure 26 – Companies in the randomized network.

Variable	**Description**
sharesOutstanding*	Total shares outstanding, calculated as issued shares minus treasury shares.
totalCash*	Total cash available to the company.
totalRevenue*	Revenue generated from the sale of goods or services through primary operations over the last twelve months.
ttmDividendRate*	Trailing twelve-month dividend rate per share.
ttmEPS*	Trailing twelve-month earnings per share.
week52change*	Percentage change based on the last 52 calendar weeks.
week52high*	Highest adjusted price during trading hours in the last 52 calendar weeks.
week52highDate*	Date corresponding to the 52-week high.
week52highDateSplitAdjustOnly*	Date corresponding to the 52-week high, adjusted for stock splits only.
week52highSplitAdjustOnly*	Highest split-adjusted price observed during the last 52 calendar weeks.
week52low*	Lowest adjusted price during trading hours in the last 52 calendar weeks.
week52lowDate*	Date corresponding to the 52-week low.
week52lowDateSplitAdjustOnly*	Date corresponding to the 52-week low, adjusted for stock splits only.
week52lowSplitAdjustOnly*	Lowest split-adjusted price observed during the last 52 calendar weeks.
year1ChangePercent*	Percentage change over the last year based on calendar days.
year2ChangePercent*	Percentage change over the last 2 years based on calendar days.
year5ChangePercent*	Percentage change over the last 5 years based on calendar days.
ytdChangePercent*	Year-to-date percentage change based on calendar days.
close	Adjusted closing price for historical dates, split-adjusted only.
high	Adjusted high price for historical dates, split-adjusted only.
low	Adjusted low price for historical dates, split-adjusted only.
open	Adjusted opening price for historical dates, split-adjusted only.
symbol	Stock ticker symbol.
volume	Adjusted trading volume for historical dates, split-adjusted only.
changeOverTime	Percentage change of each interval relative to the first value, useful for stock comparisons.
marketChangeOverTime	Percentage change of each interval relative to the first value, based on 15-minute delayed consolidated data.
uOpen	Unadjusted opening price for historical dates.
uClose	Unadjusted closing price for historical dates.
uHigh	Unadjusted high price for historical dates.
uLow	Unadjusted low price for historical dates.
uVolume	Unadjusted trading volume for historical dates.
fOpen	Fully adjusted opening price for historical dates.
fClose	Fully adjusted closing price for historical dates.
fHigh	Fully adjusted high price for historical dates.
fLow	Fully adjusted low price for historical dates.
fVolume	Fully adjusted trading volume for historical dates.
label	Human-readable date format, depending on the data range.
change	Daily change from the previous trading day.
changePercent	Daily percentage change from the previous trading day.

Table 9 – Dictionary of Variables

Panel (a) The Largest Network		**Panel (b) The Fully Connected Network**		**Panel (c) The Financials Network**	
3M	MMM	3M	MMM	Bank of America*,†	BAC
ADP	ADP	ADP	ADP	Berkshire Hathaway	BBK.B
Abbot	ABT	Abbot	ABT	BlackRock	BLK
Amazon	AMZN	Amazon	AMZN	Capital One	COF
Bank of America	BAC	Bank of America	BAK	Citigroup*,†	C
Berkshire Hataway	BBK.B	Berkshire Hataway	BBK.B	FactSet ‡	FDS
BlackRock	BLK	BlackRock	BLK	Goldman Sachs*,†	GS
Boeing	BA	Boeing	BA	Invesco	IVZ
Capital One	COF	Capital One	COF	JP Morgan Chase*,†	JPM
Carnival	CCL	Carnival	CCL	MSCI	MSCI
Cisco	CSCO	Citigroup	C	Morgan Stanley*,†	MS
Citigroup	C	Clorox	CLX	Nasdaq	NDAQ
Clorox	CLX	Costco	COST	PNC Financial Services*,‡	PNC
Comcast	CMCSA	Disney	DIS	Prudential Financial	PRU
Costco	COST	ExxonMobil	XOM	Raymond James	RJF
Disney	DIS	FactSet	FDS	Truist*,‡	TFC
ExxonMobil	XOM	General Electric	GE	Wells Fargo*,†	WFC
FactSet	FDS	Goldman Sachs	GS		
General Electric	GE	IBM	IBM	**Panel (d) The Fully Connected Financial Network**	
Goldman Sachs	GS	Intel	INTC		
IBM	IBM	Invesco	IVZ	Bank of America	BAC
Intel	INTC	JP Morgan Chase	JPM	Citigroup	C
Invesco	IVZ	Kroger	KR	Goldman Sachs	GS
JP Morgan Chase	JPM	Mastercard	MA	JP Morgan Chase	JPM
Kroger	KR	Microsoft	MSFT	Morgan Stanley	MS
MSCI	MSCI	Moderna	MRNA	Wells Fargo	WFC
Mastercard	MA	Morgan Stanley	MS	FactSet	FDS
Microsoft	MSFT	Nasdaq	NDAQ	PNC Financial Services	PNC
Moderna	MRNA	Netflix	NFLX	Truist	TFC
Morgan Stanley	MS	Nvidia	NVDA		
Nasdaq	NDAQ	PNC Financial Services	PNC		
Netflix	NFLX	PayPal	PYPL		
Nvidia	NVDA	PepsiCo	PEP		
PNC Financial Services	PNC	Pfizer	PFE		
PayPal	PYPL	Prudential Financial	PRU		
PepsiCo	PEP	Raymond James	RJF		
Pfizer	PFE	Salesforce	CRM		
Prudential Financial	PRU	Starbucks	SBUX		
Qualcomm	QCOM	Truist	TFC		
Raymond James	RJF	Verison	VZ		
Salesforce	CRM	Walmart	WMT		
Starbucks	SBUX	Wells Gargo	WFC		
Truist	TFC	eBay	EBAY		
Verison	VZ				
Walmart	WMT				
Wells Gargo	WFC				
eBay	EBAY				

Table 10 – Overview of Networks and Company Labels. We categorize companies into different networks based on their sector of activity. An asterisk ($*$) indicates that a company is part of the Financials Industry Network. A dagger (†) denotes membership in the Banking Industry Network. A double dagger (‡) signifies inclusion in the Other Financials Services Industry Network.

Symbol	Company	GICS Sector	Community
BAC	Bank of America	Financials	Community 1
GS	Goldman Sachs	Financials	Community 1
JPM	JPMorgan Chase	Financials	Community 1
MS	Morgan Stanley	Financials	Community 1
WFC	Wells Fargo	Financials	Community 1
DIS	Disney	Communication Services	Community 2
NFLX	Netflix	Communication Services	Community 2
AMZN	Amazon	Consumer Discretionary	Community 2
SBUX	Starbucks	Consumer Discretionary	Community 2
WMT	Walmart	Consumer Staples	Community 2
BLK	BlackRock	Financials	Community 2
BRK.B	Berkshire Hathaway	Financials	Community 2
C	Citigroup	Financials	Community 2
COF	Capital One	Financials	Community 2
NDAQ	Nasdaq	Financials	Community 2
PNC	PNC Financial Services	Financials	Community 2
PRU	Prudential Financial	Financials	Community 2
ABT	Abbott	Health Care	Community 2
MRNA	Moderna	Health Care	Community 2
PFE	Pfizer	Health Care	Community 2
BA	Boeing	Industrials	Community 2
GE	General Electric	Industrials	Community 2
MSFT	Microsoft	Information Technology	Community 2
VZ	Verizon	Communication Services	Community 3
CCL	Carnival	Consumer Discretionary	Community 3
EBAY	eBay	Consumer Discretionary	Community 3
CLX	Clorox	Consumer Staples	Community 3
COST	Costco	Consumer Staples	Community 3
KR	Kroger	Consumer Staples	Community 3
PEP	PepsiCo	Consumer Staples	Community 3
XOM	ExxonMobil	Energy	Community 3
FDS	FactSet	Financials	Community 3
IVZ	Invesco	Financials	Community 3
RJF	Raymond James	Financials	Community 3
TFC	Truist	Financials	Community 3
MMM	3M	Industrials	Community 3
ADP	ADP	Information Technology	Community 3
CRM	Salesforce	Information Technology	Community 3
IBM	IBM	Information Technology	Community 3
INTC	Intel	Information Technology	Community 3
MA	Mastercard	Information Technology	Community 3
NVDA	Nvidia	Information Technology	Community 3
PYPL	PayPal	Information Technology	Community 3

Table 11 – Comparision of Companies by GICS Sector and Community .

Panel (a) White's Test		**Panel (b) Error Metrics**	
Test Statistics	14.1377	Mean Absolute Error (MAE)	0.0319
Test Statistics p-value	0.1175	Mean Squared Error (MSE)	0.0013
F-Statistics	3.8417	Root Mean Squared Error (RMSE)	0.0369
F-Statistics p-value	0.04487		

Table 12 – **Panel (a)**: White's Test. **Panel (b)**: Error Metrics.

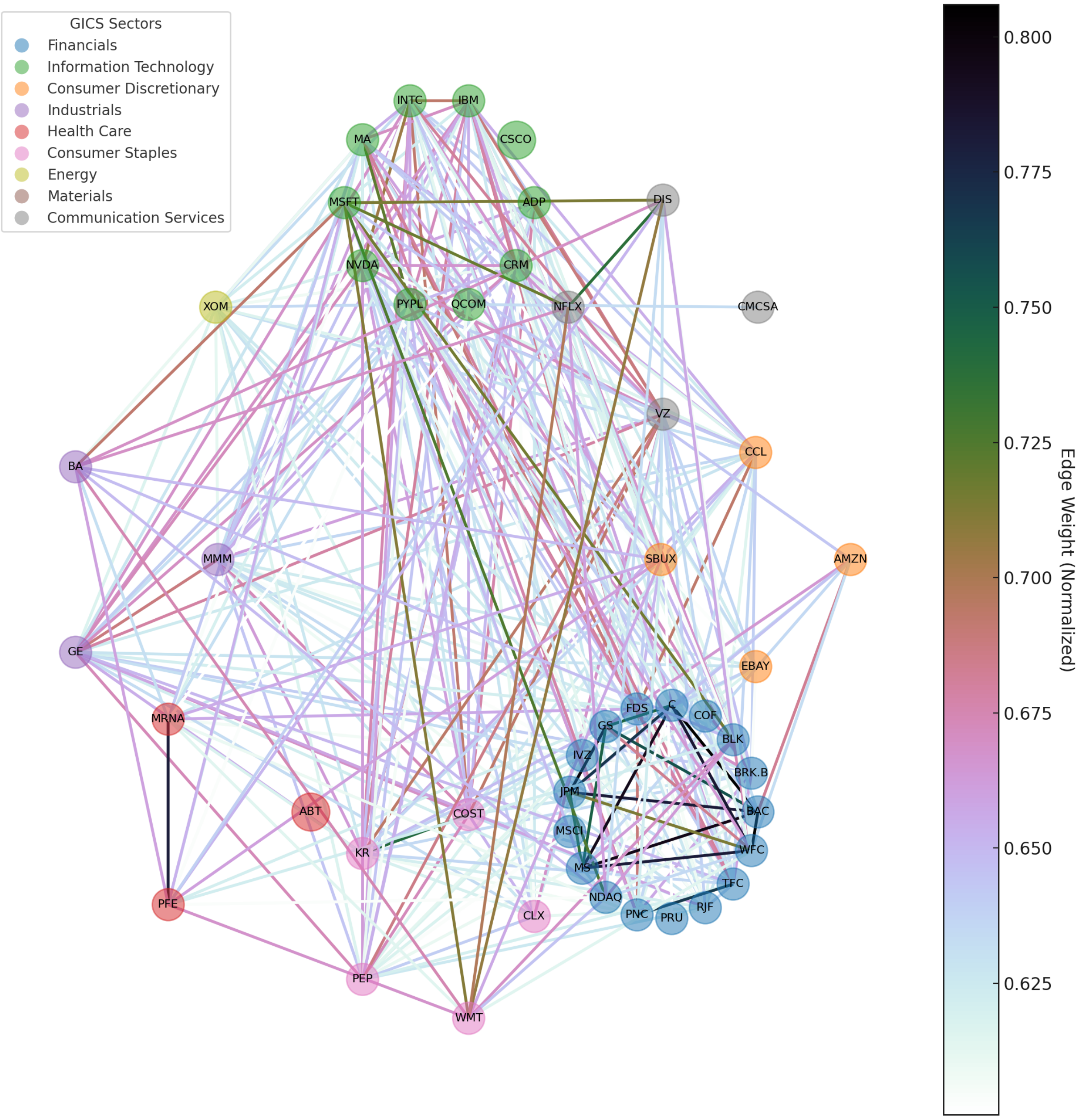


Figure 27 – The Largest Network.

APPENDIX B – DIGITAL TWINS AND NETWORK RESILIENCE IN THE EU ETS: ANALYZING STRUCTURAL SHIFTS IN CARBON TRADING

B.1 Additional Tables

The tables in this appendix provide useful reference data for the analysis. Table 13 define the variables in the transaction and account datasets, clarifying how trading relationships are recorded. Table 14 lists country codes, ensuring consistency in the interpretation of registry identifiers. Table 15 tracks how countries transition between trading communities across Phases I-IV, illustrating network reconfigurations. Finally, Table 16 ranks countries by degree centrality in each phase, revealing shifts in market influence over time. These tables support the empirical results by documenting fundamental structural elements of the EU ETS network.

Variable	**Description**
Transaction variables	
transactionID	ID of the transaction in which the transaction block took place
transferringAccount id	Identifier of account that transferred the permits
acquiringAccount id	Identifier of the account that acquired permits
amount	Number of units transferred
Account variables	
id	Unique account identifier
name	Name of account
registry id	2-letter ISO code for registry

Table 13 – Dictionary of Variables for Transaction and Account Data. The `registry_id` codes correspond to those presented in Table 14.

Code	**Description**	**Code**	**Description**
AT	Austria	AU	Australia
BE	Belgium	BG	Bulgaria
CDM	Clean Development Mechanism	CH	Switzerland
CY (CY0)	Cyprus	CZ	Czech Republic
DE	Germany	DK	Denmark
EC	European Commission	EE	Estonia
ES	Spain	EU	European Union
FI	Finland	FR	France
GB	United Kingdom	GR	Greece
HR	Croatia	HU	Hungary
IE	Ireland	IS	Iceland
IT	Italy	JP	Japan
LI	Liechtenstein	LT	Lithuania
LU	Luxembourg	LV	Latvia
MT (MT0)	Malta	NL	Netherlands
NO	Norway	NZ	New Zealand
PL	Poland	PT	Portugal
RO	Romania	RU	Russian Federation
SE	Sweden	SI	Slovenia
SK	Slovakia	UA	Ukraine
XI	Northern Ireland		

Table 14 – Country Codes and Descriptions

	Community			
Country	**Phase I**	**Phase II**	**Phase III**	**Phase IV**
AT	5	2	4	3
AU	-	5	2	-
BE	2	4	5	8
BG	-	3	3	21
CDM	-	4	5	-
CH	-	2	4	-
CY	-	5	3	7
CY0	1	6	-	-
CZ	2	3	2	21
DE	5	1	5	21
DK	4	3	1	16
EE	4	2	4	20
ES	3	4	3	2
EU	-	5	3	20
FI	4	2	4	17
FR	1	1	2	19
GB	1	5	2	20
GR	1	1	3	15
HR	-	8	1	18
HU	2	3	2	21
IE	5	5	4	10
IS	-	5	1	-
IT	3	5	1	14
JP	-	3	2	-
LI	-	2	5	-
LT	4	1	4	4
LU	1	4	5	1
LV	4	2	4	11
MT	-	5	3	21
MT0	6	7	-	-
NL	2	1	2	6
NO	-	4	4	-
NZ	-	1	2	-
PL	2	2	4	13
PT	3	5	3	9
RO	-	2	3	12
RU	-	1	5	-
SE	5	4	4	20
SI	5	1	1	5
SK	2	2	2	10
UA	-	4	4	-
XI	-	5	2	-

Table 15 – Community Transitions of Countries across Phases I - IV

Country	**Phase I**	**Country**	**Phase II**	**Country**	**Phase III**	**Country**	**Phase IV**
FR	1.88	GB	1.78	GB	1.95	EU	1.00
NL	1.88	DE	1.59	DE	1.90	DE	0.21
GB	1.79	FR	1.59	NL	1.90	BG	0.18
DK	1.71	NL	1.59	IT	1.74	SK	0.14
DE	1.58	DK	1.54	ES	1.69	IE	0.14
AT	1.46	IT	1.39	EU	1.69	HU	0.14
ES	1.25	CH	1.34	FR	1.69	CZ	0.14
CZ	1.17	PL	1.32	CH	1.44	SI	0.11
FI	1.17	ES	1.27	PL	1.41	RO	0.11
IT	1.08	AT	1.22	CZ	1.33	PT	0.11
BE	1.04	BE	1.22	BE	1.31	PL	0.11
PL	0.88	CZ	1.20	BG	1.31	NL	0.11
SE	0.88	SE	1.20	NO	1.28	LV	0.11
SK	0.83	SK	1.17	AT	1.26	LU	0.11
HU	0.79	FI	1.10	SE	1.26	LT	0.11
IE	0.79	LI	1.10	SI	1.23	IT	0.11
LT	0.79	NO	1.07	DK	1.13	HR	0.11
EE	0.75	RO	1.07	FI	1.13	GR	0.11
LV	0.71	EE	1.05	IE	1.08	FR	0.11
PT	0.63	BG	0.98	MT	1.05	FI	0.11
GR	0.42	HU	0.93	RO	1.05	ES	0.11
SI	0.38	IE	0.93	SK	1.03	EE	0.11
LU	0.25	SI	0.90	EE	0.90	DK	0.11
CY0	0.17	EU	0.80	HU	0.90	CY	0.11
MT0	0.08	PT	0.78	LT	0.85	BE	0.11
-	-	LV	0.76	PT	0.85	AT	0.11
-	-	GR	0.73	GR	0.79	MT	0.07
-	-	LT	0.73	LU	0.77	SE	0.04
-	-	JP	0.68	LV	0.77	GB	0.04
-	-	LU	0.66	JP	0.64	-	-
-	-	CDM	0.39	CY	0.59	-	-
-	-	NZ	0.34	HR	0.59	-	-
-	-	UA	0.24	IS	0.59	-	-
-	-	CY	0.17	CDM	0.44	-	-
-	-	XI	0.15	NZ	0.38	-	-
-	-	RU	0.12	XI	0.38	-	-
-	-	IS	0.10	AU	0.36	-	-
-	-	MT	0.10	RU	0.36	-	-
-	-	AU	0.05	LI	0.28	-	-
-	-	CY0	0.05	UA	0.26	-	-
-	-	HR	0.05	-	-	-	-
-	-	MT0	0.05	-	-	-	-

Table 16 – Ranking of Centrality