Leveraging The Potential of Multi-layer Networks For Subgroup Discovery

DISSERTATION

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)
genehmigt durch die Fakultät für Informatik der
Otto-von-Guericke-Universität Magdeburg
von M.Sc. Clara Ramos Teixeira Puga
geb. am 06.05.1994 in Porto, Portugal

Datum der Einreichung: 26. März 2024
Datum der Verteidigung: 27. Mai 2024

Gutachterinnen/Gutachter
Prof. Dr. rer. nat./Griechenland habil. Myra Spiliopoulou
Prof. Dr. Ernestina Menasalvas
Prof. Dr. Pedro Pereira Rodrigues

Abstract

Multi-layer networks have emerged as a powerful tool for representing complex systems that exist in various domains, such as social networks and biological systems. These networks are capable of encapsulating diverse interactions and relationships across multiple dimensions, making them an ideal framework for modeling complex systems.

This thesis explores the usefulness of multi-layer networks in modeling complex systems and highlights their advantages in capturing subtle and nuanced relationships. One domain where this approach is particularly useful is medical research, where understanding heterogeneity among patient populations is crucial for personalized treatment strategies.

Traditional clustering methods have been used though they are not able to capture feature interaction, which can be useful in developing personalized treatment plans. However, using multi-layer networks, this thesis demonstrates a novel approach to subgroup discovery in medical research. By integrating various data modalities such as genetic profiles, clinical variables, and environmental factors into a unified multi-layer framework, the intricate interplay among these factors can be elucidated.

In addition to that, the thesis focuses on the complexity of patient data and addresses the issue of cost associated with acquiring and processing features. Traditional methods tend to include all features available, which can lead to computational inefficiencies and obscure relevant patterns. To tackle this challenge, the thesis proposes a framework for feature selection within the multi-layer network paradigm, taking cost into consideration. This approach aims to balance predictive performance and resource utilization, making analyses more cost-effective. The proposed method can be applied in real-world medical settings to derive valuable insights regarding subgroups of patients/participants.

This thesis showcases the effectiveness of multi-layer networks in identifying patient subgroups with distinct prognostic or therapeutic responses through a series of case studies. The proposed methodology offers a comprehensive framework for stratifying patient populations based on complex interactions, facilitating precision medicine initiatives. Additionally, the interpretability of multi-layer networks allows for identifying actionable insights and potential biomarkers, enhancing the translational potential of research findings.

Overall, this thesis underscores the significance of multi-layer networks in unraveling the complexity inherent in medical data. By harnessing the power of multi-layer representations, researchers can gain deeper insights into the underlying mechanisms driving disease progression and treatment outcomes, ultimately paving the way for more effective and personalized healthcare interventions.

We introduce in this thesis two new techniques for identifying cost-aware subgroups in multi-layer networks for both static and temporal data. We test these methods on four different datasets with varying characteristics in terms of the number of time points and the scarcity of values. Additionally, we compare the performance of our methods with traditional clustering approaches that have been commonly used in research.

Zusammenfassung

Multi-layer networks haben sich als leistungsfähiges Instrument zur Darstellung komplexer Systeme in verschiedenen Bereichen, wie z. B. soziale Netzweke und biologische Systeme, erwiesen. Diese Netzwerke sind in der Lage, vielfältige Interaktionen und Beziehungen über mehrere Dimensionen hinweg zu erfassen, was sie zu einem idealen Rahmen für die Modellierung komplexer Systeme macht.

In dieser Arbeit wird die Nützlichkeit von mehrschichtigen Netzweke bei der Modellierung komplexer Systeme untersucht und ihre Vorteile bei der Erfassung subtiler und nuancierter Beziehungen hervorgehoben. Ein Bereich, in dem dieser Ansatz besonders nützlich ist, ist die medizinische Forschung, in der das Verständnis der Heterogenität von Patientenpopulationen entscheidend für personalisierte Behandlungsstrategien ist.

Die bisher verwendeten, herkömmlichen Clustering-Methoden sind nicht in der Lage Interaktionen von Merkmalen zu erfassen, was jedoch für die Entwicklung personalisierter Behandlungspläne nützlich sein kann. In dieser Arbeit hingegen werden mehrschichtige Netzwerke als neuartiger Ansatz zur Entdeckung von Untergruppen in der medizinischen Forschung vorgestellt. Durch die Integration verschiedener Datenmodalitäten wie genetische Profile, klinische Variablen und Umweltfaktoren in ein einheitliches mehrschichtiges System kann das komplexe Zusammenspiel dieser Faktoren aufgeklärt werden.

Darüber hinaus befasst sich die Arbeit mit der Komplexität von Patienten-/Patientinnendaten und mit der Frage der Kosten, die mit der Erfassung und Verarbeitung von Merkmalen verbunden sind. Herkömmliche Methoden sind meist so angelegt, dass sie alle verfügbaren Merkmale einbeziehen, was zu Ineffizienzen bei der Datenverarbeitung führen und relevante Muster verdecken kann. Um diese Herausforderung zu bewältigen, wird in dieser Arbeit ein Rahmen für die Merkmalsauswahl im Rahmen des Paradigmas des mehrschichtigen Netzwerkes unter Berücksichtigung der Kosten vorgeschlagen. Dieser Ansatz zielt darauf ab, ein Gleichgewicht zwischen der Vorhersageleistung und der Ressourcennutzung herzustellen, um die Analysen kosteneffizienter zu machen. Die vorgeschlagene Methode kann in realen medizinischen Umgebungen eingesetzt werden, um wertvolle Erkenntnisse über Untergruppen von verschiedener Populationen zu gewinnen.

In dieser Arbeit wird die Wirksamkeit von mehrschichtigen Netzwerken bei der Identifizierung von Untergruppen von Patienten/Patientinnen mit unterschiedlichen prognostischen oder therapeutischen Verläufen anhand einer Reihe von Fallstudien aufgezeigt. Die vorgeschlagene Methodik bietet einen umfassenden Rahmen für die Stratifizierung von Patientenpopulationen auf der Grundlage komplexer Interaktionen und erleichtert damit Initiativen der Präzisionsmedizin. Darüber hinaus ermöglicht die Interpretierbarkeit von mehrschichtigen Netzwerken die Identifizierung von umsetzbaren Erkenntnissen und potenziellen Biomarkern, was das translationale Potenzial von Forschungsergebnissen erhöht.

Insgesamt unterstreicht diese Arbeit die Bedeutung von mehrschichtigen Netzwerken bei der Entschlüsselung der Komplexität medizinischer Daten. Durch die Nutzung der Leistungsfähigkeit mehrschichtiger Darstellungen können Forschende

tiefere Einblicke in die zugrundeliegenden Mechanismen gewinnen, die Krankheitsverläufe und Behandlungsergebnisse vorantreiben, was letztlich den Weg für effektivere und personalisierte Gesundheitsmaßnahmen ebnet.

Diese Arbeit stellt zwei neue Verfahren zur Identifizierung kostenbewusster Untergruppen in mehrschichtigen Netzwerken für statische und zeitabhängige Daten vor. Diese Methoden werden an vier verschiedenen Datensätzen mit unterschiedlichen Merkmalen in Bezug auf die Anzahl der Zeitpunkte und die Knappheit der Werte getestet. Außerdem wird die Leistung dieser Methoden mit traditionellen Clustering-Ansätzen verglichen, die in der Forschung häufig verwendet werden.

Contents

List of Figures

List of Tables

List of Acronyms

Mini-TQ Mini-Tinnitus-Questionnaire

RQ Research Question

MLN Multilayer Network

MLNs Multilayer Networks

KNN K-Nearest Neighbors

CoNet Co-occurence Network Inference

EMA Ecological Momentary Assessment

COBALT Cost BAsed Layer SelecTor

MLF Maximum Likelihood Filter

Evol-COBALT Evolution-Cost BAsed Layer SelecTor

UNC Unknown Node Correspondence

KNC Known Node Correspondence

MRI Magnetic Resonance Imaging

TN Temporal Network

SN Snapshot Network

MLSN Multi-layer Snapshot Network

MLSNs Multi-layer Snapshot Networks

netLSD Network Laplacian Spectral Descriptor

RCT Randomized Clinical Trial

MDT-OS Mnemonic Discrimination Task for Objects and Scenes

ORR Object- In-Room Recall

CSR Complex Scene Recognition

MI-GRAAL Matching-based Integrative GRAph ALigner

SLN Single-layer Network

NMI Normalized Mutual Information

THI Tinnitus Handicap Inventory Questionnaire

TQ Tinnitus Questionnaire

TBF12 Tinnitus Impairment Questionnaire

TFI Tinnitus Functional Index Questionnaire

MDI Major Depression Inventory Questionnaire

CBT Cognitive Behavioral Therapy

FTQ Fear of Tinnitus Questionnaire

PHQ9 Patient Health Questionnaire

GNN Graph Neural Network

LSTMs Long Short-Term Memory artificial neural networks

MAE Mean Absolute Error

MSE Mean Squared Error

MAPE Mean Absolute Percentage Error

BMI Body Mass Index

1. Introduction

Precision medicine entails stratifying patients into more homogeneous groupings in order to customize treatment options to each patient's unique traits [40]. To collect data that characterizes patients, it is necessary to gather data from assessments such as medical tests, questionnaire data, and self-reported interview information. Some of these assessments have associated costs, such as psychological effort/patient burden, time required, and/or monetary expenses [74]. Hence, it is crucial to consider these expenses while extracting information from medical data.

Subgroup discovery is a powerful tool used for detecting patient heterogeneity by mining medical data. These subgroups can, in aftermath, be handled differently (e.g. customized treatment designs), depending on their characteristics. In medical research, the discovery of patient subgroups oftentimes requires high dimensional data mining. This problem can be translated into the discovery of subgroups of patients (or, more generally, entities) in high dimensional data, i.e. with a high number of features and time points.

Time is an important factor that needs to be considered when analyzing data. Temporal data is widely available in various applications, including medical research. This type of data provides insights that cannot be obtained through static data alone. If we have access to time series data, it is important to incorporate it into our analysis. For example, when performing subgroup discovery, which is a crucial part of this thesis, it is essential to take the temporal aspect into account.

Before presenting the problem statement, it is essential to define the problem tackled in this thesis - we aim to find subgroups in an unsupervised manner. This means that we focus on developing methods that deal with scenarios in which we have no ground truth available.

1.1. Problem Statement

While traditional unsupervised algorithms are able to separate entities into subgroups and take many features as inputs, they do not account for inter-feature relationship which might also be informative. This is one of the main reasons why Multilayer Networks (MLNs) can provide a more complete representation of entity data. The layers can represent different features of the data, a node can represent an entity with respect to a certain feature, and an edge can represent how similar the entities are.

Data representation is a crucial step to represent the system being analyzed. The definition of "what is a node, edge or layer" is highly context dependent.

In many applications, nodes are connected by a single edge, which represents the connection between them, resulting in a single-layer network. However, this approach oversimplifies the connections between nodes and does not take into account the complexity of the interactions between them [25]. To address this issue, a better approach is to introduce *layers*.

A layered network allows us to represent connections between nodes (edges) in many different perspectives, depending on the layer. Each layer represents a different

type of interaction between nodes, and the edges within a layer represent connections of the same type. By using multiple layers, we can model the complexity of the interactions more accurately.

For example, in a transportation network, we can have a layer representing the traffic between cities (nodes) and another layer representing the quantity of goods to transport between cities. The edges in the first layer might represent the number of cars, buses, or other transportation vehicles that travel between cities, while the edges in the second layer might represent the amount of goods transported between cities.

When representing a complex system, it is important to first define what the nodes represent and then decide which interactions between nodes should be represented. This will allow us to create layers that represent different perspectives on the connections between nodes, and model the complexity of the system more accurately. After, we must ensemble these concepts and build the MLN.

Two types of edges may exist: inter- and intra-layer edges. The inter-layer edges capture the relationship between features which is crucial afterwards for the detection of communities (or entities' subgroups) in the data. In comparison to traditional clustering algorithms (where there is no clear inter-feature distance), this representation provides additional information regarding the entity. In addition, MLNs are able to incorporate cost in a flexible way, i.e. cost can be modeled at feature and/or entity level. To be cost-effective, we must select only the informative layers. Some layers should be deleted from the MLN if they do not provide useful information for the subgroup discovery.

Our research is inspired by medical data to uncover subgroups and subsequently develop subgroup-specific treatments, while accounting for data acquisition cost.

The evaluation is conducted on medical and mHealth data. The approach is evaluated based on the quality of the partition of the entities into subpopulations as well as on how an outcome variable differs per subpopulation. The latter will focus on how much subpopulation specific data enhances the prediction of the outcome variable compared to a scenario in which all data are used.

In Chapter 2, we will argue in more detail why to use MLNs and not other methods in the literature.

1.2. Research Questions

This thesis tackles three Research Question (RQ)s that we hereafter present:

> **RQ1:** In which cases are MLN-discovered subgroups of higher quality than those discovered using traditional clustering techniques?

In the field of data analysis, clustering techniques are widely used to identify subgroups within a dataset in an unsupervised manner. However, when dealing with complex real-world problems, the traditional clustering techniques may not be sufficient in finding an appropriate representation of the underlying complex systems. In such cases, MLNs are considered to be a more suitable approach because they can represent complex systems more effectively.

In this research study, we aim to investigate and experiment with the transformation of a traditional clustering problem into a subgroup detection problem using MLNs. Our

goal is to determine if this approach is better than the conventional one. Specifically, we aim to explore how to transform a traditional clustering problem into an MLN problem and evaluate the effectiveness of this approach in detecting subgroups in complex systems. By doing so, we hope to contribute to the ongoing efforts to improve the accuracy of data analysis techniques and their applicability to real-world problems. For that, we include the following goals for this RQ:

- The development of a tool to represent real-world data into an MLN has to be proposed and is the first and essential step for the next steps of the approach
- A comparative evaluation has to be performed to compare the performance of MLN-discovered subgroups and traditional clustering algorithms for different datasets
- A methodology to compare networks, so that the same factors in different networks (e.g., age, gender) can be directly compared as in raw data

RQ2: How to model cost in MLN-based subgroup discovery?

In many applications, adding new features to a product or service can be a costly endeavor. As a result, we want to present an alternative method for incorporating new features that takes into account the cost component. To achieve this objective, we will first need to define cost, model it, and then incorporate it accordingly in the output of RQ1.

The cost of adding new features is often a significant factor that needs to be considered when making decisions about feature acquisition. Therefore, we will examine the cost component in greater detail and identify the different factors that contribute to it. This includes factors such as the cost of research and development, the cost of training employees to use or generate new features, and the cost of maintaining and upgrading the "product" over time.

Once we have a clear understanding of the cost component, we will then develop a model that can estimate the cost of adding new features. This model will take into account various factors, such as the complexity of the feature and the impact the feature will have on the product's overall performance.

Representing complexity in medical data and other applications is essential to unleash the full potential of the data. Medical datasets often contain intricate patterns/relationships between various factors. By incorporating complexity into the data representation, we can gain deeper insights into these interactions. This enhanced understanding leads to more accurate predictions and personalized treatment plans tailored to individual patients.

Furthermore, accounting for complexity helps to identify the most relevant features within the data. Transparent and interpretable representations of complexity ensure trust and understanding among healthcare professionals and patients. This, in turn, supports informed decision-making and improves patient outcomes while also being ethically sound.

Finally, we will incorporate the cost component into the output of RQ1. This will provide a more comprehensive understanding of the trade-offs involved in feature acquisition and help decision-makers make more informed decisions.

RQ3: How do MLN-discovered subgroups evolve with time?

In today's datasets, data is collected at multiple points in time, which makes the temporal dimension a constant. Consequently, there is a need for temporal modeling. To address this issue, we have identified three objectives for our research:

- We aim to develop a method that models time in MLNs. The temporal modeling of MLNs is crucial for capturing the temporal dependencies between events in time-series data.

- We aim to develop an approach to discover dynamic subgroups in MLNs. A subgroup is a subset of entities that share similar characteristics. Identifying subgroups that evolve over time can provide insights into the dynamics of the dataset.

- We aim to monitor the evolution of dynamic subgroups and propose visualization tools to interpret them. Visualization tools can help to understand the changes in subgroups over time and identify patterns that may be missed in the raw data.

We tackle the first objective, RQ1, in Chapters 5 and 6. In these chapters, we focus on designing all the necessary implementations for finding predictive MLN-discovered subgroups. In Chapter 7, we demonstrate the application of MLN-discovered subgroups to identify differences between samples.

RQ2 is approached inside of Chapters 5 and 6. RQ3 is tackled in Chapter 6.

1.3. Summary of Scientific Contributions

Our research delves into the use of multi-layer networks for subgroup discovery in the context of clinical and mHealth data. We conducted an extensive study to explore how data can be represented in a multi-layer network. This involved defining the network's components, such as nodes, edges, and layers, and understanding their relationships. By addressing RQ1, we were able to develop a robust method for representing medical data into an MLN.

In order to find cost-aware subgroups in multi-layer networks, we developed two methods that are related to RQ2. The first method involves assigning costs to the edges in a network, while the second method assigns costs to the layers. We conducted experiments using different datasets to test these methods and refine them further. We also evaluated the effectiveness of these methods in identifying cost-aware subgroups in a multi-layer network.

To answer Research Question 3, we included temporal dimensions and combined the methods proposed for Research Question 1 and Research Question 2. We developed a tool that discovers subgroups in multi-layer networks considering time and cost. This tool is capable of handling large and complex datasets, making it a valuable resource for researchers and practitioners.

Our contributions are:

- A method for representing medical data into a multi-layer network, which enables the detection of relationships between different layers and nodes.

- Two methods for finding cost-aware subgroups in multi-layer networks, which enable the identification of subgroups that are sensitive to cost.

- A method for finding cost-aware subgroups in multi-layer networks, which can be used to identify subgroups in situations where time is not a factor (static version).
- A tool that discovers subgroups in multi-layer networks considering time and cost, which is capable of handling large and complex datasets.
- A node assignment methodology that assigns nodes to subgroups without recomputing the whole network, which is computationally efficient and reduces the time required for subgroup discovery.
- A methodology for comparing samples from different populations through network analysis, which enables the identification of differences and similarities between populations.

Figure 1.1 illustrates a more detailed view of the scientific contributions of this thesis. There are six (6) different contributions in the approach proposed in this thesis to solve the problem of "detecting subgroups using MLNs."

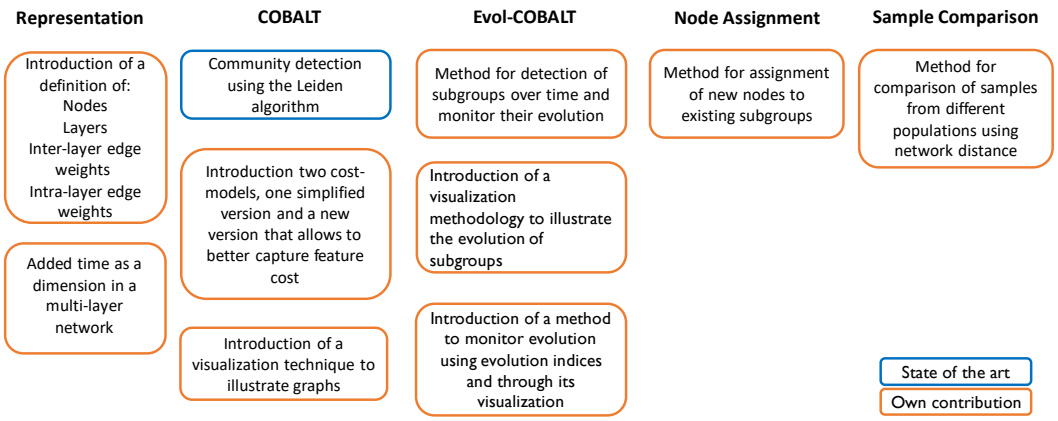


Figure 1.1.: Summary of scientific contributions. Each step is described as a column in the figure, with colored boxes below each column. The orange contoured boxes describe the contributions of this thesis, whereas the blue contoured boxes describe the part of the method that is composed by a method from the literature.

1.4. Outline of the Thesis

This thesis is composed by 7 chapters, starting with the Introduction and followed by:

- Chapter 2 (*Related Work*) introduces related literature to subgroup discovery in medical data, which is the closest to the topic of this thesis.

- Chapter 3 (*Underpinnings*) presents the basic concepts about subgroup discovery in medical data using MLNs and a comprehensive explanation about MLNs, community detection algorithms for subgroup discovery and quality metrics of subgroups in MLNs.
- Chapter 4 (*Materials*) presents all four datasets used in the experiments of this thesis. We describe each dataset in terms of its content and the variables of interest in that certain application. Each dataset has different properties, in terms of data dimensionality and sparsity.
- Chapter 5 (*COBALT for Static Data*) introduces the COBALT method for the detection of subgroups in MLNs. We propose a new methodology to find subgroups of individuals that are informative to predict a certain variable of interest. We start by proposing a method to represent data into a network. Then, we show two experiments with two different datasets.
- Chapter 6 (*Evol-COBALT for Temporal Data*) introduces a novel methodology to find subgroups in MLNs, extending the method proposed in Chapter 5 by incorporating the temporal aspect. Here, we extend the representation methodology proposed in the previous chapter by adding a new dimension to the networks - the time. We also propose a method for monitoring the evolution of subgroups with time in a qualitative and quantitative manner.
- Chapter 7 (*Cross-population Layer Matching*) proposes a workflow for identifying differences between two samples from different populations by first representing data into a network and then computing the distance between networks representing the different samples.
- Chapter 8 (Conclusion and Outlook) introduces a comprehensive conclusion that effectively summarizes the main findings and contributions of the research. In this conclusion, we not only highlight the successes and accomplishments of the project, but also address any limitations or areas for further improvement. Additionally, we outline possible avenues for future research and exploration, highlighting the potential impact and significance of this work.

2. Related Work

In this Section, related literature about "subgroup discovery" and its applications is presented, as well a summary of how different methods for subgroup discovery compare. This literature is important to position our work. Section 2.1 explains how subgroup discovery is used with medical data. Section 2.2 explains why building cost-aware subgroups is relevant. Section 2.3 explores how to model time in subgroup discovery. Section 2.4 introduces subspace clustering and positions it concerning the purpose of this thesis. The same is done but with the relevance of using networks to represent further subgroup discovery in Section 2.5.

More detailed explanations are added throughout the thesis when relevant, along with the proposed methodology.

2.1. Subgroup Discovery in Medical Data

According to [38], current methods for finding subgroups of patients focus on techniques such as K-means, hierarchical clustering analysis, and latent class analysis. For instance, in [28], the authors study diabetes disease progression and treatment response in subgroups detected using k-means. In [114], the authors extended Latent Dirichlet Allocation to the Poisson Dirichlet Model to find subgroups of patients.

In tinnitus research, phenotyping is a technique to analyze the heterogeneity of patients [38]. [81] work focuses on tinnitus subgroups, using X-means as well as visualization tools to present these subgroups. This work also shows how important subgroups are for the prediction of treatment outcomes.

However, studies on the comparison of K-means with the Louvain method (a community detection approach for networks by modularity optimization) have shown that Louvain methods produce better results when it comes to the identification of more accurate groups [113, 86]. A *community* can be described as a set of nodes that are grouped in a graph/network. In this thesis, we often call it "subgroup" or "subpopulation" when it is referred to subgroups of people. [86] has not only contrasted the Louvain method with K-means but also with a hierarchical clustering approach and concluded that the Louvain method has achieved higher accuracy.

In [115], the authors propose a method for identifying disease subtypes from electronic medical records using topic models. The approach represents patients as nodes in a graph and captures their interactions through topic modeling. By doing so, it uncovers hidden disease subgroups and their associated clinical features.

Some state of the art methods for subpopulation discovery focus on the use of a Graph Neural Network (GNN) [118] and on supervised methods such as Long Short-Term Memory artificial neural networks (LSTMs) as in [11, 123]. These methods, however, do not allow for the configuration of similarity based on the perspective being evaluated. Furthermore, LSTMs, which are used as supervised approaches in these works, are inapplicable to the situation presented in this thesis because no ground truth is available (in this thesis, it corresponds to the subgroup/community memberships). We explain this in more detail in subsection 2.5.

In order to develop a model that can be effectively utilized by medical researchers, it is necessary to establish criteria for identifying high-quality subgroups. Quantitative metrics such as silhouette coefficient for clustering and modularity[1] for community detection are commonly used, but not enough. In 2004, Lavrac et al. [68] conducted an assessment of subgroups in decision support systems, with a particular focus on those derived from medical data. Through their research, the authors proposed six key subjective metrics for evaluating subgroups: usefulness, actionability, operability, unexpectedness, novelty, and redundancy. These metrics can be applied to assess subgroups in any context, including medical data. Additionally, a quantitative approach can be used to measure the quality of subgroups.

In this Section, we have only summarized why and what in subgroup discovery, focusing on medical data. Next, we introduce why cost is important in subgroup discovery.

2.2. Building Cost-aware Subgroups

In [57, 71], the authors emphasize the importance of creating cost-effective models for clinical phenotyping. [71] specifically points out that the cost of acquiring data should be taken into account when building subgroups. Therefore, it is crucial to consider the cost factor while creating subgroups of entities, a topic that will be explored in research question 2 (RQ2).

Yu et al. [120] perform feature selection under a finite budget, in which the cost of feature acquisition is derived from suggestions by medical experts based on total financial expense, patient privacy, and patient inconvenience. Kachuee et al. [62] derive feature costs from the convenience of answering questions and performing medical examinations, such as blood and urine tests. Feature selection under budget constraints follows the principle that including irrelevant, costly features is worse than including irrelevant but inexpensive features.

When it comes to assessing and managing risks, cost-aware subgroup analysis can come in handy. This method helps identify high-risk subgroups within a population or portfolio while taking into account the cost implications of different risk factors or exposure levels. By doing so, organizations can develop targeted risk mitigation strategies, allocate resources judiciously, and minimize financial losses [22].

As a final note, when conducting cost-aware subgroup analysis, decision-makers must ensure equitable access to resources and opportunities, and consider how subgroup-specific interventions may impact vulnerable or disadvantaged populations [19].

The concept of feature cost through feature selection started to be introduced in the field of medicine [17], in which medical exams have an associated financial cost.

In the field of feature selection, there are three main types of methods: filter, embedded, and wrapper methods. The filter method is used in a pre-processing stage before the learning task. Embedded methods are used during the training of the learning algorithm. Wrapper methods use the learning algorithm to determine the best set of features that can maximize prediction quality.

The paper referenced as [91] explains several metrics that are frequently used to determine the cost of features. The authors specifically concentrate on the filter methods, which can be applied in unsupervised scenarios where no ground truth is available, unlike other methods. They compare different feature cost models

[1] Modularity is a quality metric used to evaluate how well-separated communities are.

to classify 4 Barabási–Albert [9] networks and distinguish between two protein interaction network models. For that, they use information theoretic approaches as mutual information, RelieF, and Random Forest based approaches.

In their works, [16] present a framework of metrics that define feature cost using correlation-based techniques like mutual information and Minimal-Redundancy-Maximal-Relevance. In this thesis, we focus on a similar approach, but this time applied to the similarity of layers that represent features.

In [72] investigate the effectiveness of using mutual information as a similarity metric for feature selection. The authors compare different feature selection methods based on mutual information and evaluate their ability to capture relevant features while minimizing redundancy.

In [94], the authors present a survey of feature selection techniques in bioinformatics, which includes methods that use similarity metrics. It explains how similarity-based approaches can be used to select informative features from high-dimensional biological data and how they have been applied in tasks such as gene expression analysis and biomarker discovery.

2.3. Time in Subgroup Discovery

[7] discusses the integration of time into subgroup discovery algorithms and its impact on the identification of temporal patterns in data. Clustering on subsets of features has been studied as "subspace clustering" [63], and temporal extensions have been recently proposed [84, 124]. In the domain of MLNs, temporal extensions are rare and confined to building communities over multivariate time series [101]. However, these methods are intended to group time series together rather than investigating to what extent the evolution of subgroups predicts some outcome in the pathways of the subgroup members. For subgroups in networks, there are several operations related to community evolution using dynamic community detection methods [93]. For instance, some communities may disappear and re-appear at another time point. Other examples are *"merges"* and *"splits"*. The former corresponds to when two or more communities are merged into one, and the latter to when one community is split into one or more [82, 23]. This property is defined as the community life cycle. An overview of these operations [82, 23] can be summarized as in Table 2.1. A common problem in dynamic community detection is the instability of the solutions. More specifically, when a community detection algorithm is executed in a certain network with minimal topological changes, it leads to great variation in the results (cf. [8]). This is because many algorithms are greedy and try to find a local optimum. With small modifications to the network structure (e.g., removal of a node), the local optimum found might differ. This event is caused by the instability of the algorithm and not by the evolution of the communities over time stamps [93].

In this thesis, we analyze "merges" and "splits," to monitor the evolution of entities with time.

2.4. Subgroup Discovery With Subspace Clustering

Subspace clustering, also known as clustering on subsets of features, has been a topic of study. Recently, there have been proposals for temporal extensions of this technique. These extensions have been proposed in papers such as [84, 124]. In the domain of multi-layer networks, temporal extensions are rarer and mostly confined to

Operation	Description
Birth	the first time a community appears
Death	when a community disappears
Growth	when a community increases in size of nodes
Contraction	when a community decreases in size of nodes
Merge	when two or more communities are merged into one
Split	when one community is split into one or more
Continue	when a community remains the same within a time-frame
Resurgence	when a community that had disappeared, resurges

Table 2.1.: Community operations in temporal networks

building communities over multivariate time series [101]. Evolution-based clustering algorithm of high-dimensional data streams (SE-Stream) [1] target dimension selection in streams. In [1], the authors perform a systematic review of subspace clustering in streams. They conclude that these types of methods can identify clusters within different subspaces of a dataset and are designed to handle dimensionality reduction.

Creating clusters using different subspaces can be a difficult and costly task. This is because the process of clustering involves grouping similar data points together based on their similarities. However, when multiple subspaces are involved, the process becomes more complex and may yield less meaningful outcomes. Additionally, comparing clusters created using different feature spaces at the same time may not provide useful insights, as the dissimilarities in the subspaces can lead to ambiguous or irrelevant results. Therefore, it is important to carefully choose the appropriate subspaces and feature spaces when clustering data to ensure that the resulting clusters are relevant, interpretable, and useful for analyzing the data.

The paper [50] introduces a subspace clustering approach based on two constraints - must link (ML) and not link (NL). The must link constraint is used to group instances that must belong to the same cluster, while the not link constraint is used to prevent instances that should not belong to the same cluster from being grouped together. These constraints can be derived from domain knowledge.

The article referenced as [30] delves into the difficulties of interpretability that arise when using subspace clustering for subgroup discovery. The authors highlight the challenge of interpreting clustering algorithm results, especially in spaces with high dimensions, and stress the significance of creating interpretable descriptions for subgroups.

In [31], the authors discuss the limitations of traditional subspace clustering methods in dealing with sparse data. It highlights the need for robust clustering algorithms that can effectively handle datasets with sparse and noisy features.

Sparse data can be effectively represented by MLNs without the limitations of subspace clustering. Moreover, the similarity between entities typically depends on the method used, but MLNs provide the flexibility to employ various similarity functions. We can even use different functions for different features. This is because representing data through an MLN allows us to define all aspects of the analyzed

system, providing greater flexibility.

In real-world applications, the features that characterize subgroups may vary over time and may have different costs at different time points; therefore, we need a technique that is flexible enough to consider that. Our proposed method involves representing data using an MLSN. This allows us to capture the complexity of the system being studied. As far as we know, subspace clustering has not yet been used to address the problem of using MLSNs instead of raw data. In subspace clustering, though, the data is not represented in a network, which is something fundamental that we propose in this thesis as a means to better represent complex systems.

2.5. Subgroup Discovery With Networks

MLNs - a group of interdependent networks in which each network is represented as a layer - enable the representation of the entity data in a network. Nodes and edges are properties of a network. Nodes can be in at least one layer, and if in more than one layer, the node will assume different properties in the different layers [87]. MLNs present a possible tool to uncover the intrinsic complexity of medical data. They have been used for finding phenotypes of patients [85, 64].

[69] review methods that use MLNs for the representation of biological systems' hierarchy. This work focuses on the representation of subgroup and genotype data with MLNs, but not the detection of the subgroups. [66] use two modularity-based community detection algorithms (Louvain and Leiden) to find subgroups representing the data using K-Nearest Neighbors (KNN) and Co-occurence Network Inference (CoNet). We are concentrating on community detection.

In their paper titled "Constraint-based pattern set mining framework" [46], the authors present a method for discovering subgroups from graph data. The framework is extended to subgroup discovery by applying constraints on graph-based patterns. The authors demonstrate the flexibility and scalability of the method in handling various types of graph data, making it a versatile approach to subgroup discovery. In [46] the authors introduce a constraint-based pattern set mining framework, extendable to subgroup discovery from graph data by imposing constraints on graph-based patterns. The method demonstrates flexibility and scalability in handling diverse graph data types, offering a versatile approach to subgroup discovery.

In [111], the authors have suggested a method to discover uncommon substructures in graphs, which can be used for subgroup discovery tasks. The approach uses a graph-based scoring scheme and representation to identify subgroups that deviate significantly from the norm. This helps in identifying new and fascinating patterns.

GNNs are able to process graph structures [2] and learn from them [96] mainly for supervised and semi-supervised tasks.

[44] have recently proposed a generalization of a GNN that take an MLN as input and evaluate it on a supervised learning task. In our work, we build a MLN and derive communities on it, i.e. we have an unsupervised learning task. Closer to our work is the study of [119], who propose adapting a GNN to the *unsupervised* task of discovering clusters/communities, for a given number of clusters. This seems the closest in terms of relevance for our problem setting. However, the method of [119] does not take a MLN as input; unsupervised learning with an GNN that takes an MLN as input is still an open problem. Moreover, our goal is to build an MLN and

[2]The internal architecture built up by a GNN from the input graph can be described as an MLN, but of relevance to our work are rather studies that taken a multi-layer graph as input.

derive communities on it in a *cost-aware way* rather than to deliver an MLN as input to an GNN. Therefore, we incorporate the Leiden method into our cost-aware layer selection algorithm for community learning, rather than incorporating a cost-aware layer selector mechanism into an unsupervised GNN that takes an MLN as input.

Network analysis is an important tool that handles complex systems and, therefore, data [93]. Recently, [33] studied whether graph neural network-based models of electronic health records can predict specialty consultation care needs, finding that these types of representation into graphs can improve the predictiveness of care needs.

In general, graph applications are categorized as either node-focused or graph-focused. The former is concerned with node classification or regression problems relating to node properties, whereas the latter is concerned with learning the graph structure. This indicates that applications developed with the purpose of learning node-properties belong under the “node-focused" group. When the goal is to learn graph-focused attributes, it falls under the category of “graph-focused" applications.

In [96], the authors propose a supervised neural network model that is suitable for both node-focused and graph-focused applications. In terms of approximation capability, graph neural networks have been demonstrated to be universal approximators on graphs [95]. Before that, multilayer feedforward networks have been also proven capable of approximating any measurable function [53]. Alternative techniques, such as semi-supervised learning tasks with graph neural networks, have also been proposed [65].

In our application, we have an unsupervised task (finding subgroups without ground truth) and aim to model entity similarity across all variables and across and between every pair of features. More explicitly, we aim to model the similarity between an entity in feature X and feature Y, but also their similarity in feature X and Z. Inter-layer edges can model these similarities between every pair of layers (features in our application).

In [119], the authors utilize an unsupervised GNN derived from a semi-supervised GNN. In their method, they require to set a variable as “larger than the number of the ground truth number of clusters". This is information that is not always available in some applications. Hence, graph neural network methods, including [119], cannot transfer trivially. In particular, they would need to take an MLN as input, with some ground truth about the real subgroups, which is unavailable in our application. We also aim to provide information on which features are the most important. These impositions are inadequate for our application, although they may be for many others. Hence, for this work, GNNs are not the most appropriate.

The topology of an MLN, however, differs from that of a graph neural network, which has an ordered set of layers (input, hidden, and output layers [96]). MLN's layers can be ordered or unordered, and the layers of an MLN can connect not only the following layers, as in a graph neural network structure but every pair of layers. A graph neural network is an MLN, yet not every MLN is a graph neural network. As a result, all graph neural network methods, including [119], cannot be trivially transferred. In particular, they would need to take an MLN as input.

3. Underpinnings and Advances on MLNs for Community Construction

This chapter describes the foundations necessary to comprehend MLNs and how to find subgroups in those structures through community detection algorithms. We also explain how time and evolution can be modeled in these structures.

3.1. Networks with One or Multiple Layers

3.1.1. Definition of a Network

A network has a certain topology, which may be rooted or not, directed or undirected. In addition, quantitative information might exist about edges and nodes in the network [2]. Figure 3.1 illustrates an example of a network. Note that in the figure, some edges are thicker than others. They denote *weighted* edges. Plus, note that there is no direction in the edges, which means that the edges are *undirected*. Another important fact is that a node can be isolated - not connected by edges. Plus, edges connect nodes but might only connect some of them.

More formally, let N be the set of nodes n_i with $i = 1, 2, ...N$ and $N \in \mathbb{N}$. Each node can be associated with a state, represented by a canonical vector in $\mathbb{R}^N$: $\boldsymbol{e_i} \equiv (0, ..., 1, ..., 0)^\dagger$ for node n_i and $\boldsymbol{e_j} \equiv (0, ..., 1, ..., 0)^\dagger$ for n_j, with $\dagger$ being the transposition operator. The i^{th} and j^{th} components of $\boldsymbol{e_i}$ and $\boldsymbol{e_j}$ are 1 and the others 0, respectively [25].

3.1.2. Multi-layer Networks

An MLN can be defined as multiple single-layer networks. Each single-layer network can be considered a layer of a bigger structure, which forms an MLN. Each layer can then be connected to other layer(s), by edges. In this case, there are two types of edges:

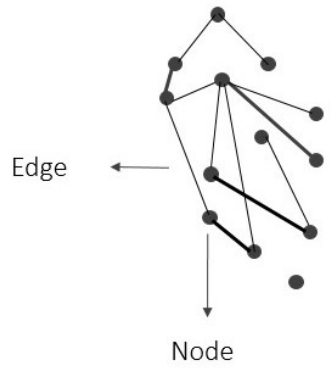


Figure 3.1.: Example of a single-layer network.

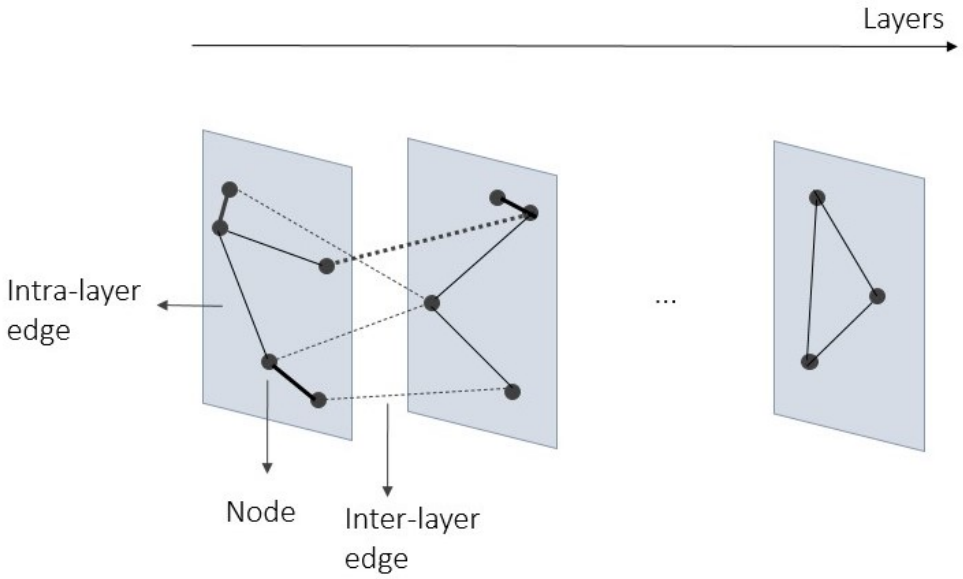


Figure 3.2.: Example of a multi-layer network.

- intra-layer edges: edges that connect nodes inside a layer
- inter-layer edges: edges that connect two nodes from different layers

which can be weighted or not, directed or not.

Figure 3.2 illustrates an MLN, with both intra- and inter-layer undirected edges.

Let $\mathcal{L}$ be a set of layers with L layers, where $\tilde{k} \in \mathcal{L}$. Two nodes n_i and n_j can be connected in the same layer $\tilde{k}$ (by an intra-layer edge), but they can also be connected in two different layers, for example $\tilde{k}$ and $\tilde{h}$. The latter case corresponds to inter-layer connections, or edges. To represent a MLN, we follow the same notation as in [25]. Since we only have one type of relationship between nodes and we model time, we slightly change the formulation of the multilayer adjacency tensor in [25] to incorporate time to the one described in Equation 3.1. Let M_t be the multilayer adjacency tensor at time point t.

$$M_t = \sum_{\tilde{h},\tilde{k}=1}^{\mathcal{L}} \sum_{i,j=1}^{N} w_{ij}(\tilde{h}\tilde{k})(t)\boldsymbol{E_{ij}}(\tilde{h}\tilde{k})(t) \tag{3.1}$$

The main goal is to find the sets of nodes $\mathcal{S}_t = S_t^1, S_t^2, ..., S_t^{NC}$ in M_t, at each time point $t \in t_0, t_1, ..., t_{NT}$, where NT is the number of time points and NC is the number of subgroups, that comprise nodes that are similar to one another but distinct to nodes in other groups. These are named subgroups or communities.

sectionDefinition of Community The term *community* started to be used in the context of social networks to name a subgroup of nodes in the network. In network science, this term describes a subgroup of nodes in a network.

Commonly, as in traditional clustering, a community can overlap with others when a node belongs to more than one community. Depending on the approach, it might

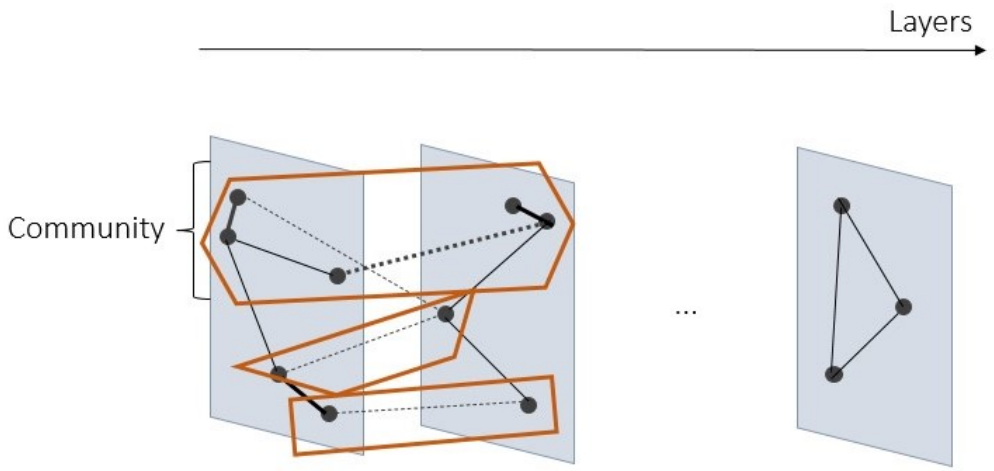


Figure 3.3.: Example of an MLN with inter-layer communities. The brown lines surrounding the nodes indicate different communities that group nodes across different layers.

be useful to consider overlapping communities. For other applications, it can be unfeasible.

Figure 3.3 illustrates communities discovered in an MLN. This is different from single-layer networks, since all layers contribute to the decision of the community membership of a node.

The detection of communities can be done using many methods, but the most commonly used is simply finding the partition that maximizes modularity or another quality metric. The partitions are found in different ways, depending on the method. Next, we explore that.

3.2. Community Detection in Multi-layer Networks

3.2.1. Notion of a Community

Many community detection algorithms perform well in single-layer networks, but are difficult to adapt to MLNs [47]. According to [47], there are three key causes that explain this complexity: (i) relevant communities may be masked by irrelevant edges, (ii) it is not possible to capture the different relations of the nodes in each layer and (iii) it is not possible to detect multiplex communities since layers are not specified. For context, multiplex communities are communities that are detected across multiple layers in a multiplex network[1].

In [54], the authors organize community detection algorithms for MLNs into three groups:

- aggregation

[1] a multilayer network that only has inter-layer edges that connect nodes that represent the same entity.

- flattening
- direct

For weighted MLNs, [54] highlights the following published studies with different approaches for the detection of communities: the variational bayes (aggregation method), Louvain (direct method), Aggregation Pan (flattening method), the ParticleGao (flattening method) and the MNLPA (direct method) algorithms. [55] also proposes an algorithm to improve the detectability of communities, which is considered to be relevant to the current work.

The Louvain is a benchmark algorithm that focuses on modularity optimization. Modularity is a quality metric to measure the quality of the partition of the nodes of a network into groups (or communities, in our application). We introduce more formally the definition of modularity later in this chapter, in section 3.4.

3.2.2. From Louvain to Leiden

A widely used algorithm for is the GenLouvain [73], also mentioned by [54] as a benchmark algorithm. It evaluates the variation of modularity using the same logic as the Louvain method, but it is able to be used with the multislice modularity [73]. [26] also introduces a generalization of the Louvain method where it is used the Weighted Edge Random Walk k-Path Centrality to compute the k-path edge centrality. After, the pairwise connectivity of nodes is computed by using the k-path edge centrality. With this pairwise connectivity as edge weights, the partition of the network is done via the Louvain method. However, it is not stated its application to the MLNs.

Even though the Louvain approach is one of the most popular among community detection algorithms, it can often lead to groups that are poorly connected. Thus, [109] propose the Leiden approach as an improvement to the Louvain method. The Leiden approach introduces a refinement phase on the partitions found. I.e., while the Louvain method assigns the node that most improves the quality function to a community, the Leiden method opts for a random approach. The node will be assigned to a community if the quality function improves, but not necessarily if this results in the largest improvement of the quality function. At the same time, the likelihood of a node being assigned to that community will be higher as higher the improvement on the quality function.

In summary, there are six steps of the Leiden method:

1. Assign a different community to each node, i.e. every node will be assigned a different community (Figure 3.4)

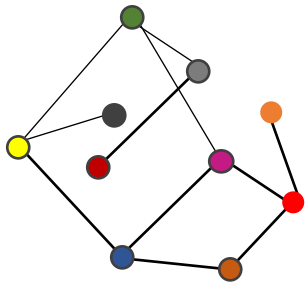


Figure 3.4.: First step Leiden

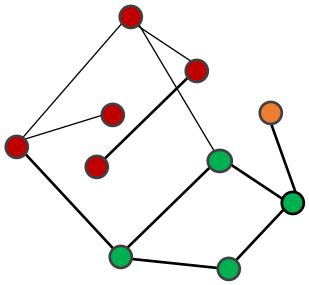


Figure 3.5.: Second step Leiden

2. Moving nodes from one community to other community (Figure 3.5)
3. Refinement phase (Figure 3.6)

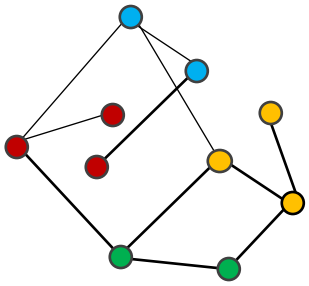


Figure 3.6.: Refinement step Leiden

4. Represent nodes assigned to a community in one single node (Figure 3.7)

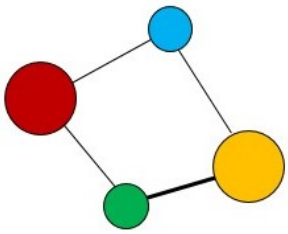


Figure 3.7.: Aggregation step

5. Move nodes (Figure 3.8)

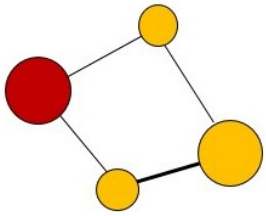


Figure 3.8.: Moving of nodes

6. Refinement

3.2.3. Further Advances on Community Detection in Multi-layer Networks

In the following section, we will provide a comprehensive overview of various community detection algorithms that have been employed in the literature. These algorithms have been designed to identify communities or clusters within a network or graph. However, it is important to note that not all algorithms are suitable for every application. Some algorithms have limitations that may make them unsuitable for certain types of networks or data. Therefore, it is crucial to carefully evaluate the properties of the network and the suitability of the algorithm before deciding to use it. In the subsequent paragraphs, we will provide a detailed description of each algorithm, including its strengths, weaknesses, and potential applications.

Variational Bayes with Stochastic Block Model

This approach proposes a method to identify shared and unshared communities in a multiplex network. In a multiplex network, inter-layer edges can only connect nodes that represent the same entity. The nodes are represented in a community-wise connectivity matrix that maps the probability of nodes being connected to other nodes. Nodes are placed in the same block if their edges are stochastically similar. The Poisson distribution is used to calculate the probability of nodes being connected.

The optimal number of blocks that represent the nodes is determined using the Bayes factor [3]. Once the optimal number of blocks is found, communities can be extracted. [3] has formulated the Weighted Stochastic Block Model (WSBM) adapted to a multiplex network. However, it is not clear if the results outperform a benchmark method, such as the Louvain.This proposed approach aims to provide a robust method for identifying both shared and unshared communities within a multiplex network. In a multiplex network, inter-layer edges are only allowed to connect nodes that represent the same entity. To represent the nodes in a community-wise connectivity matrix, a probabilistic approach is adopted that maps the probability of nodes being connected to other nodes.

The community-wise connectivity matrix is used to place nodes in the same block if their edges are stochastically similar. The probability of nodes being connected is calculated using the Poisson distribution. The optimal number of blocks that represent the nodes is determined using the Bayes factor [3]. Once the optimal number of blocks is identified, communities can be extracted.

To achieve this, [3] has formulated the Weighted Stochastic Block Model (WSBM) that is well-suited to a multiplex network. However, it is currently unclear whether the results outperform benchmark methods, such as the Louvain method. Overall, this approach provides a promising way to identify shared and unshared communities within a multiplex network.

AggregationPan

This algorithm aggregates edge weight matrices and then applies a cut-off, so that the edge weights with values below a threshold ($< \tau$) are converted to 0 [83]. This is an aggregation method, which is criticized in the literature since its simplification may mask the true nature of the initial modular patterns [54]. Plus, there is the need to define the threshold τ.

Particle Competition

This technique is based on a particle competition algorithm for multi-layer networks [37]. The main idea is to insert a certain number of particles K into nodes of the network. Each of those particles goal is to dominate as many nodes as possible as well as defending their current dominated nodes from the other particles. When a particle visits a node, it gets stronger and it weakens the other particles in the same node. At the end of the algorithm, it is expected that each particle represents a community. There are two types of moves that the particles may do: random walking or preferential walking. Random walking arbitrarily selects a node from its neighbors, and preferential walking consists in visiting a neighbor in which its dominance is high. These two types of walking are then balanced by a balancing parameter. This means that the likelihood of a node taking a random or a preferential walk will depend on this parameter. The experiments were, however, not tested in real-world networks in their published work.

MNLPA

This algorithm was introduced by [4], and it is a generalization of the label propagation algorithm to the multiplex networks. In the beginning, each node is assigned a label. Then, similarity measures are used to compare nodes. If these two nodes have a certain similarity metric higher than a given threshold, then the label of the two nodes is replaced by a common label for both. This is done until the stopping criteria are achieved. However, the survey from [54] states that this method is better suited for directed and weighted networks rather than for general MLNs. It is also mentioned that it might be unstable due to the threshold parameter defined, which strongly impacts the network partition.

Belief Propagation

This algorithm focuses on the detectability of the communities by using the Belief Propagation approach in SBM (Stochastic Block Model), which is able to deal with heterogeneous structures of communities across layers (cf. [55]). Constraints are treated as Bayesian priors. One of the reasons that lead to improved detectability is related to the labeling of the communities. They ensure that communities that represent the same nodes across layers have the same label, but the others have distinct labels. Then, the belief of each node belonging to a certain community is computed using prior knowledge about the correlation of community assignment between nodes. Then, the accuracy of the model is evaluated with the ground truth community assignment. [117] also uses the Belief Propagation method to optimize modularity, showing that an approximation of the uncertainty of the assignment at each node can be obtained from its marginal. Marginals reflect, in their work, how much modularity is impacted when a node moves from one community to another and thus test how much the node prefers a certain community.

3.3. Advances on Inter-layer Similarity

As previously mentioned, the overarching goal of this research is to develop a method that finds subgroups in a cost-effective manner, which is recognized as significant in the field (cf. Section 2.1). The cost of acquiring features is an essential cost to incorporate in such a model. Taking into consideration that features are represented

by layers in an MLN, highly similar layers are less useful for detecting subgroups than those that are very different. The main reasoning behind this statement is that considerably different layers/features provide complementary information about the nodes/patients. In contrast, layers with high similarity are deemed as redundant and, therefore, less useful.

By determining the most relevant layers, one part of the cost-effectiveness component will be introduced to the subgroup discovery model. This agrees with the goal of creating a method that detects patient subgroups with the minimum number of layers without compromising its quality.

Among the most common metrics to compute layer similarity are the Jaccard coefficient, the Normalized Mutual Information (NMI), and the adjusted Rand index. The Jaccard coefficient, introduced in [59], is often used in classification problems to compare the classifier's quality. However, it can also compare two vectors and compute their similarity.

The NMI score is often used to compute the quality of clusters and is a variation of the mutual information metric [5]. In [61], however, the authors show that this is a biased score to use for community detection and classification applications.

The adjusted Rand index measures the similarity between two data clusterings. It is a correction of the Rand Index but has the disadvantage of being sensitive to chance. The Adjusted Rand Index considers the possibility of chance agreement between two clusterings and adjusts the Rand Index accordingly [56, 106].

For the specific case of layer comparison, [20] present methods to compare distributions between layers in a multiplex network:

- Dissimilarity index
- Kullback-Leibler divergence
- Jensen-Shannon divergence
- Jeffrey distance

[108] present techniques to compare networks, both for a scenario when the nodes are aligned and the pairs are known, Known Node Correspondence (KNC) and when the networks have nodes that are not aligned and therefore different, Unknown Node Correspondence (UNC). For undirected and weighted networks, some methods are presented in Table 3.1.

Type of network	**Techniques**
KNC	Euclidean, Manhattan, Canberra distances
	Weighted Jaccard distance
UNC	Global statistics
	Spectral adjacency, Laplacian, SNL distances
	Matching-based Integrative GRAph ALigner (MI-GRAAL)
	netLSD
	Portrait divergence

Table 3.1.: Network similarity techniques for KNC and UNC

For UNC, it is stated that global statistics are not a robust method to compare layer similarity since they are too simplistic. Spectral methods also present some drawbacks concerning the co-spectrality between different graphs, dependence on the

matrix representation, and abnormal sensitivity. MI-GRAAL has a high computational cost, and the portrait divergence is mostly efficient with small to medium graphs [108].

The netLSD is based on a spectral representation of the graph. The properties of the graph are represented by the solution to the heat function using the normalized Laplacian matrix. A crucial property of the netLSD method for this work is the "size-invariance" property. This means that networks with different sizes can be compared without compromising the interpretability of the metric. The "size-invariance" property is especially helpful in the medical context; for example, when we want to compare different clinical centers and the number of patients at different clinical facilities varies greatly.

All of the above metrics focus on describing the similarity between networks based on their edges, nodes, and/or other properties. For the current work, we are interested in comparing a specific property - the community structures among layers.

Recently, [39] identified some drawbacks when using extrinsic (that require ground truth) evaluation metrics, such as the direction of the comparison. More precisely, if we have 2 layers of an MLN, l_α and l_β, there are two ways to match the communities in each one of them. One can use either l_α or l_β as the basis of the comparison, and this may lead to different results. For instance, let $C^{l_\alpha} = \{c_1, c_2, ..., c_r\}$ be the set of communities in l_α and $C^{l_\beta} = \{c_1, c_2, ..., c_k\}$ be the set of communities in l_β. Let $n^{l_\alpha}_{C_1} = \{n_1, n_2, ..., n_m\}$ be the set of nodes in c_1 of layer l_α and $n^{l_\beta}_{C_1} = \{n_1, n_2, ..., n_n\}$ the set of nodes in c_1 of layer l_β. In these communities, if we consider l_α as the basis of the comparison (the ground truth) and l_β as the layer to be compared with the group truth. If we look at node existence, we will check for the number of nodes from $n^{l_\alpha}_{C_1}$ that is in $n^{l_\beta}_{C_1}$. In the case that $n^{l_\alpha}_{C_1}$ has 15 nodes and $n^{l_\beta}_{C_1}$ has 30, from which 15 are the same as in l_α, then the similarity metric gives a perfect similarity. Therefore, a two-way matching is required, in which each layer is considered the ground truth - one at a time - and then both similarity values are combined into a single metric. Since purity and F-measure are among the most commonly used metrics for clustering evaluation, and they allow the comparison between two clustering solutions, they fit the purpose of the current work.

[39] convert the purity in a two-way matching by computing firstly the purity of l_α against l_β, $purity^{[l_\alpha \| l_\beta]}$, and vice-versa, $purity^{[l_\beta \| l_\alpha]}$. The harmonic mean between both values is computed, and the final purity is as follows:

$$purity = \frac{2 * purity^{[l_\alpha \| l_\beta]} * purity^{[l_\beta \| l_\alpha]}}{purity^{[l_\alpha \| l_\beta]} + purity^{[l_\beta \| l_\alpha]}} \tag{3.2}$$

With this metric, we can investigate how "pure" the clustering is in terms of node presence.

The recall and precision of the clusters are used to compute the F-measure. A cluster's precision is the same as its purity. The recall metric calculates the percentage of common nodes between the ground truth and system-generated clusters.

Finally, a cluster's F-measure is the harmonic mean of its precision and recall. Considering $F^{[l_\alpha \| l_\beta]}$ as the F-measure when l_α is the ground truth and l_β as the system-generated clustering solution and $F^{[l_\beta \| l_\alpha]}$ as the F-measure of the clustering solution when l_β is the ground-truth and l_α as the system-generated clustering solution. The harmonic mean of both values gives the overall F-measure:

$$F = \frac{2 * F^{[l_\alpha \| l_\beta]} * F^{[l_\beta \| l_\alpha]}}{F^{[l_\alpha \| l_\beta]} + F^{[l_\beta \| l_\alpha]}} \tag{3.3}$$

3.4. Evaluation of Community Detection Algorithms

There are distinct ways to measure the quality of a community. In [105], the authors mention some metrics usually used for the validation/evaluation of communities. Even though the most widely used metric is modularity [79], the author also mentions accuracy, Rand index, adjusted Rand index, and normalized mutual information. However, these metrics demand a ground truth, except modularity.

The GenLouvain algorithm is a modularity optimization algorithm widely used and considered a benchmark algorithm [54].

Modularity is a metric that evaluates the quality of the network partition into communities. The multislice modularity is a metric adapted to MLNs introduced in [77]. This metric can be defined as follows:

$$Q_M = \sum_{ij\alpha\beta} \frac{(A_{ij\alpha} - \gamma \frac{k_{i\alpha}k_{j\alpha}}{2m\alpha})\delta_{\alpha\beta} + \delta_{ij}C_{j\alpha\beta}}{2\mu} \delta(g_{ij}, g_{j\beta}) \tag{3.4}$$

where μ is the number of edges in the MLN, γ is a resolution parameter, $A_{ij\alpha}$ is the value of the edges between i and j in the layer α, $C_{j\alpha\beta}$ is an inter-layer edge between the same node j that belongs to layer α and β and $k_{i\alpha}$ represents the degree of node i in the layer α.

The primary difference between the GenLouvain algorithm and the conventional Louvain algorithm is that the modularity measure has been replaced by the multislice modularity metric [73]. Despite being one of the benchmark algorithms for community detection based on modularity optimization, Louvain can result in suboptimal partitions, as demonstrated by [109]. To address this, the authors propose the Leiden algorithm as an enhancement to the Louvain algorithm. They include a refinement step in which the nodes to be relocated to another community do not have to deliver the greatest improvement in the quality function. This is a noteworthy difference from Louvain, where nodes are only assigned to another community if it results in the greatest increase in the quality function. This is a significant difference from Louvain, where the strategy is greedy, with nodes only being assigned to another community if it leads to the highest increase in the quality function. Nonetheless, there is a likelihood linked with the decision on whether nodes should be relocated to other communities in the Leiden algorithm. The greater the increase in the quality function caused by shifting a node, the more likely that node will be chosen.

It is crucial to highlight that while modularity assesses the quality of node partitioning into communities, this does not imply that the communities reflect the ground truth entirely, as cited by [105].

Conductance [103, 102] is another metric widely used to evaluate cluster quality in graphs and looks at the edges between the nodes of a cluster and the nodes of other clusters. Considering a cluster $\mathcal{S}$ of a graph G, the conductance $\emptyset(S)$ of that cluster is:

$$\phi(S) = \frac{|\delta(S)|}{\min(vol(S), 2m - vol(S))} \tag{3.5}$$

where m is the number of undirected edges, $vol(\mathcal{S})$ is the volume of edges in S, $vol(S) = \sum_{v \in S} w_v$, w_v denotes the weight of edge v. The boundary of S is $\delta(S) \in \{(i,j) \in E | i \in S \texttt{ and } j \notin S\}$.

Aside from modularity, domain-specific applications necessitate an in-depth assessment of the evaluation strategy, for example, by including domain-specific evaluation procedures.

3.5. The Temporal Aspect in Multi-layer Networks

3.5.1. Representation of Time in Multi-layer Networks

Rossetti and Cazabet present two distinct ways of modeling time for community discovery [93]. In a *temporal network* (TN), time is one of the dimensions of a community, while in a *snapshot network* (SN) time is expressed as a sequence of snapshots – communities are built at each snapshot and linked together. Snapshot networks, also referred to as static networks or cross-sectional networks, are visual representations of the relationships or interactions among different entities at a particular point in time. These networks are static and do not capture the temporal dynamics of interactions or changes in the network structure over time. Nodes in snapshot networks typically represent different entities such as individuals, organizations, or objects, while edges represent relationships or interactions between them [10].

More formally, a *snapshot network* G_T can be expressed as an ordered set of snapshots $G_T = (G_{t_0}, G_{t_1}, ..., G_{t_T})$, with $G_t = (V_t, E_t)$ where V_i is the set of nodes and E_i is the set of edges from G_t. SNs agrees with our notion of subgroup evolution monitoring, so we use SNs as basis of our approach. For our task of community evolution monitoring, we use SNs. We can model time in networks for community discovery by using a Temporal Network (TN) or a Snapshot Network (SN) [93]. TNs and SNs are intended for different problems though:

A temporal network can be expressed as a graph $G = (V, E, T)$, V denotes the set of vertices between time of birth and time of death of v. E is the set of edges over time, $E = (n_i, n_j, t_s, t_e)$ with $n_i, n_j \in V$ and in this case, t_s, t_e denote the time of birth and death of the edge connecting n_i to n_j, respectively. We can categorize temporal networks like a normal graph as directed or undirected, depending on the nature of the edges.

In certain contexts, we might have aggregated data in the form of daily or monthly measurements. In this case, snapshot networks are the most common representation.

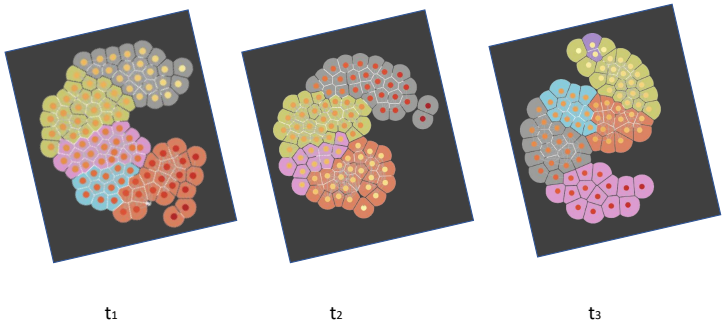
Since SNs are sets of single networks and thus represent an aggregated set of networks, traditional single-network analysis methods can be applied, thus resulting in a lower complexity compared to TNs. Moreover, the The complexity of SNs is determined by the number of time points [93].

Subfigure 3.9a shows an example of a snapshot network (without edges between time points), and Figure 3.9b shows an example of a temporal network, where time-stamped data is connected by inter-layer edges. In these figures edges represent the similarity between nodes.

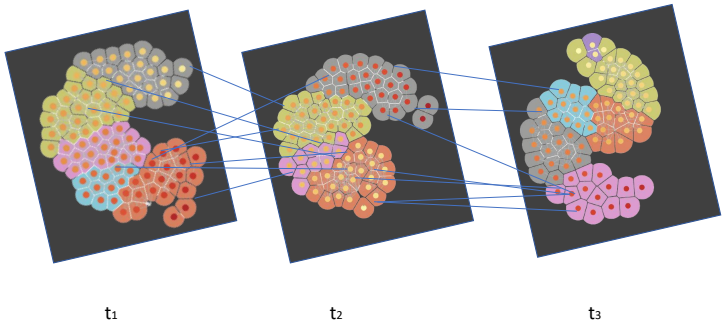
The selection criteria to choose between TNs or SNs depend on the data representation and complexity [93]:

- Representation: when the data is aggregated (e.g. monthly/yearly), SNs are the most commonly used. If there is non-aggregated time point data, then both SNs and TNs can be used.
- Complexity: the complexity of SNs is determined by the number of time points. To identify structures like communities, a community detection algorithm must be run at each point in time, and the results must be consolidated in some way. The number of network modifications that occur over time dictates the complexity of community detection approaches in TNs.

It is important to note that snapshot networks only provide a snapshot of the



(a) Snapshot network



(b) Temporal network

Figure 3.9.: Snapshot and temporal network. Each time point (t_1, t_2, t_3) in the figure represents a single-layer network at that point in time. Colored dots represent nodes and the colored surrounding circles around them denote the subgroup of that node. Each color represents a subgroup. The temporal network illustrated in subfigure 3.9b has edges across time points, which connect pairs of nodes from different points in time.

network structure at a specific moment in time and do not show how this structure changes over time. Therefore, they cannot capture the evolution or history of interactions between entities. Snapshot networks have numerous applications in various domains, including social network analysis, biological networks, transportation networks, and communication networks. They are particularly useful in analyzing the network structure, identifying communities, measuring centrality, and studying diffusion processes [116, 10].

In social network analysis, snapshot networks are used to study the structure of social groups, identify key players, and analyze social dynamics. In biological networks, they are used to study protein-protein interactions, gene regulatory networks, and metabolic pathways. In transportation networks, they are used to analyze traffic flow and identify congestion points. In communication networks, they are used to analyze email communication patterns, social media interactions, and phone call networks. Overall, snapshot networks are an important tool for understanding the structure and dynamics of complex networks at a specific moment in time [80].

3.5.2. Community Evolution in Multi-layer Networks

In [112], the authors present a method to quantify how much communities expand, shrink, merge, and split. For that, they create a migration matrix T_t, in which the rows are the communities of a timepoint t and the columns are the communities of the subsequent timepoint $t+1$.

Let $\vec{C} = \{C_1, ..., C_t\}$ be the vector of the set of communities at each timepoint t. At each time point, we have a set of communities in the network $C_t = \{c_{1,t}, ..., c_{n,t}\}$, where $c_{i,t}$ is a community i in timepoint t and n is the total number of communities of C_t.

The cells of the matrix reveal the number of members of a community $c_{i,t}$ that migrated to $c_{j,t+1}$. For the cases in which members of a $c_{i,t}$ disappear in $t+1$, the matrix has an extra row for members that disappear, $Null_t$. For members that are "born" in $t+1$, there is a column $Null_{t+1}$.

Then, with these values, we can compute the percentage of members that transitioned from one community to the other:

$$\psi_{i,j} = \frac{f_t(a_{i,j})}{f_t(c_{t,i})}, \text{ and } \eta_i = \frac{f_t(b_i)}{f_t(c_{t,i})} \tag{3.6}$$

where $f_t(a_{i,j})$ denotes the number of members of $c_{t,i}$ that migrated to $c_{t+1,j}$, $f_t(c_{t,i})$ denotes the total number of members of $c_{t,i}$, $f_t(b_i)$ denotes the number of members of $c_{t,i}$ that did not migrate to the $c_{i,t+1}$.

For cases in which members transition from many communities to one, we have:

$$\phi_{i,j} = \frac{f_{t+1}(a_{i,j})}{f_{t+1}(c_{t+1,j})}, \text{ and } \mu_j = \frac{f_{t+1}(d_j)}{f_{t+1}(c_{t+1,j})} \tag{3.7}$$

where $f_{t+1}(a_{i,j})$ denotes the number of members of $c_{i,t}$ that migrated to $c_{j,t}$, $f_{t+1}(c_{j,t+1})$ the sum of members of $c_{j,t+1}$ and $f_{t+1}(d_j)$ denotes the number of members of $c_{t+1,j}$ that appear for the first time ("are born") in $t+1$ and did not exist in t. Considering $m+1$ (the $+1$ is the rest of the nodes that appeared in $t+1$, $Null_{t+1}$) communities in $t+1$, which denote the m communities in C_{t+1} and $Null_{t+1}$ with the respective weights of the nodes' migration $[\psi_{i,1}, ..., \psi_{i,m}, \eta_i]$. More specifically, $\psi_{i,1}$ measures the number of nodes from $c_{t,i}$ that migrated to $c_{t+1,1}$, which is community 1 in t. Then, the weights are normalized as in:

$$\hat{\psi}_{i,j} = \frac{\psi_{i,j}}{\sum_{j=1}^{m} \psi_{i,j}} \tag{3.8}$$

The split index can be calculated as follows:

$$I^{\psi}_{c_{t,i}} = (1 - \eta_i) H_{c_{t,i}} \tag{3.9}$$

and the shrink index can be calculated as follows:

$$I^{\eta}_{c_{t,i}} = \eta_i (H^{*}_{t \to t+1} - I^{\psi}_{c_{t,i}} + \sigma_{\eta_i}) \tag{3.10}$$

where

$$H_{c_{t,i}} = -\sum_{j=1}^{m} \hat{\psi}_{i,j} \log_2(\hat{\psi}_{i,j}), \text{ and } H^{*}_{t \to t+1} = -\log_2(\frac{1}{m}) \tag{3.11}$$

$$\sigma_{\eta_i} = \begin{cases} 0.5\eta_i, \text{ if } m = 1 \\ 0, \text{ otherwise} \end{cases} \tag{3.12}$$

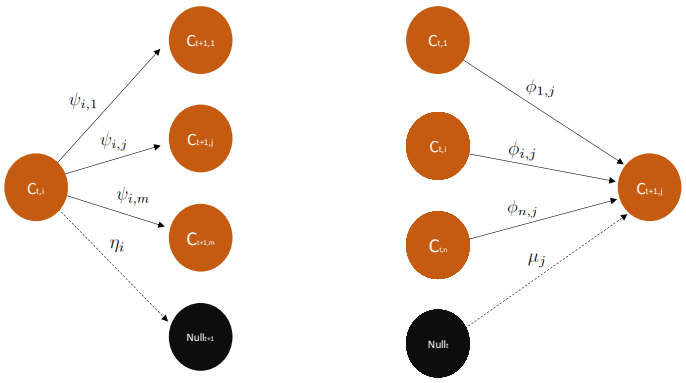


Figure 3.10.: Graphical explanation of the metrics used to compute the shrink and split indices. $c_{t,i}$ denotes a subgroup/community i at time point t. Similarly, $c_{t+1,1}$ denotes subgroup 1 at the next time point, $t+1$.

The graphical representation shown in Figure 3.10 provides an explanation of how the metrics used to calculate the indices are computed.

In practical terms, $I^{\psi}_{c_{t,i}}$ refers to how much community i at time points t split, with comparison to the subsequent time point $t+1$. Higher values mean that the community split more than others with a lower split value. Lower split index values are better if we aim for stability, but if we aim for change, higher values are better. This is highly dependent on the application - if we want stable subgroups or subgroups that change rapidly (e.g., many features of a particular product are introduced and customers start to buy different but more personalized/expensive products.) Similarly, $I^{\eta}_{c_{t,i}}$ measures how much a community i shrank from t to $t+1$.

4. Materials

In our work, we use 4 different datasets, which we introduce hereafter:

- Dataset 1: questionnaire and socio-demographic data from tinnitus patients of a clinical center in Regensburg, Germany
- Dataset 2: questionnaire and socio-demographic data from a Randomized Clinical Trial (RCT) of tinnitus patients
- Dataset 3: Ecological Momentary Assessment (EMA) of mHealth App users that assesses cognitive performance by completion cognitive task digitally
- Dataset 4: questionnaire and socio-demographic data from two clinical centers

We use data from three datasets of patients who have been diagnosed with tinnitus. Tinnitus is a medical condition that can be characterized as a phantom perception of sound, as first described in [60]. This condition can manifest in a variety of ways, such as ringing, hissing, or buzzing in the ears, and can have a significant impact on the quality of life of those affected. Tinnitus is multifactorial and is associated with further morbidities and co-morbidities, which affect treatment planning and treatment outcome [81]. Therefore, clinicians use several questionnaires and assessments for diagnosis. However, in practice, patients do not fill all of them as we show in [89]. Hence, we use tinnitus datasets with high or low amount of missingness to showcase the potential of cost-aware feature selection for community detection, where a feature is a questionnaire which is modeled by a layer. These three datasets are 4.1, 4.2 and 4.3.

Next, we explain each of the datasets in more detail. In the other chapters, we refer to these datasets as they are used in the experiments. We also present a table with a summary of the most essential information of each dataset.

4.1. UHREG: Data Before and After Treatment

This dataset refers to chronic tinnitus patients admitted to the University Hospital of Regensburg in Germany. The data was gathered between January 3, 2016 and May 28, 2020.

The studies involving human participants were reviewed and approved by the ethics committee of the University Regensburg. The patients/participants provided written informed consent to participate in this study.

At admission, each patient fills out a series of questionnaires meant to assess some of the patient's mental and physiologic symptoms. These questionnaires are endorsed in the guidelines for chronic tinnitus (cf. [104]).

The questionnaire data used in this research was gathered from five questionnaires: TQ [42],THI [75], Tinnitus Functional Index Questionnaire (TFI) [76], Major Depression Inventory Questionnaire (MDI) [12], and Tinnitus Impairment Questionnaire (TBF12) [45]. Table 4.1 shows the description of the dataset. One important insight from this brief overview is that the majority of the patients are male.

	screening (N=1141)	interim_visit (N=177)	final_visit (N=382)
Age			
Mean (SD)	53.8 (12.9)	55.1 (11.9)	55.3 (11.7)
Median [Min, Max]	55.0 [19.0, 91.0]	57.0 [27.0, 79.0]	56.0 [26.0, 85.0]
Sex			
f	413 (36.2%)	52 (29.4%)	105 (27.5%)
m	728 (63.8%)	125 (70.6%)	277 (72.5%)
THI score			
Mean (SD)	49.5 (22.9)	49.4 (23.5)	48.9 (22.7)
Median [Min, Max]	48.0 [0, 100]	48.0 [6.00, 98.0]	46.0 [4.00, 100]
Missing	38 (3.3%)	51 (28.8%)	113 (29.6%)
TFI score			
Mean (SD)	53.4 (21.2)	62.9 (19.1)	52.6 (20.1)
Median [Min, Max]	53.5 [1.00, 100]	63.0 [25.0, 82.0]	54.0 [8.00, 95.0]
Missing	309 (27.1%)	170 (96.0%)	155 (40.6%)
Mini-TQ score			
Mean (SD)	14.1 (5.50)	16.2 (6.77)	13.2 (5.58)
Median [Min, Max]	14.0 [0, 24.0]	19.0 [3.00, 21.0]	13.0 [1.00, 24.0]
Missing	848 (74.3%)	171 (96.6%)	261 (68.3%)
CGI score			
Mean (SD)	NA (NA)	3.89 (1.09)	3.51 (1.20)
Median [Min, Max]	NA [NA, NA]	4.00 [-1.00, 7.00]	4.00 [-1.00, 7.00]
Missing	1141 (100%)	8 (4.5%)	22 (5.8%)

Table 4.1.: Source data: Dataset UHREG 4.1. Data description of the socio-demographics and questionnaire scores per treatment visit.

Two-time points are considered: t_0 denotes the so-called "screening", where all questionnaires are answered, and the treatment is scheduled to start; t_1 denotes the moment of the last visit of the patient at the end of the treatment, whereby some of the questionnaires are answered again, and the scores are compared. We also use the expressions "pre-treatment" and "before treatment" (moment) synonymously for t_0 and "post-treatment" and "after treatment" (moment) synonymously for t_1.

4.2. RCT: With Three Time Points

This dataset contains data from an RCT with tinnitus patients [97]. Each arm of the RCT participants received a different treatment: four single treatments and six pairwise combinations of these four treatments. Among them was Cognitive Behavioral Therapy (CBT), designed explicitly for tinnitus patients, according to [35]. This treatment requires clinician certification, which may still be needed for each tinnitus facility. The patients have a baseline visit in which the treatment starts, and data about their symptoms is gathered. Then, there is an interim visit at which the patients fill out the same set of questionnaires, as well as at the final visit. We use data from three clinical centers (Berlin and Regensburg, Germany, and Granada, Spain).

	Berlin		Regensburg	
	Test set (N=24)	**Train set (N=65)**	**Test set (N=20)**	**Train set (N=61)**
Age				
Mean (SD)	53.3 (11.3)	50.8 (14.4)	56.6 (10.0)	54.9 (14.6)
Median [Min, Max]	55.0 [33.0, 75.0]	53.0 [24.0, 78.0]	57.5 [33.0, 73.0]	58.0 [0, 75.0]
Sex				
female	14 (58.3%)	34 (52.3%)	6 (30.0%)	16 (26.2%)
male	10 (41.7%)	31 (47.7%)	14 (70.0%)	45 (73.8%)
THI score				
Mean (SD)	34.0 (19.0)	30.3 (17.4)	33.7 (19.3)	38.4 (21.2)
Median [Min, Max]	27.0 [8.00, 78.0]	26.0 [6.00, 82.0]	28.0 [10.0, 82.0]	32.0 [10.0, 88.0]
TFI score				
Mean (SD)	35.0 (21.4)	33.7 (18.7)	40.1 (23.9)	40.4 (22.0)
Median [Min, Max]	33.5 [10.0, 82.0]	32.0 [7.00, 82.0]	43.0 [2.00, 87.0]	38.0 [8.00, 85.0]

Table 4.2.: Source data: Dataset RCT 4.2. Data description of the target variable of the train set used in the prediction task of predicting THI at t_2 using the information at t_0.

Questionnaires used in this RCT are also endorsed in the guidelines for chronic tinnitus [104]. The questionnaire data used from this dataset was gathered from five questionnaires: tinnitus questionnaire (mini-TQ) [51], tinnitus handicap inventory (THI) [75], tinnitus functional index (TFI) [76], major depression inventory (MDI) [12], tinnitus impairment questionnaire (TBF12) [45], patient health questionnaire (PHQ9) [67] and the fear of tinnitus questionnaire (FTQ) [36]. Table 4.4 describes the data used in the experiments.

Table 4.2 shows the train and test set used later in this thesis for the prediction of the THI questionnaire at t_2, for both clinic centers, using for training data at time point t_0.

Table 4.3 displays the train and test sets used for predicting the THI questionnaire at t_2, for both clinic centers, using training data at time point t_1 in this thesis.

	Berlin		Regensburg	
	Test set (N=25)	**Train set (N=64)**	**Test set (N=26)**	**Train set (N=55)**
Age				
Mean (SD)	51.3 (11.4)	51.5 (14.5)	57.1 (15.8)	54.5 (12.5)
Median [Min, Max]	53.0 [29.0, 69.0]	55.0 [24.0, 78.0]	60.5 [0, 75.0]	58.0 [24.0, 74.0]
Sex				
female	16 (64.0%)	32 (50.0%)	7 (26.9%)	15 (27.3%)
male	9 (36.0%)	32 (50.0%)	19 (73.1%)	40 (72.7%)
THI score				
Mean (SD)	30.0 (18.3)	31.8 (17.8)	36.9 (20.4)	37.4 (21.1)
Median [Min, Max]	26.0 [8.00, 78.0]	27.0 [6.00, 82.0]	31.0 [10.0, 82.0]	32.0 [10.0, 88.0]
TFI score				
Mean (SD)	33.9 (19.7)	34.1 (19.4)	40.8 (22.1)	40.1 (22.6)
Median [Min, Max]	35.0 [7.00, 73.0]	30.5 [10.0, 82.0]	34.0 [8.00, 84.0]	42.0 [2.00, 87.0]

Table 4.3.: Source data: Dataset RCT 4.2. Data description of the target variable of the train set used in the prediction task of predicting THI at t_2 using the information at t_1.

	Berlin			Granada			Regensburg		
	baseline (N=99)	interim_visit (N=90)	final_visit (N=91)	baseline (N=71)	interim_visit (N=56)	final_visit (N=50)	baseline (N=100)	interim_visit (N=89)	final_visit (N=81)
Age									
Mean (SD)	51.3 (13.4)	51.3 (13.6)	51.5 (13.5)	49.1 (10.9)	50.7 (10.7)	51.1 (10.7)	55.6 (12.8)	55.3 (13.3)	55.3 (13.6)
Median [Min, Max]	55.0 [24.0, 78.0]	54.5 [24.0, 78.0]	55.0 [24.0, 78.0]	49.0 [26.0, 69.0]	52.5 [26.0, 69.0]	52.5 [26.0, 69.0]	58.0 [0, 75.0]	58.0 [0, 75.0]	58.0 [0, 75.0]
Sex									
female	52 (52.5%)	49 (54.4%)	50 (54.9%)	32 (45.1%)	29 (51.8%)	25 (50.0%)	24 (24.0%)	22 (24.7%)	22 (27.2%)
male	47 (47.5%)	41 (45.6%)	41 (45.1%)	39 (54.9%)	27 (48.2%)	25 (50.0%)	76 (76.0%)	67 (75.3%)	59 (72.8%)
THI score									
Mean (SD)	46.8 (19.0)	35.2 (17.1)	31.3 (17.8)	46.1 (21.3)	39.6 (20.4)	37.8 (21.5)	51.3 (20.3)	43.0 (21.7)	37.3 (20.7)
Median [Min, Max]	44.0 [10.0, 96.0]	32.0 [2.00, 98.0]	26.0 [6.00, 82.0]	42.0 [10.0, 96.0]	34.0 [6.00, 92.0]	30.0 [8.00, 94.0]	48.0 [18.0, 96.0]	40.0 [8.00, 90.0]	32.0 [10.0, 88.0]
Missing	0 (0%)	1 (1.1%)	2 (2.2%)	0 (0%)	1 (1.8%)	0 (0%)	1 (1.0%)	1 (1.1%)	0 (0%)
TFI score									
Mean (SD)	47.5 (19.6)	36.8 (17.8)	34.0 (19.4)	52.7 (21.3)	46.7 (21.3)	43.6 (20.8)	49.7 (20.3)	42.6 (21.3)	40.3 (22.3)
Median [Min, Max]	51.0 [10.0, 90.0]	34.0 [1.00, 89.0]	32.0 [7.00, 82.0]	54.0 [11.0, 94.0]	44.0 [6.00, 100]	40.5 [14.0, 99.0]	51.0 [8.00, 92.0]	40.0 [5.00, 95.0]	40.0 [2.00, 87.0]
Missing	0 (0%)	1 (1.1%)	2 (2.2%)	0 (0%)	1 (1.8%)	0 (0%)	0 (0%)	1 (1.1%)	0 (0%)
Mini-TQ score									
Mean (SD)	11.3 (4.74)	9.00 (4.53)	7.74 (4.68)	13.1 (4.81)	11.0 (4.90)	11.1 (4.46)	13.0 (5.19)	11.0 (5.32)	9.44 (5.31)
Median [Min, Max]	11.0 [1.00, 23.0]	8.00 [0, 22.0]	7.00 [0, 20.0]	14.0 [4.00, 23.0]	10.0 [3.00, 21.0]	11.0 [4.00, 23.0]	13.0 [4.00, 24.0]	10.0 [1.00, 24.0]	8.00 [2.00, 21.0]
Missing	0 (0%)	1 (1.1%)	2 (2.2%)	8 (11.3%)	1 (1.8%)	0 (0%)	0 (0%)	1 (1.1%)	0 (0%)
FTQ score									
Mean (SD)	6.44 (3.36)	4.53 (3.09)	4.33 (3.36)	7.41 (3.91)	6.32 (4.27)	5.92 (3.95)	7.00 (2.98)	5.57 (3.44)	4.84 (3.31)
Median [Min, Max]	6.00 [0, 16.0]	4.00 [0, 17.0]	3.00 [0, 16.0]	7.00 [2.00, 15.0]	5.00 [1.00, 15.0]	5.00 [0, 16.0]	7.00 [0, 14.0]	5.00 [0, 16.0]	4.00 [0, 14.0]
Missing	1 (1.0%)	9 (10.0%)	2 (2.2%)	1 (1.4%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
PHQ9 score									
Mean (SD)	7.85 (4.55)	7.04 (4.24)	6.37 (4.56)	7.42 (5.52)	6.27 (5.69)	5.54 (5.43)	7.66 (4.96)	7.65 (4.92)	6.05 (4.63)
Median [Min, Max]	7.00 [0, 23.0]	6.00 [0, 20.0]	6.00 [0, 24.0]	6.00 [0, 27.0]	4.00 [0, 25.0]	4.00 [0, 25.0]	7.00 [0, 22.0]	6.00 [0, 22.0]	5.00 [0, 23.0]
Missing	0 (0%)	0 (0%)	2 (2.2%)	0 (0%)	1 (1.8%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
CGI score									
Mean (SD)	NA (NA)	3.55 (1.17)	3.43 (1.09)	NA (NA)	3.29 (0.896)	3.14 (1.09)	NA (NA)	3.51 (0.871)	3.51 (0.882)
Median [Min, Max]	NA [NA, NA]	4.00 [1.00, 7.00]	3.00 [1.00, 6.00]	NA [NA, NA]	3.00 [1.00, 5.00]	3.00 [1.00, 6.00]	NA [NA, NA]	4.00 [1.00, 5.00]	4.00 [1.00, 5.00]
Missing	99 (100%)	1 (1.1%)	2 (2.2%)	71 (100%)	1 (1.8%)	0 (0%)	100 (100%)	1 (1.1%)	0 (0%)

Table 4.4.: Source data: Dataset RCT 4.2. Data description for each clinical center and time point.

4.3. COGN: With Many Time Points

Week	N	4, N = 261†	5, N = 218†	6, N = 162†	7, N = 142†	8, N = 124†	9, N = 108†	10, N = 93†	11, N = 83†	12, N = 74†	13, N = 61†
Static variables											
Age	1,310	59 (52, 67)	59 (52, 66)	59 (52, 67)	59 (53, 66)	59 (54, 67)	60 (54, 69)	60 (54, 67)	61 (55, 69)	63 (55, 69)	61 (55, 68)
Unknown		5	4	1	2	2	1	1	0	0	0
Sex	1,326										
female		62 (24%)	46 (21%)	41 (25%)	35 (25%)	30 (24%)	29 (27%)	21 (23%)	22 (27%)	23 (31%)	20 (33%)
male		199 (76%)	172 (79%)	121 (75%)	107 (75%)	94 (76%)	79 (73%)	72 (77%)	61 (73%)	51 (69%)	41 (67%)
BMI	1,318	25.4 (22.7, 28.7)	25.2 (22.8, 29.1)	25.1 (22.9, 28.7)	25.8 (23.0, 29.4)	25.9 (23.3, 29.0)	25.8 (23.3, 28.6)	25.9 (23.3, 28.7)	25.9 (23.3, 28.6)	25.8 (23.4, 28.6)	25.7 (23.4, 28.4)
Unknown		3	2	1	1	1	0	0	0	0	0
Variables associated with test session that might influence performance											
TimeOfDayLearning		16.1 (12.3, 19.2)	15.7 (11.6, 18.6)	15.4 (11.1, 18.1)	16.1 (11.9, 18.5)	13.9 (10.4, 17.7)	15.4 (10.2, 18.3)	13.3 (9.0, 17.6)	15.0 (10.6, 18.2)	16.0 (11.6, 18.3)	15.6 (12.3, 18.0)
pc_delay											
1		147 (56%)	117 (54%)	87 (54%)	77 (54%)	72 (58%)	63 (58%)	54 (58%)	43 (52%)	44 (59%)	30 (49%)
2		74 (28%)	75 (34%)	50 (31%)	36 (25%)	32 (26%)	29 (27%)	26 (28%)	29 (35%)	23 (31%)	20 (33%)
3		34 (13%)	23 (11%)	20 (12%)	24 (17%)	15 (12%)	12 (11%)	11 (12%)	9 (11%)	2 (2.7%)	8 (13%)
4		6 (2.3%)	2 (0.9%)	5 (3.1%)	5 (3.5%)	5 (4.0%)	4 (3.7%)	2 (2.2%)	1 (1.2%)	5 (6.8%)	3 (4.9%)
5		0 (0%)	1 (0.5%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (1.2%)	0 (0%)	0 (0%)
Screen		11.9 (11.7, 14.8)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	12.3 (11.7, 14.9)	12.3 (11.7, 14.9)	11.9 (11.7, 14.9)
Unknown		54	36	14	10	3	0	1	3	4	7
Outcome variable											
pct_corr		0.56 (0.44, 0.68)	0.52 (0.44, 0.64)	0.56 (0.44, 0.68)	0.56 (0.44, 0.68)	0.52 (0.44, 0.60)	0.52 (0.44, 0.65)	0.56 (0.44, 0.68)	0.52 (0.44, 0.64)	0.52 (0.44, 0.64)	0.56 (0.44, 0.68)
Part I: Self-assessment questionnaire - Questions regarding test session											
Concentration											
Unknown		51	35	14	10	3	0	1	3	4	7
Distraction											
Unknown		51	35	14	10	3	0	1	3	4	7
Subjective Performance											
Unknown		51	35	14	10	3	0	1	3	4	7
Part II: Self-assessment questionnaire - Questions concerning the previous 8 days											
HappinessWk											
Unknown		78	57	39	25	26	18	22	22	24	31
EnergyWk											
Unknown		78	57	39	25	27	19	22	22	24	31
InterestWk											
Unknown		78	57	39	25	26	19	22	22	24	31
SubjectiveMemoryWk											
Unknown		78	57	39	25	26	19	22	22	24	31
ConcentrationWk											
Unknown		78	58	39	25	26	19	22	22	24	31
ThinkingWk											
Unknown		78	58	39	25	26	19	22	22	24	31
PlanningWk											
Unknown		78	58	39	26	26	19	22	22	24	31
Calmness											
Unknown		78	57	39	25	26	19	22	22	24	31
Freshness											
Unknown		78	57	39	25	26	19	22	22	24	31
Mood											
Unknown		95	58	39	25	26	19	22	22	24	31

† Median (IQR); n (%)

Table 4.5.: Overview of the dataset distribution of the variables of the dataset over all time points (weeks)

Dataset 3 contains mHealth data from users that are exposed to cognitive tasks. The experiment is introduced in [14] and is composed of three tasks:

- Mnemonic Discrimination Task for Objects and Scenes (MDT-OS): this task assesses pattern separation in a short-term memory task

- Object- In-Room Recall (ORR): assesses pattern completion in both short- and long-term memory tasks
- Complex Scene Recognition (CSR): assesses long-term memory with recognition of photographic scenes

Each participant is requested by the mHealth app to complete one of the three randomly selected memory tasks every week. The participants are asked to complete the tasks by push notifications on their mobile devices. After completing each task, each participant is asked how well they perceive that they performed. Other questions about how the participant perceives their well-being are asked and stored. The actual performance of the task(s) is also available for analysis. Table 4.5 shows the number of users across the 16 weeks and briefly describes age, sex and Body Mass Index (BMI).

4.4. CLINICS: With Data From Two Different Clinics

This dataset contains data from patients with chronic tinnitus who were admitted to the University Hospital of Regensburg and the Tinnitus Center, Charité Universitätsmedizin Berlin. The Regensburg University Hospital collected the data between January 3, 2016 and May 28, 2020. The data from the Tinnitus Center, Charité Universitätsmedizin Berlin were gathered between January 1, 2011 and October 15, 2015. Despite the fact that both datasets contain data from many questionnaires and socio-demographic data, some variables appear only in one dataset and vice versa. Table 4.6 shows the number of patients with available data at different time points as well as the number of different treatments that were assigned. Two time points were considered: t_0 and t_1. The former denotes the time point at admission, while the latter represents the time at the final visit of the patient to the clinical centre. For the sake of simplicity, University Hospital of Regensburg is denoted by UHREG, and the Tinnitus Center, Charité Universitätsmedizin Berlin, is denoted by CHA.

Specific data pre-processing steps are required depending on the type of analysis performed. The three research questions require different filters. I.e., for RQ1, we consider all patients from both centres who filled out the age and gender questions among all questionnaires at admission. RQ2 and RQ3 require a match on the questionnaire data being compared. This phase is critical for juxtaposing the medical centres' questionnaires and ensuring the feasibility of the comparison.

Table 4.6 summarizes the available questionnaires per centre. In order to compute the treatment outcome, the scores at admission and after treatment are used. As a consequence, the shared questionnaire (in this case, TQ) has to be available at both time points (t_0 and t_1) in order to learn a model that predicts the treatment outcome.

Table 4.7 shows the description of the patients from the clinical center "CHA". Please note that the patients from "UHREG" in this dataset are the same ones as in 4.1, with precisely the same data. In this dataset, we aggregate the data shown in Table 4.7 with the ones shown in 4.1.

Category	Questionnaire	CHA	REG	Citation
Tinnitus distress	TQ	✓	✓	[42]
	TLQ	✓		[43]
	THI		✓	[75]
	TFI		✓	[76]
	TBF12		✓	[45]
	CGI		✓	[121]
Physical strain	BI	✓		[18]
Depressivity	ADSL	✓		[49]
	BSF	✓		[52]
	MDI		✓	[41]
Stress	PSQ	✓		[70, 32]
Quality of life	SF8	✓		[13]
Coping	SWOP	✓		[98]
Socio-demographics	SOZK	✓		
	[age, gender]	✓	✓	

Table 4.6.: Questionnaire categories and the available questionnaire data per centre. TQ: tinnitus questionnaire, TL: tinnitus localization and quality questionnaire, THI: tinnitus handicap inventory, TFI: tinnitus functional index, TBF12: tinnitus impairment questionnaire, CGI: clinical global impression, BI: Berlin complaint inventory, ADSL: general depression scale—long form, BSF: Berlin mood questionnaire, MDI: major depression inventory, PSQ: perceived stress questionnaire, SF8: short-form 8 health survey, SWOP: self-efficacy- optimism-pessimism scale, SOZK: socio-demographics questionnaire.

	final_visit (N=500)	screening (N=500)
Age		
Mean (SD)	50.3 (12.2)	50.3 (12.2)
Median [Min, Max]	51.0 [18.0, 83.0]	51.0 [18.0, 83.0]
Sex		
f	260 (52.0%)	260 (52.0%)
m	240 (48.0%)	240 (48.0%)
TQ_score		
Mean (SD)	31.6 (18.0)	38.8 (18.1)
Median [Min, Max]	28.5 [0, 82.0]	37.0 [1.00, 82.0]

Table 4.7.: Source data: Dataset CLINICS 4.4. Data description of the 500 patients from the clinical center “CHA”.

5. COBALT for Static Data

5.1. Overview

We present COBALT, a cost-based layer selector that finds subgroups in MLNs in an improved cost-aware manner. COBALT has a successor named Evol-COBALT, which includes a temporal dimension and is introduced in the next chapter.

The method can be summarized into the steps illustrated in Figure 5.1.

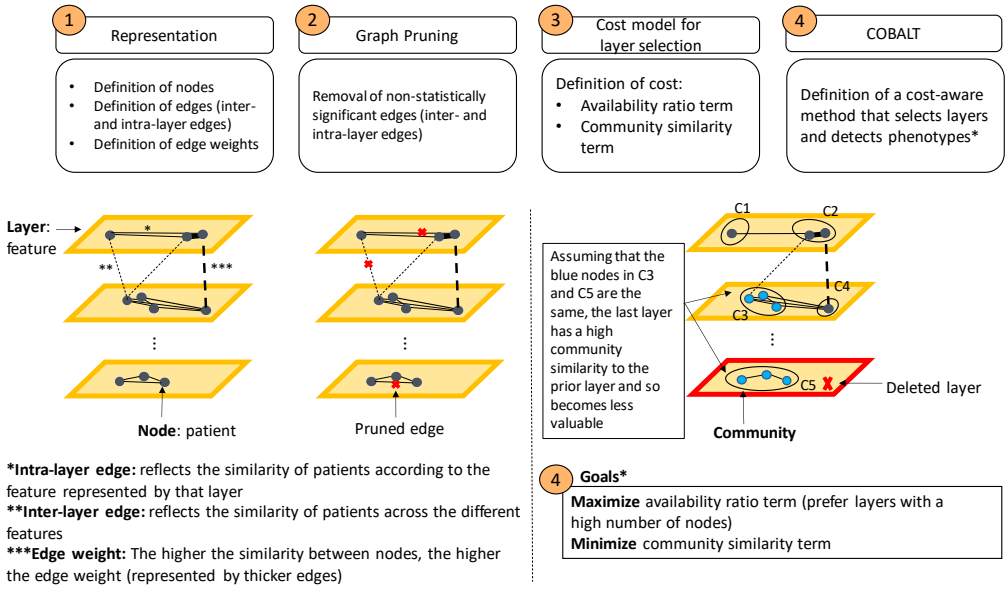


Figure 5.1.: Content structure of the proposed method, COBALT. Each block represents a stage of the approach, which are presented in the next sections.

In this workflow, we have 4 main blocks, which occur in order from left to right. First, the representation of data in a network. Then, pruning of edges. These two blocks refer to the pre-community detection stage, which begins with block 3 (Cost model for layer selection). In this block, we first incorporate the notion of feature/layer cost into a quantitative cost function. After, we incorporate it in the community detection algorithm, leading to COBALT. The cost model should consider the previously stressed problem of considering data missingness and feature cost.

This chapter is based on the following publications:

- C. Puga, U. Niemann, V. Unnikrishnan, M. Schleicher, W. Schlee, and M. Spiliopoulou, "Discovery of Patient Phenotypes through Multi-layer Network Analysis on the Example of Tinnitus," 2021 IEEE 8th Int. Conf. Data Sci. Adv. Anal., pp. 1–10, 2021, doi: 10.1109/dsaa53316.2021.9564158.

- C. Puga, U. Niemann, W. Schlee, and M. Spiliopoulou, "A cost-based multi-layer network approach for the discovery of patient phenotypes," Int. J. Data Sci. Anal., 2023, doi: 10.1007/s41060-023-00431-7.

5.2. Entity Similarity Representation in Sparse Multi-layer Networks

The first step towards the discovery of subgroups using multilayer networks is to represent the data into an MLN. The main goal is that the MLNs are able to represent the real system.

The first step is to model the essential components of an MLN:

- Nodes
- Edges

  Inter-layer edges (weighted/unweighted, directed/undirected)

  Intra-layer edges (weighted/unweighted, directed/undirected)
- Layers

Figure 5.2 illustrates the steps we use to represent real-world data into an MLN.

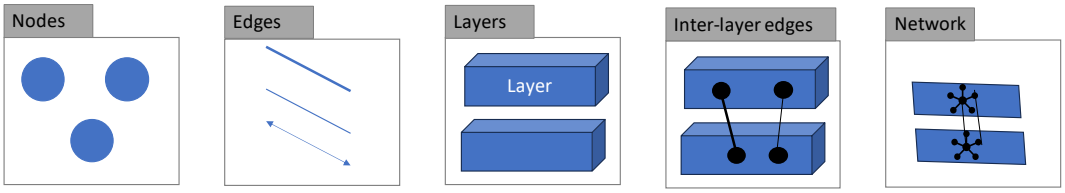


Figure 5.2.: Steps for representation in an MLN.

For this chapter, we use dataset 4.1 as an example and use the questionnaire scores as layers.

When building the inter-layer edges, two types of edges may exist:

1. edges of a node with itself between two layers
2. edges between different nodes located in different layers

We generate the inter-layer edges to incorporate the different perspectives given by different features of a node. Hence, only inter-layer edges between the same nodes in different layers are incorporated [88, 90].

The distances between nodes within a layer are represented by intra-layer edges. A layer represents a feature of the dataset. These features can be questionnaires (as in datasets 4.1, 4.2,4.3) or can be simply a numerical feature that represents the system.

Considering two nodes p_i and p_j in the same layer, the edge between them is defined by the difference between their feature values. Assuming a numerical feature $l \in L$, $value_{i,l}$ and $value_{j,l}$ denote the values of nodes i and j in layer l, respectively.

The larger the discrepancy in values between nodes, the lower their edge weight. This ensures that their connection is represented by a “weak" edge weight if their values are not close. To accommodate for this, we define the weight as in Equation 5.1, which describes the transformation $1/x$.

$$w_{i,j,l} = \frac{1}{|value'_{i,l} - value'_{j,l}|} \tag{5.1}$$

In Equation 5.1, $w_{i,j,l}$ denotes the edge weight that connects i and j in layer l. The values are then normalized by subtracting the mean and dividing by the standard deviation.

Equation 5.2 shows how to compute the weight $w_{(i,l_\alpha),(i,l_\beta)}$ of an inter-layer edge that connects (i, l_α) and (i, l_β).

$$w_{(i,l_\alpha),(i,l_\beta)} = \frac{1}{|value'_{i,l_\alpha} - value'_{i,l_\beta}|} \tag{5.2}$$

$value'_{i,l_\alpha}$ and $value'_{i,l_\beta}$ denote the normalized value of layer l_α and l_β for node i, respectively. $l_\alpha, l_\beta \in L$ correspond to layers α and β, respectively.

5.3. Graph Pruning

Graph pruning is a network simplification filtering strategy [58]. Graph pruning improves computation for algorithms that have difficulty in large networks. Pruning is a dimensionality reduction technique used to remove noisy and unneeded edges/nodes from networks. These methods are divided into three categories: (i) centrality-based, (ii) node-layer relevance-based, and (iii) model-based. The first two are node removal and edge pruning, respectively. In this study, we use the Maximum Likelihood Filter (MLF) introduced in [29], as described in [58], as an unbiased edge-filtering technique with maximum entropy.

The fundamental concept of MLF is to develop a null model that generates a "realized graph," which is a graph with the same total weight and degree sequence as the original graph. Then, the "realized graph" is used to compute the statistical significance of each edge compared to the "realized graph". Edges with p-values greater than a certain *alpha* (significance level) are pruned. The nodes in the "realized graph" are the same as in the original graph, with the same degree sequence. Edges are assigned to a pair of nodes with a probability of being selected. For example, a node with a high degree is more likely than a node with a low degree to be allocated an edge. The binomial distribution used to compute this likelihood is as follows:

$$Pr(\sigma_{ij} = m|k_i, k_j, E) = \binom{E}{m} p^m (1-p)^{E-m} \tag{5.3}$$

where m is an edge from the set of edges E that connected the pair of nodes (i, j); k_i, k_j are the degrees of nodes i and j, respectively; σ_{ij} is the weight of the undirected edge between i and j, $p = \frac{k_i k_j}{2E^2}$ and $E = \frac{1}{2}\sum_i k_i$. The p-value of an edge connecting nodes i and j with a weight of w_{ij} is denoted as $s_{ij}(w_{ij})$. Equation 5.4 shows the calculation of the p-value of an edge with weight w_{ij}.

$$s_{ij}(w_{ij}) = \sum_{m \geq w_{ij}} Pr(\sigma_{ij} = m|k_i, k_j, E) \tag{5.4}$$

5.4. Incremental Cost-aware Layer Selection

In this section, we propose two versions for the feature/layer cost model of static COBALT. The first, we name the "simplified version" and the enhanced version we call the "full-fledged version". Both versions though have the common goal of maximizing the diversity of information in the network, so that the system can be

Symbol	Description	Algorithm
G	Set of graphs/layers	5.1
Q	Modularity	5.1
$g_{bestsingle}$	first graph chosen for the MLN, with the partition with the best modularity	5.1
$corr$	Set of correlation values between $g_{bestsingle}$ and $g_i \in G \backslash g_{bestsingle}$	5.1
S_{all}	Set of partitions	5.1
Q_{all}	Set of modularity values of S_{all}	5.1
$\mathcal{L}$	the network structure: a single or MLN	5.1
$\mathcal{S}_{single}$	set of community memberships of the nodes of each single-layer network $g \in G$	5.2

Table 5.1.: Notation table of the static COBALT.

best represented, but always considering minimization of redundancy. We assume the following statement:

$$\max \text{diversity} \equiv \min \text{feature redundancy} \tag{5.5}$$

Following the definition of the cost for each version, we then show how we initialize COBALT.

5.4.1. Cost Models

Simplified Cost Model

We start by defining the way in which we compute the redundancy. Considering that the goal is to minimize redundancy, we define two redundant layers(features) as two layers that have a high association between them. Considering that the edges of the networks denote the similarity between nodes, then we focus on the edge weights to understand how layers compare. For the numerical case, we propose to use Pearson's correlation between the edges of two layers.

the cost computation between two layers, l_{inc} and l_α can be computed as follows:

$$C_{l_{inc},l_\alpha} = CS_{l_{inc},l_\alpha} = similarity(l_{inc}, l_\alpha) \tag{5.6}$$

where C_{l_{inc},l_α} is the cost between layers l_{inc} and l_α, CS_{l_{inc},l_α} is the community/subgroup similarity between both layers and $similarity(l_{inc}, l_\alpha)$ is the similarity function chosen between the three mentioned: Jaccard coefficient, adjusted rand index of normalized mutual information.

To conclude, the higher the similarity between two layers, the higher the cost C.

Full-fledged Cost Model

In the previously introduced "simplified cost model" [88], the Pearson coefficient was used to compute layer similarity. However, this metric requires a direct correspondence of edges (e.g., an edge connecting two nodes i and j in both layers so that they are comparable). In many real-world scenarios, networks are not fully connected; therefore, a more sophisticated approach is necessary. This was tackled in [90], which will be explained later.

In [90], only edges that connect the same pair of nodes are filtered and compared, disregarding the other edges. We modify the method and use instead the "F-measure,"(cf. Section 3.3) a metric that does not require correspondence and is therefore better suited to the problem.

Data availability is also a factor to consider - the edge weights of one layer may be highly correlated with the edge weights of another layer, which has a higher number of edges. However, the latter has more data, and those exceeding edges may provide helpful information about the nodes. In [90], we propose COBALT, a cost-sensitive community detection algorithm with a different metric for layer similarity. We analyse the impact of missingness (missing nodes) on the quality of the communities discovered.

Particular layers may be more likely to be available than others (due to how easy it is to gather that data). This aspect should be considered while deciding on the next layer to be added to the MLN. For example, in the medical setting, a Magnetic Resonance Imaging (MRI) might be more difficult to gather (higher cost) than a blood test. Due to the difficulty of having an MRI for every patient, this data has a low availability.

The similarity of communities between layers is a second criterion of relevance. We intend to add a layer that contains additional information about the nodes. In other words, we avoid adding redundant information. The more distinct the layers are concerning their community structure, the less redundant they are. The goal is to add a layer with a low community similarity.

Hence, the cost of a layer is calculated by combining these two ideas into two terms: an availability ratio term (A) and a community similarity term (CS).

We formulate the function to measure the cost of a layer l_α, C_{l_{inc},l_α}, with respect to the incumbent set of layers in the network (l_{inc}) as:

$$C_{l_{inc},l_\alpha} = \frac{1}{A_{l_{inc},l_\alpha}} + CS_{l_{inc},l_\alpha} \tag{5.7}$$

where A_{l_{inc},l_α} denotes the availability ratio term and CS_{l_{inc},l_α} the community similarity term. These two terms are described hereafter.

We term the percentage of nodes in which a feature is available as the availability ratio. A_{l_{inc},l_α} denotes the availability ratio of layer l_α against l_{inc}. It is the ratio of nodes that are not missing in l_α from the nodes that are already in l_{inc}.

Considering the set of nodes in layer l_α as $p^{l_\alpha} = \{p_1, p_2, ..., p_m\}$ and in the incumbent layer or layer set, l_{inc}, as $p^{l_{inc}} = \{p_1, p_2, ..., p_n\}$, $A_{l_{inc}l_\alpha}$ is given by:

$$A_{l_{inc},l_\alpha} = \frac{|p^{l_\alpha} \cap p^{l_{inc}}|}{|p^{l_{inc}}|} \tag{5.8}$$

The higher the number of missing nodes in a layer, the fewer the number of nodes added to the MLN. The goal is to add the maximum information about the nodes in each iteration. Therefore, we aim to maximize the availability ratio term.

We define the community similarity term as a measure of how similar communities are between layers with respect to the assigned nodes. More specifically, we focus on quantifying the shared nodes between communities of different layers.

We use the bi-directional F-measure to quantify this term. There are two reasons for this choice: (i) the metric includes both the purity/precision and the recall of the solution, and (ii) it handles the absence of ground truth by considering ground truth one layer at a time.

The community similarity between layers l_α and the incumbent layer or set of layers l_{inc} is modeled as:

$$CS_{l_{inc},l_\alpha} = \frac{2 * F^{[l_{inc}\|l_\alpha]} * F^{[l_\alpha\|l_{inc}]}}{F^{[l_{inc}\|l_\alpha]} + F^{[l_\alpha\|l_{inc}]}} \tag{5.9}$$

As previously stated, we intend to use layers that provide the maximum additional information about the nodes. If the community structure of two layers is substantially similar, they are seen as redundant. We intend the exact opposite: to add layers with a community structure that differs from that of the incumbent (current) network. As a result, CS_{l_{inc},l_α} should be minimized.

5.4.2. Initialization: Simplified Cost Model

In this subsection we present the initialization of our community-detection algorithm using the simplified cost model previously described (cf. subsection 5.4.1) Firstly, we select the first layer of the MLN by choosing the graph $g \in G$ with the higher modularity value q. In this phase, where we have no layer, we choose the layer with the higher modularity q. For that, we need to detect the subgroups in each g, which we do by using the Leiden (cf. 3.4) algorithm. The Leiden algorithm detects the subgroups in g, which we denote by s, which is stored in S_{all}, and then we compute the q of s, which is also stored in Q_{all}.

The layer with the highest q indicates the best partition s, meaning that it is the layer in which subgroups are the most dissimilar. With this criterion, we aim to obtain the layer/feature that can better separate the nodes.

After the first layer is chosen, g_1, the next layer to add should be the one that provides the most additional information to the current layer. Therefore, we use the Pearson correlation between the edge weights of g_1 and the other layers from G. The layer with the lower correlation with g_1 is selected. The Pearson correlation can be computed in two directions:

- correlation of the edge weights of $g_{bestsingle}$ with the other layers $g_i \in G \backslash g_{bestsingle}$
- correlation of the original values of the variables represented by the layers, i.e. all values before pruning

The two options above differ because the nodes in one layer, for example $g_{bestsingle}$ might not be all represented in other layers. Therefore, the direction of the comparison matters here. In our experiments, we try both and perform a comparison analysis.

The output is then a two-layer network. Algorithm 5.1 shows how the initial solution is generated.

After the selection of $g_{bestsingle}$ and $g_{bestsecond}$, the algorithm proceeds on detecting the subgroups in the two-layer network, which is presented hereafter.

After, the cost of each layer is computed by using the Jaccard coefficient, adjusted rand index or the normalized mutual information. These metrics evaluate the similarity between two subgroup partitions. We aim to minimize the similarity between the layers that are added to the structure $\mathcal{L}$, since we aim to add only additional information and minimize redundancy.

Please note that in this initialization, we have already selected the first 2 layers. The main idea behind this is to use the Pearson correlation in the first step because it is simpler and faster to compute than the full-fledged initialization that we present later. However, the limitation of this Pearson correlation is that it does not consider

Algorithm 5.1 COBALT Initialization Simplified: Pseudo Code for generating the first multi-layer network

Require: G
Ensure: $g_{bestsingle}; g_{bestsecond}$
$\mathcal{S}_{single} \leftarrow \emptyset$
$corr \leftarrow \emptyset$
for each $g \in G$ **do**
    $S, q \leftarrow Leiden(g)$ ▷ compute Leiden for all g
    $\mathcal{S}_{single} \leftarrow \mathcal{S}_{single} \cup S$
    $Q_{all} \leftarrow Q_{all} \cup q$
$g_{bestsingle} \leftarrow$ g with $max(Q_{all})$
▷ iterate over each combination of 2 graphs
for each $g_i \in G \backslash g_{bestsingle}$ **do**
    $corr_{g_i} \leftarrow correlation(g_{bestsingle}, g_i)$ ▷ compute correlation
    add $corr_{g_i}$ to $corr$
$g_{bestsecond} \leftarrow g$ with $min(corr)$ ▷ minimum correlation with $g_{bestsingle}$
Output: $g_{bestsingle}, g_{bestsecond}$

the availability component (missingness rate). In the following section, we will present the initialization algorithm that we use when considering the full-fledged model.

5.4.3. Initialization: Full-fledged Cost Model

On the basis of the full-fledged cost model defined in subsection 5.4.3, we now present our algorithm COBALT for cost-based layer selection and community construction.

Algorithm 5.2 shows the pseudo-code for the generation of the starting solution.

Algorithm 5.2 COBALT Initialization Full-fledged Cost Model Version: Pseudo code for choosing the first layer

Input: G
Output: $g_{bestsingle}$
$\mathcal{S}_{single} \leftarrow \emptyset$
$g_{bestsingle} \leftarrow NULL$
$S_{bestsingle} \leftarrow NULL$; $q_{bestsingle} \leftarrow -\infty$
for $g \in G$ **do**
    $S, q \leftarrow Leiden(g)$
    $\mathcal{S}_{single} \leftarrow \mathcal{S}_{single} \cup \{S\}$
    if $q_{bestsingle} < q$ **then**
        $g_{bestsingle} \leftarrow g$
        $S_{bestsingle} \leftarrow S$; $q_{bestsingle} \leftarrow q$

Output: $g_{bestsingle}, S_{bestsingle}, \mathcal{S}_{single}$

The input of Algorithm 5.2 is the set of all single-layer graphs G. For each $g \in G$, COBALT invokes the Leiden algorithm (cf. Section 3.2.2 for a detailed explanation of this algorithm) which builds a set of communities and returns two objects: the 'partition' S, which encompasses the community membership of each node for the incumbent graph $g \in G$, and q – the modularity of S. The partition $S_{bestsingle}$ with

the largest modularity and the corresponding graph $g_{bestsingle}$ are returned as initial solution of COBALT, together with the set of all single-layer partitions.

Please note that at this point *are not yet computing cost because we are simply selecting the first layer*. It would be *impossible* to compute the community similarity component described earlier, since we need at least 2 layers for that.

The main difference between this initialization and the initialization of the simplified cost model is that for the simplified cost model, the initialization corresponds to selecting the first 2 layers already.

5.5. Community Detection

The next step after initialization (cf. algorithms 5.1 and 5.2) is the mounting of the MLN structure and the global search for the structure that provides the subgroup solution with the least cost. For that, we implement our cost model by adding one layer at a time, in an ascending order of cost. The cost computation depends on the methodology previously mentioned.

Algorithm 5.3 presents the pseudo-code of the global search for the final solution. The input of the algorithm differs depending on the chosen cost model. For the simplified version, we have already the initialization with a two-layer network ($g_{bestsingle}$ and $g_{secondbest}$). The solution that comes from the initialization algorithm becomes now $\mathcal{L}$, the variable that stores MLNs. Then a list of candidate layers is created, $G_{candidateS}$, which are the set of layers that are to be added to the structure. Naturally, the layers that were already in the initial solution cannot be added again and therefore we define $G_{candidates}$ according to the selected initialization. For each

Algorithm 5.3 Iterative Layer Selection: Pseudo-code of the subsequent iterations of COBALT

Input: $G, g_{bestsingle}, S_{bestsingle}, \mathcal{S}_{single}$ and $g_{bestsecond}$ if simplified version of cost
Output: S_{best} (best set of communities)

if simplified version **then**
  $G_{candidates} \leftarrow G \setminus \{g_{bestsingle}, g_{bestsecond}\}$
  $\mathcal{L} \leftarrow MLN(\{g_{bestsingle}, g_{bestsecond}\})$ ▷ multi-layer network composed by $\{g_{bestsingle}, g_{bestsecond}\}$
else if full-fledged version **then**
  $G_{candidates} \leftarrow G \setminus \{g_{bestsingle}\}$
  $\mathcal{L} \leftarrow g_{bestsingle}$
while $G_{candidates} \neq \emptyset$ and SC is False **do**
  $g \leftarrow NULL$; $c \leftarrow +\infty$
  for $u \in G_{candidates}$ **do**
    $c_u \leftarrow cost(\mathcal{L}, u, \mathcal{S}_{single}, S_u)$
    if $c_u < c$ **then**
      $g \leftarrow u$; $c \leftarrow c_u$
  extend $\mathcal{L}$ with g
  $(S_{\mathcal{L}}, q_{\mathcal{L}}) \leftarrow Leiden(\mathcal{L})$
  $G_{candidates} \leftarrow G_{candidates} \setminus \{g\}$
return $S_{\mathcal{L}}$

candidate, a cost is computed. The cost computation requires 4 variables: the $\mathcal{L}$ which is the incumbent Single-layer Network (SLN) or MLN structure, the candidate

u, the subgroups of each layer $\mathcal{S}_{single}$ and of the subgroups of the candidate layer S_u. The algorithm iterates over all the candidates $g \in G_{candidates}$ to find the g that has the lower cost to the structure. The candidate with the lower cost is selected and added to $\mathcal{L}$.

This procedure continues until there are no more candidates ($G_{candidates} =$) and while the stopping criteria (SC) is not met (in the algorithm, this corresponds to False).

In the simplified cost model, there is a significant drawback that has been mentioned several times. The similarity between layers does not guarantee that we are adding a layer that contains a substantial amount of new information to improve the subgroups. This is because the availability of data is not taken into consideration.

For the full-fledged cost model, layers are chosen on the basis of availability ratio *and* community similarity, and since both factors take positive values, cost cannot be negative for the full-fledged cost model. Availability ratio may increase or drop from one iteration to the next, though.

The stopping criteria can are distinct for the simplified and the full-fledged cost model version:

- Simplified Cost Model

  SC1: COBALT stops when the modularity decreases compared to the previous iteration

  SC2: COBALT stops when the similarity score is lower than a threshold

- Full-fledged Cost Model

  SC1: COBALT stops when the availability ratio decreases towards the previous iteration

  SC2: COBALT stops when the availability ratio drops *and* the community similarity increases

where the "previous iteration" is the 2nd (i.e. the 1st after the initialization) or a later one.

For the simplified version, we choose to stop when the modularity of each iteration and stop when it decreases, compared to the previous iteration (*SC1*).

For the full-fledged cost model version, we choose $SC1$ as the stopping condition in the **while**-loop. In our experiments, $SC1$ is not used because we study the behavior of COBALT as each layer is added. We instead report at which iteration COBALT would have stopped and what would have been the effect on the communities' contribution to predictive power.

5.6. Experiment Design

In this section, we will discuss how we evaluate the results obtained from COBALT, which was presented previously. As we do not have ground truth, we have developed an evaluation workflow that includes not only the typical subgroup quality metrics (such as modularity) but also domain-specific information. Our aim is to create a comprehensive evaluation process that takes into account as much relevant information as possible.

Figure 5.3 provides an overview of the evaluation strategy carried out in the experiments of COBALT. We use 4 different quality strategies in order to cover the

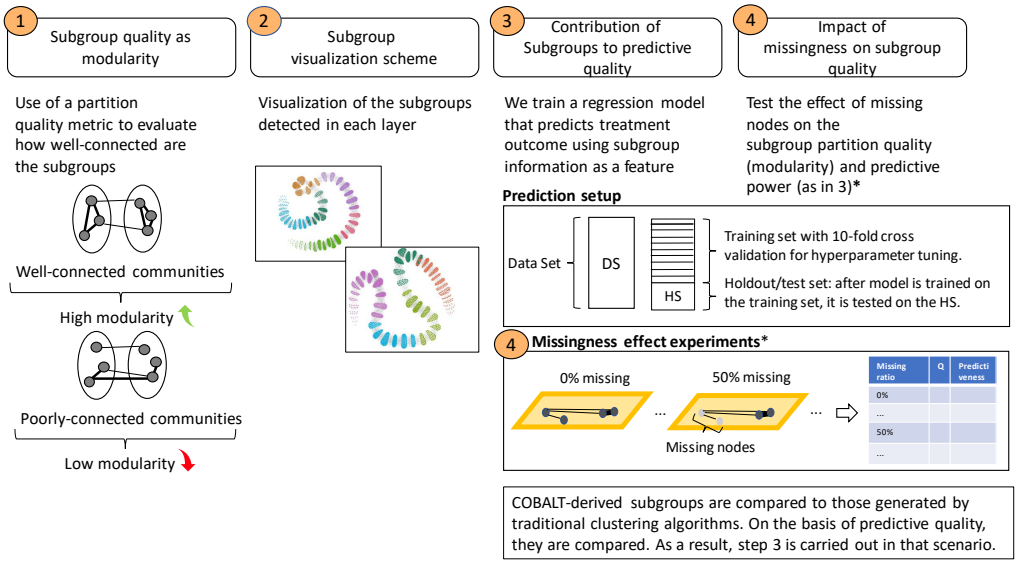


Figure 5.3.: Evaluation workflow of the COBALT algorithm.

majority of dimensions that can provide information about how good the subgroups are.

As can be interpreted from Figure 5.3, we propose both qualitative and quantitative evaluation steps. The workflow proposed aims to define a good solution in scenarios without ground truth. Hereafter, we describe each step of the evaluation workflow in detail.

5.6.1. Subgroup quality as modularity

The first step of the evaluation strategy is the calculation of modularity (cf. Equation 3.4) as a quality metric of the discovered subgroups. This metric assesses how well-connected the nodes inside each subgroup are and how well-separated the different subgroups are. This metric depends solely on how we model the system in an MLN. We cannot rely on the modularity values if the system is poorly represented. On the other hand, if the system is well described, then modularity can assess the quality of the subgroups.

Regarding the interpretation of values, higher modularity values are better because they suggest that subgroups are well-separated.

5.6.2. Subgroup visualization scheme

We use the Fruchterman-Reingold layout [34] as the basis for network visualization to gain insights into how communities develop after selecting each new layer: Nodes that are near to one another have stronger connections than those that are far apart. In this type of design, we use colors to represent communities. Thus, each node takes on the color of the subgroup to which it belongs. As a result, "excellent" communities are depicted as graph partitions with only one hue. On the other hand, multiple colors in one location of the depicted network show that the subgroups are mixed up.

To demonstrate how the colors/subgroups vary across levels, we contrast the visualizations of the layers used in each iteration of community detection. The absolute horizontal and vertical positions of nodes have no relevance; only their relative

positions do, with closer nodes positioned near each other and solid connections between them (represented by darker edges).

5.6.3. Contribution of Subgroups to Predictive Quality

The predictive quality of subgroups relies on their contribution to a specific domain-specific variable. Defining this variable is crucial, as it is highly dependent on the application. The aim is to create subgroups that enhance the accuracy of predicting this variable of interest. By doing so, we hope to gain further insights into this variable of interest.

Therefore, another step of our evaluation strategy focus on the domain of the application. More specifically, we add a step that uses domain knowledge to assess the quality of the subgroups.

We propose to define a target variable that is of interest to the specific application and then build the following subsets:

- *Baseline subset*: subset of variables without subgroup information
- *COBALT-based subset*: subset of variables in "Baseline" plus the COBALT-discovered subgroups
- *Clusterer-based subset*: subset of variables in "Baseline" plus the clusterer-discovered subgroups

The first subset refers to the subset of variables which are domain specific that are used to predict the chosen target variable, in that application.

The second subset corresponds to the "Baseline" subset plus a variable/feature that contains information about the subgroups discovered on that dataset, using COBALT.

The "Clusterer-based subset" corresponds to the "Baseline" subset plus a variable that contains information about subgroups detected using traditional clustering algorithms on that dataset.

We train one regression model per feature subset mentioned above to predict the defined target variable. With this experiment, we aim to analyze to what extent is subgroup information predictive of the target variable. Plus, we also use subgroups discovered using traditional clustering algorithms to understand how they compare to the COBALT-detected subgroups in terms of predictive power.

For regression we use linear regression, ridge, LASSO and SVR (support vector regression). To set the hyperparameters, we apply a grid search. We perform 10-fold cross validation on the training set, and we evaluate with mean absolute error (MAE), mean squared error (MSE) and R^2. The dataset is split into a test set of 30% and a training set of 70%.

The clustering techniques used are: (i) agglomerative hierarchical clustering [24], which we denote as 'AHS'; (ii) BIRCH [122], (iii) Gaussian Mixture Models with the Expectation Maximization algorithm [15, 27], which we denote as 'GMM_EM'; (iv) HDBSCAN [21], (v) k-means [48] and (vi) OPTICS [6]. It must be stressed that these clustering models serve only as baselines, because they operate under different conditions than COBALT: they are cost-insensitive, since they exploit *all* questionnaires/layers, while COBALT uses only the layers up to a given iteration.

Moreover, these clustering algorithms do not handle missing values, hence they are trained only on nodes present in all layers. There is an interplay between modularity and predictive power. Modularity is not independent of quality. Higher modularity reflects well connected subgroups, which allows for better interpretability.

5.6.4. Evaluating the Effects of Missingness

COBALT is designed to deal with layers that contain only few nodes, i.e. layers for which only few entities have delivered data. To measure the influence of missingness, we gradually introduce missingness into a dataset that originally has no missing data. We perform this experiment to dataset 4.1. When removing a node, we remove its corresponding node from all layers.

The complete workflow is as follows. If the dataset in use contains missing values, we extract from it the subset D that has no missing values on the target variable. In the next step, we specify the maximum and minimum of the "missingness ratio", which we define the percentage of nodes to be randomly selected and eliminated from D. Next, we perform a grid search between the minimum and maximum percentage and derive the corresponding subset of D for each value in the grid; for our experiment, we vary the missingness ratio from 0.1(10% of entities removed) to 0.9(90% of the entities removed), with a step of 0.1, i.e. of 10%. Finally, we run COBALT on each derived dataset and measure quality as (i) modularity at each iteration and (ii) predictive quality for the iteration with the best modularity.

To conduct a sensitivity analysis of the impact of missing data in our model, we examine the modularity of the partitions with different missing data ratios, as well as the performance of the prediction of the target variable, using the COBALT-based subset.

5.7. Results

In this section, we have two main blocks: subsections 5.7.1 and 5.7.2. They correspond to two different experiments, in which we apply the simplified cost model version of the algorithm presented in Section 5.4.1 and the full-fledged cost model version presented in Section 5.4.3.

5.7.1. Simplified Cost Model

For this experiment, we use dataset 4.1 to collect questionnaire data from tinnitus patients who visit a clinic center. The patients are assigned to a treatment and respond to the questionnaires at different times between the beginning and end of treatment.

To represent this system into an MLN, we consider the patients as nodes of a network and the questionnaires as the layers of the MLN. Since there are multiple time points (pre-treatment and post-treatment) and we are now focusing on the static algorithm, we represent the system using the pre-treatment data to represent the system.

The structure can be defined as follows:

- *Nodes:* patients
- *Layers:* questionnaire scores (1 layer per questionnaire) which means 5 layers (THI, TFI, TQ, TBF12 and MDI questionnaires)
- *Intra-layer edges:* weighted edges that represent the similarity of pairs of nodes according to the same questionnaire/layer
- *Inter-layer edges:* weighted edges that represent the similarity of the same node across two different questionnaires/layer

Both intra- and inter-layer edge weights are computed using the Equations 5.1 and 5.2, where $value_{i,l_\alpha}$ represents a score of questionnaire l_α of patient i. For example, $value_{i,THI}$ is the THI score of patient i.

Edge Pruning

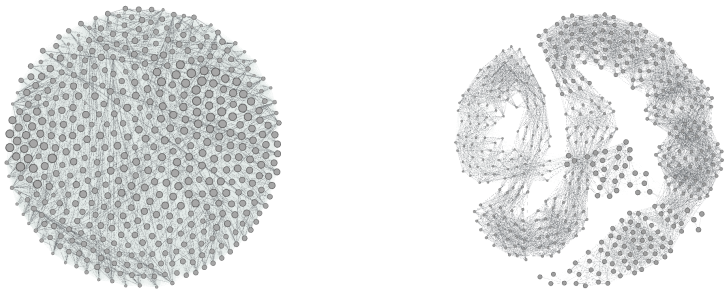
Edge pruning is applied to every single layer, using the MLF to filter only statistical significant edges. When applying this filter, two scenarios may occur:

1. No edges are pruned and the network stays as is.

2. Some edges are pruned and the network gets less dense, leading to a not-fully connected network.

3. All edges are pruned. When all edges are pruned and the network is not connected, it is a significant piece of information. This indicates that the feature being represented by that network is not able to distinguish the nodes, especially when all edge weights are very similar and there is little difference between them.

Figures 5.4a and 5.4b show one example of a layer before pruning and after pruning.



(a) Layer before pruning (b) Layer after pruning

Figure 5.4.: Two networks, one before pruning (subfigure 5.4a) and therefore fully connected and the other after pruning (subfigure 5.4b) . The nodes are represented by gray dots and the edges are represented by gray links.

The former represents the fully-connected layer before pruning. The latter corresponds to the pruned layer.

We apply the MLF method to every feature in 4.1: THI, TQ, TBF12, TQ and MDI scores.

Figure 5.5 shows the proportion of kept and pruned edges per layer. In this dataset (cf. 4.1), most layers have a pruning ratio $> 50\%$. This means that most of the edges were not statistically significant.

Figure 5.6 shows the distribution of the questionnaires scores across each subgroup/community for dataset 4.1.

All subgroups comprise a majority of male patients, but there are differences in the relative proportion of female patients. For instance, subgroup $C5$ has a higher-than-average proportion of female patients. Looking at the bar plots, we can conclude that $C5$ is also the subgroup with the highest scores among all questionnaires.

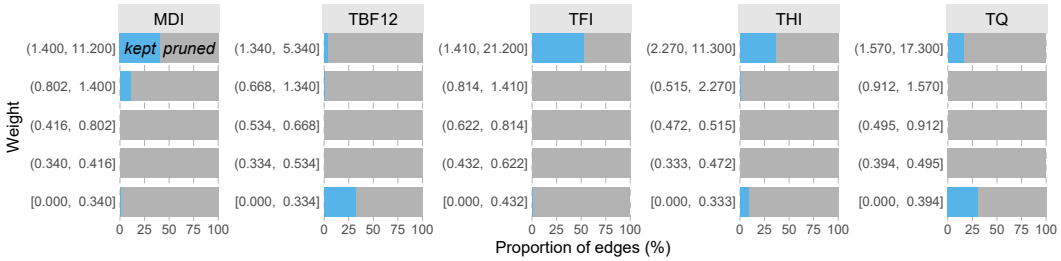


Figure 5.5.: Proportion of "kept" edges per layer and per weight. A set of barplots is illustrated, per layer. For instance, for the MDI layer, in the first barplot, we see that the higher proportion of kept/not pruned edges have a weight between 1.4 and 11.2. In contrast, for the TBF12 layer, the majority of the edges kept are the ones with weight between 0 and 0.334.

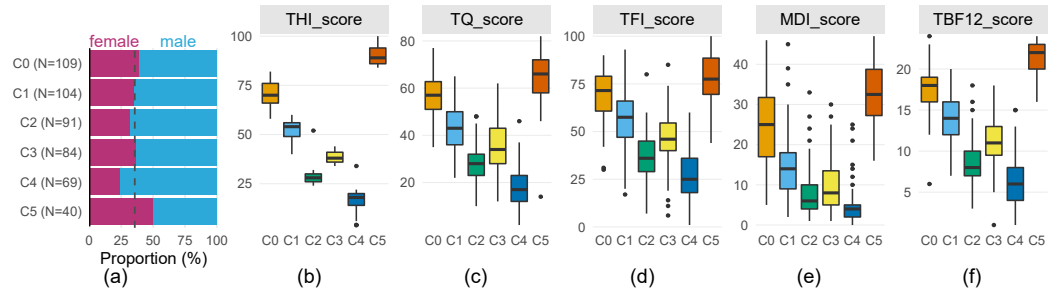


Figure 5.6.: Description of each community per variable. On the leftmost part of the figure, each horizontal bar represents each community/subgroup and the pink part of the barplot denotes the proportion of female patients and the blue part denotes the proportion of male patients. For each questionnaire, we represent the distribution of that questionnaire per community using boxplots, each one with a color that represents the correspondent subgroup.

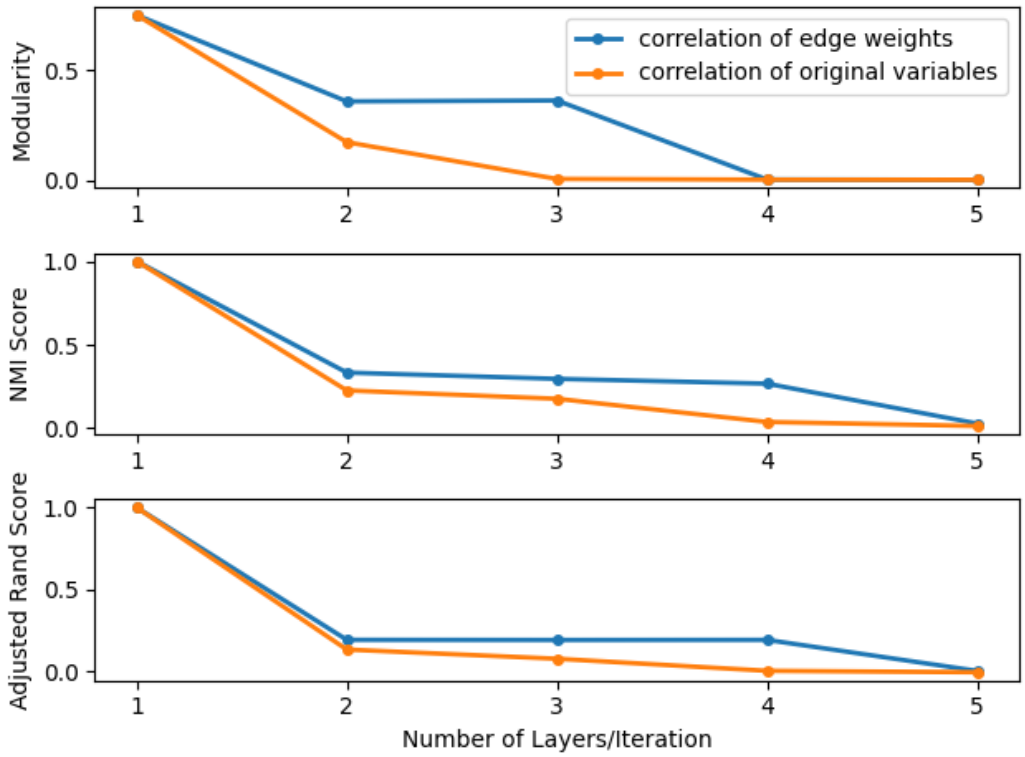


Figure 5.7.: Evolution of Modularity, NMI and Adjusted Rand Score values as layers are incrementally added. There are 5 iterations, because there are 5 possible layers to be added.

Evolution of Modularity as Layers are Added

Figure 5.7 shows the results from the COBALT using both types of correlation mentioned in 5.4.1. The graph shows the modularity score over the 5 iterations, for both cases where the NMI score is used and the Adjusted Rand score is used for computing the layer similarity/layer cost. From Figure 5.7, we can understand that modularity decreases or keeps the same value with the iterations when using the correlation of edge weights as a criterion to compute layer cost. The NMI and the adjusted rand scores decrease more when using the correlation of original values for the cost model. However, we can see that by iteration 4, both models achieve a partition into subgroups that have the same Q. Regarding modularity, using the correlation of edge weights is a better choice when comparing subgroup structures. This conclusion is used as an insight for the next version that is introduced and the later works.

Evolution of Communities as Layers are Added

The figures presented below represent the graphs of each layer of each iteration. At the first iteration, only one layer is selected, which for the dataset 4.1 was the layer that represents the THI questionnaire of all patients. Figure 5.8 illustrates this. We see that nodes colored with the same color are placed next to each other. This means that the nodes that belong to the same subgroup have a higher similarity (strong connection/higher edge weight) according to their THI score similarity. For example, nodes colored in pink are mostly placed next to one another, with the expectation of

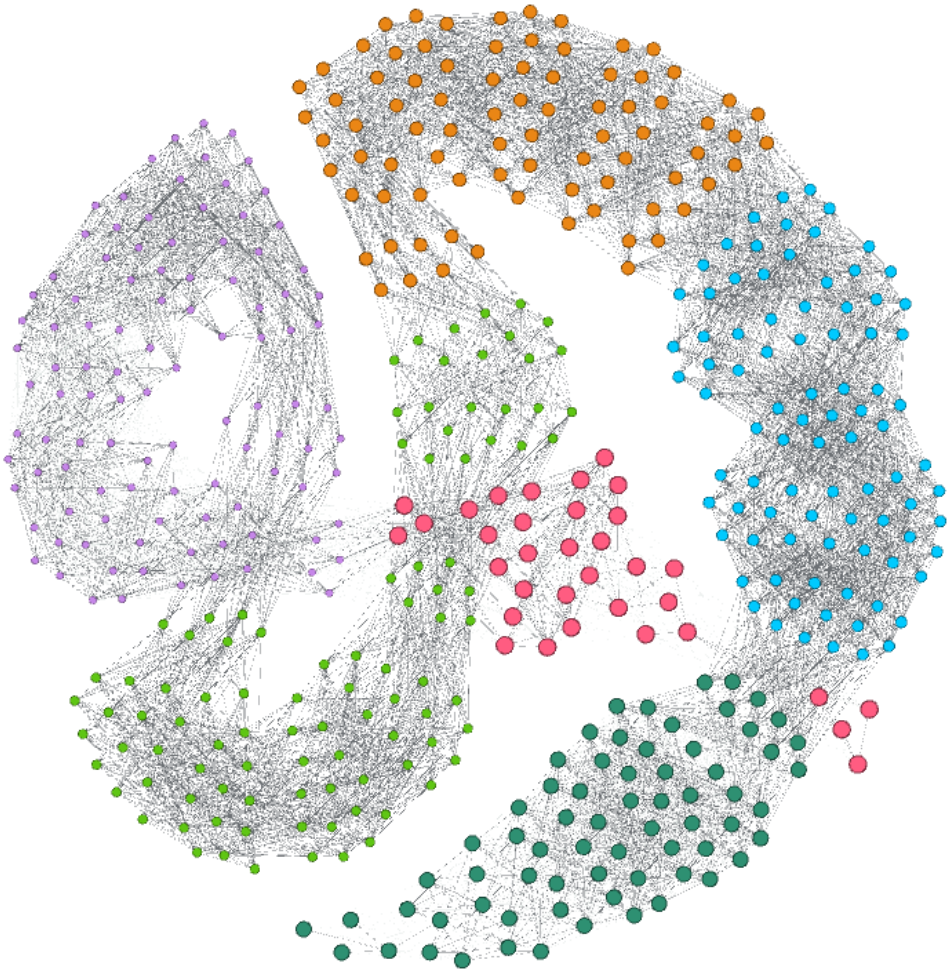
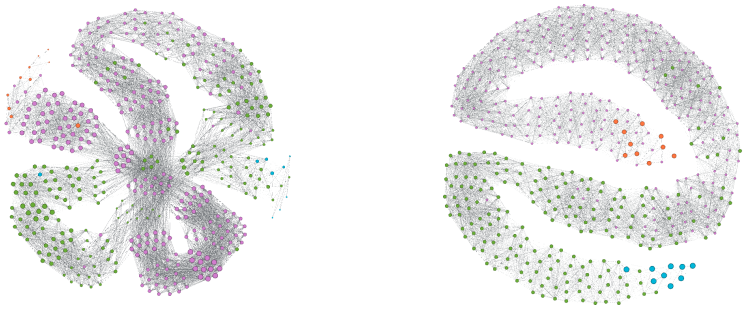
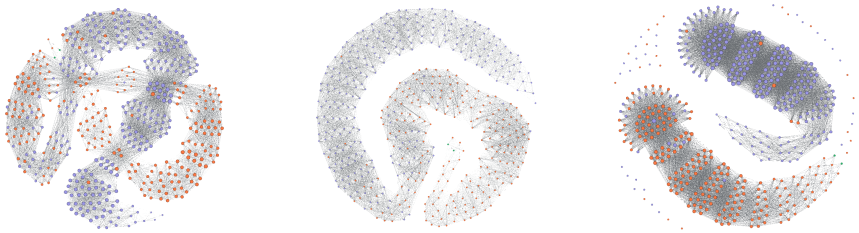


Figure 5.8.: Layer that displays the THI score of patients with detected communities at iteration 1 with modularity value of $Q = 0.752$. The nodes are represented by colored dots. The colors represent the community to which the node belongs to. The edges are represented by gray links.



(a) THI graph (b) TQ graph

Figure 5.9.: THI and TQ layers with detected communities at iteration 2 with modularity value of $Q = 0.358$. The nodes are represented by colored dots. The colors represent the community to which the node belongs to. The edges are represented by gray links.

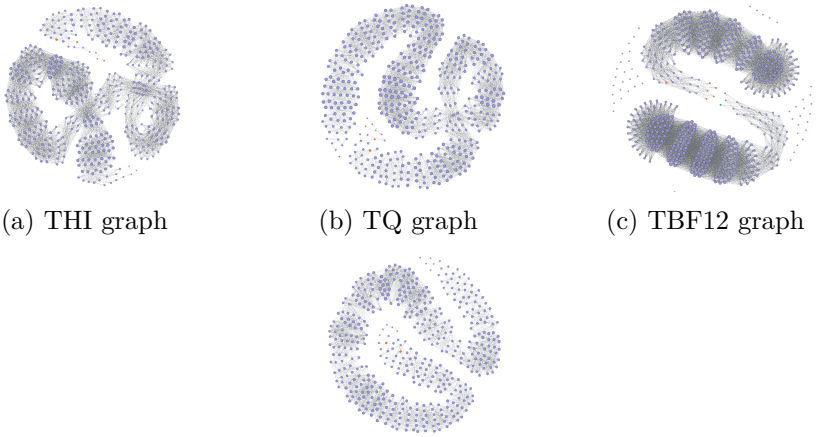


(a) THI graph (b) TQ graph (c) TBF12 graph

Figure 5.10.: THI, TQ and TBF12 layers with detected communities at iteration 3 with modularity value of $Q = 0.362$. The nodes are represented by colored dots. The colors represent the community to which the node belongs to. The edges are represented by gray links.

4 nodes that appear in the border between the green and the blue subgroups. The green and the blue subgroups are better connected according to their THI score since they group together nodes that are closer to each other. Figure 5.9 shows the second iteration, in which the TQ layer is added. In this iteration, it is noticeable that the nodes are more dispersed, and nodes with different colors are sometimes placed in nearby areas of the graph. This happens because the algorithm is now not using simply one layer but two, trying to separate the nodes according to both layers. In a 2D graph, it is hard to visualize the partition in these cases. For example, the violet nodes in the middle of Figure 5.9(a) are very similar, regarding their THI score, to some of the green nodes since they are connected by a strong edge weight. However, according to the TQ score, they are not that similar, and that makes the violet nodes in 5.9(a) appear next to the green nodes. Looking at Figure 5.9(b), the colors are well separated, leading us to believe that the algorithm had to allocate the nodes in Figure 5.9(a) like that because it optimizes the modularity Q.

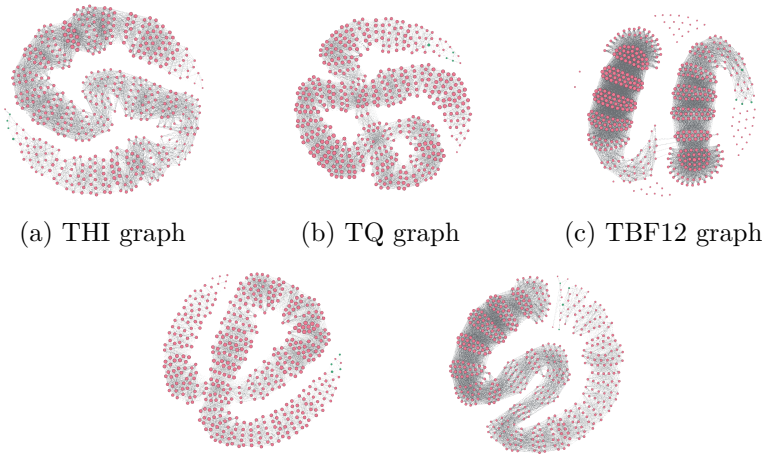
In the next iteration, the TBF12 layer is added. The number of different subgroups detected is again lower: we can observe only the orange and the purple subgroups. Figure 5.11 shows iteration 4 in which TFI layer is added. Figure 5.12 shows iteration 5 with the last layer, with the MDI information. For both these iterations, the modularity Q is close to 0, indicating that there is no underlying structure in the MLN.



(a) THI graph (b) TQ graph (c) TBF12 graph

(d) TFI graph

Figure 5.11.: THI, TQ, TBF12 and TFI layers with detected communities at iteration 4 with modularity value of $Q = 0.002$. The nodes are represented by colored dots. The colors represent the community to which the node belongs to. The edges are represented by gray links.



(a) THI graph (b) TQ graph (c) TBF12 graph

(d) TFI graph (e) MDI graph

Figure 5.12.: THI, TQ, TBF12, TFI, and MDI layers with detected communities at iteration 5, with a modularity value of $Q = 0.001$. The nodes are represented by colored dots. The colors represent the community to which the node belongs to. The edges are represented by gray links.

Layer	N	C	Model	MAE	MSE	R^2
THI	17	1	Lasso	9.7	110.0	**0.183**
THI	17	2	Ridge	15.2	274.5	0.034
TQ, THI	28	1	Lasso	13.2	280.5	**0.189**
TQ, THI	28	0	Lasso	13.1	245.0	-0.152
THI, TQ, TBF12	32	0	Ridge	11.1	197.7	-0.115
THI, TQ, TBF12, TFI	57	0	Ridge	19.3	535.3	0.049
THI, TFI, TQ, TBF12, MDI	57	0	Lasso	19.7	573.7	0.049
Baseline	247	All	Lasso	9.1	529.9	-0.027

Table 5.2.: Performance of the prediction of TQ score after treatment with and without subgroup information. We train different models with subgroups detected using different layers. The column "Layers" denotes the layer(s) of the MLN used to detect afterward the subgroups, which here we call communities, or "C". "Baseline" corresponds to a model that does not use subgroup information."N" denotes the number of data points and "C" the community identifier.

Subgroups for Prediction

We present in table 5.2 the performance results of a regression model trained to predict TQ score at the final visit.

The results are presented per community, i.e., per subgroup. For the network at iteration 1, with only the THI, we present the results for two communities, $C = 1$ and $C = 2$. The others could not be used because there are not enough data points to train a regression model using cross-validation. For the MLN with layers TQ and THI, we report only communities 0 and 1 for the same reason.

The "Baseline" prediction model achieves relatively poor performance when compared with the regression models that use subgroup information. The regression model that has the highest R^2 is trained with subgroup information discovered by detecting communities in an MLN with the TQ and THI layers, which is trained only using the patients allocated in subgroup 1 ($C = 1$). The separation between subgroup 1 and the other subgroups makes the model achieve a higher performance, which means that these patients are easier to predict, according to the used variables than the ones in the other subgroups.

5.7.2. Full-fledged Cost Model

In this subsection, we present the results for the full-fledged cost model, and we use the same dataset 4.1.

Layer Selector Results

Figure 5.13 shows the evolution of the modularity Q per iteration using COBALT as in [90], as well as the layer cost and availability ratio of the selected layer at that iteration. Note that there are 5 iterations because there are five possible layers to be added.

Q starts around 0.75 in the first iteration and then decreases until the last layer is added, in which the Q is 0.6. This evolution is expected since we start by optimizing by Q when the first layer added is the one with the highest Q. Therefore, all other

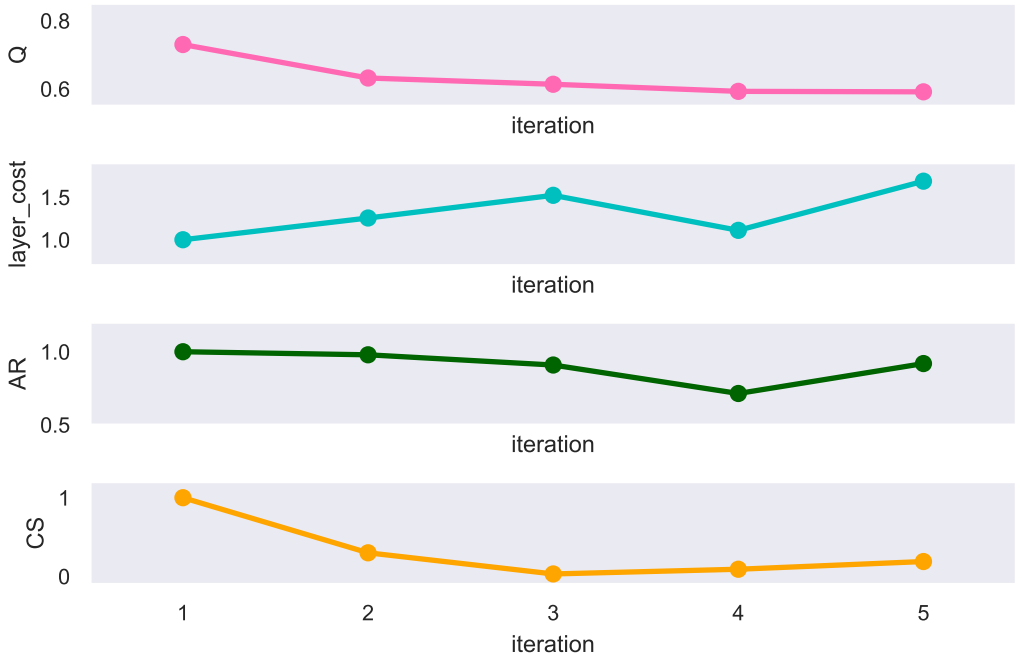


Figure 5.13.: Evolution of modularity (uppermost subfigure), cost (middle upper subfigure), availability ratio (middle lower subfigure) and community similarity (lowermost subfigure), computed as layers are added by COBALT, one at a time.

layers that are added have, as single layers, a lower Q. Even though there is a drop in the Q value, the values are still above 0.55, which means that there are underlying structures in the MLN.

Layer cost has a decrease at iteration 4, which points to the fact that, at iteration 4, the layer that is added has one of the following properties:

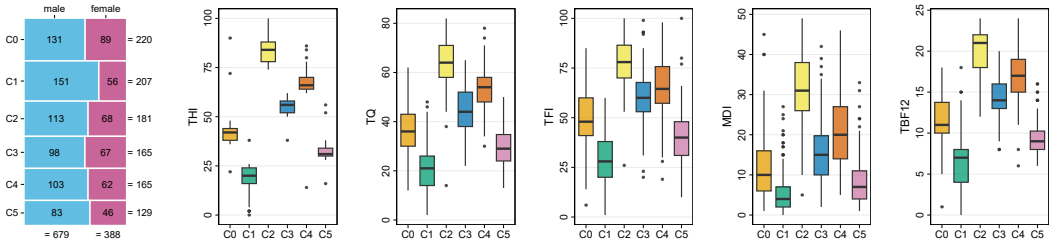
- Higher data availability than the layer at iteration 3
- Lower subgroup similarity, CS, with the MLN at iteration 3
- the sum of the inverse of AR and CS is lower than the one at iteration 3

From the CS yellow line plot, we observe that the cost component of AR in iteration 4 is more than in iteration 3, while the CS component is similar. Therefore, the data availability of the layer added at iteration 4 is higher than that added at iteration 3. One question remains, though, why layer cost decreased, which can be understood by understanding the equation in 5.8, in which we compare the nodes of the layer to be added with the last selected layer. It is then possible that the AR cost decreases because it depends on the layer with the lowest number of nodes.

The stopping criteria *SC1* could be activated, for example, when CS increases again (iteration 4).

Figure 5.14 depicts the distribution of the questionnaire scores for each of the 6 communities of this 1st iteration of COBALT.

The first column of Figure 5.14 shows the value distribution for sex, with one row per community. We present the data in a tabular format, with one column representing each questionnaire and one boxplot for each community. Our analysis



(a) sex (b) THI (c) TQ (d) TFI (e) MDI (f) TBF12

Figure 5.14.: The 6 communities of the THI network added by COBALT as 1st layer, depicting the value distributions inside each questionnaire.

indicates that the communities are quite consistent in their responses to the THI questionnaire, as evidenced by the small boxes indicating low variance. However, we observed an increase in variance for the other questionnaires, which is not surprising as COBALT used only the THI data to create the communities.

There are notable differences among some of the communities. Community C1, represented by green boxes in the questionnaires, has the lowest average score in each questionnaire, while community C2, represented by faded yellow boxes, has the highest average score. This suggests that C1 and C2 have significantly different characteristics. Based on the data used, it appears that C1 groups patients with mild tinnitus symptoms, whereas C2 groups those with severe tinnitus symptoms. Furthermore, there are differences in gender between the communities, with C1 having a lower proportion of female patients.

Evolution of Communities as Layers are Added

Figures 5.15 to 5.19 show the subgroups as COBALT adds layers. The visualization is two-dimensional, as the one presented for the "simplified cost model version". Each subfigure illustrates a layer of the network.

Figure 5.15 depicts the 6 communities in the THI layer, which is chosen by COBALT in the first iteration. The colors of the nodes are well-separated, which means that same-subgroup nodes have similar THI scores, while nodes in different subgroups can be distinguished by their different THI scores.

There are two exceptions, which concern the blue and green communities and the purple and pink communities. There are areas of the graph where there is a mix between green and blue nodes and purple and pink nodes. However, this is minor in comparison with the total number of nodes.

Figure 5.16 illustrates the 5 communities discovered in the 2nd iteration, where COBALT adds the MDI layer with the lowest cost. The COBALT algorithm identifies communities by analyzing data from both layers, which results in a different community structure when compared to iteration 1. In iteration 2, the MDI layer is chosen to be added as it has the lowest cost. In Figure 5.13, the evolution curves show that iteration 2 caused a cost increase and a slight decrease in modularity. This would have stopped the iterative procedure of COBALT in a real setting. The visualization of the communities in the THI and MDI networks makes this evident. There is a clear separation of colors in the THI network, while the colors in the MDI network are mixed. This indicates that the use of the inter-layer similarities and the similarities inside the MDI network did not contribute to a good modularity score.

In Figure 5.17, the 4 communities found in the third iteration are shown, where

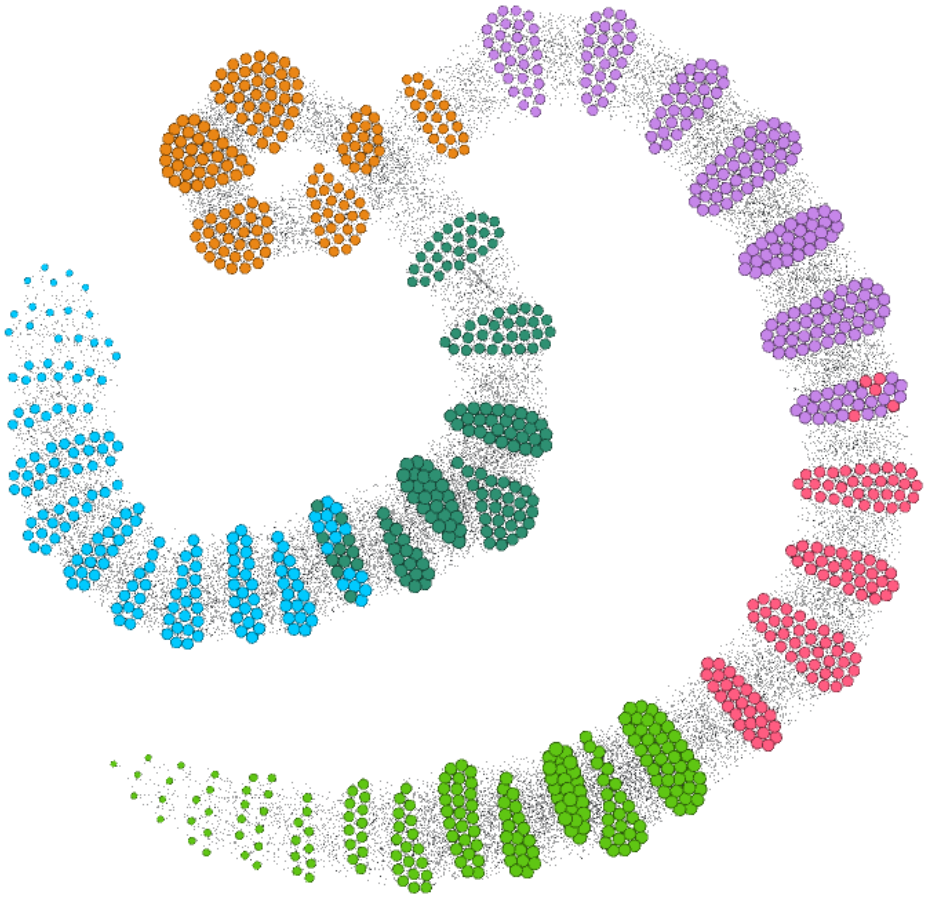
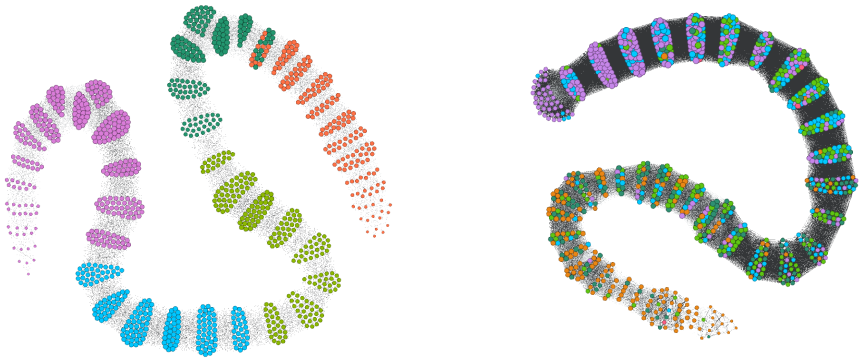


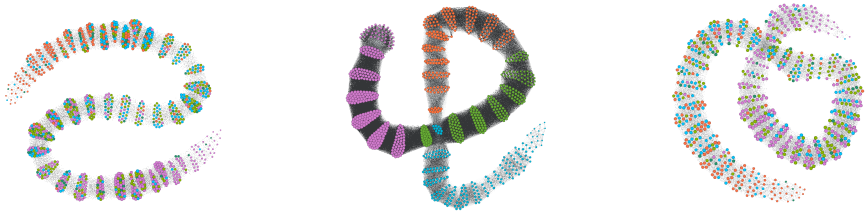
Figure 5.15.: First iteration: 6 communities on the THI at t_0 layer . The modularity value is of $Q = 0.730$. Different colors represent different communities.



(a) THI layer (b) MDI layer

Figure 5.16.: 2nd iteration with MDI questionnaire at t_0 as second layer: 5 communities detected. The modularity value is of $Q = 0.632$.

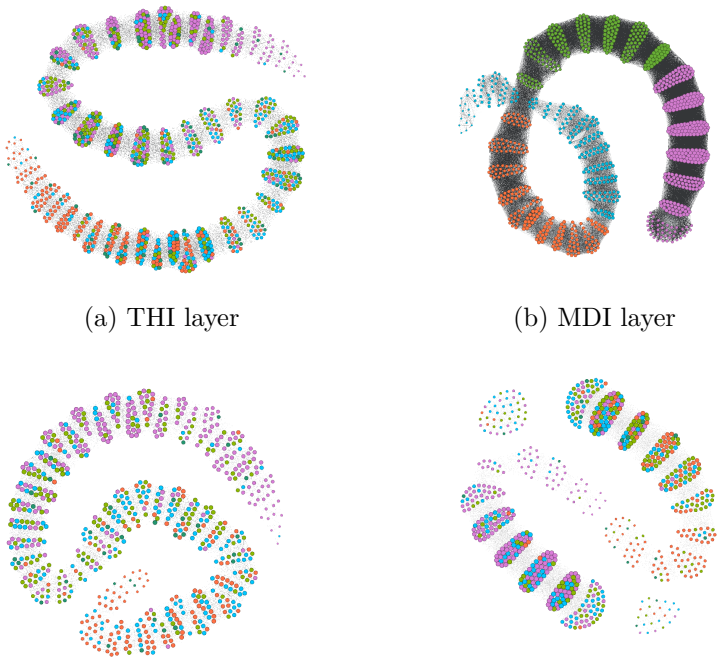
COBALT added the TQ layer. From this iteration, community induction is driven by the node similarities inside the MDI layer, leading to more homogeneous communities in the MDI layer. In contrast, the community colors in the other layers are mixed.



(a) THI layer (b) MDI layer (c) TQ layer

Figure 5.17.: 3rd iteration with TQ questionnaire at t_0 added as 3rd layer: 4 communities detected. The modularity value is of $Q = 0.613$.

Figure 5.17 reveals a noteworthy observation: the MDI network has a high density, which indicates that the patients are quite similar to each other within this layer and could lead to communities that are not clearly separated. This could explain why the graph pruning step had a smaller effect when compared to other layers. In contrast, the partition in Figure 5.18, which shows the 4 communities found when COBALT adds TBF12 as the 4th layer, appears to be well separated with respect to the MDI layer. On the other hand, the colors in the other two layers appear dispersed in many areas of the networks. Similar to the third iteration, the nodes are still well-separated



(a) THI layer (b) MDI layer

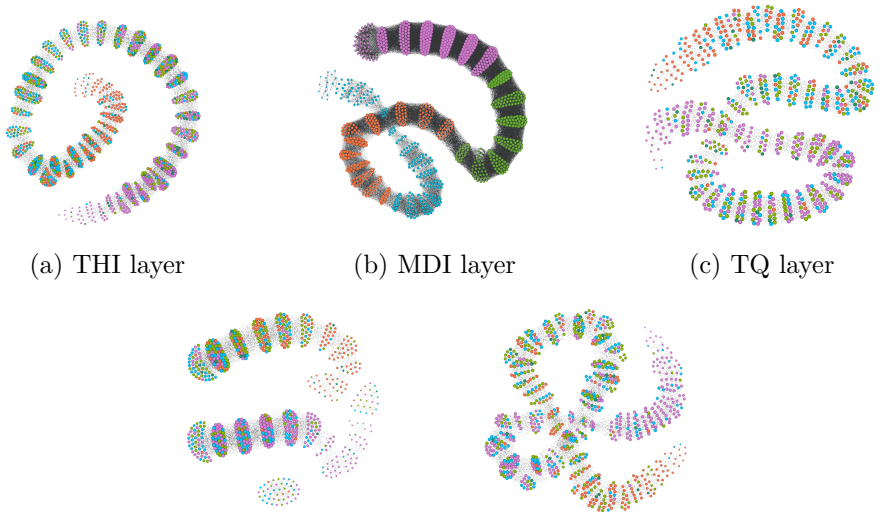
(c) TQ layer (d) TBF12 layer

Figure 5.18.: 4th iteration with TBF12 questionnaire at t_0 added as 4th layer: 4 communities detected. The modularity value is of $Q = 0.592$.

into communities based on the MDI layer but not in the other layers. The number of communities remains the same as before, although this does not guarantee that the communities are exactly the same. The addition of the TBF12 layer did not have a significant impact on the community structure detected. However, this does not

mean that the communities are identical to those in iteration 3. Our primary goal when using a cost-based model is to keep the layers that provide the least additional information for the final iterations. The results of the fourth iteration suggest that adding the TBF12 layer does not contribute much relevant information to the previous MLN.

Finally, Figure 5.19 shows the 4 communities found in the 5th iteration, when the TFI layer is added as the last one.



(a) THI layer (b) MDI layer (c) TQ layer

(d) TBF12 layer (e) TFI layer

Figure 5.19.: 5th iteration, last layer (TFI questionnaire at t_0) added: 4 communities detected. The modularity value is of $Q = 0.591$.

The MDI network shows clear separation between communities, while the other networks do not. The communities discovered using the 5 layers are displayed, with the TFI layer being the last and most expensive addition. The algorithm identified 4 communities, which are well separated in the MDI network. The modularity value of this partition is 0.591, which is very close to the modularity value of the 4th iteration (0.592).

In summary, the visualization scheme shows a deterioration of community quality as layers are added. The evolution of layer cost and its two factors capture this deterioration well; the *SC1* stopping criterion would have stopped the MLN expansion after the 3rd iteration by the latest.

In our scenario, communities are phenotypes for prediction. Here we report their contribution to predictive quality.

Subgroups for Prediction

As per our evaluation design, we compare the quality of predictions achieved using COBALT communities to that achieved by the "Baseline" regression model and traditional clustering algorithms instead of COBALT.

We evaluate the extent to which we can predict a post-treatment score for a layer without using the pre-treatment data of that layer. To achieve this, we use the communities learned over the layers l_1 (first iteration) and l_2 (second iteration) as input and the post-treatment score (at t_1) of each layer as a target.

For instance, suppose that the target is the post-treatment score of question-

Questionnaire	Total	Training set	Test set
THI	123	86	37
MDI	109	76	33
TQ	70	49	21
TBF12	35	24	11
TFI	87	61	26

Table 5.3.: Number of data points in the train and test set size used, per questionnaire score, for the prediction of the treatment outcome (post-treatment questionnaire score).

naire/layer l, and the inputs are the communities learned over the layers l_1 and l_2, where $l \notin \{l_1, l_2\}$. Suppose the prediction quality is high in the evaluation setting for a given phenotype. In that case, we expect that we can predict the score of l at t_1 without recording layer l for this phenotype at all.

In contrast to the simplified cost model, we expand the experiment by incorporating several target variables instead of solely relying on the TQ score. Consequently, this experiment is more comprehensive and aims to determine which questionnaires are easier to predict post-treatment scores for. Table 5.3 displays the training and testing sizes of each questionnaire prediction task.

We present our findings in Table 5.4, and will now explain them in detail. For each of the 5 scores at t_1, we have provided the performance in terms of MSE and MAE (where lower values are better), as well as the explained variance R^2 (where higher values are better). The best values for predicting a questionnaire score are highlighted in **boldface**.

To evaluate how much the subgroups identified by COBALT contribute to phenotype-sensitive treatment, we created a Baseline' group of regression models. Each model predicts the score of a given questionnaire (or layer) at t_1. For each patient, the models use the following features: age, gender, and their score for that questionnaire at t_0.

In our study, we perform a comparison of "Baseline" models with the regression models that incorporate community information. The regression models comprise the ID of the patient's community for each patient/layer node. This information proved to be particularly useful as COBALT adds layers, causing a patient's community to change. To account for this, we train a regression model for each iteration using community-augmented data. By doing so, we are able to analyze the impact of community information on the accuracy of the models. This approach allows us to evaluate the effectiveness of incorporating community data into the regression models.

In Table 5.4, the upper section displays the quality of the phenotypes in the MLN network for each COBALT iteration in the first column. The quality is measured by modularity Q in the third column. The best values achieved in a COBALT iteration are underlined for each predicted score. The COBALT-augmented regression model outperforms the corresponding "Baseline" regression model for each score, as seen by the underlined values. When observing the phenotypes of each iteration, it is clear that the COBALT-augmented regression model consistently performs better.

- **iteration 1:** According to the evaluation results, the performance of COBALT is better than the Baseline when it comes to measuring the THI score, MDI, and TBF12 scores. The MAE values for these measures are identical for both

	Partition quality		THI at t_1 (n=123)			MDI at t_1 (n=109)			TQ at t_1 (n=70)			TBF12 at t_1 (n=35)			TFI at t_1 (n=87)		
Model	silhouette	Q	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE	R^2
Baseline			13.1	287.6	0.569	5.1	43.0	0.686	9.9	159.9	0.394	3.1	13.2	0.182	10.2	158.4	0.637
COBALT																	
iteration 1		0.730	9.3	130.9	0.720	4.9	43.0	0.707	11.3	280.3	0.284	2.8	13.2	0.613	11.1	346.4	0.344
iteration 2		0.632	12.3	228.2	0.439	5.4	62.6	0.532	9.6	143.4	0.521	2.4	9.8	0.798	10.1	148.2	0.730
iteration 3		0.613	9.6	154.6	0.563	6.2	75.4	0.227	9.2	123.7	0.573	2.1	5.9	0.866	13.1	294.2	0.354
iteration 4		0.592	13.0	265.8	0.606	4.4	36.9	0.715	7.6	93.9	0.622	2.5	8.7	0.738	11.2	177.9	0.621
iteration 5		0.591	8.6	103.5	0.748	5.9	68.7	0.302	11.6	227.1	0.379	3.2	13.4	0.595	14.9	356.8	0.359
Clusterers																	
AHC	0.479		9.9	198.9	0.565	8.8	116.8	0.203	8.2	110.2	0.355	2.1	6.4	0.765	14.8	389.3	−0.085
BIRCH	0.482		10.1	183.1	0.576	5.3	53.6	0.599	6.7	61.5	0.455	3.8	15.7	0.671	11.7	325.6	0.108
GMM_EM	0.349		8.4	138.5	0.564	6.8	82.1	0.545	5.6	40.4	0.758	3.4	12.7	0.752	12.3	213.1	0.557
HDBSCAN	0.134		11.7	216.5	0.738	5.6	59.2	0.572	10.6	195.0	0.520	3.0	9.4	0.318	16.7	533.9	0.095
k-means	0.481		8.2	107.2	0.883	5.4	53.9	0.370	9.1	140.4	0.735	2.5	8.9	0.777	16.0	415.3	0.309
OPTICS	0.406		9.1	148.4	0.650	3.8	21.2	0.704	10.4	166.6	0.458	4.1	20.2	−4.564	10.7	184.4	0.410

Table 5.4.: Prediction of questionnaire scores at t_1 given age, gender, score at t_0 (Baseline, first row of values) and, additionally, the phenotype IDs returned by COBALT (upper part), respectively by the clustering algorithms (lower part); for convenience, we ordered the columns with the same order as layers were added by COBALT, but this ordering has no effect on the way the data are read by the predictor. 'n' represents the number of patients used for this evaluation. Note that 'n' is then partitioned into two: train and holdout sets.

COBALT and the Baseline. However, the Baseline outperforms COBALT when it comes to measuring the TQ and TFI scores. Overall, the evaluation results suggest that COBALT is a more reliable and accurate tool for measuring certain aspects of the performance, while the Baseline performs better in other areas.

- **iteration 2:** COBALT performs better than the Baseline for TQ, TFI, and TBF12 scores. Regarding the THI score, COBALT outperforms the Baseline in terms of MSE and MAE. The results show that for the MDI score, the Baseline outperforms COBALT. This is surprising because the MDI layer was included in the 2nd iteration. However, upon examining the communities in Figure 5.16, it is evident that they are more focused on THI.

- **iteration 3:** COBALT outperforms the Baseline for the TQ and TBF12 scores and for the THI score with respect to MSE and MAE. With respect to the MDI score and the TFI score, the Baseline is superior.

- **iteration 4:** COBALT outperforms the "Baseline" model for all scores except TFI.

- **iteration 5:** COBALT outperforms the "Baseline" model for the THI score, but it is inferior to it for all other questionnaire scores.

To summarize, the COBALT phenotypes on the THI and MDI layers are sufficient to predict 4 out of 5 scores at t_1 with better Mean Absolute Error (MAE) and Mean Squared Error (MSE) than the "Baseline" that uses the scores of all 5 questionnaires at t_0. The THI layer alone is enough to predict 3 out of 5 scores at t_1. This shows that using phenotypes during prediction is advantageous, and the advantage is even greater when adding the least-cost layer.

The influence of the MDI layer may be considered an artifact because this layer improves predictive performance but not for the MDI score itself. This could be due to the density of the layer, which may have resulted in poor-quality communities inside it.

Table 5.4 shows the performance of clustering algorithms in constructing phenotypes for prediction instead of COBALT on MLNs. We modified the number of clusters to optimize silhouette and reported the clusters found for that optimal number. For instance, for k-means, the best silhouette was found for $k = 2$.

At least one clustering algorithm outperformed the Baseline model for each of the five scores. This suggests that taking advantage of subgroups during prediction is beneficial, similar to the COBALT approach. However, unlike COBALT, no clustering algorithm was significantly better than the others. For the THI score, all algorithms delivered models superior to the Baseline. For the TFI score, none of them did. For the remaining scores, some models were better than the Baseline, while others were worse.

Regarding the prediction of the TFI score (iteration 2) and the TBF12 score (iteration 3), COBALT outperformed the clustering approaches. However, this was before the corresponding layers were added. On the other hand, *ML* outperformed all clustering algorithms while predicting the MDI score, TFI score, and TBF12 score.

For the MDI score, iteration 4 delivers the best R^2 value, but OPTICS is superior to *MLN* with respect to MSE and MAE. For the TFI score, the 2-layer network provides the best results, and for the TBF12 score, it is the 3-layer network.

When it comes to the THI score, K-Means returns the best results. However, there is no clear winner among the clustering approaches with respect to phenotype

	Partition quality	THI score at t_1					MDI score at t_1					TQ score at t_1				
	Q (best)	n	NL	MSE	MAE	R^2	n	NL	MSE	MAE	R^2	n	NL	MSE	MAE	R^2
Ratio of node missingness																
0%	0.740	48	1	8.0	102.3	0.850	47	1	7.1	90.0	0.601	29	1	7.5	88.6	0.703
10%	0.741	48	4	9.1	149.0	0.789	47	4	5.5	45.4	0.574	29	4	9.8	142.5	0.635
20%	0.742	48	4	7.9	89.9	0.847										

Table 5.5.: Prediction quality for each percentage value of missingness, showing the iteration (equiv. number of layers NL) that achieves the best modularity. Number of patients used for training (n) is different for each questionnaire, since only patients that filled the questionnaire represented in each column at t_1 are considered if and only if they files all questionnaires at t_0.
Note that 'n' is then partitioned into two: train and holdout sets (cf. Figure 5.3 and subsection 5.6.3)).

contribution. For each of the 5 scores, another algorithm is best for one or more of the three measures. Unlike COBALT, all clustering algorithms are trained on all scores at t_0. In contrast, the phenotypes returned by COBALT on the first two layers outperform the baseline for all scores except the MDI.

The comparison of COBALT and clustering based on community homogeneity is not feasible. Therefore, we resort to using silhouette instead of modularity for clustering purposes. Our analysis reveals that BIRCH produces subgroups with the highest silhouette value among different clustering algorithms. However, these subgroups have limited potential for prediction compared to those generated by other clustering algorithms. This observation aligns with our findings on modularity for COBALT: although the subgroups obtained at iteration 1 have the highest modularity, the regression models utilizing them are of inferior quality.

To provide more detailed insight, the COBALT method has been found to produce more consistent and reliable predictive results when compared to individual clustering algorithms. Despite the latter method using all scores at t_0, the results are not as accurate as those generated by COBALT. This is particularly evident in iteration 2, where a layer is selected in a cost-sensitive manner before the stopping criterion SC1 is triggered. It is interesting to note that the quality of the subgroups, such as the modularity in MLNs or the silhouettes in clustering, is not a reliable indicator of predictive performance. However, using a cost-sensitive approach to create subgroups with COBALT leads to highly competitive predictive performance. Therefore, the COBALT method is recommended for generating subgroups with superior predictive performance.

Impact of missingness

In Figure 5.20, we can observe the trend in modularity as data missingness increases from 10% to 90%. This figure provides information about the changes in modularity values across different iterations. The top-left subfigure corresponds to the initial iteration, while the bottom-right subfigure corresponds to the final iteration. It is important to note that each subfigure displays a dashed line, which represents the modularity value for data with 0% missingness. By analyzing this figure, one can gain a deeper understanding of the relationship between data missingness and modularity values, and how this relationship changes over time.

The plotted curves suggest that the modularity of the MLN is not significantly impacted by missing data. The curves always decline towards 0.4 and remain close to the reference line of 0% missing data. Although sometimes the curves fall below

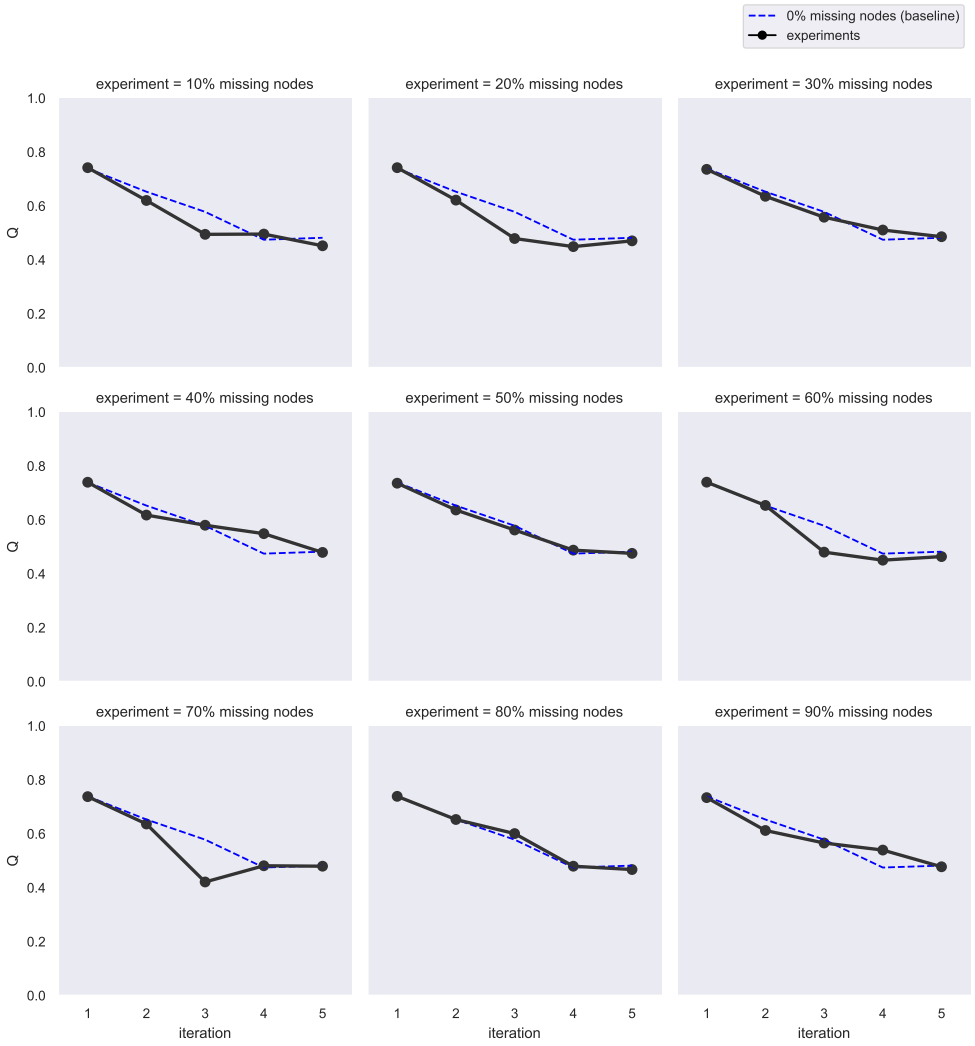


Figure 5.20.: Modularity per iteration as missingness ratio is increased from 10% to 90%.

the reference line, they are mostly above it.

The findings of the study suggest that specific subsets of nodes within the MLN have intra/inter-layer edges and weights that create communities of similar quality to the ones that do not have any missing data. This indicates that the missing data in the MLN can be predicted accurately to a certain extent.

The prediction quality table (Table 5.5) demonstrates that COBALT was only able to accurately predict THI, MDI, and TQ scores at time point 1 when the missing data was 0% and 10%. However, when the missing data was increased to 20%, only THI could be predicted with sufficient accuracy. Unfortunately, in this dataset there was not enough data to train and test for the TFI and TBF12 questionnaires.

To select the best iteration, we choose the one with the highest modularity for each missing data value. This value is then presented in the "NL" column, which indicates the number of layers used. The number of patients used for training varied depending on the questionnaire score being predicted and is shown in the "n" column. Overall, these findings provide useful insights into the predictability of the missing data in the MLN and suggest that specific subsets of nodes can be used to create accurate predictions.

As shown in Table 5.4, decreasing modularity does not necessarily indicate a decline in prediction quality measures. This is supported by the results presented in Table 5.5, where sets of communities with the best modularity were selected, but prediction quality measures varied in both directions. Overall, the prediction quality is good, indicating that a small increase in missingness does not lead to a significant deterioration of the quality.

When the number of missing nodes increased from 0% to 10%, the THI score prediction quality dropped, although not by a significant margin, based on R^2. However, it increased again when node missingness was increased to 20%. For MDI and TQ questionnaire score predictions, prediction quality decreased with node missingness. However, the amount of training data was so small that generalization was not possible.

5.8. Discussion

The results mentioned above raise the question of whether modularity is equally important as prediction quality. Although modularity maximization is not an end goal, creating well-connected communities has contributed to better prediction quality. Communities play a crucial role in the cost-aware selection of questionnaires, thereby reducing the cost and eventually the burden on patients without compromising prediction quality.

Additionally, the impact of missing data on modularity was not significant, indicating that communities can be identified even with a smaller sample size. This points out that the removal of nodes does not significantly impact the relative strength of the edges in the structure for this specific dataset. More specifically, communities that were before very well-connected remain well-connected despite the removal of several nodes. We can say that the system is then "robust" to the removal of nodes. This can be an indication that the system was well represented into an MLN, since some properties of the network are maintained.

However, this only applies to the dataset studied here. Please note that different systems require different representations, and the first step that we emphasized in

the beginning of this thesis - the representation - is the most important to ensure a robust system representation.

Two modalities are presented to define cost, and they are tested in the same dataset. The first one, the simplified cost model version, does not allow for missing data, i.e. only considers patients that have available data for every questionnaire. In the full-fledged cost model version, we allow for missing data. This is noticeable already in the first iteration, as seen in Table 5.6.

	Simplified version	**Full-fledged version**	**Simplified version**	**Full-fledged version**
Iteration number	**Layers**		**Q**	
iteration 1	THI	THI	0.752	0.730
iteration 2	THI, TQ	THI, MDI	0.358	0.632
iteration 3	THI, TQ, TBF12	THI, MDI, TQ	0.362	0.613
iteration 4	THI, TQ, TBF12, TFI	THI, MDI, TQ, TBF12	0.002	0.592
iteration 5	THI, TQ, TBF12, TFI, MDI	THI, MDI, TQ, TBF12, TFI	0.001	0.591

Table 5.6.: Modularity values, per iteration, of the "simplified version" and "Full-fledged version". The column "Layers" denotes the used layers at each iteration. For instance, at iteration 1, both models used the same layer.

In the first iteration, both models choose the same layer (named THI), but the Q values are different. The only explanation for this is the different subset of nodes used since the dataset is the same. As previously stated, the "simplified version" is less complex not only because of the cost model but also because it does not account for missing values. Therefore, at iteration 1 of the "simplified cost model version," we have only nodes that have information about all layers. On the other hand, we have in the "full-fledged cost model version" all the nodes irrespective of their values on the other layers.

Upon comparing the modularity values and selected features of both methods, it can be observed that the order of selected layers/features differs, with the previous work selecting THI, TQ, TBF12, TFI, MDI, and the current work selecting THI, MDI, TQ, TBF12, TFI. The difference in modularity obtained in the first layer can be attributed to the fact that COBALT uses all patient data, whereas [88] only uses patients with available data in all layers. Essentially, COBALT's modularity in the first layer is different due to the inclusion of all patient data, while [88] only includes patients with available data in all layers.

Regarding prediction performance, it is not possible to compare the "simplified version" [88] with the "Full-fledged version" because the former predicted for each community separately and only for the TQ score at t_1, considering only those communities that had sufficient data for learning and testing. The difference in the layer selection order between the "simplified version" and the "Full-fledged version" is a result of a cost-aware approach. If the layer order were the same and missing nodes were allowed, the predictive quality would be the same, given that the detected communities would also agree. Both studies use the Leiden algorithm and the same dataset.

The prediction task conducted in [88] produced a R^2 of 0.183 with data from only one community. In contrast, we achieved a R^2 of 0.720 when predicting THI at t_0 but using communities as a feature in our work. This new prediction task uses data from all communities as features, whereas in [88], the prediction task only uses patient data from one community at a time. As a result, there are fewer data points per regression model.

Nonetheless, we can state that COBALT is superior to the predecessor approach of [88] by design since it allows for missing values.

6. Evol-COBALT for Temporal Data

6.1. Overview

In this chapter, we present Evol-COBALT, which extends COBALT by considering the temporal dimension. Not only does it add the temporal aspect to it, but it also captures the evolution of communities.

Figure 6.1 illustrates where Evol-COBALT stands in the workflow previously introduced.

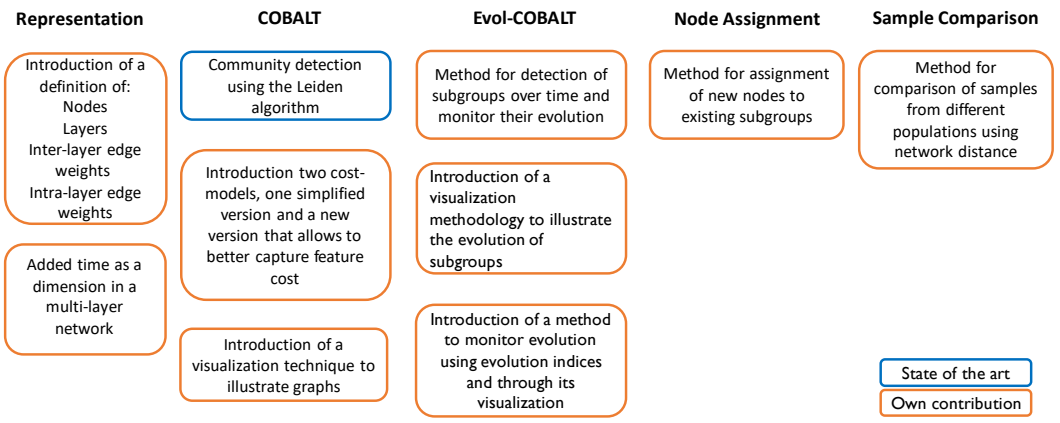


Figure 6.1.: Evol-COBALT: content structure of the approach.

The addition of a temporal aspect extends COBALT to Evol-COBALT. A new step is added for that in block number 5 (block on the rightmost part of Figure 6.1).

In this chapter, part of the work is published in:

- C. Puga, L. Basso, J. Simoes, B. Mazurek, and J. A. Lopez-escamez, "Predicting Treatment Outcome Through Patient Subgroup Evolution - A Multi-Layer Snapshot Network Approach," submitted to the Machine Learning Journal, 2024.
- C. Puga, M. Spiliopoulou, and D. Berron, "MINERVA : Multi-layer Network Subgroup Evolution Tracker To Predict the Cognitive," submitted to JDSA, 2024.

6.2. Representation of Entities in the Temporal Space

We modify the representation presented in subsection 5.2 of the static approach to account for adding the time dimension to the structure.

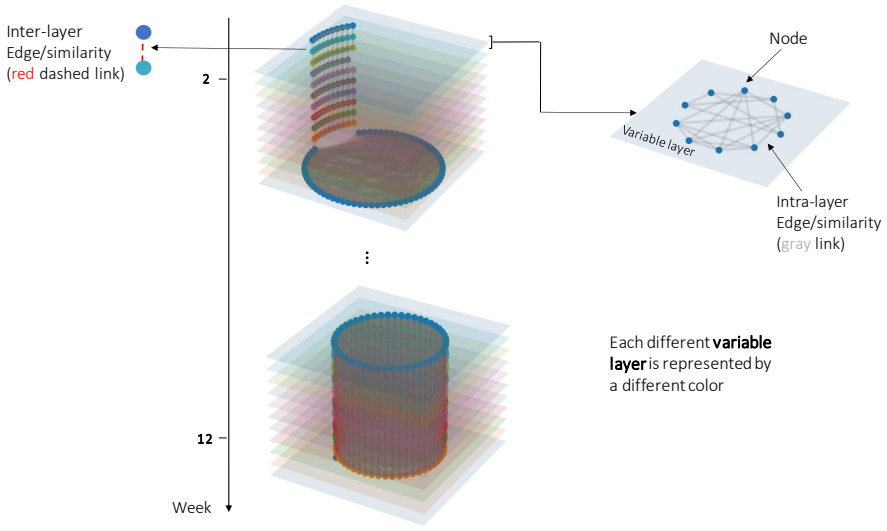


Figure 6.2.: The figure displays an MLSN in a graphical form. The vertical axis represents time, measured in weeks. Each week is represented by a separate MLSN in 3D format. Each MLSN has a unique set of colored layers that represent different variables that define the nodes. The nodes themselves are shown as spheres on each layer, colored with the same color as their respective layer. If data is available for a node, the similarity between nodes is illustrated with gray edges.

Figure 6.2 illustrates how the representation in COBALT can be incorporated with the time dimension. In the example, the time axis is weeks, and for each week (in Figure 6.2, one for week 2 and one for week 12), there is an MLN that has the data of the nodes for that time point. This mapping type is as a snapshot-MLN. This is because we take a snapshot of the data at each point in time. In summary, we have the following structure:

- *Nodes:* represent an entity, which in our application is a mHealth application user or a patient
- *Feature layers:* each feature layer is a network with nodes and edges that represent a specific feature of the node
- *Time layers:* each time layer is a network with nodes and edges that represent a specific feature of the node
- *Edges:* links between nodes

  Intra-feature-layer edges: weighted, undirected edges that connect two nodes inside the same layer

  Inter-feature-layer edges: weighted, undirected edges that connect two nodes located in different layers

The intra-layer and inter-layer edges that we use are regarding the feature layers since we have no edges across time layers.

Figure 6.3 illustrates more formally the structure of the network.

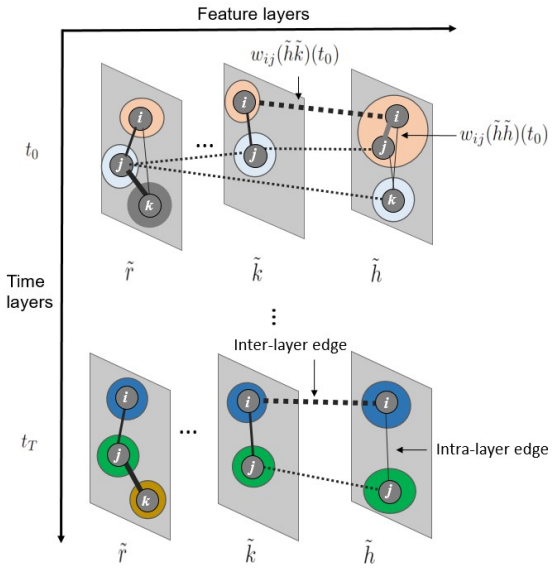


Figure 6.3.: Formal representation of a set of MLSNs with detected communities. The y-axis represents the time layers, and the x-axis represents the feature layers. Each layer has a set of circles surrounding nodes. These circles illustrate the communities/subgroups at each variable and time-layer.

Consider a layer $l_\alpha \in \mathcal{L}$ where $\mathcal{L}$ is the set of feature layers. As in the previous chapter, questionnaire scores are represented by feature layers that measure different aspects of the node numerically. To compute edge weights, we standardize the feature values. We denote by $value_{i,l_\alpha}$ and $value_{j,l_\alpha}$ the feature value of nodes i and j in l_α. The standardization is done by subtracting the mean from each value and dividing it by the standard deviation of the sample.

The larger the difference in values between nodes, the weaker their edge weight, which results in a lower edge weight. The weight of an intra-layer edge between i and j in layer l_α is denoted by $w_ij(l_\alpha, l_\alpha)$ in Equation 6.1.

$$w_{ij}(l_\alpha, l_\alpha) = \frac{1}{|value_{i,l_\alpha} - value_{j,l_\alpha}|} \tag{6.1}$$

Inter-layer edges are computed as in Equation 6.2.

$$w_{ij}(l_\alpha, l_\beta) = \frac{1}{|value_{i,l_\alpha} - value_{i,l_\beta|}} \tag{6.2}$$

where $w_{ij}(l_\alpha, l_\beta)$ denotes an inter-layer edge that connects node i in variable layer l_α and node j in variable layer l_β.

Note that both $w_{ij}(l_\beta, l_\beta)$ and $w_{ij}(l_\alpha, l_\beta)$ represent the similarity between nodes. The similarity should be the maximum when both nodes i and j have the same value for a certain variable layer. Hence, we account for this case to both formulations of the edge weights in equations 6.1 and 6.2.

Figure 6.3 illustrates our proposed representation of the data into a set of MLSNs. Notice that each node belongs only to one subgroup in each time point. For the different time points, subgroups are assigned to the same node. One example is node i, which belongs to the orange subgroup in all layers of t_0, but belongs to the blue subgroup in t_T. Each MLN in each time point corresponds to a snapshot network. Thicker edges indicate greater similarity between nodes.

Symbol	Description	Algorithm
$w_{ij}(l_\alpha, l_\alpha)$	intra-layer edge weight of the edge connecting nodes i in layer l_α and j in layer l_α	
$w_{ij}(l_\alpha, l_\beta)$	inter-layer edge weight of the edge connecting nodes i in layer l_alpha and j in layer l_β	
t	vector of time points	6.1,6.3
M_{t_n}	multi-layer network tensor, at time point t_n it contains the structure of the MLN	6.1
$Q(\mathcal{S}_{t_n})$	modularity of the partition $\mathcal{S}_{t_n}$	6.1
$c_{t_n,i}$	community i of partition $\mathcal{S}_{t_n}$	6.1,6.2, 6.3
$\emptyset(c_{t_n,i})$	conductance of community $c_{t_n,i}$ in partition $\mathcal{S}_{t_n}$	6.1
$I^{\eta}_{c_{t_n,i}}$	shrink index of the community $c_{t_n,i}$	6.2, 6.3
$I^{\psi}_{c_{t_n,i}}$	split index of the community $c_{t_n,i}$	6.2, 6.3
$\psi_{i,j}$	refers to $\psi_{c_{t_n,i},c_{t_n,j}}$	6.2
$\phi_{i,j}$	refers to $\phi_{c_{t_n,i},c_{t_n,j}}$	6.2
$\eta_{i,j}$	refers to $\eta_{c_{t_n,i},c_{t_n,j}}$	6.2
$\mu_{i,j}$	refers to $\mu_{c_{t_n,i},c_{t_n,j}}$	6.2

Table 6.1.: Notation table for Evol-COBALT. We describe briefly the symbol and make reference in which algorithms these symbols are used. The other symbols that are not time-dependent are presented in Table 5.1.

6.3. Graph Pruning

The graph pruning step is similar to the one mentioned for COBALT. The main difference is that we apply the graph pruning for each layer of the MLN of each time point. Therefore, the MLF method is applied for every layer of each MLSN.

6.4. Mathematical Formulation

We run COBALT for each time point, which uses the Leiden algorithm [109] to identify communities. Therefore, we filter the most important layers based on the cost model presented in COBALT.

Algorithm 6.1 Evol-COBALT

Require: M_{t_n}, t
1: $Q \leftarrow$ {}; $\emptyset(\mathcal{S}) \leftarrow$ {}; $\mathcal{S} \leftarrow$ {}
2: **for each** $t_n \in t$ **do**
3: $\quad \mathcal{S}_{t_n} \leftarrow \text{COBALT}(M_{t_n})$ ▷ Get partition
4: $\quad Q_{t_n} \leftarrow Q(\mathcal{S}_{t_n})$ ▷ Get modularity
5: $\quad$ Add $Q(\mathcal{S}_{t_n})$ to Q
6: $\quad$ Add S_{t_n} to $\mathcal{S}$
7: $\quad$ **for each** $c_{t_n,i} \in \mathcal{S}_{t_n}$ **do** ▷ Iterate over communities
8: $\quad\quad$ Compute $\emptyset(c_{t_n,i})$ ▷ Get conductance
9: $\quad\quad$ Add $\emptyset(c_{t_n,i})$ to $\emptyset(\mathcal{S}_{t_n})$
10: **Output:** Q, $\emptyset(\mathcal{S}), \mathcal{S}$

Algorithm 6.1 shows our approach to detect subgroups for each time point $t_n \in T$. After having the representation of the multi-tensors for each t_n, M_{t_n}, we run COBALT for that snapshot multi-layer network. We then compute modularity Q for each subgroup partition and conductance $\emptyset(\mathcal{S}_{t_n})$ for each community $c_{t_n,i}$ in $\mathcal{S}_{t_n}$.

Both modularity and conductance are evaluation metrics used to evaluate the quality of partitions of nodes into groups. However, but conductance is applied at the level of the community/subgroup, whereas modularity is applied at the partition level. This is why we need to compute the conductance for each community in partition $\mathcal{S}_{t_n}$, which is itself a partition of a single snapshot MLN since it refers only to time point t_n.

6.5. Computation of Subgroup Evolution

After detecting the subgroups over time, it is also relevant to monitor how the subgroups change over time. For that, we compute the split and the shrink index per community and timepoint in order to understand how a community changes in size.

Algorithm 6.2 Compute metrics

Require: $t_n, t_{n+1}, \mathcal{S}_{t_n}, \mathcal{S}_{t_{n+1}}$
1: **for** $t_{inc} \in [t_n, t_{n+1}]$ **do**
2: **for each** $c_{t_{inc},i} \in \mathcal{S}_{t_{inc}}$ **do**
3: **if** $t_{inc} = t_n$ **then**
4: `Compute` $\psi_{i,j}$ ▷ migration ratio of nodes from $c_{t_{inc},i}$ to $c_{t_{inc}+1,j}$
5: `Compute` $\phi_{i,j}$ ▷ migration ratio from other communities to $c_{t_{inc}+1,i}$
6: **else**
7: `Compute` η_i
8: `Compute` μ_j
9: `disloyalty index` $\leftarrow \eta_i$
10: `Compute`$I^{\eta}_{c_{t_{inc},i}}, I^{\psi}_{c_{t_{inc},i}}$ ▷ Compute shrink and split indices
11: `Compute` $\hat{\psi}_{i,j}$ ▷ Average of nodes that migrated to other communities in t_{n+1}, across all communities
12: **Output:** $I^{\eta}_{c_{t_n,i}}, I^{\psi}_{c_{t_n,i}}$, `disloyalty index`

Please note that, for the sake of readability, we simplify $\psi_{c_{t_n,i},c_{t_n,j}}$ to $\psi_{i,j}$. The same applies for $\phi_{i,j}$, η_i and μ_j.

We use η_i to compute an index that we introduce as the "disloyalty index". It can be computed as presented in Equation 6.3.

$$\texttt{disloyalty index} = \eta_i \tag{6.3}$$

This index is computed per time point for each subgroup solution.

Consider $t \in [t_0, t_1, ..., t_{tp}]$, which denote the tp time points in the dataset. In 6.3, we explain the process of computation of the metrics necessary for the computation of the shrink and split indices.

6.6. Experiment Design

We test Evol-COBALT in two datasets presented in subsections 4.2 and 4.3. The two datasets are of different natures, which helps to evaluate to what extent our method can apply to completely different tasks.

Algorithm 6.3 Computation of migration matrices

Require: $\mathcal{S}, \mathcal{L}, t$

$t \Rightarrow [t_0, t_1, ..., t_{tp-1}]$ ▷ all timepoints except the last

$n = 0$

for each $t_n, t_{n+1} \in t$ **do** ▷ for each subsequent pair of timepoints

    $I^{\eta}_{c_{t_n,i}}, I^{\psi}_{c_{t_n,i}} \leftarrow$ `compute_metrics`$(t_n, t_{n+1}, \mathcal{S}_{t_n}, \mathcal{S}_{t_{n+1}})$

    $n = n + 1$

Output: $I^{\eta}_{c_{t_n,i}}, I^{\psi}_{c_{t_n,i}}$, `disloyalty index`

6.6.1. Predictiveness of Subgroups

Table 6.2 summarizes the experiments that are performed in 4.2 and 4.3 regarding the predictiveness analysis of subgroups. We train a regression model as in 5.6.3 to predict a target variable.

The target variable differs for the two datasets:

- Target variable dataset 4.2: the TFI score of the patients at the final visit of the treatment (t_2)
- Target variable dataset 4.3: "pct_corr_total," the percentage of correct answers to the cognitive tasks by participants, at a certain time point t.

Dataset	Set	Time	Feature I	Feature II	Domain related feautres	Communities	Clusters
Dataset 2	Baseline		age	sex	t_0	✗	✗
	Evol-COBALT	t_0	age	sex	t_0	t_0	✗
	Clustering		age	sex	t_0	✗	t_0
	Baseline		age	sex	t_1	✗	✗
	Evol-COBALT	t_1	age	sex	t_1	t_1	✗
	Clustering		age	sex	t_1	✗	t_1
Dataset 3	Baseline		todl	*pc_delay*	✗	✗	✗
	Evol-COBALT	t	todl	*pc_delay*	✗	t	✗
	Clustering		todl	*pc_delay*	✗	✗	t

Table 6.2.: The table outlines the feature sets used for prediction-based evaluation. Variables that are included in the feature set are marked with a checkmark (✓), while variables that are not included are marked with a cross (✗). The notation t_0 indicates that the feature value is taken from the pre-treatment visit, t_1 indicates that the feature value corresponds to the value at time point t_1, and t represents any time point in the series.

The "Set" in Table 6.2 refers to the feature sets used to train the regression model. We propose to use three, each with different characteristics:

- *Baseline Set*: it includes feature I, feature II, and domain-specific features (for dataset 4.2, since medical researchers asked for it). It does not contain any information about subgroups (neither from communities nor clusters).
- *Evol-COBALT Set*: it includes the "Baseline Set" plus the subgroups detected by Evol-COBALT
- *Clustering Set*: it includes the "Baseline Set" features plus the subgroups detected using traditional clustering algorithms

Please note that for dataset 4.2, we train two regression models using domain-related features. One model is trained using data from t_0, and the other one is trained using data from t_1. This is because we have three-time points in this dataset. On the other hand, for dataset 4.3, we mention t because we have multiple time points, specifically 16 weeks. In this case, t takes the value of each time point.

6.6.2. Subgroup Evolution Visualization

There are many components of the structure that require interpretation in order to draw conclusions. We developed a visualization strategy so that we could show the results in an intelligible manner.

To begin with, it is crucial to visualize the subgroups at every point in time in order to understand the differences between them. To accomplish this, we opt to create a radial plot with a separate region for each subgroup. These regions are then equipped with a barplot that displays the variables that differentiate that particular subgroup.

It is important to note that variables that characterize subgroups often have varying ranges, which can make it difficult to compare and contrast them. To overcome this challenge, we apply z-score normalization to each variable of each subgroup. This normalization technique ensures that the z-score of a variable of a specific subgroup is 0 if the subgroup's average variable value is equivalent to that of the entire population. A z-score value greater than 0 indicates that the data points of that variable in that subgroup are above the population's average. This normalization technique helps to standardize the variables, making them more comparable across subgroups.

The main objective of these plots is to make it easier to compare the composition of each subgroup. By visualizing each subgroup's unique characteristics in a clear and concise manner, we can better understand the factors that differentiate them from one another. Ultimately, this information can be used to make informed decisions and take appropriate actions based on the needs and priorities of each subgroup.

Afterward, we include the temporal aspect in the visualizations. For that, we opt for an alluvial diagram that represents the transition of the nodes/users across time/weeks. We have represented a subgroup as a white rectangle for each time point. The size of the rectangle represents the size of the subgroup. Then, we draw chords that connect rectangles in subsequent time points. These chords exit a subgroup/rectangle at a one-time point and enter one or many in the subsequent time. The color of these chords corresponds to the color of the subgroup that it exits. The width of the chord is proportional to the number of users that exit and enter the subgroups at each end of the chord. The absence of a chord between two subgroups means there is no user migration between them.

After computing the evolution indices as described in Section 6.5, we create a multi-line plot for each subgroup. In each plot, there are three lines, each with a different color, representing the shrink, split, and loyalty indices. These indices are calculated for each time point.

Two datasets are tested with Evol-COBALT: datasets 4.2 and 4.3. Those data come from different sources and represent completely different systems. However, we use the same representation method and show that it is possible to generalize it.

6.7. Results

6.7.1. Subgroups for Prediction

Hereafter, we present the prediction performance of the regression models trained with and without subgroup information. We divide it per dataset since the prediction tasks are entirely different.

Dataset 2

The Evol-COBALT set produced the best results for predicting treatment outcomes using t_0 data of patients from center B, for both patients who received CBT and those who did not. It outperformed other models by using t_1 questionnaire data for patients who did not receive CBT, and by using "hdbscan"-discovered subgroups with data from t_1 for those who did receive CBT.

When predicting treatment outcomes at center “R” using subgroups discovered at t_0, the “Evol-COBALT Set" produced the best results for patients who did not receive CBT. The “Ridge" regression model was used for this prediction.

For predicting treatment outcomes using subgroups discovered at t_1, the “knn" and “ahc" methods produced the most predictive subgroups for center R. The “Lasso" regression model was used for this prediction. The clustering algorithms identified subgroups based on four questionnaire variables: TFI , Mini-Tinnitus-Questionnaire (Mini-TQ), Fear of Tinnitus Questionnaire (FTQ), and Patient Health Questionnaire (PHQ9) score. On the other hand, Evol-COBALT only used the TFI score to identify all subgroups, except for center B at t_0, which used TFI and PHQ9 score. This suggests that the communities provide valuable information that helps the model predict the treatment outcome more accurately. In fact, they outperform the treatment outcome prediction without subgroup information (referred to as "Baseline Set") and those found with traditional clustering algorithms for four out of the eight prediction tasks presented.

It is also worth noting that Evol-COBALT uses fewer features to build subgroups, as it is cost-aware. In contrast, traditional clustering algorithms and the "Baseline Set" do not account for cost.

In conclusion, the study shows that models using subgroup information (whether from Evol-COBALT or traditional clustering) perform better than models that do not use subgroup information for all prediction tasks.

Note that communities are not discovered at this stage for center “G”. Instead, we use it as a test center, employing the node assignment approach outlined below.

The performance of Evol-COBALT was evaluated by adding new nodes and training on centers B and R, then adding nodes from center G. The results were compared to a “Baseline” and other subgroup discovery algorithms, as shown in Table 6.2. The table indicates that using a subgroup discovery algorithm leads to better performance than the “Baseline Set."

For the test center G, clustering algorithms produced better results for subgroups found, even though, for example, for subgroups assigned using center B and t_1, Evol-COBALT achieved 0.77 ± 0.01 and the best clustering set achieved 0.78 ± 0.03. Looking at the standard deviation of the predictions, we can conclude that both models produce comparable results.

Unlike the clustering approach, our methodology finds subgroups with fewer features. Since Evol-COBALT optimizes both the number of features used and the partition quality (modularity), we can conclude that even when using less information

Center	t	Treatment	Baseline	Evol-COBALT		ahc	birch	em	hdbscan	km	optics
B	t_0	with CBT	0.12 ± 0.05	PHQ9, TFI	**0.56±0.02**	0.55 ± 0.01	0.03 ± 0.08	0.44 ± 0.11	0.42 ± 0.10	0.39 ± 0.02	0.38 ± 0.10
		without CBT	0.51 ± 0.12		**0.52±0.01**	0.48 ± 0.01	0.26 ± 0.08	0.37 ± 0.04	0.51 ± 0.16	0.31 ± 0.11	0.52 ± 0.14
	t_1	with CBT	0.29 ± 0.00	TFI	0.17 ± 0.02	0.36 ± 0.09	-0.21 ± 0.07	-0.06 ± 0.17	**0.38±0.23**	0.33 ± 0.16	0.32 ± 0.10
		without CBT	0.40 ± 0.01		**0.53±0.01**	0.18 ± 0.08	0.41 ± 0.04	0.42 ± 0.01	0.26 ± 0.03	0.31 ± 0.00	0.01 ± 0.08
R	t_0	with CBT	0.46 ± 0.05	TFI	**0.67±0.01**	0.40 ± 0.03	0.44 ± 0.18	0.51 ± 0.14	0.10 ± 0.09	0.65 ± 0.17	0.42 ± 0.06
		without CBT	0.34 ± 0.02		0.29 ± 0.02	0.48 ± 0.04	0.35 ± 0.09	-0.19 ± 0.11	**0.53±0.16**	0.62 ± 0.01	-0.06 ± 0.12
	t_1	with CBT	0.48 ± 0.07	TFI	0.36 ± 0.02	0.42 ± 0.17	-0.64 ± 0.03	0.50 ± 0.07	0.52 ± 0.00	**0.55±0.02**	0.51 ± 0.00
		without CBT	0.45 ± 0.03		0.38 ± 0.04	**0.59±0.20**	0.25 ± 0.02	0.39 ± 0.07	0.18 ± 0.14	0.44 ± 0.06	0.58 ± 0.12

Table 6.3.: Results of the prediction of THI at t_2 using three different sets of features as in Table 6.2. The columns labeled with clustering algorithms correspond to the "Clustering Set," which includes the subgroups discovered by the correspondent algorithm. Underlined are the models that produced the highest R^2. The clustering algorithms represented by abbreviations are the following: agglomerative hierarchical clustering [24] (ahs); BIRCH [122], Gaussian mixture models with the Expectation Maximization algorithm [15, 27] ("em"), HDBSCAN [21], K-means [48] and OPTICS [6].

Test Center	t	Baseline	Center	Evol-COBALT	ahc	birch	em	hdbscan	km	optics
G	t_0	0.37 ± 0.07	B	0.45 ± 0.02	0.46 ± 0.04	0.47 ± 0.04	0.46 ± 0.09	0.45 ± 0.14	**0.49 ± 0.09**	0.46 ± 0.04
	t_0		R	0.46 ± 0.01	0.48 ± 0.05	**0.49 ± 0.07**	0.27 ± 0.01	0.45 ± 0.05	0.48 ± 0.16	0.48 ± 0.08
	t_1	0.56 ± 0.09	B	0.77 ± 0.01	0.77 ± 0.06	0.65 ± 0.09	0.70 ± 0.01	**0.78 ± 0.03**	0.75 ± 0.06	0.77 ± 0.10
	t_1		R	0.74 ± 0.03	0.77 ± 0.01	0.77 ± 0.01	0.77 ± 0.03	0.77 ± 0.05	**0.78 ± 0.01**	0.76 ± 0.03

Table 6.4.: Predictions for the test center G using a subgroup assignment strategy. The aim is to predict THI at t_2. The models that produced the highest R^2 are highlighted in blue. The term "Center" refers to the name of the clinical center to which the pre-existing subgroups belong and to which new patients were assigned. It is important to note that the best regression model varies per table entry. A grid search was performed. The results in this table only refer to the best prediction.

to generate subgroups, subgroups identified using Evol-COBALT are comparable in terms of their predictiveness of treatment outcome for some of the presented experiments.

Dataset 3

In order to facilitate a deeper comprehension of the experimental procedure and the corresponding outcomes, we have included Figure 6.4. This figure is intended to provide a more detailed visual representation of the setup of the experiments and the analysis described in this section.

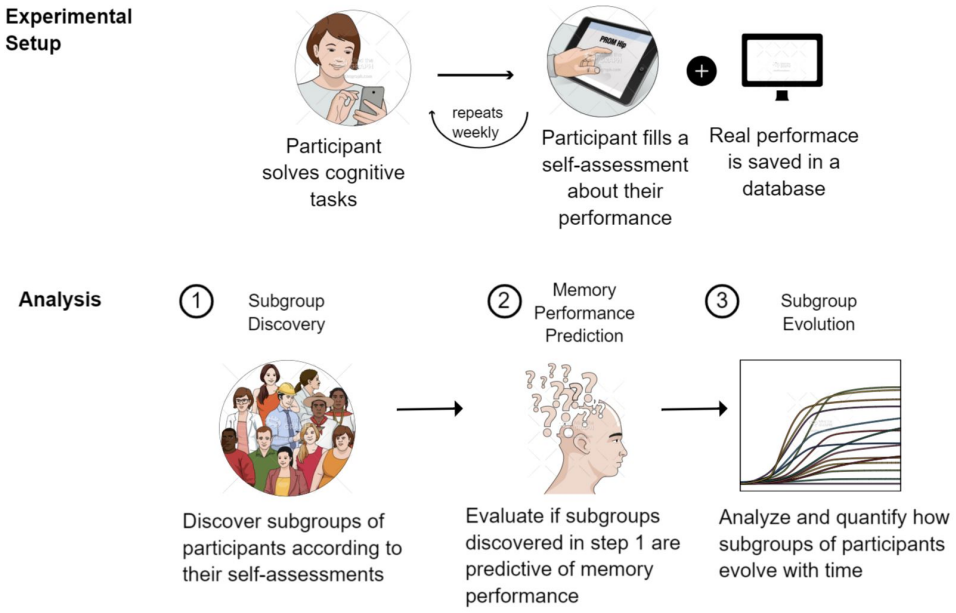


Figure 6.4.: Overview of the experimental setup and analysis using Evol-COBALT for dataset 4.3.

As previously mentioned, dataset 4.3 has multiple time points, more specifically 10. Evol-COBALT is applied and different layers are used to build subgorups at each week. For that reason, we start by presenting Table 6.5 with the variables/layers selected by Evol-COBALT to be used for subgroup discovery, which may vary weekly. We can see that layer "Freshness" is the most often selected, which means that it is the most informative layer concerning building subgroups with the highest modularity, meaning with the subgroups that can be better separated.

Figure 6.5 displays the modularity of the subgroups identified using the set of layers that achieved the highest modularity. The bubble plot shows a bubble for each week, with the size of the bubble representing the number of users with available data for that week.

We observe that the largest bubbles occur between week 4 and week 11, while after that, the size of bubbles decreases significantly. This observation is consistent with the dataset description presented earlier (refer to dataset 4.3). The highest modularity is achieved in week 12, followed by week 5. The lowest is at week 13, with number of data points lower than 50.

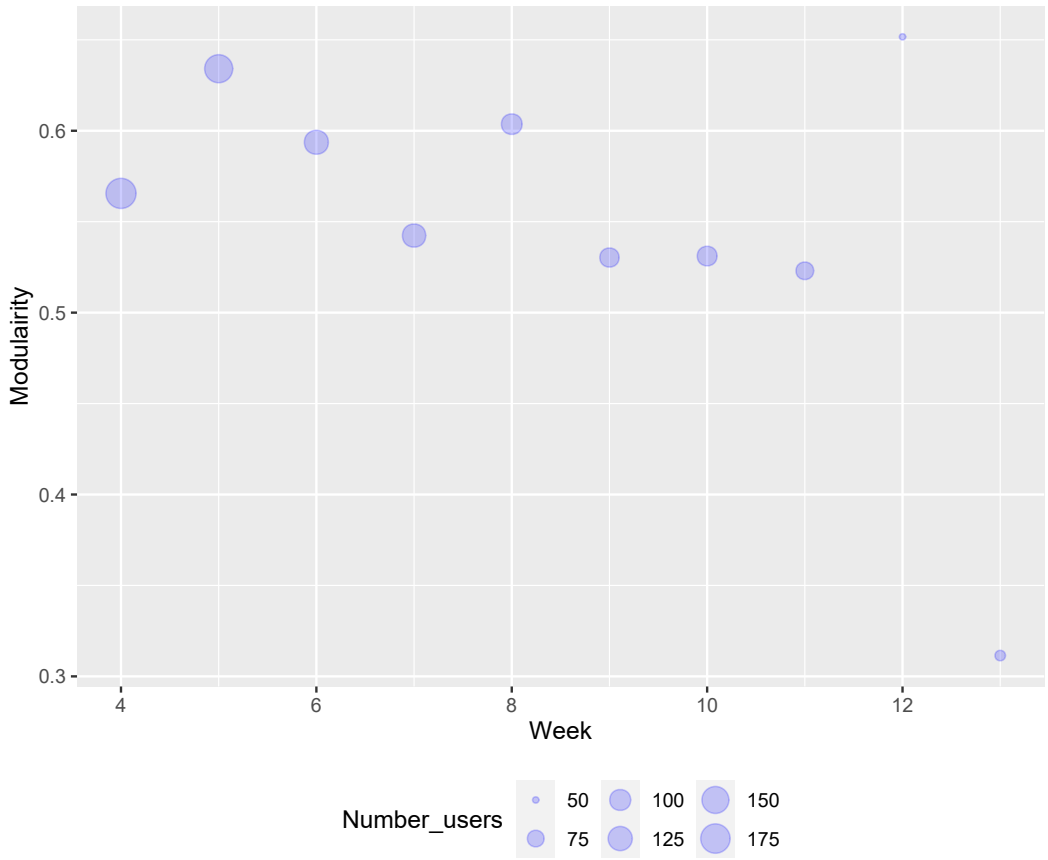


Figure 6.5.: Modularity of subgroups discovered at each week.

Variable	week 4	week 5	week 6	week 7	week 8	week 9	week 10	week 11	week 12	week 13
Static variables										
Age										
Sex										
BMI										
Variables associated with test session that might influence performance										
TimeOfDayLearning										
Delay Period										
Self-assessment questionnaires										
Part I: Questions regarding test session										
Concentration										
Distraction										
Subjective Performance							✓	✓		✓
Part II: Questions concerning the previous 8 days										
Memory Performance										
HappinessWk										
Calmness										
EnergyWk				✓						
Freshness	✓	✓	✓	✓	✓	✓				
InterestWk									✓	
SubjectiveMemoryWk										
ConcentrationWk										
ThinkingWk										
PlanningWk										
Mood										

Table 6.5.: List of variables selected by Evol-COBALT for subgroup discovery, updated on a weekly basis. A crosscheck denotes that the correspondent variables (in that row of the table) was used to find subgroups at the corresponding week.

Next, we present Table 6.6, which summarizes the predictiveness of "Memory Performance" of subgroups per week, comparing two types of scenarios: with and without subgroup information.

In most weeks, incorporating subgroup information (discovered with Evol-COBALT) leads to better predictive performance of the user's cognitive task performance compared to not using subgroup information. However, for weeks 10, 12 and 13, the model that does not use subgroup information achieves a higher R^2 than the one using subgroup information. The cause may be that we have a lower number of data points for later weeks. The most common reason for this is that participants drop out of the study and stop completing the cognitive tasks.

6.7.2. Subgroup Visualization and Evolution

In this subsection, we visualize the subgroups discovered for each dataset in order to characterize the subgorups in a qualitative way. Sections 6.7.2 and 6.7.2 show the qualitative evaluation for the dataset 4.2 and 4.3, respectively. Please note that dataset 4.3 has more time points than 4.2 and therefore the visualizations are more complex.

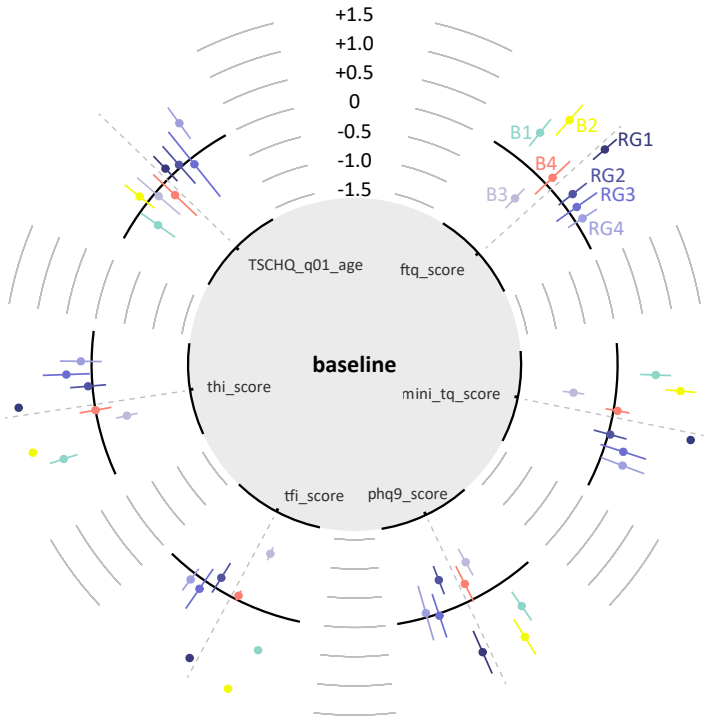
Dataset 2

The dataset discussed in Section 4.2 contains comprehensive data from an RCT that focused on patients with tinnitus who received various therapies. Our primary interest is in the data collected during the "baseline," "interim," and "final" visits, which offer insights into the patients' questionnaire scores and other relevant information.

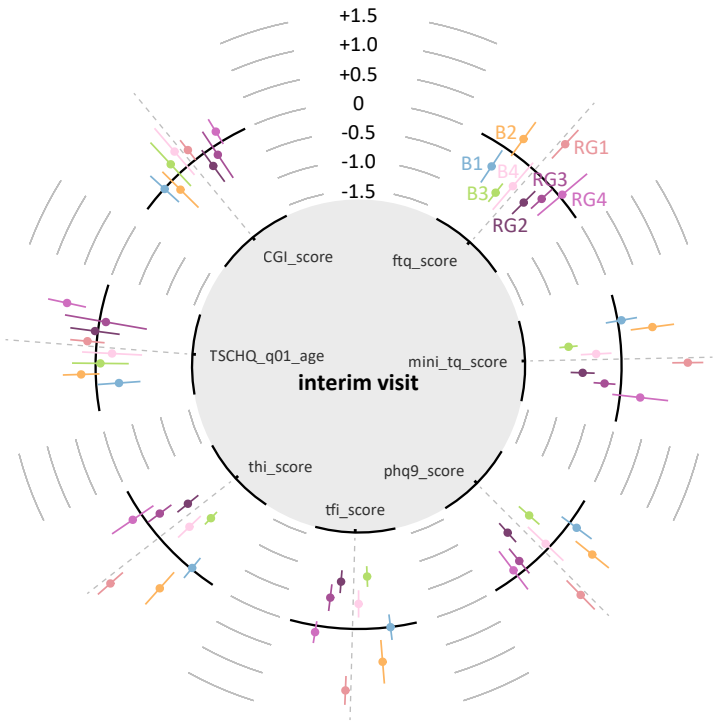
Figure 6.6 visualizes the questionnaire score values of each subgroup, with radial plots highlighting the differences between the subgroups. During the first visit, patients in subgroup "B3" had the lowest scores on the five questionnaires, indicating

week	type	q	exp_var	MAE	MSE	MAPE
4	with subgroup info	0.19	0.209	3.470	17.171	9.236
4	no subgroup info	✖	0.192	4.166	32.486	16.587
5	with subgroup info	0.19	0.135	3.072	13.824	8.298
5	no subgroup info	✖	0.122	2.790	11.980	7.243
6	with subgroup info	0.25	0.097	3.352	17.697	8.738
6	no subgroup info	✖	0.053	3.588	17.720	9.128
7	with subgroup info	0.16	0.139	3.252	14.962	8.521
7	no subgroup info	✖	0.128	3.353	17.076	9.207
8	with subgroup info	0.22	0.174	2.609	11.753	7.118
8	no subgroup info	✖	0.078	3.596	36.534	16.303
9	with subgroup info	0.22	0.203	3.121	13.744	8.097
9	no subgroup info	✖	0.171	3.353	16.896	8.518
10	with subgroup info	0.53	0.101	3.418	20.242	9.350
10	no subgroup info	✖	0.191	3.315	16.280	9.475
11	with subgroup info	0.33	0.020	3.332	15.790	8.923
11	no subgroup info	✖	0.006	3.275	18.630	8.121
12	with subgroup info	0.27	0.010	3.542	25.957	9.650
12	no subgroup info	✖	0.096	4.105	30.170	9.546
13	with subgroup info	0.11	0.051	4.462	25.443	11.890
13	no subgroup info	✖	0.107	3.749	20.373	9.622

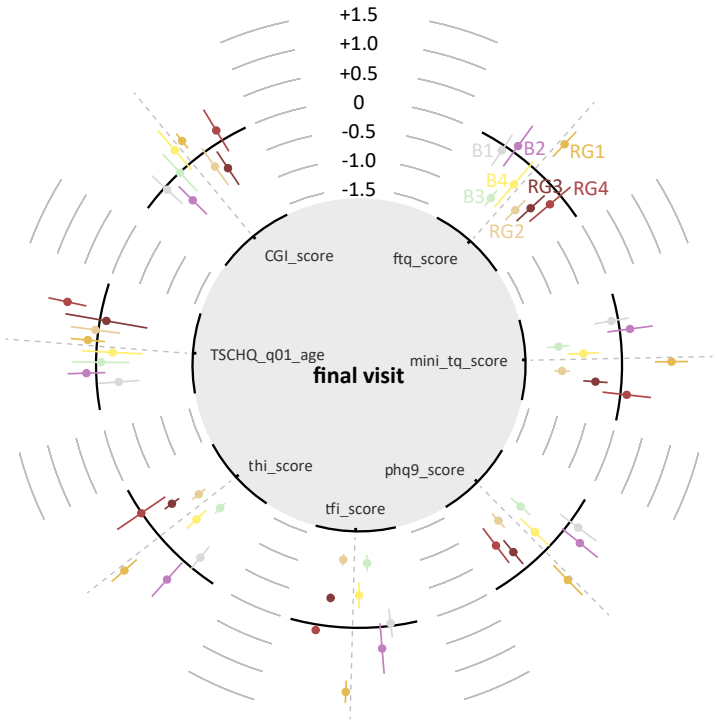
Table 6.6.: Table with prediction results of a learning model that predicts "Memory Performance" using subgroup information and not using it. The column "type" contains information on whether the learning model used subgroup information as a feature of the model or not. Column "q" denotes the modularity of the subgroups. Naturally, prediction models not trained with subgroup information have no "q" value. The explained variance ("exp_var") column has a pink bar, which is as long as the value of the "exp_var". The notation of MAE, MSE, and Mean Absolute Percentage Error (MAPE) can be found in the acronyms section.



(a) First visit



(b) Interim visit



(c) Final visit

Figure 6.6.: Radial plots of different subgroups that were discovered at different time points using Evol-COBALT. We show baseline (a), interim (b), and final (c) visits. For each variable, subgroup, and clinical center, error bars are generated to represent the z-score of the subset. The higher the z-score, the higher the variable value is for that particular subset. Each error bar has a different color and is accompanied by a label starting with "B" or "R". The error bars with the label starting with "B" represent subgroups from the clinical center "B", while those starting with "R" represent clinical center "R".

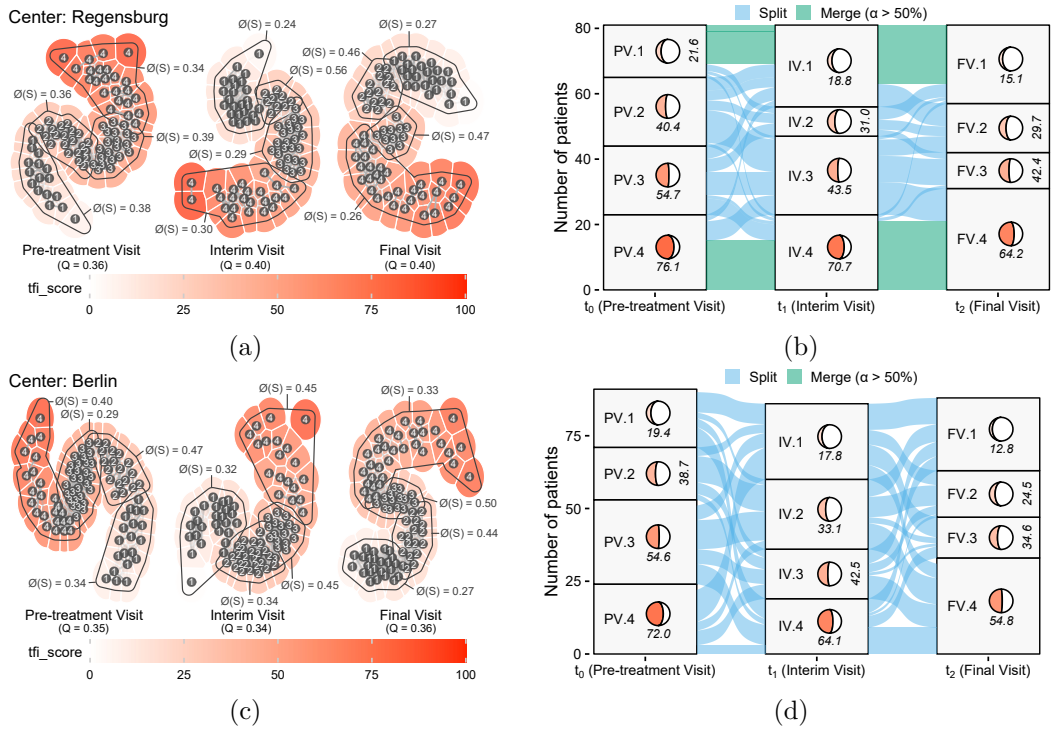


Figure 6.7.: Subgroups and their transitions across time for the centers Regensburg (subfigures (a),(b)) and Berlin ((c),(d)). (a,c) Dynamic communities in an SN from questionnaire data. (b,d) Patient transitions between communities from t_0 to t_1 and t_1 to t_2. Moon charts depict within-community TFI score averages, whereas larger gibbous portions represent higher scores. *"Merge"* transitions where the majority of a community's patients move to the same community at the subsequent time point are highlighted by chord color.

that they may have been struggling with tinnitus symptoms more than other subgroups. Interestingly, these patients' age was close to the average, suggesting that age may not be a significant factor in tinnitus severity.

At center R, patients in subgroup "RG1" had the highest questionnaire scores, indicating that they may have had the most severe tinnitus symptoms among all subgroups. In contrast, patients in "RG4" were the oldest on average, compared to center "B," suggesting that age may play a role in tinnitus severity.

Regarding the interim visit, subgroups "B2" and "RG1" had the highest questionnaire scores, indicating that their tinnitus symptoms may have been more severe than other subgroups during this period.

Finally, during the final visit, the subgroup with the lowest average score on the questionnaire was "B3" (except for the **CGI!** (**CGI!**) score), along with "RG2," which may suggest that the therapies provided to these subgroups may not have been as effective as other subgroups.

Next, we illustrate the subgroups over time. Figure 6.7 shows on the left, per clinical center ("Regensburg" and "Berlin") the "TFI" questionnaire score layer of each MLN for the pre-treatment, interim, and final visit.

In Figures 6.7a and 6.7c, we can observe three layers that represent the TFI score at different time points for the two clinical centers. The subgroups detected (black dots surrounded by a black fine line) are composed of tiles with similar colors. The

color gradient of red denotes the TFI score of the nodes, which indicates that nodes with similar TFI scores are grouped together. Therefore, we can conclude that the subgroups detected are formed by nodes with similar TFI scores.

Figures 6.7b and 6.7d display alluvial diagrams for each clinical center. The diagram for center Regensburg shows that the subgroups “PV.1" and “PV.4" (subgroups 1 and 4 at pre-treatment visit) have the majority of their members migrating together to the interim visit. Similarly, subgroups “IV.1" and “IV.4" have the majority of their members migrating together to the final visit subgroups, “FV.1" and “FV.4". These subgroups have members with the highest (subgroup 1) and the lowest average (subgroup 4) TFI score, and they are also the ones with the highest stability in terms of members.

Dataset 3

Figure 6.8 shows the radial plots that show the composition of the subgroups discovered with Evol-COBALT with the dataset presented in Section 4.3.

Subgroup “D” of week 3 is the one composed of users with the highest-than-average weight, body mass index (BMI), and height. On top of that, it is also the subgroup in which users learned the task later in the day since the time of day learning (TODL) z-score is higher than the other subgroups.

In week 9, the composition of subgroups was analyzed based on variables that were not considered by the model. The results are shown in Figure 6.8k. Subgroup “C" was found to have lower-than-average task performance.

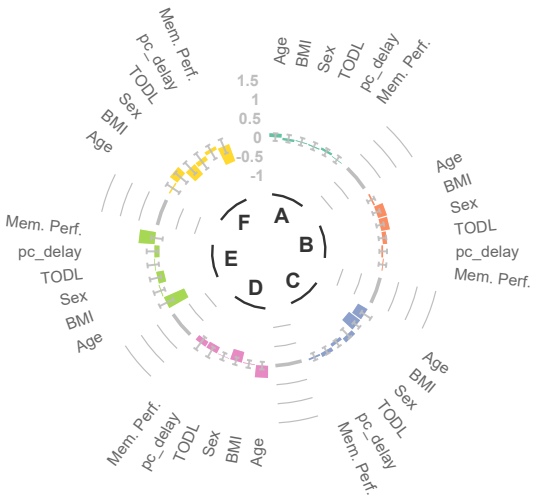
Overall, the z-scores are higher for subgroups with a few members. This is expected because it compares the variable values of a small number of members. This can still be interesting since these members could not fit into any other subgroup and thus were separated, even if that means they were in a minimal subgroup. Examples of those subgroups are the following:

- subgroup “F” (yellow): subfigure 6.8c
- subgroup “F” (yellow): subfigure 6.8e
- subgroup “F” (yellow): subfigure 6.8g
- subgroup “D” (pink): subfigure 6.8m
- subgroup “E” (green): subfigure 6.8q

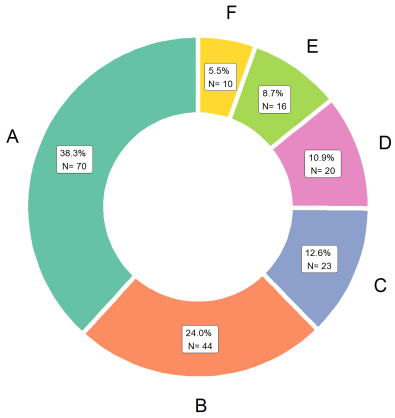
The subgroups previously mentioned contain a small number of members when compared to the other subgroups. These are subgroups that are specially interesting to study because they have members that present properties very different from the others.

Figure 6.9 illustrates the evolution of the subgroups across the 10 weeks. Users in subgroup “A" stay in the same subgroup over time, while subgroups “E" and “D" appear, disappear, and reappear. Subgroup “F" emerges at time point 4 and disappears at week 10, possibly due to data availability and subgroup merging.

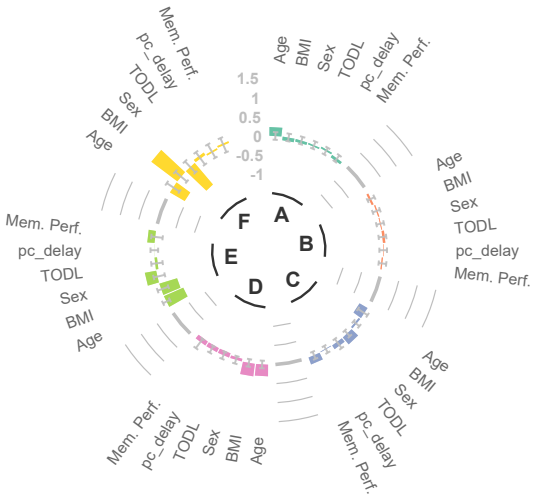
Most users of subgroup “D” in week 5 migrate to subgroup “C” in week 6. This is an example of a subgroup with a significant change in its membership. Subgroup “D” at week 6 comprises users of many other subgroups but also has new users that were not at week 5. Note that if the total width of all chords that enter a specific rectangle is not equal to the rectangle’s lateral dimension, then new users in this



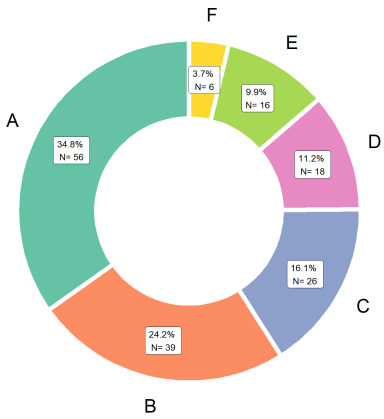
(a) Radial plot for week 4.



(b) Number of users per subgroup at week 4.



(c) Radial plot for week 5.



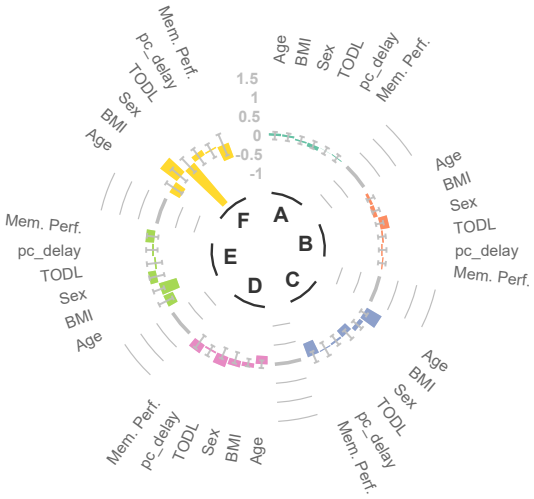
(d) Number of users per subgroup at week 5.

subgroup appear at this point. Figure 6.10 complements the understanding of the evolution of subgroups visualized in Figure 6.9.

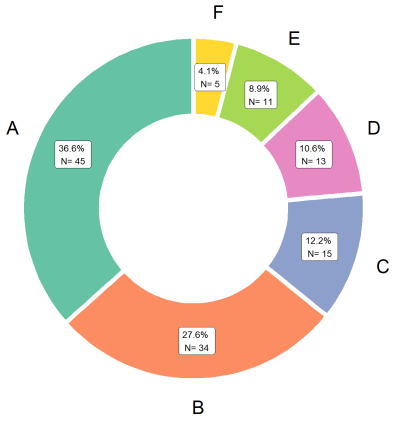
In Figure 6.10a, we can observe that subgroups that shrink the most are not necessarily the largest in size. For instance, during week 5, subgroup “D" has the same shrink index as subgroup “F" in week 6. However, subgroup “F" is already quite small, while “D" is much bigger. This suggests that, in this context, subgroup "F" is closer to the phenomenon known as “subgroup death" than subgroup “D".

It is worth noting that the subgroup labeled “A" frequently has lower shrink index values compared to the other subgroups, except for week 2. This indicates that the subgroup has relatively low shrinkage. Furthermore, by examining the size of the bubbles of subgroup A over the weeks, we can infer that the size fluctuates but is consistently smaller than the bubbles of the other subgroups.

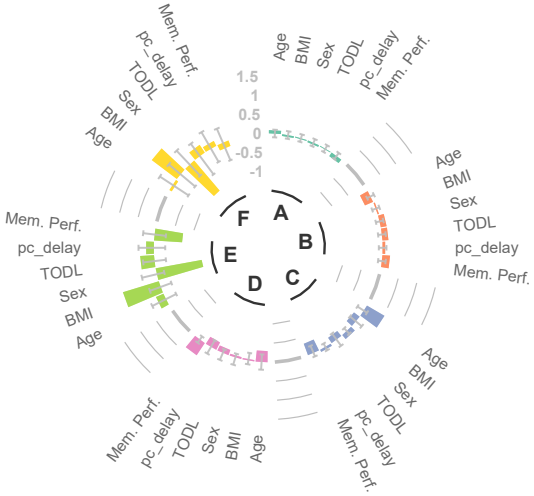
The graph labeled as Figure 6.10b shows a relatively high split index for subgroup A during the period of weeks 4 to 8. Although this subgroup is not shrinking significantly,



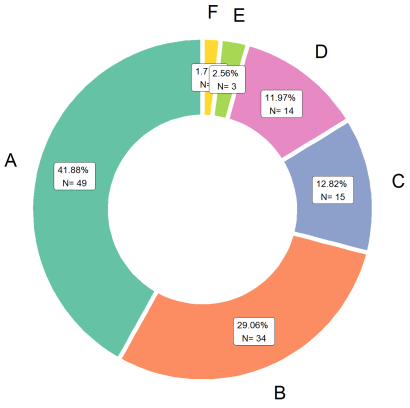
(e) Radial plot for week 6.



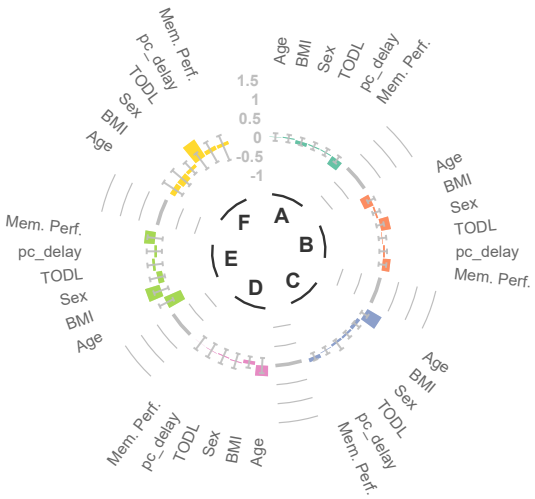
(f) Number of users per subgroup at week 6.



(g) Radial plot for week 7.

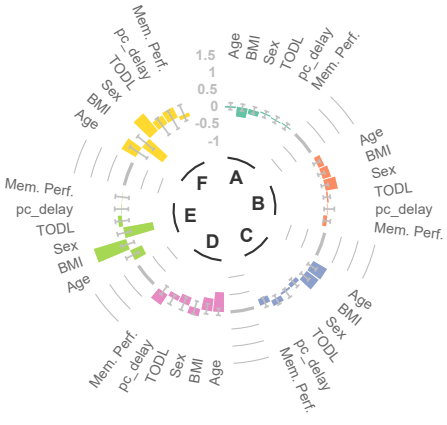


(h) Number of users per subgroup at week 7.

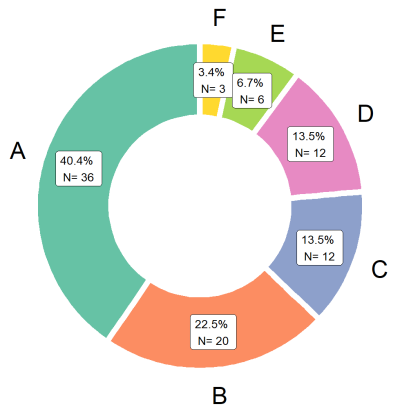


(i) Radial plot for week 8.

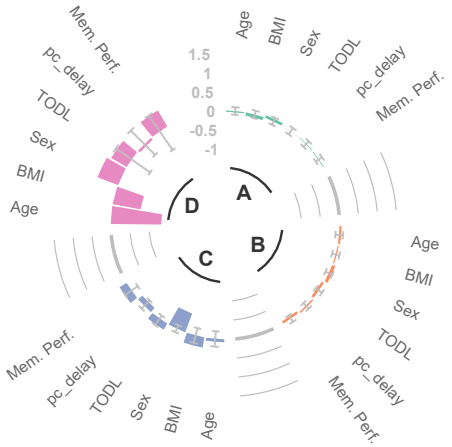
(j) Number of users per subgroup at week 8.



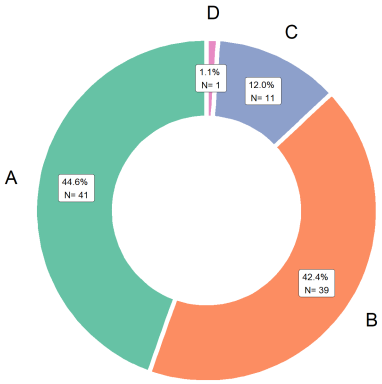
(k) Radial plot for week 9.



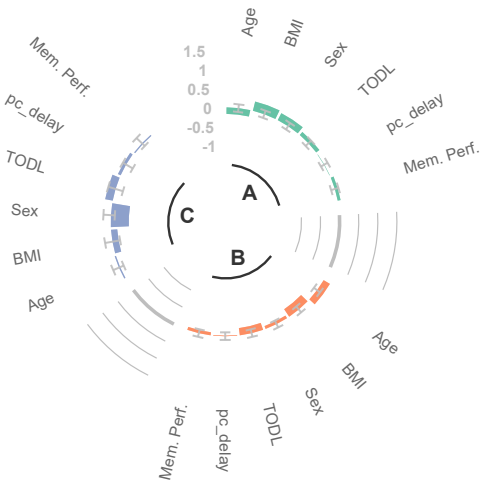
(l) Number of users per subgroup at week 9.



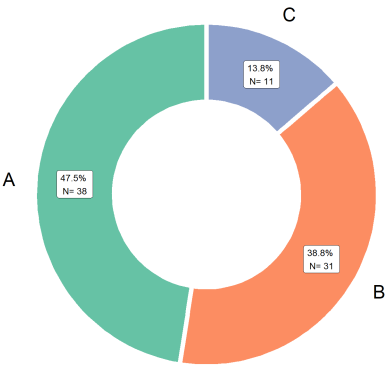
(m) Radial plot for week 10.



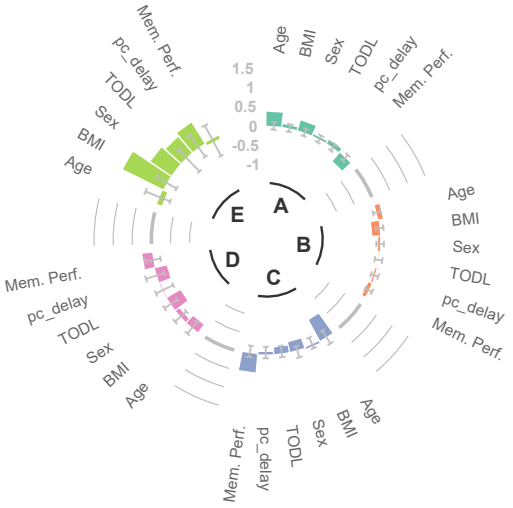
(n) Number of users per subgroup at week 10.



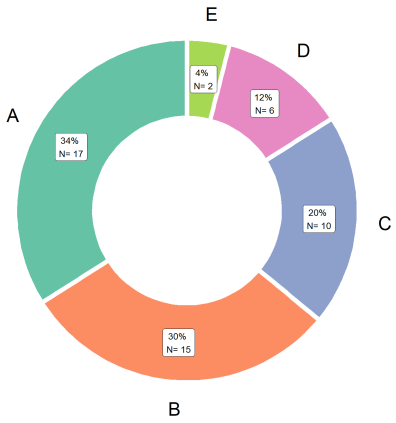
(o) Radial plot for week 11.



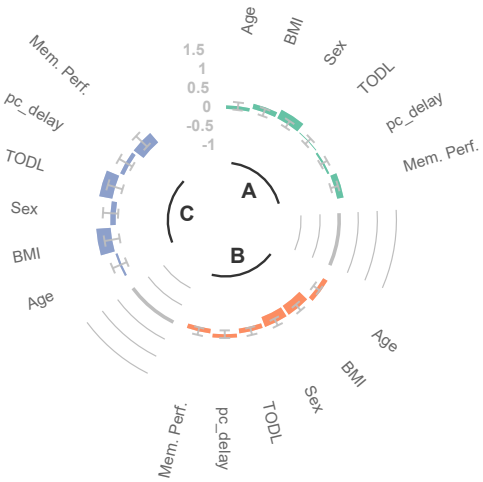
(p) Number of users per subgroup at week 11.



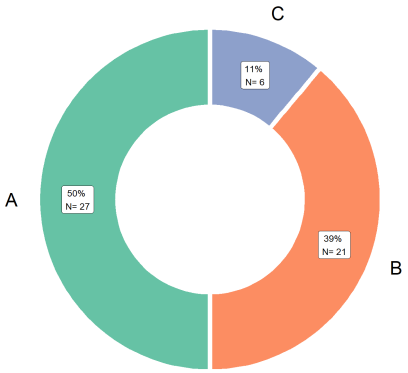
(q) Radial plot for week 12.



(r) Number of users per subgroup at week 12.



(s) Radial plot for week 13.



(t) Number of users per subgroup at week 13.

Figure 6.8.: Subgroup composition. Role of hidden variables in the model in explaining the discovered subgroups. Each subgroup is labeled by a letter. For each subgroup, a set of variables is represented by its z-score and a color that represents the subgroup. The opacity of the bars is proportional to the number of users in that subgroup. The lower the opacity, the lower the number of users. For example, subgroup "D" in week 3 is described by unseen variables that are colored pink. The gray error bars represent the confidence interval of the value range of each variable. Next to the radial plot, we have a pie plot, which illustrates the number of users per subgroup.

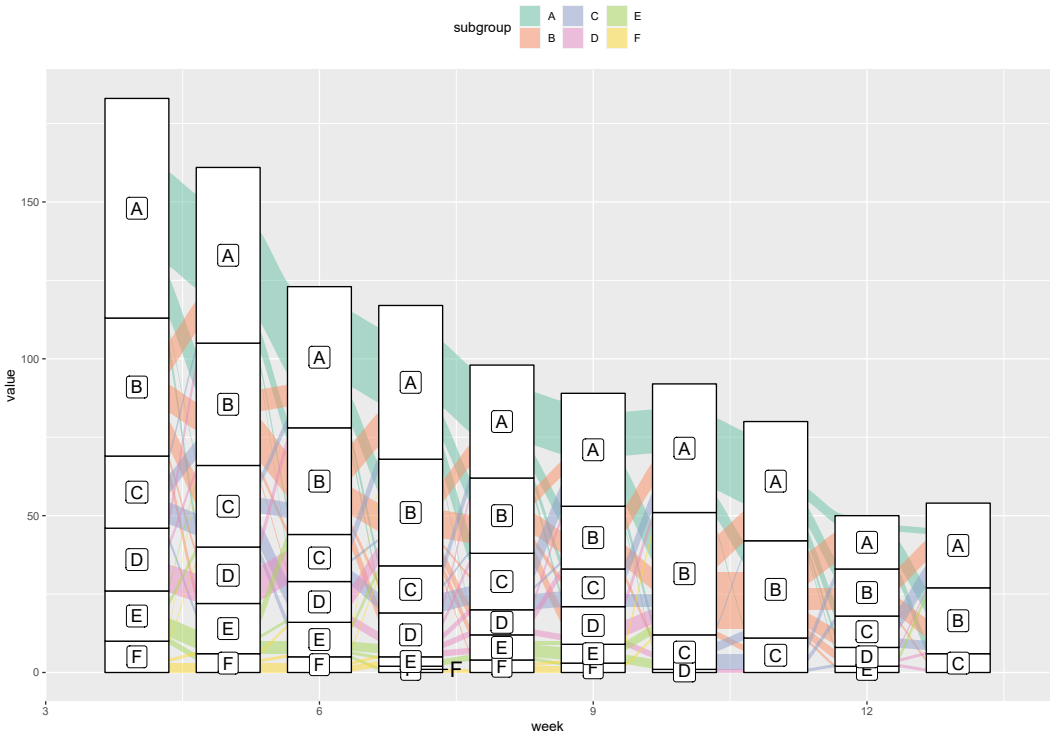
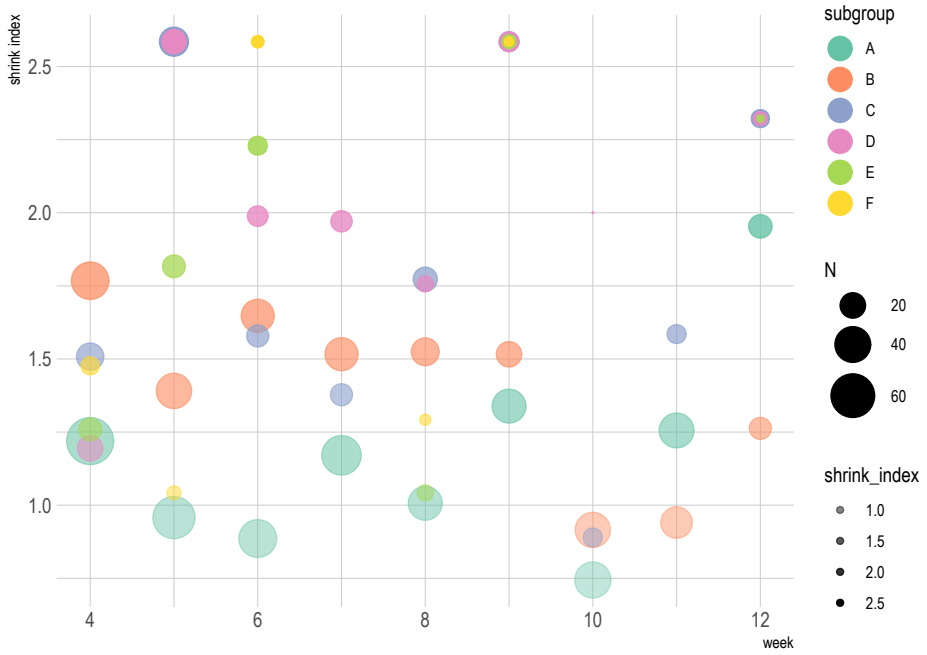
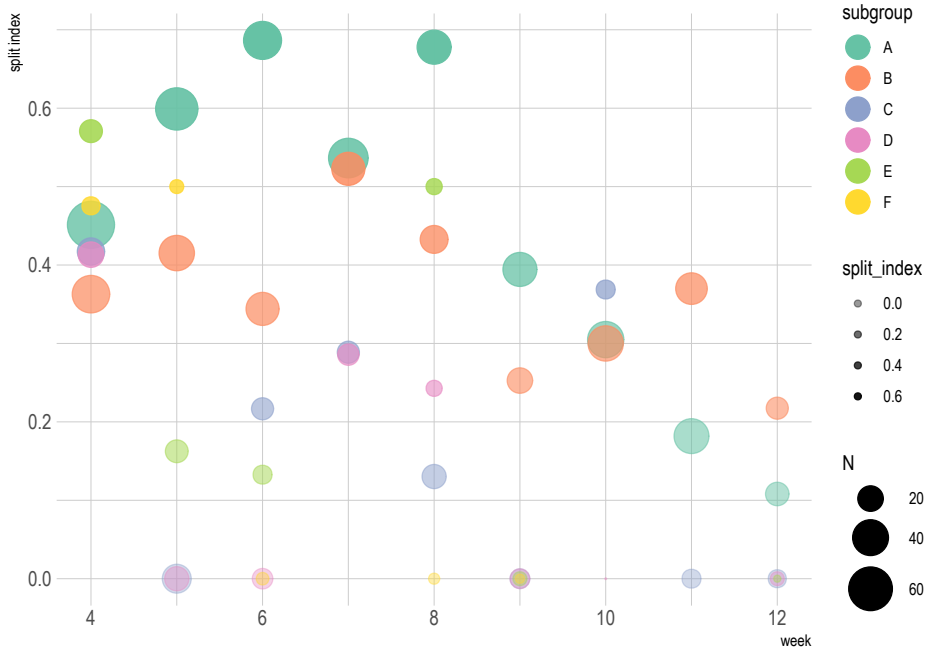


Figure 6.9.: Alluvial diagram on the evolution of subgroups. Each white rectangle at each week denotes the subgroup. If a label A, B, and C appears at two subsequent time points, then the subgroups have survived. The y-axis denotes the number of users. Each subgroup is associated with a color. For example, subgroup A is represented in pale green. From each rectangle, we see chords of the subgroup's color that exit the rectangle and enter a rectangle of the next week. These chords represent the migration of users from a subgroup to subgroup(s) of the next week. For example, in week 4, there are two thick pale green chords exiting the rectangle of subgroup A. One of them enters the rectangle of subgroup A in week 5, and the other the rectangle of subgroup B in week 5. The thickness of each chord reflects the proportion of users of the original subgroup that arrive/migrate to the corresponding subgroup of the next week.



(a) Shrink index



(b) Split index

Figure 6.10.: Shrink and split indices of each subgroup over time. Each bubble is assigned a color that corresponds to a specific subgroup, and its size varies in proportion to the size of the subgroup. Therefore, the larger the bubble, the larger the subgroup. Additionally, the opacity of the bubbles varies in proportion to the indices. The more opaque, the higher the index value.

it is splitting. This indicates that new members are joining this subgroup every week, thereby maintaining its size at a relatively stable level.

The shrink and split indices help to understand which subgroups mutate and how. There are different ways to interpret these values.

- Shrink index > 0: During a specific week, some users from a subgroup either disappeared or split into different subgroups, resulting in a smaller subgroup in the following week.
- Shrink index $= 0$: The users of a subgroup in a given week either all moved to the same subgroup in the following week or split into different subgroups.
- Split index > 0: During a certain week, some or all users in a subgroup moved to different subgroups the following week.
- Split index $= 0$: All the users who did not disappear were moved to the same subgroup. Additionally, if there are no users from a certain subgroup in a given week who appear in the next week, it means that the shrink or split index is missing.
- Eta/disloyalty index > 0: some of the users of that subgroup migrated to other subgroups
- Eta/disloyalty index $= 0$: no user migrated to another subgroup, which means a high loyalty

In this analysis, we are conducting a thorough examination of a metric called the disloyalty index, which is represented by the symbol η_i. The subscript i represents one of the subgroups, which we have categorized as "A," "B," "C," "D," "E," or "F". The disloyalty index is an important metric that helps us understand how loyal members are to their respective subgroups over time.

We have noticed that subgroup "A" has a high split index, which indicates that there is a significant amount of subgroup splitting occurring. However, despite the high split index, the disloyalty metric for subgroup "A" is relatively low between weeks 3 and 10. This suggests that members of subgroup "A" are quite loyal to their subgroup during this period.

However, from weeks 11 to 15, the disloyalty index for subgroup "A" increases, despite the subgroup having a low splitting index during this period. This may appear contradictory at first, but it actually means that even though fewer members are leaving the subgroup, more members are not staying in it at the next time point, resulting in a high disloyalty index. This can be due to various reasons, such as dissatisfaction with the subgroup or a lack of engagement.

On the other hand, from weeks 4 to 8, the splitting index of subgroup "A" is high, indicating that many members are leaving the subgroup during this period. However, there were relatively fewer members who transitioned to other subgroups during that period. This indicates that while members of subgroup "A" may be splitting, they are not necessarily moving to other subgroups.

Overall, examining both the split index and disloyalty metric for each subgroup can provide valuable insights into the dynamics of these subgroups and help us identify patterns and trends over time.

In the previous radial plots, we did not illustrate the differences between participants based on their responses to self-assessments. To gain a more comprehensive understanding of the subgroups, we present the self-assessment values for subgroup

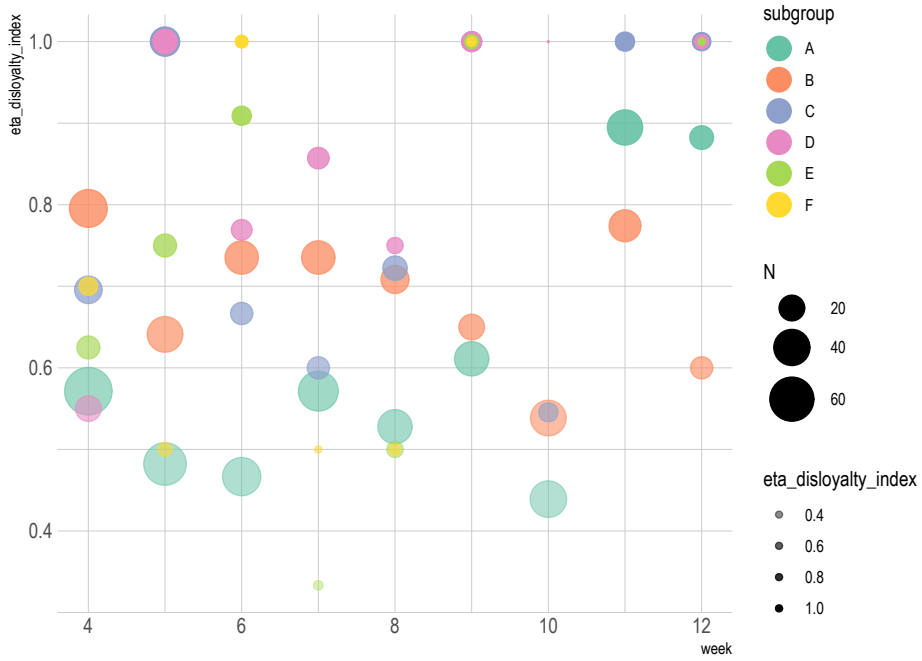


Figure 6.11.: Eta values, named as disloyalty index. Each bubble is assigned a color that corresponds to a specific subgroup, and its size varies in proportion to the subgroup size. The larger the bubble, the larger the subgroup. The opacity of the bubbles varies in proportion to the indices. The more opaque, the higher the disloyalty index value.

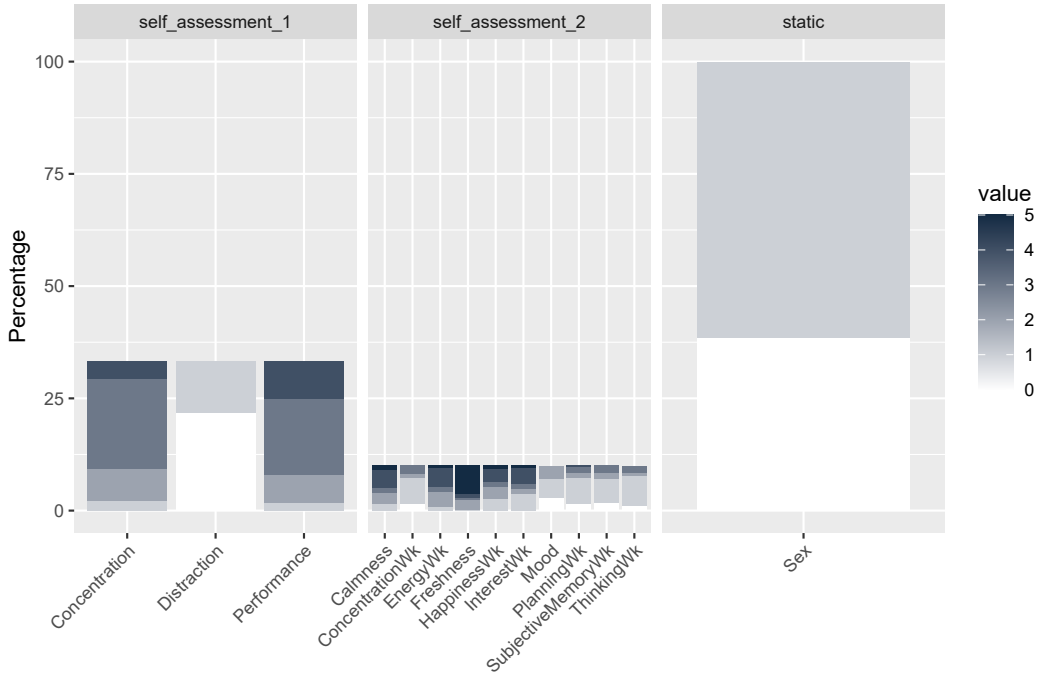


Figure 6.12.: Distribution of self-assessment questionnaires for the participants of subgroup F at week 4. On the right side of the plot, we see a gradient of colors which denote the answer to the questionnaire. The higher the number for self-assessments, the better. For example, when "Concentration" equals to 5, this means the participant thinks that their concentration level was high when completing the task. The only exception in this plot is "Sex", in which 1 denotes the male and 0 the female.

"F." This particular subgroup consists of participants who scored lower in cognitive tasks, specifically in "Memory Performance" (refer to Figure 6.8a).

The data presented in Figure 6.12 provides insightful details about the participants belonging to a specific subgroup. The analysis shows that most individuals in this subgroup are male (represented by sex=1). Another significant observation is that more than half of the participants rated their performance higher than 3, indicating that they believe that they performed well. Interestingly, despite reporting lower cognitive performance, this group's self-assessment suggests otherwise, which can be seen as a notable finding. These statistics could help researchers in understanding the underlying reasons behind the discrepancy between self-assessment and actual performance and can aid in developing effective intervention strategies.

6.8. Discussion

In this chapter, we present Evol-COBALT, a cost-aware algorithm that efficiently detects dynamic subgroups in MLNs. Our method captures the evolution of subgroups in MLNs and improves the prediction of target variables in real-world applications. We provide details on two such applications, and demonstrate that subgroups are not only predictive, but also offer valuable insights. Overall, our results show that subgroup analysis is a powerful tool for understanding complex systems.

We introduced Evol-COBALT, a method that identifies dynamic subgroups to predict a variable of interest. It builds subgroups based on multi-layer networks and captures their evolution and dynamics.

Firstly, we experimented on medical data from tinnitus patients, in which we discovered that Evol-COBALT are predictive of treatment outcome and mostly more predictive than the subgroups discovered with clustering methods.

We also presented our strategy for allocating new nodes to pre-existing subgroups, outperforming predicting treatment outcomes without subgroup knowledge. This strategy is of great importance in real-world applications, since re-computing a network requires much higher computational effort than simply using the node assignment tool proposed.

Secondly, we tested our method with mHealth data with cognitive performance data from many users. We also discovered that subgroups detected with Evol-COBALT offer insightful information about the users' cognitive performance detected at different points in time. In summary, our main conclusions are as follows:

1. Subgroup evolution can be described visually and quantitatively and improves the interpretability of user similarity

2. User characteristics unseen by the method are also statistically significantly different among the majority of the subgroups of the majority of the weeks

3. Description of a "disloyalty index" that helps interpret how stable are the members in a subgroup

Evol-COBALT is appropriate for datasets that include one or more numerical features as well as one or more timestamps. The critical advantage of Evol-COBALT is that it discovers subpopulations using the minimum set of features (layers).

However, the proposed technique has some limitations. For example, Evol-COBALT cannot directly represent categorical features. To address this, edge weights must be computed differently, using an appropriate function for categorical data similarity. Secondly, the number of nodes, layers, and time points heavily influence edge computation and computational effort. Regarding design, Evol-COBALT is suitable when time is modeled as snapshots. An adaptation to a temporal network is required to model time points continuously.

Furthermore, the interpretation and analysis of communities is challenging, given that communities are detected through layers and time periods, making it difficult to interpret. To increase interpretability, we focused on visualizing them in our study. However, additional work in this direction is required.

In summary, we highlight the importance of the architecture of MLNs when modeling a real-world problem, which we demonstrate with tinnitus patients. Representing a complex system is a challenging task, which we tackle in this chapter by modeling intra- and inter-feature interaction with edge weights. This is a significant component of the proposed approach and distinguishes it from existing designs.

In summary, our method has the potential to be a foundation for personalizing treatment plans for specific patient subgroups in clinical settings, as demonstrated by our research. Additionally, we have explored other potential applications of our method, such as using an mHealth app to monitor cognitive status in participants. Our method can also be adapted for use in various other fields beyond healthcare. The key is to ensure that the system of analysis is accurately represented within a machine learning social network and then apply Evol-COBALT to identify subgroups within that system.

7. Cross-population Layer Matching

7.1. Overview

In this chapter, our primary focus is on comparing samples using network analysis. The comparison of samples from different populations is not straightforward, but it is sometimes of great relevance. In this chapter, we propose a methodology that first transforms raw data into a network and then uses that representation to infer the distance between different samples.

We are interested in finding the differences between tinnitus patients' characteristics and the overall population. Figure 7.1 summarizes the proposed methodology in this chapter.

In [89], we carry out a network similarity analysis to compare tinnitus patients across and within clinical centers and genders. This comparison allows the comparison of patient data with different characteristics and sizes. This chapter explores how network-based data representations can be used to juxtapose clinical centers, which is also a valuable approach for medical researchers. We use the netLSD, a metric that allows comparing networks from different population bases. Networks that represented patients from different patients are compared using the dataset described in Section 4.4.

The work presented in this chapter is published in:

- C. Puga et al., "Juxtaposing Medical Centers Using Different Questionnaires Through Score Predictors," Front. Neurosci., vol. 16, no. March, pp. 1–12, Mar. 2022, doi: .

7.2. Proposed Method

In this section we explain the proposed method to compare samples from different populations. We organize it into three main blocks: (i) data description and statistical testing, (ii) statistical testing and (iii) network comparison.

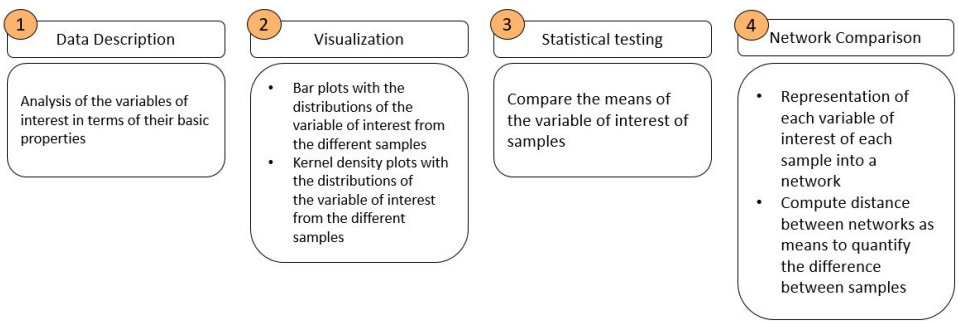


Figure 7.1.: Proposed methodology to compare samples from different populations.

7.2.1. Data Description and Statistical Testing

In order to examine how age variables differ between the general German population and tinnitus patients, we use a method that compares the two distributions by creating a kernel density estimation plot. This helps us identify the similarities and differences between the two distributions. We also analyze the age distribution differences between genders and centers. For instance, we compare the age distribution of female tinnitus patients in one center to that of another center. The same comparison is made for male tinnitus patients. We also compare gender between centers and with the distribution in Germany. To compare the centers with each other and with the general population, we use statistical testing and visualization tools.

We analyze the age distributions of two samples of tinnitus patients, s_1 and s_2, which were collected from two different clinical centers. We perform this analysis separately for each gender and correct it for multiple testing over the six comparisons that are made using the Bonferroni correction method. To check whether the samples follow a normal distribution, we use the Shapiro-Wilk test [100]. If the samples follow a normal distribution, we apply the student's test [107] on the means of the samples. However, if the samples do not follow a normal distribution, we conduct the Mann–Whitney U test [78] instead. All tests are performed with a significance level of $\alpha = 0.05$.

7.2.2. Visualization

We use kernel density estimation plots to visualize the distributions. Specifically, we plot the age distributions of the German population and the tinnitus patients from both clinical centers. To make a comparison, we calculate the percentage of people at a certain age for both data sets. We use a kernel density estimation plot to identify the age intervals where both the German population and tinnitus patients have similar percentages. This helps us to understand the zones where both distributions behave similarly. We aim to identify the age intervals where the percentage of people in both populations is similar or different. By doing so, we can explore the extent to which the two distributions agree.

We also plot the female and male ratio in Germany and each clinical center using a bi-directional bar plot. This helps us to understand the gender distribution in both populations.

7.2.3. Network Comparison

We utilize a network-based approach to compare two populations. We use it in the example of tinnitus, by examining the similarity between tinnitus patients, taking into account gender and clinical center.

Data Representation

each patient is represented as a node, with the distance between them serving as an edge. To calculate edge weights, we employ the TQ score. Prior to this, we standardize scores using a transformation equation as shown in Equation 7.1.

$$value_{i,X} = \frac{v_{i,X} - \mu_{f,X}}{\sigma_{f,X}} \tag{7.1}$$

where i, X denotes a patient i from the clinical center X, $v_{i,X}$ is the original value of the questionnaire score of that patient i_X, $\mu_{f,X}$ and $\sigma_{f,X}$ are the mean and standard

deviation of the questionnaire scores (or feature) f in the clinical center (or sample) X, respectively.

The higher the score difference between patients, the weaker the connection between them. To account for that, a transformation $1/x$ is applied, as shown in Equation 7.2, representing the edge weight.

$$w_{ij(X,X)} = \frac{1}{|value_{i,X} - value_{j,X}|} \tag{7.2}$$

where $w_{ij,(X,X)}$ as the edge weight between patient p_i and p_j in clinical center X.

Network-Based Approach for Cross-Population Matching

According to a study by [110], there are three main approaches to comparing graphs or networks: direct methods, kernel methods, and statistical representations. However, Tsitsulin's approach is different and is based on spectral representation.

Tsitsulin [110] introduced the netLSD method, which creates a vector for each network using the "heat equation." The method computes the difference between these vectors, and the final distance is the NetLSD metric. This method has the ability to meet three important properties simultaneously, which sets it apart from other approaches. These properties are permutation invariance, scale-adaptivity, and size-invariance.

Permutation invariance ensures that the distance between two isomorphic graphs is always zero. Scale-adaptivity is based on the representation of both local and global graph properties. Lastly, size-invariance takes into account the magnitude of the graphs and can differentiate between graphs with similar features but different magnitudes. It is important to note that datasets from various clinical centers may differ in size, resulting in graphs of varying sizes. The size-invariance feature of netLSD is crucial in this context since it allows for the comparison of graphs of different sizes.

7.3. Results

Previous research into tinnitus has focused on analyzing demographic variables, such as age, and their relationship with the condition. However, there has been a lack of exploration into the possibility that age distribution may be influenced by other variables, such as life expectancy and population fluctuations. For example, in Germany, the population over the age of 70 is comparatively smaller due to life expectancy.

To address this gap, we propose conducting a detailed analysis of the age distribution of tinnitus patients from clinical centers and comparing it to the age distribution in Germany. By doing so, we aim to gain a better understanding of the factors that contribute to the prevalence of tinnitus among different age groups.

To gather data on the age distribution in Germany, we have accessed statistics from a reliable and publicly available source, the German Federal Statistical Office ("Statistisches Bundesamt") [1]. This will help us to accurately compare the age distribution of tinnitus patients with the age distribution of the general population in Germany.

[1] `https://www.destatis.de`, site accessed in March 2021.

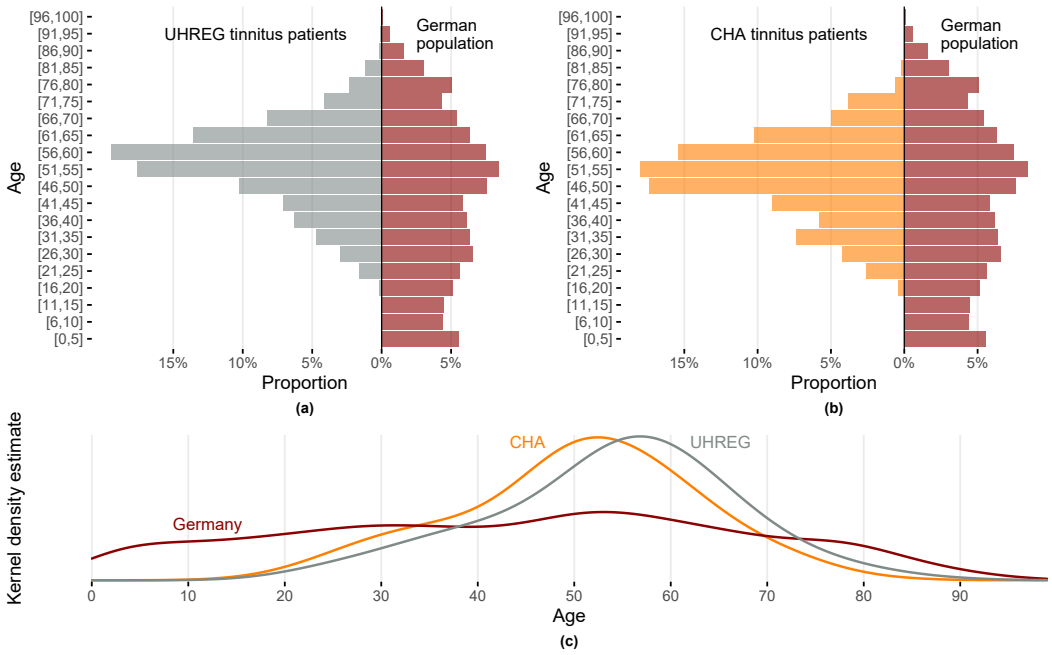


Figure 7.2.: Barplot and a Kernel density estimation plot that compare the age distribution in Germany with that of two clinical centers. The bar plots will show the age distribution of the German population in red, and the age distribution of patients from the clinical center “UHREG" in gray (subfigure (a)). The age distribution of tinnitus patients from the “CHA" clinical center will be shown in yellow. The Kernel density estimation plot will be used to visualize the age distribution in each of the three groups.

7.3.1. Comparison of Data Distributions

Figure 7.2 shows the age distribution of the tinnitus patients of two clinical centers compared to the age distribution of the German population.

The figures, Figure 7.4 and Figure 7.2, present histograms and kernel density plots displaying the age distribution of three groups: two clinics and the German population. First, it is noticeable that the centers of all three distributions are within the age range of 50 to 60 years. However, the distributions themselves differ significantly in their shape and spread.

Upon careful inspection, we observe that the curves of the two clinics intersect with the curve of the German population in two age ranges: the first range is between 30 to 40 years of age, and the second range is slightly before 80 years of age. These observations suggest a difference in the age distribution patterns between the two clinics and the German population.

We conduct statistical tests to compare the age distributions between different centers and determine if there is any statistical evidence to confirm that they are different. As the hypotheses of normality (Shapiro Wilk test) are rejected for all subsets in Table 7.1, we have chosen the Mann-Whitney test to compare location measures between samples.

Specifically, we test the following:

- whether female (1st line of Table 7.2) and male tinnitus patients (2nd line) have similar age distributions in the two centers

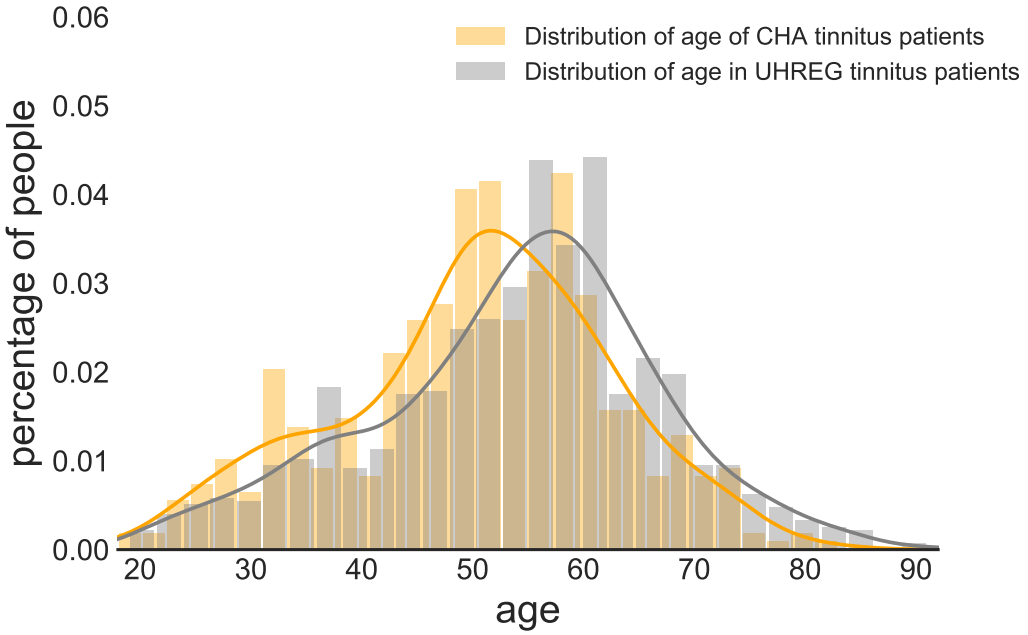


Figure 7.3.: Comparison of age distribution between clinical centers. The age distribution of tinnitus patients at the "UHREG" clinical center is displayed in gray, while the age distribution of those at the "CHA" clinical center is displayed in yellow.

- whether female and male patients have similar age distributions within each center (3rd line for "UHREG", 4th line for "CHA")
- whether female tinnitus patients in one center have similar age distributions as male patients in the other center (last two lines)

In the table 7.2, it is evident that male and female patients in the same center have a similar age distribution (H_0 cannot be rejected). However, the other null hypotheses are rejected.

After analyzing the histograms, we can conclude that the age distribution in "UHREG" is consistently higher than that in "CHA", regardless of gender.

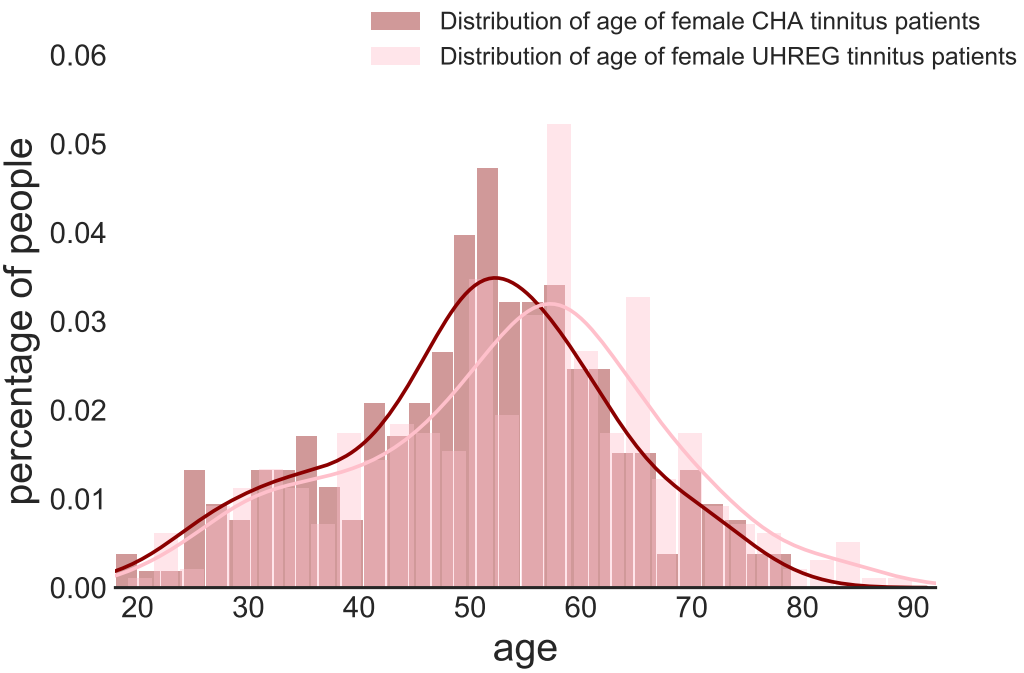
Moreover, when comparing the age distribution between male and female tinnitus patients from the same center, the null hypothesis is not rejected at 95% confidence. Therefore, we cannot claim that the age distribution of male patients in the same clinical center is significantly different from that of female patients.

Figure 7.5 displays the gender distribution in Germany, sourced from the "Statistisches Bundesamt" [2] as well as in both clinical centers. The data shows that the gender distribution in "CHA" is similar to that of Germany. However, "UHREG" has a higher percentage of male individuals compared to the general German population.

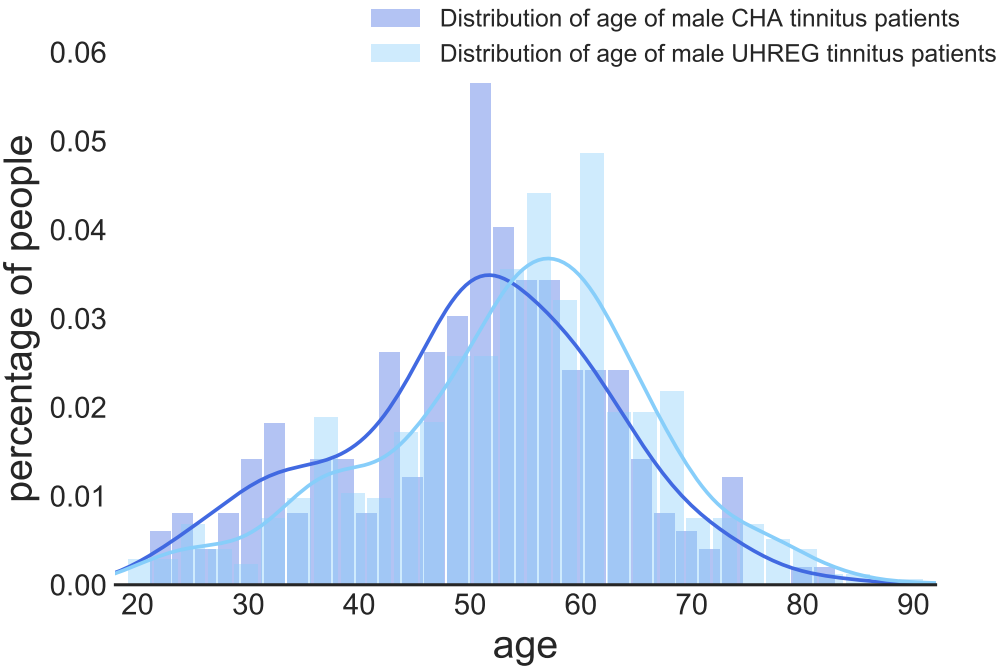
7.3.2. Network Analysis

Figure 7.6 shows the four networks and the netLSD distance between them.

[2] `https://www.destatis.de`, accessed in March 2021.



(a) Female patients



(b) Male patients

Figure 7.4.: Age distribution of patients is shown per gender and clinical center. The age distribution of female patients from both centers is displayed on the left in vivid pink (labeled as “CHA") and light pink (labeled as “UHREG"). On the right side (subfigure (b)), the age distribution of male patients is illustrated in vivid blue (labeled as “CHA") and light blue (labeled as “UHREG").

Data	N	min	max	mean	SD (σ)	p-value (Shapiro)
age_{uhreg}	1087	19	91	53.7	12.9	$3.796 * 10^{-8}$
age_{cha}	500	18	83	50.3	12.2	$0.001 * 10^{-1}$
$age_{uhreg,f}$	397	19	90	53.5	13.8	0.001
$age_{uhreg,m}$	690	19	91	53.9	12.5	$2.080 * 10^{-6}$
$age_{cha,f}$	260	18	79	50.3	12.4	0.006
$age_{cha,m}$	240	21	83	50.4	12.1	0.021

Table 7.1.: Descriptive statistics of age distributions for tinnitus patients at each clinical center, categorized by gender. For instance, age_{uhreg} represents the age distribution of all tinnitus patients at the clinical center "UHREG", while $age_{uhreg,f}$ refers only to the female patients within age_{uhreg}. SD: standard deviation, f: female, m: male, N: number of data points.

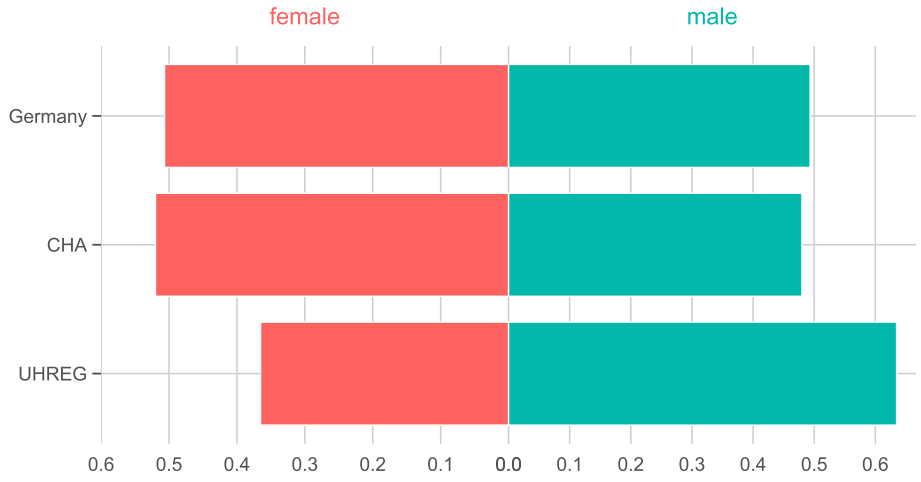


Figure 7.5.: Barplot with the gender distribution per clinical center and for Germany. The pink bars correspond to the sample with female patients and the green-blue one to the male patients.

Sample 1 (s_1)	**Sample 2** (s_2)	**Median** (s_1)	**Median** (s_2)	**U (statistic)**	**p-value**
$age_{uhreg,f}$	$age_{cha,f}$	55	51	59233.0	< 0.01*
$age_{uhreg,m}$	$age_{cha,m}$	55	51	97328.5	< 0.01*
$age_{uhreg,m}$	$age_{uhreg,f}$	55	55	138974.5	0.69
$age_{cha,m}$	$age_{cha,f}$	51	51	31163.0	0.98
$age_{uhreg,m}$	$age_{cha,f}$	55	51	105214.0	< 0.01*
$age_{uhreg,f}$	$age_{cha,m}$	55	51	54711.5	< 0.01*

Table 7.2.: Medians, Mann-Whitney U-statistic and p-value of a Mann-Whitney two-sided test for comparison of two samples. An asterisk* indicates statistical significance after the Bonferroni correction of the critical value. Therefore, the $p_{crit} = \alpha / n_{comparisons} = 0.05/6 \approx 0.008$. $age_{uhreg,f}$ denote the age of female tinnitus patients in "UHREG", $age_{cha,f}$ the age of female tinnitus patients in "CHA", $age_{uhreg,m}$ denotes the age of male tinnitus patients in "UHREG" and $age_{cha,m}$ the age of male tinnitus patients in "CHA".

As previously mentioned, in the network representation, each node represents a patient, and the edges connecting them indicate the similarity between them. The denser areas in each network reflect patients who are similar to each other in terms of their TQ score. The darker and thicker edges indicate a stronger connection between patients, indicating a high similarity.

The netLSD score provides a measure of the distance between networks. The lower the netLSD score, the higher the similarity between the networks. In this chapter, we focus on the difference in the distance values only. Figure 7.6 shows that, compared to the respective other networks, female patient networks in "UHREG" and "CHA" are the most similar, while male patient networks in "UHREG" and "CHA" are the most dissimilar with respect to the TQ score.

The distance values in this analysis are only used for comparison purposes. As a result, the absolute values are not the focus of the analysis, but rather the difference between them.

Despite the statistically significant differences in the age distributions of female patients within centers, as shown in Table 7.2, the TQ scores indicated that the female patients' networks were the most similar. On the other hand, male tinnitus patients, who also had significant differences in their age distributions, differed the most in terms of their TQ scores.

The blank spaces on the graphs indicate the absence of connections between nodes located on opposite sides of the graph. This is due to the graph pruning phase, which is a process of removing non-statistically significant edges. During this phase, edges that do not contribute significantly to the overall structure of the graph are eliminated in order to simplify the visualization and analysis of the graph.

The empty areas on the graphs are unique to each graph and vary according to their characteristics. Some graphs may have more statistically significant edges than others. These edges create more connections between nodes, making the graph more connected. As a result, the empty spaces on these graphs would be fewer, and there would be a greater number of nodes with connections between them. Conversely, graphs with fewer statistically significant edges would have more empty spaces, indicating fewer connections between the nodes.

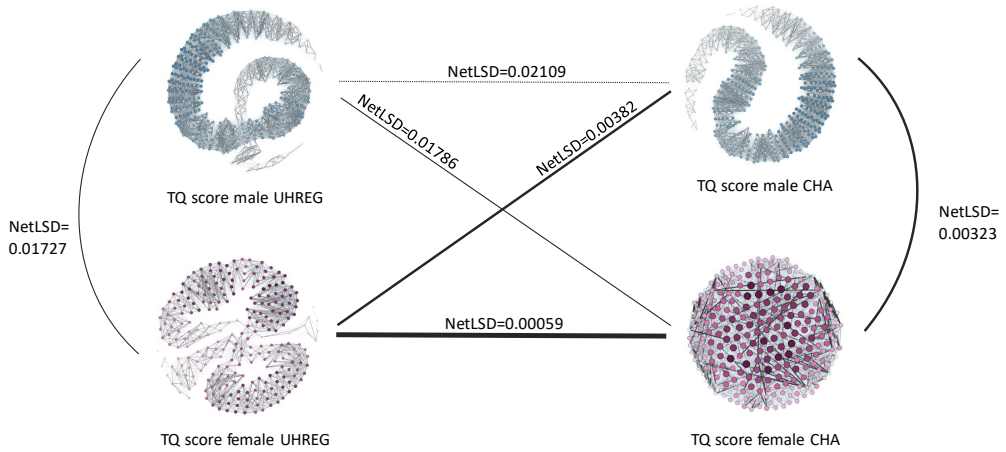


Figure 7.6.: netLSD distances of graphs with TQ_{t_0} per clinical center and gender. Networks with blue nodes represent the male patients of the correspondent clinical center. Magenta nodes represent the female patients. The edges connecting the networks represent the similarity between the networks. The thicker the edge, the more similar the networks are, and therefore, the lower the netLSD distance.

7.4. Discussion

In this chapter, we conducted an analysis to compare the TQ scores of male and female patients across different clinical centers. They found that the scores at the initial assessment (t_0) were similar for both genders, indicating that males and females experience similar levels of tinnitus-related distress.

The research on tinnitus has identified age and gender as essential variables to consider. A recent study by [81] explored the relationship between gender and tinnitus-related distress. They found that women were more likely to experience depression and higher levels of tinnitus-related distress than men. Another study by [99] investigated the factors that predict tinnitus distress, and age and gender were found to be the most significant predictors.

[92] conducted a study to investigate the impact of various factors on the success of internet-based CBT for tinnitus patients. The study included age, gender, and education level as features. Interestingly, the authors found that education level had the most significant impact on the outcome of the treatment, with higher education levels associated with better treatment outcomes. This highlights the importance of considering a patient's educational background when designing and implementing interventions for tinnitus.

We observed that the age distribution of tinnitus patients in the two centers partially reflects the age distribution of the general population in Germany.

All three distributions show a decrease in the age group of 75 years and above. This could be due to the fact that elderly patients, especially those living in rural areas, are less mobile. Another possible explanation is that as age increases, the likelihood of having other medical conditions also increases, which may lead to a

perception of tinnitus as being less distressing. However, these are just hypotheses and further investigation is required to fully understand this pattern among elderly patients with chronic tinnitus.

It has been observed that middle-aged individuals are more likely to seek medical care for tinnitus compared to other age groups. This is evident from the fact that the tinnitus centers have a higher proportion of middle-aged patients than what is expected based on the general population of Germany. This trend cannot be attributed to the demographic characteristics of the German population.

Although patients of the same gender within a center showed significant differences in age, female patients from different centers were found to be the most similar in terms of their TQ score, according to the results from the netLSD distances. Additionally, the ages of female and male patients from different centers did not show any significant statistical difference.

In summary, this chapter provides valuable insights into the similarities and differences between male and female patients with tinnitus, the impact of gender and age on tinnitus-related distress, and the importance of education levels in predicting treatment outcomes.

8. Conclusion and Outlook

Subgroup discovery is a widely researched topic, although the methods are complicated to evaluate. In this thesis, we propose a novel method for subgroup discovery and evaluate it with domain-specific data. Real-life data is often a complex system with interactions between entities.

Our approach acknowledges that complex systems are difficult to represent as data and proposes that instead of using raw data and clustering on it, we first take the time to understand the system and represent it as closely as possible to reality.

8.1. Main Contributions

8.1.1. Contributions to the Computer Science Field

Our main contributions can be organized according to the RQs. We answer to RQ1, by proposing a novel method named COBALT and presented in Chapter 5 that finds subgroups on data represented into MLNs. The representation of complex systems into multi-layer networks is proposed by us and is based on trying to reflect the system as well as possible. Then, we apply a community detection algorithm to it and find subgroups in those data. We thoroughly evaluate by comparing COBALT with traditional clustering approaches. We show how our method works evaluating it not only using traditional partition quality metrics, but also using domain-specific knowledge. We could conclude that COBALT outperforms the traditional clustering methods for the dataset in which we tested it.

These methods are used when an MLN represents data from the same population. However, in some cases, we might want to compare cross-population data; for that, we also propose a method for comparing cross-population networks.

For RQ2, we developed two cost models that model feature cost, which, in our case, we denote as layer cost. We propose two feature cost models, one simplified and a full-fledged one. Both cost models are based on the principle that redundant features are costly since they bring little information to the system, considering their acquisition cost. With these models, we concluded that we could find high-quality subgroups using a small feature set, which is, in some cases, higher quality than those found by traditional clustering algorithms that use all feature space.

Finally, for RQ3, we developed a novel method to find subgroups over time, Evol-COBALT, presented in Chapter 6. With this novel approach, we propose a different type of data representation into an MLSN. Evol-COBALT is evaluated using two datasets with different temporal data sizes - one with small temporal data, and the other with more time points. This enabled us to have a broader view of how applicable our method is. Our findings revealed that for time points in which we have a relatively high amount of data, subgroups found with Evol-COBALT are more predictive of domain-specific target variables than traditional methods.

In order to keep track of the changes in subgroups within an MLSN, we use a number of different metrics to measure the extent of their evolution over time. This

helps us to gain a better understanding of how these subgroups change and adapt with time. Additionally, we have developed a novel metric named “disloyalty index", which provides a more direct measure of how much a subgroup changes over time. By using these various measures, we can gain comprehensive insights into the evolution of subgroups in a more detailed way, since we focus on each subgroup. With this, we can better understand in what way are the subgroups evolving and, eventually, help a medical researcher/stakeholder make more informed decisions based on these findings.

Finally, we produce a set of visualizations for a qualitative evaluation of the subgroups, an in-detail view of subgroup composition using radial plots, and the evolution of subgroups over time.

8.1.2. Contributions to Different Stakeholders

The proposed methods that this thesis entails also contribute to different stakeholders of different types:

1. **Medical Researchers:** Our innovative method enables a deeper understanding of the characteristics of patients and therefore can improve diagnosis for patients. Medical researchers can utilize the COBALT-output static subgroups to perform a comprehensive analysis of patients, identifying unique factors that set them apart from others. This approach deepens their comprehension of the diagnostic process, enabling them to create customized treatment plans that cater to the specific needs of each subgroup, rather than relying on a generic approach.

   Moreover, to complement COBALT, the Evol-COBALT model enables medical researchers to monitor subgroups over time or treatment, leveraging this data to create more efficient treatment plans tailored to each patient's unique needs.

2. **Medical Doctors:** The data output from COBALT enables medical doctors to provide more personalized treatment to their patients. Firstly, the doctors can evaluate the questionnaire and demographic data of a patient and let COBALT assign them to a particular subgroup. By doing so, the physician can identify the subgroup to which the patient is most similar and provide a tailored treatment based on the subgroup-specific treatments developed by medical researchers. This approach can lead to better outcomes for the patient.

   With Evol-COBALT, physicians can monitor the patient's progress and determine if they are still in the same subgroup or have transitioned to a different subgroup. They can then analyze the data and make informed decisions. For example, if the patient remains in the same subgroup and their treatment is improving, it is a positive scenario. However, if there is an abrupt transition to a different subgroup with more severe symptoms, then the doctor may need to adjust the treatment plan accordingly.

3. **Other Stakeholders:** Although this thesis was initially motivated by the medical field, the proposed models can be applied to other fields as well. In the representation step, nodes, edges, and layers can be defined differently. For example, in machine maintenance, nodes and components can be used, or even one component that has sub-components, which can be defined as a node. Components' similarities can be characterized based on their interactions while functioning, for example. Subgroups can then help to understand which groups of components should be maintained first.

Another example could be in the transportation or logistics field. Nodes can be cities, countries, or even neighborhoods, and edges can define the traffic between the nodes, the goods to be transported, or the number of flights. Each of them would be a layer- one for flights, one for car traffic and another for transported goods. The subgroups could group nodes according to traffic, for instance. Then, the model's insights could help to make decisions about highway construction, for instance.

8.2. Limitations

The proposed method has some limitations regarding the type of data it handles. The current configuration does not allow for the representation of categorical variables. One would have to use a distance function for categorical values, such as the Jaccard index, to account for categorical features. This distance function would then be converted as a similarity function, for example, by using the inverse of that function. Then, this could be used as an edge weight, and the rest of the proposed method could be applied.

Another case is for mixed-type datasets, in which we have both categorical and numerical values. In that case, the inter-layer edges are difficult to draw since they denote the similarity between two entities across a categorical and a numerical feature. One possible solution to this scenario is ignoring inter-layer edges between categorical/numerical layers. The subgroups will be discovered only by considering the similarity of the entities within each layer, but both categorical and numerical layers will be considered. The only difference would be that the interaction between categorical and numerical features would not be accounted for.

Finally, the interpretation of subgroups/communities in MLNs is not a straightforward task and demands much effort on the visualization and explainability of the solutions. We tackle this with elaborate visualizations, but an interactive user interface where we could click on each subgroup and analyze its composition would be the next step to make it easier to understand.

8.3. Future Work

For future work, we plan to extend Evol-COBALT to account for categorical data by using a similarity function between categorical variables. This would be an extension of the first phase of the proposed method, the representations, in which we define edge weights. After, we plan to evaluate it with appropriate datasets in which we have mixed-type data.

In this thesis, even though we use datasets from different sources, all are directly related to health. In the future, we will evaluate our approach using a variety of datasets since our methodology is not limited to a particular domain. For example, we can use this method in predictive maintenance in which the entities/nodes are machine elements, and the edges measure the dependency on each other. Subgroups of machine elements might be interesting for predicting the next maintenance/substitution of an element or set of elements.

It would be helpful to rank the variables based on their significance or predictiveness to enhance the analysis. This would help identify the order of importance of factors that distinguish the entities/patients/participants and are crucial for the analysis.

In analyzing networks, the number of nodes, layers, and time points significantly

impact edge computation and computational effort. An optimization can be implemented to improve computation efficiency. One solution is to use parallel computing when computing the intra-layer edges. This is possible because the memory of one layer is not needed to produce the other during the first part of the representation phase.

Furthermore, when modeling time as snapshots, Evol-COBALT is an appropriate choice. However, to model time points continuously, an adaptation to a temporal network is necessary. This depends highly on the domain and is a future task directly related to experimenting with datasets from other real-world problems.

A. Appendix

A.1. Materials

A.1.1. Dataset 3: Detailed Materials

Week	N	4, N = 261[1]	5, N = 218[1]	6, N = 162[1]	7, N = 142[1]	8, N = 124[1]	9, N = 108[1]	10, N = 93[1]	11, N = 83[1]	12, N = 74[1]	13, N = 61[1]
Static variables											
Age	1,310	59 (52, 67)	59 (52, 66)	59 (52, 67)	59 (53, 66)	59 (54, 67)	60 (54, 69)	60 (54, 67)	61 (55, 69)	63 (55, 69)	61 (55, 68)
Unknown		5	4	1	2	2	1	1	0	0	0
Sex	1,326										
female		62 (24%)	46 (21%)	41 (25%)	35 (25%)	30 (24%)	29 (27%)	21 (23%)	22 (27%)	23 (31%)	20 (33%)
male		199 (76%)	172 (79%)	121 (75%)	107 (75%)	94 (76%)	79 (73%)	72 (77%)	61 (73%)	51 (69%)	41 (67%)
BMI	1,318	25.4 (22.7, 28.7)	25.2 (22.8, 29.1)	25.1 (22.9, 28.7)	25.8 (23.0, 29.4)	25.9 (23.3, 29.0)	25.8 (23.3, 28.6)	25.9 (23.3, 28.7)	25.9 (23.3, 28.6)	25.8 (23.4, 28.6)	25.7 (23.4, 28.
Unknown		3	2	1	1	1	0	0	0	0	0
Variables associated with test session that might influence performance											
TimeOfDayLearning		16.1 (12.3, 19.2)	15.7 (11.6, 18.6)	15.4 (11.1, 18.1)	16.1 (11.9, 18.5)	13.9 (10.4, 17.7)	15.4 (10.2, 18.3)	13.3 (9.0, 17.6)	15.0 (10.6, 18.2)	16.0 (11.6, 18.3)	15.6 (12.3, 18.
pc_delay											
1		147 (56%)	117 (54%)	87 (54%)	77 (54%)	72 (58%)	63 (58%)	54 (58%)	43 (52%)	44 (59%)	30 (49%)
2		74 (28%)	75 (34%)	50 (31%)	36 (25%)	32 (26%)	29 (27%)	26 (28%)	29 (35%)	23 (31%)	20 (33%)
3		34 (13%)	23 (11%)	20 (12%)	24 (17%)	15 (12%)	12 (11%)	11 (12%)	9 (11%)	2 (2.7%)	8 (13%)
4		6 (2.3%)	2 (0.9%)	5 (3.1%)	5 (3.5%)	5 (4.0%)	4 (3.7%)	2 (2.2%)	1 (1.2%)	5 (6.8%)	3 (4.9%)
5		0 (0%)	1 (0.5%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (1.2%)	0 (0%)	0 (0%)
Screen		11.9 (11.7, 14.8)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	11.9 (11.7, 14.9)	12.3 (11.7, 14.9)	12.3 (11.7, 14.9)	11.9 (11.7, 14.
Unknown		54	36	14	10	3	0	1	3	4	7
Outcome variable											
pct_corr		0.56 (0.44, 0.68)	0.52 (0.44, 0.64)	0.56 (0.44, 0.68)	0.56 (0.44, 0.68)	0.52 (0.44, 0.60)	0.52 (0.44, 0.65)	0.56 (0.44, 0.68)	0.52 (0.44, 0.64)	0.52 (0.44, 0.64)	0.56 (0.44, 0.6
Part I: Self-assessment questionnaire - Questions regarding test session											
Concentration											
0		0 (0%)	0 (0%)	1 (0.7%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
1		6 (2.9%)	5 (2.7%)	3 (2.0%)	0 (0%)	5 (4.1%)	1 (0.9%)	2 (2.2%)	1 (1.3%)	1 (1.4%)	2 (3.7%)
2		22 (10%)	30 (16%)	29 (20%)	20 (15%)	11 (9.1%)	13 (12%)	16 (17%)	15 (19%)	17 (24%)	6 (11%)
3		94 (45%)	84 (46%)	54 (36%)	57 (43%)	55 (45%)	49 (45%)	35 (38%)	31 (39%)	35 (50%)	26 (48%)
4		88 (42%)	64 (35%)	61 (41%)	55 (42%)	50 (41%)	45 (42%)	39 (42%)	33 (41%)	17 (24%)	20 (37%)
Unknown		51	35	14	10	3	0	1	3	4	7
Distraction		36 (17%)	46 (25%)	41 (28%)	23 (17%)	23 (19%)	18 (17%)	17 (18%)	24 (30%)	15 (21%)	10 (19%)
Unknown		51	35	14	10	3	0	1	3	4	7
Performance											
0		0 (0%)	0 (0%)	1 (0.7%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
1		3 (1.4%)	2 (1.1%)	3 (2.0%)	2 (1.5%)	3 (2.5%)	1 (0.9%)	1 (1.1%)	0 (0%)	2 (2.9%)	1 (1.9%)
2		14 (6.7%)	17 (9.3%)	12 (8.1%)	14 (11%)	8 (6.6%)	9 (8.3%)	11 (12%)	11 (14%)	9 (13%)	5 (9.3%)
3		89 (42%)	77 (42%)	66 (45%)	53 (40%)	40 (33%)	53 (49%)	39 (42%)	31 (39%)	33 (47%)	21 (39%)
4		104 (50%)	87 (48%)	66 (45%)	63 (48%)	70 (58%)	45 (42%)	41 (45%)	38 (48%)	26 (37%)	27 (50%)
Unknown		51	35	14	10	3	0	1	3	4	7
Part II: Self-assessment questionnaire - Questions concerning the previous 8 days											
HappinessWk											
0		14 (7.7%)	11 (6.8%)	11 (8.9%)	15 (13%)	14 (14%)	5 (5.6%)	9 (13%)	10 (16%)	6 (12%)	5 (17%)
1		112 (61%)	100 (62%)	74 (60%)	73 (62%)	60 (61%)	60 (67%)	40 (56%)	31 (51%)	28 (56%)	18 (60%)
2		28 (15%)	32 (20%)	24 (20%)	15 (13%)	10 (10%)	16 (18%)	11 (15%)	8 (13%)	9 (18%)	4 (13%)
3		19 (10%)	13 (8.1%)	6 (4.9%)	8 (6.8%)	9 (9.2%)	4 (4.4%)	7 (9.9%)	7 (11%)	5 (10%)	1 (3.3%)
4		7 (3.8%)	5 (3.1%)	8 (6.5%)	5 (4.3%)	5 (5.1%)	4 (4.4%)	3 (4.2%)	3 (4.9%)	2 (4.0%)	2 (6.7%)
5		3 (1.6%)	0 (0%)	0 (0%)	1 (0.9%)	0 (0%)	1 (1.1%)	1 (1.4%)	2 (3.3%)	0 (0%)	0 (0%)
Unknown		78	57	39	25	26	18	22	22	24	31
EnergyWk											
0		16 (8.7%)	10 (6.2%)	11 (8.9%)	15 (13%)	15 (15%)	11 (12%)	8 (11%)	8 (13%)	6 (12%)	6 (20%)
1		81 (44%)	70 (43%)	52 (42%)	49 (42%)	39 (40%)	44 (49%)	30 (42%)	27 (44%)	22 (44%)	15 (50%)
2		43 (23%)	45 (28%)	35 (28%)	34 (29%)	24 (25%)	19 (21%)	22 (31%)	11 (18%)	8 (16%)	4 (13%)
3		26 (14%)	27 (17%)	18 (15%)	14 (12%)	12 (12%)	7 (7.9%)	5 (7.0%)	8 (13%)	10 (20%)	2 (6.7%)
4		13 (7.1%)	9 (5.6%)	7 (5.7%)	3 (2.6%)	7 (7.2%)	6 (6.7%)	5 (7.0%)	6 (9.8%)	4 (8.0%)	3 (10%)
5		4 (2.2%)	0 (0%)	0 (0%)	2 (1.7%)	0 (0%)	2 (2.2%)	1 (1.4%)	1 (1.6%)	0 (0%)	0 (0%)
Unknown		78	57	39	25	27	19	22	22	24	31
InterestWk											
0		21 (11%)	16 (9.9%)	22 (18%)	20 (17%)	23 (23%)	15 (17%)	13 (18%)	10 (16%)	10 (20%)	6 (20%)
1		84 (46%)	72 (45%)	48 (39%)	55 (47%)	39 (40%)	41 (46%)	31 (44%)	24 (39%)	17 (34%)	16 (53%)
2		41 (22%)	49 (30%)	32 (26%)	22 (19%)	20 (20%)	20 (22%)	14 (20%)	13 (21%)	15 (30%)	6 (20%)
3		19 (10%)	12 (7.5%)	11 (8.9%)	11 (9.4%)	9 (9.2%)	8 (9.0%)	8 (11%)	9 (15%)	6 (12%)	1 (3.3%)
4		16 (8.7%)	12 (7.5%)	9 (7.3%)	8 (6.8%)	7 (7.1%)	5 (5.6%)	5 (7.0%)	5 (8.2%)	2 (4.0%)	1 (3.3%)
5		2 (1.1%)	0 (0%)	1 (0.8%)	1 (0.9%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Unknown		78	57	39	25	26	19	22	22	24	31

SubjectiveMemoryWk										
0	53 (29%)	50 (31%)	40 (33%)	35 (30%)	29 (30%)	24 (27%)	21 (30%)	18 (30%)	12 (24%)	9 (30%)
1	104 (57%)	86 (53%)	62 (50%)	68 (58%)	53 (54%)	48 (54%)	36 (51%)	30 (49%)	30 (60%)	17 (57%)
2	23 (13%)	17 (11%)	15 (12%)	9 (7.7%)	13 (13%)	14 (16%)	10 (14%)	10 (16%)	7 (14%)	4 (13%)
3	3 (1.6%)	8 (5.0%)	6 (4.9%)	5 (4.3%)	3 (3.1%)	3 (3.4%)	4 (5.6%)	3 (4.9%)	1 (2.0%)	0 (0%)
Unknown	78	57	39	25	26	19	22	22	24	31
ConcentrationWk										
0	48 (26%)	47 (29%)	38 (31%)	34 (29%)	27 (28%)	22 (25%)	19 (27%)	20 (33%)	12 (24%)	9 (30%)
1	107 (58%)	84 (53%)	70 (57%)	68 (58%)	54 (55%)	53 (60%)	40 (56%)	32 (52%)	30 (60%)	16 (53%)
2	21 (11%)	20 (13%)	10 (8.1%)	9 (7.7%)	13 (13%)	10 (11%)	8 (11%)	6 (9.8%)	7 (14%)	5 (17%)
3	7 (3.8%)	9 (5.6%)	5 (4.1%)	6 (5.1%)	4 (4.1%)	3 (3.4%)	3 (4.2%)	3 (4.9%)	1 (2.0%)	0 (0%)
4	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	1 (1.1%)	1 (1.4%)	0 (0%)	0 (0%)	0 (0%)
Unknown	78	58	39	25	26	19	22	22	24	31
ThinkingWk										
0	57 (31%)	52 (33%)	45 (37%)	39 (33%)	29 (30%)	25 (28%)	21 (30%)	20 (33%)	14 (28%)	10 (33%)
1	96 (52%)	82 (51%)	63 (51%)	63 (54%)	57 (58%)	51 (57%)	38 (54%)	30 (49%)	31 (62%)	16 (53%)
2	25 (14%)	19 (12%)	10 (8.1%)	11 (9.4%)	6 (6.1%)	9 (10%)	8 (11%)	9 (15%)	4 (8.0%)	3 (10%)
3	5 (2.7%)	6 (3.8%)	5 (4.1%)	3 (2.6%)	5 (5.1%)	4 (4.5%)	3 (4.2%)	2 (3.3%)	1 (2.0%)	1 (3.3%)
4	0 (0%)	1 (0.6%)	0 (0%)	1 (0.9%)	1 (1.0%)	0 (0%)	1 (1.4%)	0 (0%)	0 (0%)	0 (0%)
Unknown	78	58	39	25	26	19	22	22	24	31
PlanningWk										
0	69 (38%)	60 (38%)	47 (38%)	49 (42%)	38 (39%)	28 (31%)	23 (32%)	23 (38%)	16 (32%)	12 (40%)
1	87 (48%)	75 (47%)	62 (50%)	51 (44%)	51 (52%)	52 (58%)	36 (51%)	25 (41%)	28 (56%)	15 (50%)
2	17 (9.3%)	20 (13%)	9 (7.3%)	13 (11%)	5 (5.1%)	5 (5.6%)	8 (11%)	9 (15%)	4 (8.0%)	3 (10%)
3	8 (4.4%)	5 (3.1%)	5 (4.1%)	2 (1.7%)	3 (3.1%)	3 (3.4%)	3 (4.2%)	3 (4.9%)	2 (4.0%)	0 (0%)
4	2 (1.1%)	0 (0%)	0 (0%)	1 (0.9%)	1 (1.0%)	1 (1.1%)	1 (1.4%)	1 (1.6%)	0 (0%)	0 (0%)
Unknown	78	58	39	26	26	19	22	22	24	31
Calmness										
0	15 (8.2%)	7 (4.3%)	9 (7.3%)	14 (12%)	9 (9.2%)	9 (10%)	7 (9.9%)	4 (6.6%)	3 (6.0%)	4 (13%)
1	76 (42%)	74 (46%)	57 (46%)	54 (46%)	48 (49%)	44 (49%)	35 (49%)	33 (54%)	25 (50%)	17 (57%)
2	47 (26%)	59 (37%)	34 (28%)	30 (26%)	26 (27%)	22 (25%)	17 (24%)	14 (23%)	12 (24%)	5 (17%)
3	28 (15%)	15 (9.3%)	11 (8.9%)	12 (10%)	7 (7.1%)	8 (9.0%)	9 (13%)	3 (4.9%)	7 (14%)	1 (3.3%)
4	12 (6.6%)	6 (3.7%)	11 (8.9%)	7 (6.0%)	7 (7.1%)	6 (6.7%)	2 (2.8%)	7 (11%)	3 (6.0%)	3 (10%)
5	5 (2.7%)	0 (0%)	1 (0.8%)	0 (0%)	1 (1.0%)	0 (0%)	1 (1.4%)	0 (0%)	0 (0%)	0 (0%)
Unknown	78	57	39	25	26	19	22	22	24	31
Freshness										
0	20 (11%)	18 (11%)	15 (12%)	16 (14%)	18 (18%)	12 (13%)	9 (13%)	10 (16%)	8 (16%)	5 (17%)
1	70 (38%)	56 (35%)	45 (37%)	50 (43%)	36 (37%)	36 (40%)	30 (42%)	22 (36%)	20 (40%)	13 (43%)
2	44 (24%)	39 (24%)	34 (28%)	26 (22%)	24 (24%)	20 (22%)	14 (20%)	12 (20%)	9 (18%)	7 (23%)
3	23 (13%)	26 (16%)	13 (11%)	12 (10%)	8 (8.2%)	12 (13%)	10 (14%)	7 (11%)	6 (12%)	1 (3.3%)
4	16 (8.7%)	16 (9.9%)	11 (8.9%)	8 (6.8%)	4 (4.1%)	3 (3.4%)	4 (5.6%)	7 (11%)	4 (8.0%)	1 (3.3%)
5	10 (5.5%)	6 (3.7%)	5 (4.1%)	5 (4.3%)	8 (8.2%)	6 (6.7%)	4 (5.6%)	3 (4.9%)	3 (6.0%)	3 (10%)
Unknown	78	57	39	25	26	19	22	22	24	31
Mood										
0	108 (65%)	102 (64%)	71 (58%)	83 (71%)	66 (67%)	55 (62%)	43 (61%)	40 (66%)	33 (66%)	23 (77%)
1	50 (30%)	54 (34%)	44 (36%)	29 (25%)	28 (29%)	29 (33%)	26 (37%)	18 (30%)	16 (32%)	6 (20%)
2	8 (4.8%)	4 (2.5%)	8 (6.5%)	5 (4.3%)	4 (4.1%)	5 (5.6%)	2 (2.8%)	3 (4.9%)	1 (2.0%)	1 (3.3%)
Unknown	95	58	39	25	26	19	22	22	24	31

[1] Median (IQR); n (%)

Table A.1.: Detailed overview of dataset 4.3.

A.2. Evol-COBALT

A.2.1. Quantitaive Evaluation

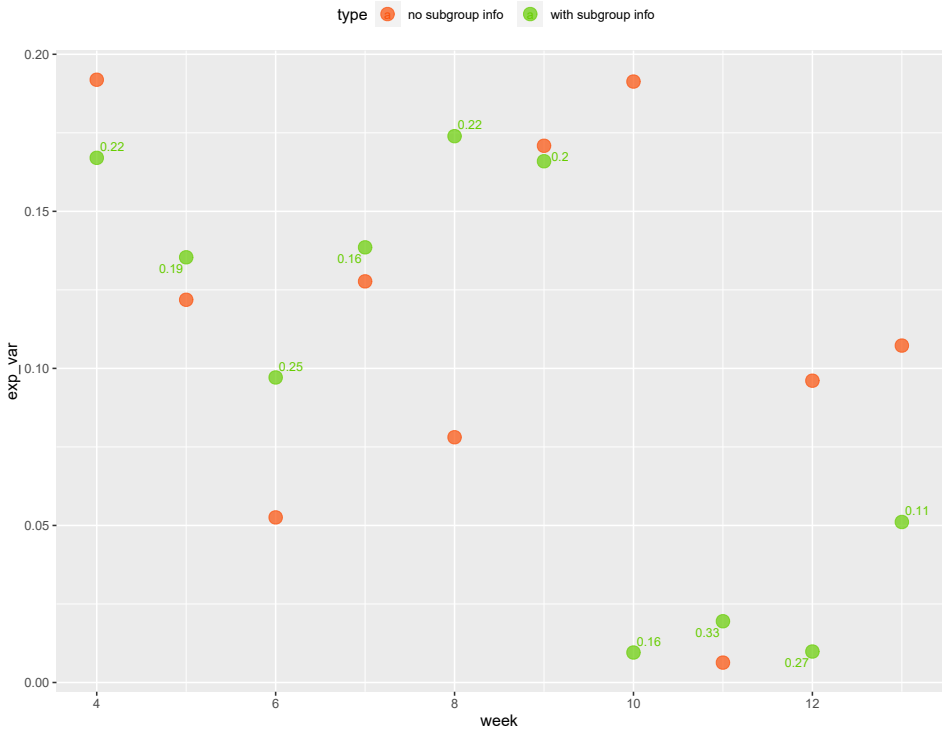


Figure A.1.: Explained variance of the regression models that use subgroup information (in green) and that do not use subgroup information (in red) per week. Each yellow point in the graph has a label that denotes the modularity value of the subgroups used on the regression model represented by that point. For example, for week 4, the green point represents the regression model that uses subgroup information to predict "Memory Performance". That subgroup solution has a modularity value of 0.22. The red point represents the regression model that does not use subgroup information to predict "Memory Performance", which, in that case, has a lower explained variance.

A.2.2. Qualitative Evaluation

For the sake of simplicity, we show all subgroups from week 4 and 9 below only.

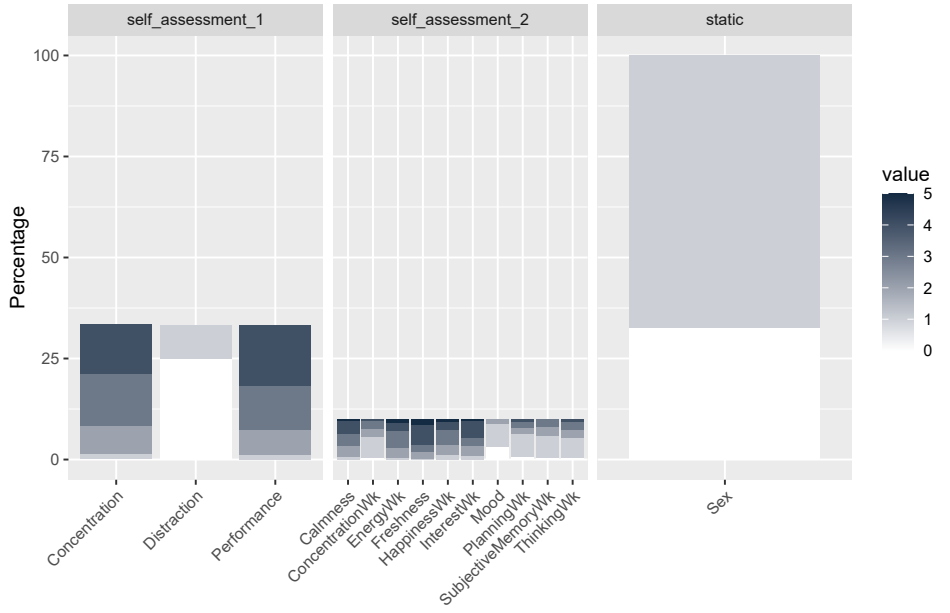


Figure A.2.: Description of the self-assessment questionnaires of subgroup E at week 4.

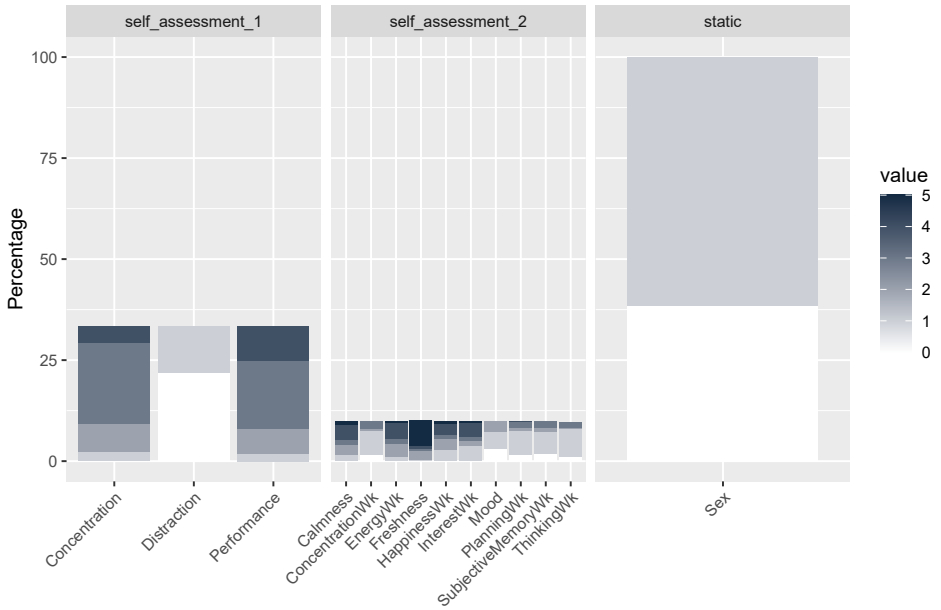


Figure A.3.: Description of the self-assessment questionnaires of subgroup F at week 4.

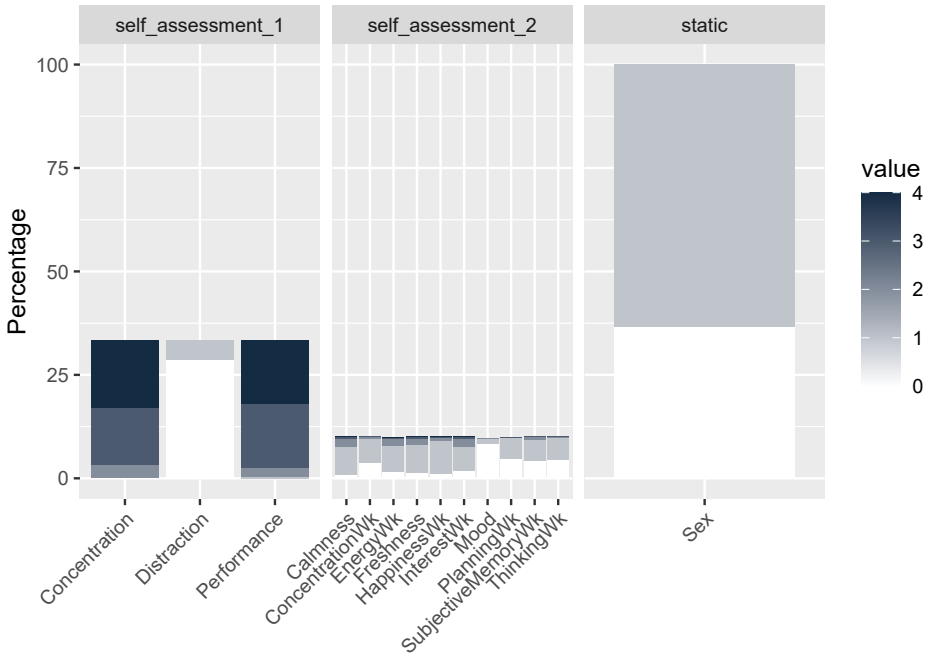


Figure A.4.: Description of the self-assessment questionnaires of subgroup A at week 9.

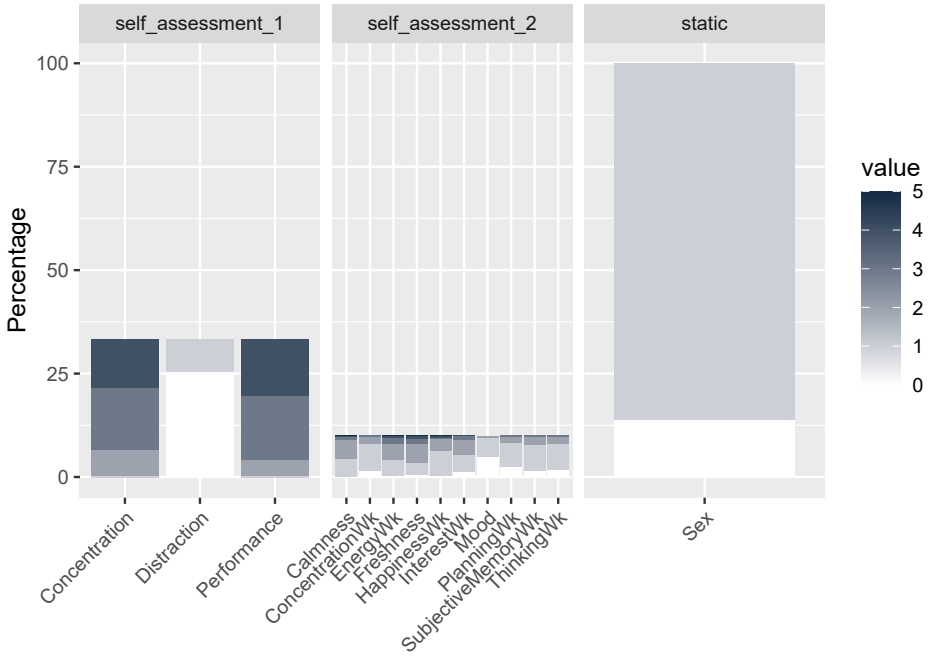


Figure A.5.: Description of the self-assessment questionnaires of subgroup B at week 9.

Figure A.6.: Description of the self-assessment questionnaires of subgroup C at week 9.

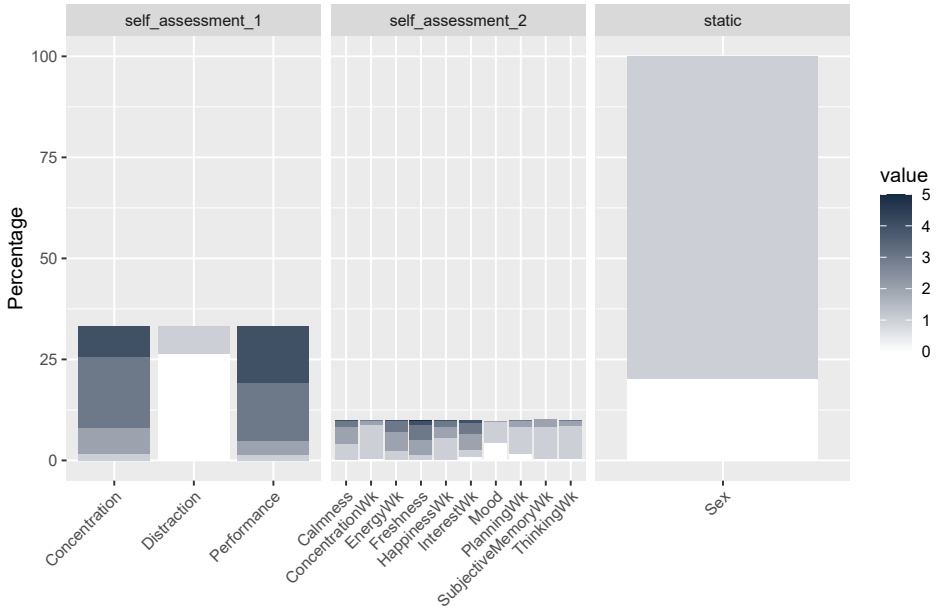


Figure A.7.: Description of the self-assessment questionnaires of subgroup D at week 9.

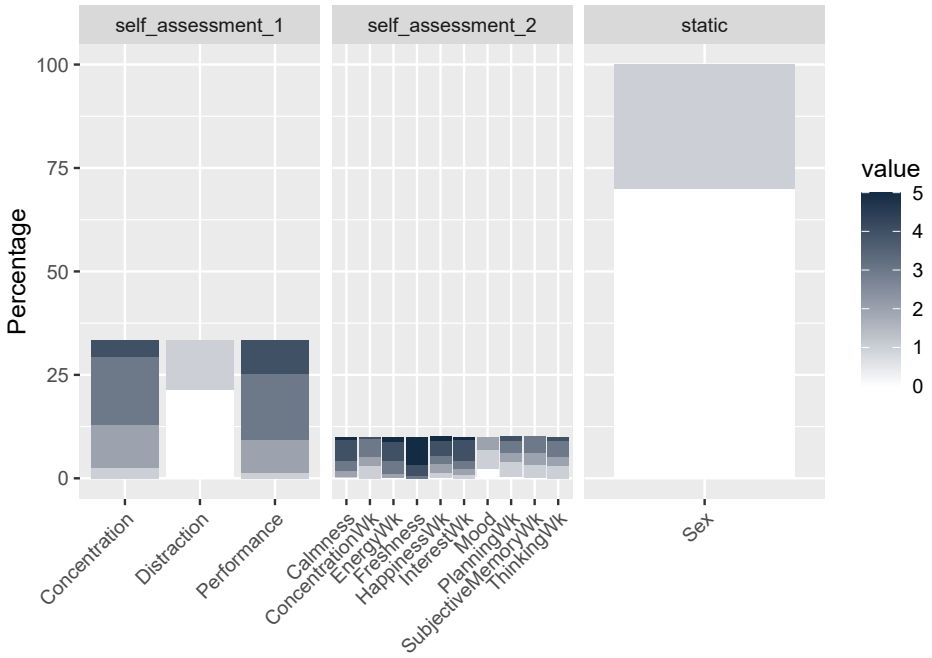


Figure A.8.: Description of the self-assessment questionnaires of subgroup E at week 9.

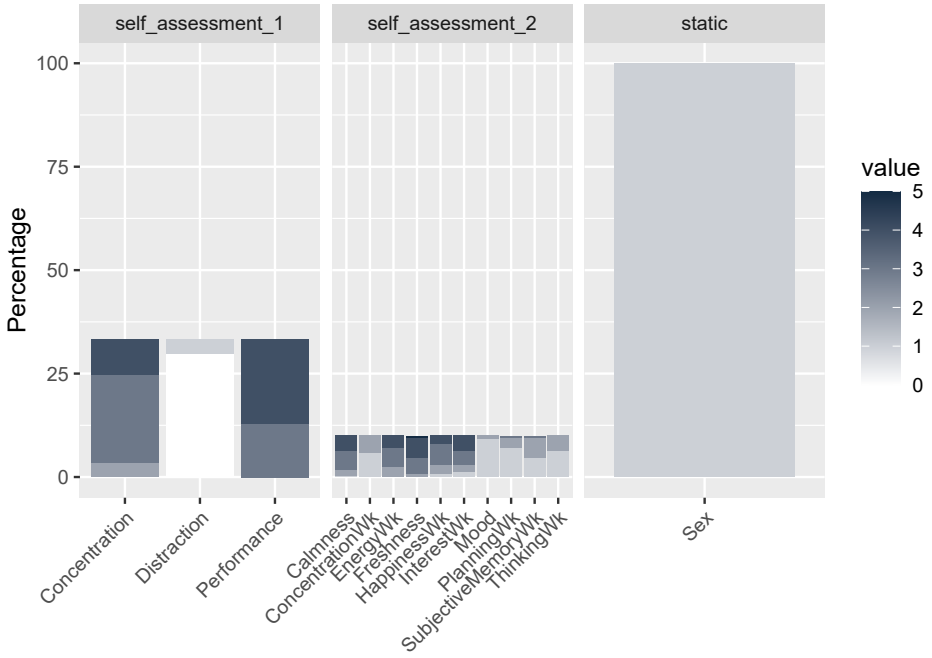


Figure A.9.: Description of the self-assessment questionnaires of subgroup F at week 9.

A.2.3. Subgroup Evolution

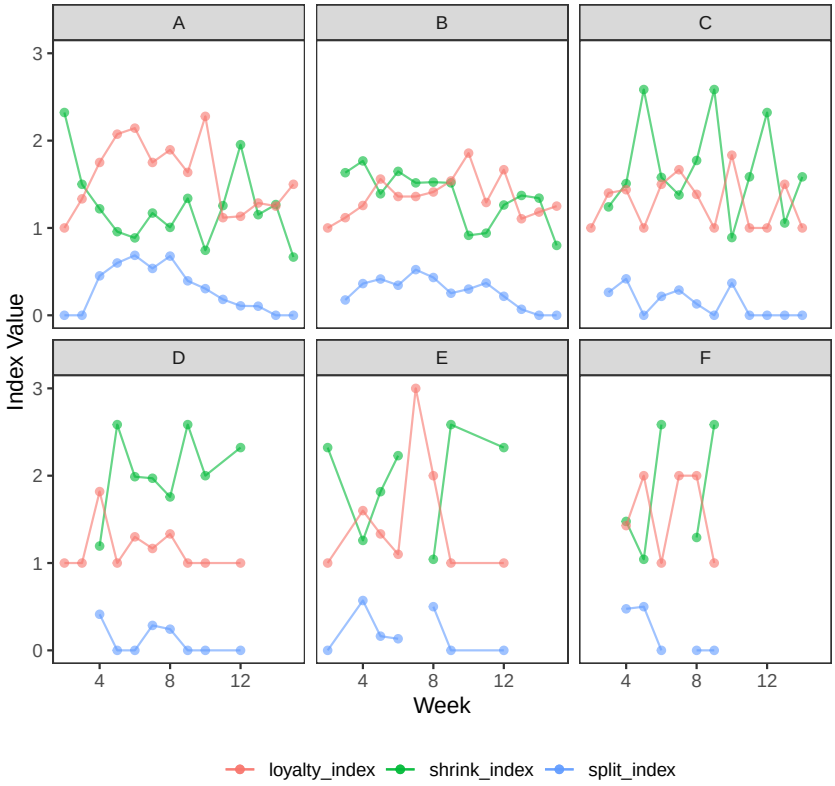


Figure A.10.: Evolution of the shrink, split and loyalty index per subgroup and per timepoint.

Bibliography

[1] Ab Ghani, Nur Laila, Aziz, Izzatdin Abdul, and AbdulKadir, Said Jadid. "Subspace Clustering in High-Dimensional Data Streams: A Systematic Literature Review". In: *Computers, Materials and Continua* 75.2 (2023), pp. 4649–4668. ISSN: 15462226. DOI: 10.32604/cmc.2023.035987.

[2] Alhajjar, Elie. "Network Science". In: *Mathematics in Cyber Research*. Vol. 41. Boca Raton: Chapman and Hall/CRC, 2022, pp. 207–232. ISBN: 9781000542691. DOI: 10.1201/9780429354649-6.

[3] Ali, Hafiz Tiomoko et al. "Latent Heterogeneous Multilayer Community Detection". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 8142–8146. ISBN: 978-1-4799-8131-1. DOI: 10.1109/ICASSP.2019.8683574.

[4] Alimadadi, Fatemeh, Khadangi, Ehsan, and Bagheri, Alireza. "Community detection in facebook activity networks and presenting a new multilayer label propagation algorithm for community detection". In: *International Journal of Modern Physics B* 33.10 (2019). ISSN: 17936578. DOI: 10.1142/S0217979219500899.

[5] Amelio, Alessia and Pizzuti, Clara. "Is Normalized Mutual Information a Fair Measure for Comparing Community Detection Methods?" In: Aug. 2015. DOI: 10.1145/2808797.2809344.

[6] Ankerst, Mihael et al. "OPTICS: Ordering points to identify the clustering structure". In: *ACM Sigmod record* 28.2 (1999), pp. 49–60.

[7] Atzmueller, Martin and Puppe, Frank. "SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery". In: *Knowledge Discovery in Databases: PKDD 2006*. Springer Berlin Heidelberg, 2006, pp. 6–17. ISBN: 9783540460480. DOI: 10.1007/11871637_6.

[8] Aynaud, Thomas and Guillaume, Jean-Loup. "Static community detection algorithms for evolving networks". In: *WiOpt*. 2010, pp. 513–9.

[9] Barabási, Albert-László and Albert, Réka. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (1999), pp. 509–512. DOI: 10.1126/science.286.5439.509. eprint: https://www.science.org/doi/pdf/10.1126/science.286.5439.509.

[10] Barabási, Albert-László and Pósfai, Márton. *Network science*. Cambridge: Cambridge University Press, 2016. ISBN: 9781107076266 1107076269. URL: http://barabasi.com/networksciencebook/.

[11] Baytas, Inci M. et al. "Patient Subtyping via Time-Aware LSTM Networks". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. Halifax, NS, Canada: Association for Computing Machinery, 2017, pp. 65–74. ISBN: 9781450348874. DOI: 10.1145/3097983.3097997.

[12] Bech, P. et al. “The sensitivity and specificity of the Major Depression Inventory, using the Present State Examination as the index of diagnostic validity”. In: *Journal of Affective Disorders* 66.2-3 (2001), pp. 159–164.

[13] Beierlein, Volker et al. “Messung der gesundheitsbezogenen Lebensqualität mit dem SF-8”. In: *Diagnostica* (2012).

[14] Berron, David et al. “Feasibility of Digital Memory Assessments in an Unsupervised and Remote Study Setting”. In: *Frontiers in Digital Health* 4.May (2022), pp. 1–14. DOI: 10.3389/fdgth.2022.892997.

[15] Biernacki, Christophe, Celeux, Gilles, and Govaert, Gérard. “Assessing a mixture model for clustering with the integrated completed likelihood”. In: *IEEE transactions on pattern analysis and machine intelligence* 22.7 (2000), pp. 719–725.

[16] Bolón-canedo, V. “A framework for cost-based feature selection”. In: *Pattern Recognition* 47.7 (2014), pp. 2481–2489. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2014.01.008.

[17] Bolón-Canedo, Verónica et al. “Real-Time Tear Film Classification Through Cost-Based Feature Selection”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 78–98. ISBN: 9783319275437. DOI: 10.1007/978-3-319-27543-7_4.

[18] Brähler, Elmar and Scheer, Jörn W. “Scaling of psychosomatic by means of the Giessen inventory (GBB)(author’s transl)”. In: *Psychotherapie, medizinische Psychologie* 29.1 (1979), p. 14.

[19] Braveman, P. “Defining equity in health”. In: *Journal of Epidemiology; Community Health* 57.4 (2003), pp. 254–258. ISSN: 0143-005X. DOI: 10.1136/jech.57.4.254.

[20] Bródka, Piotr et al. “Quantifying layer similarity in multiplex networks: A systematic study”. In: *Royal Society Open Science* 5.8 (2018). ISSN: 20545703. DOI: 10.1098/rsos.171747. arXiv: 1711.11335.

[21] Campello, Ricardo JGB et al. “Hierarchical density estimates for data clustering, visualization, and outlier detection”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 10.1 (2015), pp. 1–51.

[22] Carleton, Tamma A. and Hsiang, Solomon M. “Social and economic impacts of climate”. In: *Science* 353.6304 (2016), aad9837. DOI: 10.1126/science.aad9837. eprint: https://www.science.org/doi/pdf/10.1126/science.aad9837.

[23] Cazabet, Rémy et al. *Encyclopedia of Social Network Analysis and Mining*. Ed. by Reda Alhajj and Jon Rokne. Springer New York, 2017. ISBN: 978-1-4614-7163-9. DOI: 10.1007/978-1-4614-7163-9.

[24] Day, William HE and Edelsbrunner, Herbert. “Efficient algorithms for agglomerative hierarchical clustering methods”. In: *Journal of classification* 1.1 (1984), pp. 7–24.

[25] De Domenico, Manlio et al. “Mathematical formulation of multilayer networks”. In: *Physical Review X* 3.4 (2014), pp. 1–15. ISSN: 21603308. DOI: 10.1103/PhysRevX.3.041022. arXiv: 1307.4977.

[26] De Meo, Pasquale et al. "Generalized Louvain method for community detection in large networks". In: *2011 11th International Conference on Intelligent Systems Design and Applications*. August. IEEE, 2011, pp. 88–93. ISBN: 978-1-4577-1676-8. DOI: 10.1109/ISDA.2011.6121636. URL: http://ieeexplore.ieee.org/document/6121636/.

[27] Dempster, Arthur P, Laird, Nan M, and Rubin, Donald B. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[28] Dennis, John M. et al. "Disease progression and treatment response in data-driven subgroups of type 2 diabetes compared with models based on simple clinical features: an analysis using clinical trial data". In: *The Lancet Diabetes and Endocrinology* 7.6 (2019), pp. 442–451. ISSN: 22138595. DOI: 10.1016/S2213-8587(19)30087-7.

[29] Dianati, Navid. "Unwinding the hairball graph: Pruning algorithms for weighted complex networks". In: *Physical Review E* 93.1 (2016), p. 012304. ISSN: 2470-0045. DOI: 10.1103/PhysRevE.93.012304. arXiv: 1503.04085.

[30] Duivesteijn, Wouter et al. "Subgroup discovery meets Bayesian networks - An Exceptional Model Mining approach". In: *Proceedings - IEEE International Conference on Data Mining, ICDM* (2010), pp. 158–167. ISSN: 15504786. DOI: 10.1109/ICDM.2010.53.

[31] Elhamifar, Ehsan and Vidal, Rene. "Sparse subspace clustering: Algorithm, theory, and applications". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.11 (2013), pp. 2765–2781. ISSN: 01628828. DOI: 10.1109/TPAMI.2013.57. arXiv: 1203.1005.

[32] Fliege, Herbert et al. "The Perceived Stress Questionnaire (PSQ) reconsidered: validation and reference values from different clinical and healthy adult samples". In: *Psychosomatic medicine* 67.1 (2005), pp. 78–88.

[33] Fouladvand, Sajjad et al. "Graph-based clinical recommender: Predicting specialists procedure orders using graph representation learning". In: *Journal of Biomedical Informatics* 143 (2023), p. 104407. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2023.104407. URL: https://www.sciencedirect.com/science/article/pii/S1532046423001284.

[34] Fruchterman, Thomas M. J. and Reingold, Edward M. "Graph drawing by force-directed placement". In: *Software: Practice and Experience* 21.11 (1991), pp. 1129–1164. DOI: 10.1002/spe.4380211102.

[35] Fuller, Thomas et al. "Cognitive behavioural therapy for tinnitus". In: *Cochrane Database of Systematic Reviews* 2020.1 (Jan. 2020). DOI: 10.1002/14651858.cd012614.pub2.

[36] Fuller, Thomas E. et al. "The Fear of Tinnitus Questionnaire: Toward a Reliable and Valid Means of Assessing Fear in Adults with Tinnitus". In: *Ear and hearing* 40.6 (Apr. 2019), pp. 1467–1477. DOI: 10.1097/aud.0000000000000728.

[37] Gao, Xubo et al. "Particle Competition for Multilayer Network Community Detection". In: *Proceedings of the 2019 11th International Conference on Machine Learning and Computing - ICMLC '19*. Vol. Part F1481. New York, New York, USA: ACM Press, 2019, pp. 75–80. ISBN: 9781450366007. DOI: 10.1145/3318299.3318320.

[38] Genitsaridi, Eleni et al. “A review and a framework of variables for defining and characterizing tinnitus subphenotypes”. In: *Brain Sciences* 10.12 (2020), pp. 1–21. ISSN: 20763425. DOI: 10.3390/brainsci10120938.

[39] Ghawi, Raji and Pfeffer, Jürgen. “A community matching based approach to measuring layer similarity in multilayer networks”. In: *Social Networks* 68.April 2021 (2022), pp. 1–14. ISSN: 03788733. DOI: 10.1016/j.socnet.2021.04.004.

[40] Ginsburg, Geoffrey S and Phillips, Kathryn A. “Precision medicine: from science to value”. In: *Health Affairs* 37.5 (2018), pp. 694–701.

[41] Gislén, Anna et al. “Superior underwater vision in a human population of sea gypsies”. In: *Current Biology* 13.10 (2003), pp. 833–836.

[42] Goebel, G and Hiller, W. “Tinnitus-Fragebogen (TP)-Handanweisung”. In: *Hogrefe, Göttingen* (1998).

[43] Goebel, Gerhard and Hiller, Wolfgang. *Tinnitus-Fragebogen:(TF); ein Instrument zur Erfassung von Belastung und Schweregrad bei Tinnitus; Handanweisung*. Hogrefe, Verlag für Psychologie, 1998.

[44] Grassia, Marco, Domenico, Manlio De, and Mangioni, Giuseppe. “mGNN : Generalizing the Graph Neural Networks to the Multilayer Case”. In: (2021), pp. 1–10. arXiv: arXiv:2109.10119v2.

[45] Greimel, Karoline V. et al. “Ist Tinnitus meßbar? Methoden zur Erfassung tinnitusspezifischer Beeinträchtigungen und Präsentation des Tinnitus-Beeinträchtigungs-Fragebogens (TBF-12)”. In: *HNO* 47.3 (1999), pp. 196–201.

[46] Guns, Tias and Dries, Anton. “MiningZinc: A modeling language for constraint-based mining”. In: *IJCAI International Joint Conference on Artificial Intelligence* (2013), pp. 1365–1372. ISSN: 10450823.

[47] Hanteer, Obaida et al. “Community detection in multiplex networks”. In: *arXiv* (2019). arXiv: 1910.07646.

[48] Hartigan, John A and Wong, Manchek A. “Algorithm AS 136: A k-means clustering algorithm”. In: *Journal of the royal statistical society. series c (applied statistics)* 28.1 (1979), pp. 100–108.

[49] Hautzinger, Martin and Bailer, Maja. *ADS, Allgemeine Depressions Skala. Manual*. Tech. rep. 1993.

[50] Hielscher, Tommy et al. “Identifying relevant features for a multi-factorial disorder with constraint-based subspace clustering”. In: *Proceedings - IEEE Symposium on Computer-Based Medical Systems* 2016-August (2016), pp. 207–212. ISSN: 10637125. DOI: 10.1109/CBMS.2016.42.

[51] Hiller, Wolfgang and Goebel, Gerhard. “Beliefs and attitudes among Swedish workers regarding the risk of hearing loss”. In: *International Journal of Audiology* 43.10 (Jan. 2004), pp. 600–604. DOI: 10.1080/14992020400050077.

[52] Hoerhold, Michael, Klapp, Burghard F, and Schimmack, Ulrich. “Testing the invariance and hierarchy of a multidimensional model of mood by means of repeated measurement with student and patient sample”. In: *Z med Psychol* 1 (1993), pp. 27–35.

[53] Hornik, Kurt, Stinchcombe, Maxwell, and White, Halbert. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 08936080. DOI: 10.1016/0893-6080(89)90020-8.

[54] Huang, Xinyu et al. *A survey of community detection methods in multilayer networks*. Springer US, 2020. ISBN: 1061802000. DOI: 10.1007/s10618-020-00716-6.

[55] Huang, Yuming et al. "Community Detection and Improved Detectability in Multiplex Networks". In: *arXiv* 7.3 (2019), pp. 1697–1709. ISSN: 23318422.

[56] Hubert, Lawrence and Arabie, Phipps. "Comparing partitions". In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 1432-1343. DOI: 10.1007/bf01908075.

[57] Huckvale, Kit, Venkatesh, Svetha, and Christensen, Helen. "Toward clinical digital phenotyping: a timely opportunity to consider purpose, quality, and safety". In: *npj Digital Medicine* 2.1 (2019). ISSN: 2398-6352. DOI: 10.1038/s41746-019-0166-1.

[58] Interdonato, Roberto et al. "Multilayer network simplification: Approaches, models and methods". In: *arXiv* May (2020). ISSN: 23318422. arXiv: 2004.14808.

[59] Jaccard, Paul. "Etude de la distribution florale dans une portion des Alpes et du Jura". In: *Bulletin de la Societe Vaudoise des Sciences Naturelles* 37 (Jan. 1901), pp. 547–579. DOI: 10.5169/seals-266450.

[60] Jastreboff, Pawel J. "Phantom auditory perception (tinnitus): mechanisms of generation and perception". In: *Neuroscience Research* 8.4 (1990), pp. 221–254. ISSN: 01680102.

[61] Jerdee, Maximilian, Kirkley, Alec, and Newman, M E J. "for classification and community detection". In: (), pp. 1–14. arXiv: arXiv:2307.01282v1.

[62] Kachuee, Mohammad et al. *Cost-Sensitive Diagnosis and Learning Leveraging Public Health Data*. 2019. arXiv: 1902.07102.

[63] Kelkar, Bhagyashri A and Rodd, Sunil F. "Subspace clustering—A survey". In: *Data Management, Analytics and Innovation*. Springer, 2019, pp. 209–220.

[64] Kim, Myungjun, Nam, Yonghyun, and Shin, Hyunjung. "An inference method from multi-layered structure of biomedical data". In: *BMC Medical Informatics and Decision Making* 17.S1 (May 2017). DOI: 10.1186/s12911-017-0450-4.

[65] Kipf, Thomas N. and Welling, Max. "Semi-supervised classification with graph convolutional networks". In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (2017), pp. 1–14. arXiv: 1609.02907.

[66] Kramer, Jordan et al. "Analysis of Medical Data Using Community Detection on Inferred Networks". In: *IEEE Journal of Biomedical and Health Informatics* 24.11 (2020), pp. 3136–3143. ISSN: 21682208. DOI: 10.1109/JBHI.2020.3003827.

[67] Kroenke, Kurt, Spitzer, Robert L., and Williams, Janet B. W. "The PHQ-9". In: *Journal of General Internal Medicine* 16.9 (Sept. 2001), pp. 606–613. DOI: 10.1046/j.1525-1497.2001.016009606.x.

[68] Lavrač, Nada et al. "Decision support through subgroup discovery: Three case studies and the lessons learned". In: *Mach. Learn.* 57.1/2 (Oct. 2004), pp. 115–143.

[69] Lee, Bohyun et al. “Heterogeneous Multi-Layered Network Model for Omics Data Integration and Analysis”. In: 10.January (2020), pp. 1–11. DOI: 10.3389/fgene.2019.01381.

[70] Levenstein, Susan et al. “Development of the Perceived Stress Questionnaire: a new tool for psychosomatic research”. In: *Journal of psychosomatic research* 37.1 (1993), pp. 19–32.

[71] Liang, Yunji, Zheng, Xiaolong, and Zeng, Daniel D. “A survey on big data-driven digital phenotyping of mental health”. In: *Information Fusion* 52.July 2018 (2019), pp. 290–307. ISSN: 15662535. DOI: 10.1016/j.inffus.2019.04.001.

[72] Liu, Huawen, Liu, Lei, and Zhang, Huijie. “Feature Selection Using Mutual Information: An Experimental Study”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 235–246. ISBN: 9783540891970. DOI: 10.1007/978-3-540-89197-0_24.

[73] Lucas G. S. Jeub Marya Bazzi, Inderjit S. Jutla and Mucha, Peter J. *A generalized Louvain method for community detection implemented in MATLAB*. http://netwiki.amath.unc.edu/GenLouvain/GenLouvain. Lucas G. S. Jeub, Marya Bazzi, Inderjit S. Jutla, and Peter J. Mucha (2011-2019).

[74] Maes, Iris HL et al. “Tinnitus: A Cost Study”. In: *Ear and Hearing* 34.4 (2013), pp. 508–514.

[75] McCombe, A. et al. “Guidelines for the grading of tinnitus severity: the results of a working group commissioned by the British Association of Otolaryngologists, Head and Neck Surgeons, 1999”. In: *Clinical Otolaryngology and Allied Sciences* 26.5 (2001), pp. 388–393.

[76] Meikle, M. B., Henry, J. A., Griest, S. E., et al. “The Tinnitus Functional Index”. In: *Ear & Hearing* 33.2 (2012), pp. 153–176.

[77] Mucha, Peter J et al. “Community structure in time-dependent, multiscale, and multiplex networks”. In: *Science* 328.5980 (2010), pp. 876–878. ISSN: 00368075. DOI: 10.1126/science.1184819. arXiv: 0911.1824.

[78] Neuhäuser, Markus. “Wilcoxon–Mann–Whitney Test”. In: *International Encyclopedia of Statistical Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1656–1658. ISBN: 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_615. URL: https://doi.org/10.1007/978-3-642-04898-2_615.

[79] Newman, M. E.J. “Equivalence between modularity optimization and maximum likelihood methods for community detection”. In: *Physical Review E* 94.5 (2016), pp. 1–8. ISSN: 24700053. DOI: 10.1103/PhysRevE.94.052315.

[80] Newman, Mark. *Networks*. Oxford University Press, July 2018. ISBN: 9780198805090. DOI: 10.1093/oso/9780198805090.001.0001. URL: https://doi.org/10.1093/oso/9780198805090.001.0001.

[81] Niemann, Uli et al. “Phenotyping chronic tinnitus patients using self-report questionnaire data: cluster analysis and visual comparison”. In: *Scientific Reports* 10.1 (2020), pp. 1–10. ISSN: 20452322. DOI: 10.1038/s41598-020-73402-8. URL: https://doi.org/10.1038/s41598-020-73402-8.

[82] Palla, Gergely et al. “Quantifying social group evolution”. In: *Nature* 446.7136 (2007), pp. 664–7. ISSN: 14764687. DOI: 10.1038/nature05670.

[83] Pan, Zhisong, Hu, Guyu, and Li, Dong. "Detecting communities from multi-layer networks". In: *Proceedings of the International Conference on Intelligent Science and Technology - ICIST '18*. New York, New York, USA: ACM Press, 2018, pp. 6–11. ISBN: 9781450364614. DOI: 10.1145/3233740.3233742.

[84] Paoletti, Giancarlo et al. "Subspace clustering for action recognition with covariance representations and temporal pruning". In: *ICPR*. 2021, pp. 6035–42.

[85] Petti, Manuela and Farina, Lorenzo. "Network medicine for patients' stratification: From single-layer to multi-omics". In: *WIREs Mechanisms of Disease* 15.6 (2023), e1623. DOI: https://doi.org/10.1002/wsbm.1623. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wsbm.1623.

[86] Pradana, C., Kusumawardani, S. S., and Permanasari, A. E. "Comparison Clustering Performance Based on Moodle Log Mining". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 722. 1. 2020. DOI: 10.1088/1757-899X/722/1/012012.

[87] Pramanik, S. et al. "Discovering Community Structure in Multilayer Networks". In: *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 2017, pp. 611–620. DOI: 10.1109/DSAA.2017.71.

[88] Puga, Clara et al. "Discovery of Patient Phenotypes through Multi-layer Network Analysis on the Example of Tinnitus". In: *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)* (2021), pp. 1–10. DOI: 10.1109/dsaa53316.2021.9564158.

[89] Puga, Clara et al. "Juxtaposing Medical Centres Using Different Questionnaires Through Score Predictors". In: *Frontiers in Neuroscience* (2022). DOI: 10.3389/fnins.2022.818686.

[90] Puga, Clara et al. "A cost-based multi-layer network approach for the discovery of patient phenotypes". In: *International Journal of Data Science and Analytics* (July 2023). ISSN: 2364-4168. DOI: 10.1007/s41060-023-00431-7.

[91] Raynal, Louis, Hoffmann, Till, and Onnela, Jukka Pekka. "Cost-based Feature Selection for Network Model Choice". In: *Journal of Computational and Graphical Statistics* 32.3 (2023), pp. 1109–1118. ISSN: 15372715. DOI: 10.1080/10618600.2022.2151453. arXiv: 2101.07766.

[92] Rodrigo, Hansapani et al. "Exploratory Data Mining Techniques (Decision Tree Models) for Examining the Impact of Internet-Based Cognitive Behavioral Therapy for Tinnitus: Machine Learning Approach". In: *Journal of Medical Internet Research* 23.11 (Nov. 2021), e28999. ISSN: 1438-8871. DOI: 10.2196/28999.

[93] Rossetti, Giulio and Cazabet, Rémy. "Community Discovery in Dynamic Networks". In: *ACM Computing Surveys* 51.2 (2018), pp. 1–37. ISSN: 0360-0300. DOI: 10.1145/3172867.

[94] Saeys, Yvan, Inza, Iñaki, and Larrañaga, Pedro. "A review of feature selection techniques in bioinformatics". In: *Bioinformatics* 23.19 (Aug. 2007), pp. 2507–2517. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btm344.

[95] Scarselli, Franco et al. "Computational capabilities of graph neural networks". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 81–102. ISSN: 10459227. DOI: 10.1109/TNN.2008.2005141.

[96] Scarselli, Franco et al. "The graph neural network model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. ISSN: 10459227. DOI: 10.1109/TNN.2008.2005605.

[97] Schlee, Winfried et al. "Using Big Data to Develop a Clinical Decision Support System for Tinnitus Treatment". In: *The Behavioral Neuroscience of Tinnitus*. Springer International Publishing, 2021, pp. 175–89. ISBN: 9783030855024. DOI: 10.1007/7854_2021_229.

[98] Scholler, Gudrun, Fliege, Herbert, and Klapp, Burghard F. "Questionnaire of self-efficacy, optimism and pessimism: reconstruction, selection of items and validation of an instrument by means of examinations of clinical samples". In: *Psychotherapie, Psychosomatik, medizinische Psychologie* 49.8 (1999), pp. 275–283.

[99] Seydel, Claudia et al. "Gender and Chronic Tinnitus". In: *Ear and Hearing* 34.5 (2013), pp. 661–672.

[100] Shapiro, S. S. and Wilk, M. B. "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3-4 (1965), pp. 591–611.

[101] Al-sharoa, Esraa, Al-khassaweneh, Mahmood, and Aviyente, Selin. "Tensor Based Temporal and Multilayer Community Detection for Studying Brain Dynamics During Resting State fMRI". In: *IEEE Transactions on Biomedical Engineering* 66.3 (2019), pp. 695–709. DOI: 10.1109/TBME.2018.2854676.

[102] Shi, Jianbo and Malik, Jitendra. "Normalized cuts and image segmentation". In: *PAMI* 22.8 (2000), pp. 888–905. ISSN: 01628828. DOI: 10.1109/34.868688.

[103] Shun, Julian et al. "Parallel local graph clustering". In: *VLDB Endowment* 9.12 (2016), pp. 1041–52. ISSN: 21508097. DOI: 10.14778/2994509.2994522. arXiv: 1604.07515.

[104] Society, German et al. "S3 Guideline : Chronic Tinnitus". In: July (2022), pp. 795–827. DOI: 10.1007/s00106-022-01207-4.

[105] Steinhaeuser, Karsten and Chawla, Nitesh V. "Identifying and evaluating community structure in complex networks". In: *Pattern Recognition Letters* 31.5 (2010), pp. 413–421. ISSN: 01678655. DOI: 10.1016/j.patrec.2009.11.001.

[106] Steinley, Douglas, Brusco, Michael J, and Hubert, Lawrence. "The variance of the adjusted Rand index." In: *Psychological methods* 21.2 (2016), p. 261.

[107] Student. "The probable error of a mean". In: *Biometrika* (1908). (Author is 'William Sealey Gosset' naming himself 'Student'), pp. 1–25.

[108] Tantardini, Mattia et al. "Comparing methods for comparing networks". In: *Scientific Reports* 9.1 (2019), pp. 1–19. ISSN: 20452322. DOI: 10.1038/s41598-019-53708-y.

[109] Traag, V. A., Waltman, L., and Eck, N. J. van. "From Louvain to Leiden: guaranteeing well-connected communities". In: *Scientific Reports* 9.1 (2019), pp. 1–12. ISSN: 20452322. DOI: 10.1038/s41598-019-41695-z. arXiv: 1810.08473.

[110] Tsitsulin, Anton et al. "NetLSD: Hearing the shape of a graph". In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2018), pp. 2347–2356.

[111] Wale, Nikil and Karypis, George. "Comparison of Descriptor Spaces for Chemical Compound Retrieval and Classification". In: (2006). ISSN: 1550-4786. DOI: 10.1109/icdm.2006.39.

[112] Wang, Liang et al. "Quantifying community evolution in developer social networks". In: *ESEC/FSE 2022 - Proceedings of the 30th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering* (2022), pp. 157–169. DOI: 10.1145/3540250.3549106.

[113] Wang, Shenghui and Koopman, Rob. "Clustering articles based on semantic similarity". In: *Scientometrics* 111.2 (2017), pp. 1017–1031. ISSN: 15882861. DOI: 10.1007/s11192-017-2298-x. arXiv: 1702.04946.

[114] Wang, Yanshan et al. "Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records". In: *Journal of Biomedical Informatics* 102.December 2019 (2020), p. 103364. DOI: 10.1016/j.jbi.2019.103364.

[115] Wang, Yanshan et al. "Unsupervised machine learning for the discovery of latent disease clusters and patient subgroups using electronic health records". In: *Journal of Biomedical Informatics* 102 (2020), p. 103364. ISSN: 1532-0464. DOI: https://doi.org/10.1016/j.jbi.2019.103364. URL: https://www.sciencedirect.com/science/article/pii/S1532046419302849.

[116] Wasserman, Stanley and Faust, Katherine. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.

[117] Weir, William H. et al. "Multilayer Modularity Belief Propagation to Assess Detectability of Community Structure". In: *SIAM Journal on Mathematics of Data Science* 2.3 (2020), pp. 872–900. DOI: 10.1137/19m1279812. arXiv: 1908.04653.

[118] Wu, Zonghan et al. "A Comprehensive Survey on Graph Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (2021), pp. 4–24. ISSN: 21622388. DOI: 10.1109/TNNLS.2020.2978386. arXiv: 1901.00596.

[119] Yang, Liang et al. "Toward Unsupervised Graph Neural Network : Interactive Clustering and Embedding via Optimal Transport". In: Icdm (2020), pp. 1358–1363. DOI: 10.1109/ICDM50108.2020.00177.

[120] Yu, Guo, Witten, Daniela, and Bien, Jacob. *Controlling Costs: Feature Selection on a Budget*. 2020. arXiv: 1910.03627.

[121] Zeman, Florian et al. "Tinnitus handicap inventory for evaluating treatment effects: which changes are clinically relevant?" In: *Otolaryngology–Head and Neck Surgery* 145.2 (2011), pp. 282–287.

[122] Zhang, Tian, Ramakrishnan, Raghu, and Livny, Miron. "BIRCH: an efficient data clustering method for very large databases". In: *ACM sigmod record* 25.2 (1996), pp. 103–114.

[123] Zhang, Xi et al. "Data-Driven Subtyping of Parkinson's Disease Using Longitudinal Clinical Records: A Cohort Study". In: *Scientific Reports* 9.1 (Jan. 2019). DOI: 10.1038/s41598-018-37545-z.

[124] Zheng, Jianwei et al. "Enhanced low-rank constraint for temporal subspace clustering and its acceleration scheme". In: *Pattern Recognition* 111 (2021), p. 107678.

Ehrenerklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, den