university of groningen

faculty of science and engineering

From Data to Knowledge: Temporal Rule Learning from Electronic Health Records for Modelling Patient Histories

Nicolas Schulz

university of groningen

faculty of science and engineering

University of Groningen

From Data to Knowledge: Temporal Rule Learning from Electronic Health Records for Modelling Patient Histories

Nicolas Schulz (s3141608)

October 18, 2023

Master's Thesis

To fulfill the requirements for the degree of Master of Science
in Artificial Intelligence at University of Groningen
under the supervision of:
K. (Kerstin) Bunte, Prof. Dr. (Computer Science, University of Groningen)
and
M.B. (Marleen) Schippers, Dr. (Artificial Intelligence, University of Groningen)

Contents

Acknowledgments

I would like to thank all individuals who enabled me to work on this interesting topic during my master's thesis research project, as I sincerely enjoyed working on the subject. Especially, I would like to genuinely thank my University supervisors Marleen Schippers and Kerstin Bunte, who made it possible to seek an external thesis project in another country. I am grateful for the organizational and methodological support that you provided to me during the course of this project, as it turned this challenging task into a smooth experience that I will keep in good memory.

I would also like to thank the entire Institute for Applied Medical Informatics in Hamburg, especially my supervisor Christopher Gundler, for their open culture and bright minds. Only through continuous support in meetings and thought-provoking discussions did this project take its final form.

Last but not least, I would like to thank my father, Norbert Schulz, for proof-reading and giving me feedback on this thesis.

Abstract

Large patient data is gathered continuously at every hospital. If the digital representation thereof is in the form of a standardized Electronic Health Record (EHR), reusable and scalable analytics pipelines using Artificial Intelligence can be developed to uncover data-driven insights in patient populations. In a sensitive context such as medical decision-support, ideally, these insights are generated with interpretable models and result in a visual representation which can be assessed by medical professionals without technical expertise. Digitization of the health care sector lags behind other economy sectors due to industry specific obstacles like patient privacy, patient safety, data availability and a lack of allocated resources. Due to these reasons, neither a nationwide adaptation of EHR nor the following large-scale data analytics pipelines are currently an everyday reality in European hospitals.

However, in this thesis the foundation is laid to robustly extract data from standardized EHR, transform it into a time series, learn temporal rules from these hospital encounter histories and visualize the results in a hierarchical and directed graph. Three models of computational intelligence, namely (1) a baseline transition matrix, (2) a Temporal Association Rule Mining and (3) a Dynamic Bayesian Network structure learning approach, are implemented in separate pipelines and the results are compared across models, datasets and hyperparameters using real-world lung cancer patient data from Germany.

The baseline transition matrix was found to be suitable for exhaustive representations of small datasets ($N \leq 10$), Temporal Association Rule mining is computationally the most efficient and thus most suitable for very large data ($N \geq 10000$) and the Dynamic Bayesian Network structure learning approach was identified to be the most robust computational model and resulted in the most meaningful rules for medical decision-making. While all models were found to have their potential use cases, if sufficient computational resources are available, learning the structure of a Dynamic Bayesian Network from data in cooperation with a medical expert should be the preferred option.

Table 1: List of relevant Abbreviations.

Abbreviation	**Meaning**
UKE	Universitätsklinikum Hamburg-Eppendorf
IAM	Institute of Applied Medical Informatics
DAG	Directed Acyclical Graph
JPD	Joint Probability Distribution
SDG	Synthetic Data Generation
EHR	Electronic Health Records
FHIR	Fast Healthcare Interpretable Resources
OMOP	Observational Medical Outcomes Partnership
CDM	Common Data Model
PADASER	Publicly Available Data Approach to the Realistic Synthetic EHR
ESMO	European Society of Medical Oncology
MM	Markov Model
ARM	Association Rule Mining
TARM	Temporal Association Rule Mining
DBN	Dynamic Bayesian Network
SVAR	Structural Vector Autoregressive Model
SCM	Structural Causal Model
SEM	Structural Equation Modeling
GCM	Graphical Causal Model

Chapter 1

Introduction

Recent years have brought immense technological advancements by Data Science and Artificial Intelligence (AI), progressing the effectiveness and precision of work by use of Big Data in industry sectors like autonomous driving [1, 2], finance [3, 4, 5] and marketing [6, 7, 8]. However, the healthcare industry received the least attention and benefit so far, which becomes evident through official government reports in Germany [9], the country this project is performed in.

Besides the healthcare sector holding a special industry standard in terms of customer privacy and safety [10, 11, 12], the development of AI in healthcare still needs to gain more traction compared to the rest of industry [9]. The arguably most prevalent use case of AI in healthcare are computer vision tasks such as medical image analysis [13, 14, 15], but the digital use of local patient records is less established [16]. This could be due to the fact that for the efficient usage of patient records technological and policy requirements have to be in place. For example, all medical information of a patient has to be in a centralized and digital resource. Also, this digital resource has to be standardized to enable second-use analytical tools to be efficient and scalable. However, given these prerequisites, the large patient data which continuously accumulates within a hospital could be used to analyze disease, comorbidity, treatment, prescription and recovery history of anonymous patient subpopulations and thereby serve as a knowledge base and mirror the hospital's daily work.

The past resistance observed in the healthcare sector to incorporate analytic tools of AI on patient records and thereby close to medical decision-making stands to reason, as trust in such a system has to be established beforehand. In order to establish trust, the transparency of the model is non-negotiable. In other words, it is imperative that healthcare providers, policymakers, and patients understand the decision procedure of the AI model and have access to information about how the model operates. These so-called "white-box" models are essential to ensure that the decisions made by the system are well-founded and ethically sound. Actually, with the General Data Protection Regulation (GDPR) in force in Europe [17], an individual has the right to an explanation when an algorithm makes a decision about her or him [18]. While AI models should and likely never will make medical decisions without a human expert, the need for transparency is highlighted once more by legal restrictions. Therefore, to ensure that the benefits of AI are introduced to the realm of medical patient records safely and ethically, the first step necessary is to approach the issue with an interpretable and transparent white-box model.

If the technological prerequisites are fulfilled and the transparency of the model is guaran-

teed, the use cases and benefits of AI tools can be wide-ranging. Firstly, statistical analyses of patient subpopulations can extract useful knowledge for clinicians and researchers while securing the privacy of an individual's sensible information. Secondly, the extracted knowledge can be used to improve patient treatment on a structural level. Lastly, the extracted knowledge can be used to mirror the hospital's inner workings, validate or falsify cause-effect relationships and uncover unexpected dependencies between medications, treatments and disease progression. While it is noteworthy that such applications still have a long way to go before they become real-world medical tools, learning and visualizing the disease and treatment process of individual disorders can constitute a promising starting point.

Building up on this, the current research project focuses on learning and extracting rules from medical observational data as a time series to comprehensively present the disease progression in a data-driven fashion. To achieve this, 11641 real-world pulmonary cancer patient records registered at the Universitätsklinikum Hamburg Eppendorf (UKE) in Germany will be mapped onto a digital and standardized format for Electronic Health Records (EHR). Subsequently, three transparent models of computational intelligence capable of learning rules from a time series will be implemented, compared and evaluated. For each model, an extract, transform and load (ETL) process is designed which enables rule learning directly from the standardized EHR. After learning, these rules will be visualized in a digital ontology which ideally mirrors the disease and treatment progression in pulmonary cancer patients in a comprehensible manner. Hence, the system is able to learn and visualize disease progressions across patients, disease patterns and institutions.

1.1 Research Questions

To summarize, this thesis focuses on the following objectives:

Theoretical Research Objectives

- Q1. Is it possible to learn and extract temporal rules from standardized EHRs such that healthcare professionals can interpret them without prior knowledge about the underlying model of computational intelligence?
- Q2. Are there models which demonstrate relatively robust performance for varying data set sizes and varying data dimensionality using real-world medical records?

Medical Research Objectives

- Q1. Are the inferred dependencies in the patient data what we would expect considering the official medical treatment guidelines in Germany?

1.2 Thesis Outline

The thesis is structured as follows. In chapter 2, relevant concepts from the scientific literature are presented. Chapter 3 summarizes the different data formats and datasets used in this research, while chapter 4 outlines the chosen models mathematically and their respective implementation details. In chapter 5 the experiments and their evaluation methods are described. In chapter 6 the experimental results are portrayed before the discussion and conclusion of the study is presented in chapter 7.

Chapter 2

Background

The theoretical background of this research can roughly be divided into two sub-parts. Firstly, the need and context of this research is highlighted by outlining the status quo of data interoperability and AI usage in the healthcare sector in Germany. Afterwards, the reader will be familiarized with relevant concepts of digital healthcare. For this, the Fast Healthcare Interoperability Resources (FHIR®) [19] and The Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) will be introduced as standards for Electronic Health Records (EHR). In addition, it is important to investigate the Synthetic Data Generation (SDG) tool Synthea to highlight its role in the motivation and utility of this research project. In the second step, the concept of rule learning on time series using Artificial Intelligence (AI) will be addressed in general terms, before eligible models for this research will be assessed conceptually. The section will end by incorporating the background knowledge into the current research objective to concisely familiarize the reader with the intentions of performing this study.

2.1 Digital Healthcare & Tools

The official report from the German Ministry of Economy and Energy has released statistics [9] displaying the Health Care industry in Germany to be the worst-performing industry in terms of digitization and digital innovation. The rankings per industry are shown in Figure 2.1. Noteworthy, queried healthcare institutions in Germany reported no relevance or application of Artificial Intelligence (AI) in their work [9] so far. However, the UKE in Germany is viewed as one of the most progressive health care institutions in Europe. With the newly founded IAM (2021) the aim is to further progress towards digitization, interoperability and application of AI through research to advance on the aforementioned systematical disadvantage of the healthcare sector. In the following, two digital healthcare tools used at the IAM will be presented which lay the foundation for this research project, namely standardized EHR and Synthea.

2.1.1 FHIR & OMOP

An important form of digitization in the healthcare industry displays the shift from paper-based to digital documentation. The benefits stand to reason, as patient data in a digital format are more readily available, distributed and accessible than traditional patient records. In addition, representing the data in digital format allows to build tools and applications on

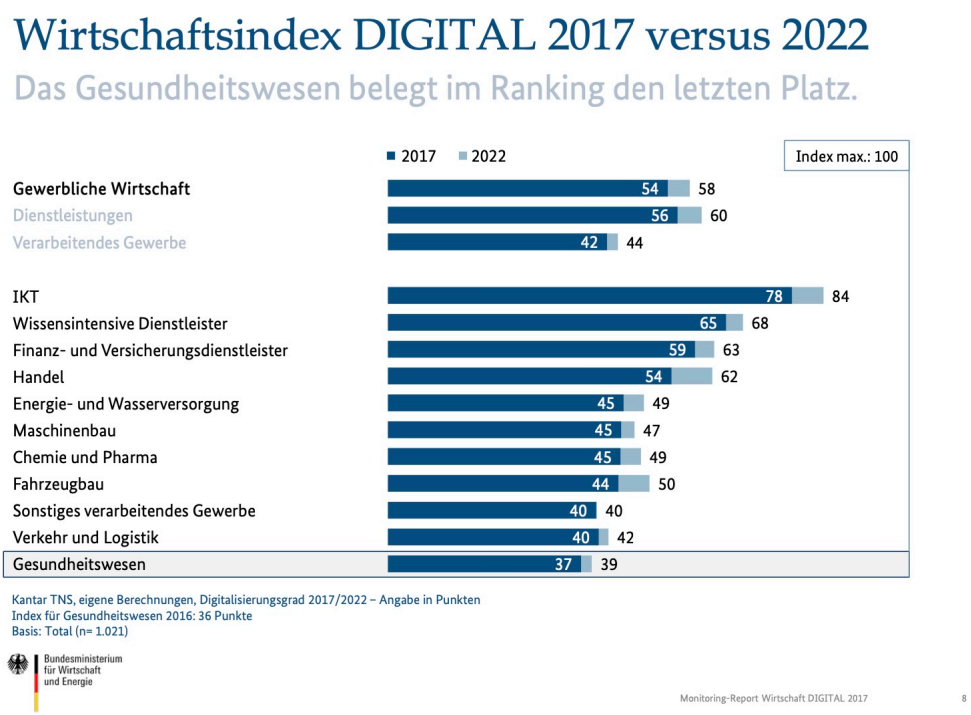


Figure 2.1: Official statistics of the German Ministry ranking the digital development per industry [9]. The healthcare sector ("Gesundheitswesen") ranks last for 2017 and 2022 with a digitization score of 37-39 out of 100.

top of it which can exploit the data. Standardized digital health records like FHIR and OMOP hereby facilitate the scalability of such tools and applications across domains and distributions. In the following, both data formats will be presented to assess their individual qualities.

FHIR has the objective of reducing implementation complexity without losing information integrity [20]. To achieve this a set of so-called resources, the basic components of FHIR (e.g. patient, observation, condition), is built and application programming interfaces (API) are developed to access and use these resources [20]. An EHR of a patient in FHIR thus is an aggregation of resources and can be represented in either JSON, XML, or RDF files [21]. The fact that the user is able to access and perform operations on the resources on a granular level sets FHIR apart from other standardization formats for EHR. The FHIR standard may thus be one of the most promising digital formats for EHR, highlighted by the fact that in 2018 world-leading tech companies including Microsoft, Google, IBM, and Amazon signed a commitment to remove the barriers of healthcare interoperability in which FHIR was explicitly mentioned as emerging standard [20] [22].

Another data standard for EHR is the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM), or the OMOP cdm in short. The OMOP cdm is standardized in terms of content and structure, such that it facilitates efficient analysis [23]. Medical terminology regarding drugs, conditions, procedures, devices, observations and measurements are encoded in a vocabulary which further allows for standardization and organization of medical terms within the database [24]. The vocabulary was created by the OHDSI, the Observational Health Data Science and Informatics initiative, which intended to optimize secondary use of ob-

servational data by harmonizing and standardizing clinical data and creating scalable analytical tools [16, 25, 26].

Noteworthy, the two mentioned data standards belong to the most common formats in clinical practice. Also, EHRs in one data standard are easily transferred into the other and vice versa [27]. In this project, the unstructured hospital data will be mapped into OMOP cdm and the analytical data pipelines are thus built assuming the OMOP data standard. However, the pipelines developed during this project may also be used if the data is in another clinical data standard by prior mapping of, for example, the data within a FHIR database to OMOP cdm.

2.1.2 Synthea

Synthea is a SDG tool that is able to synthesize large patient data sets from disease modules in the form of digital ontologies. The Synthea community, which is part of the larger MITRE corporation, has built several such modules based on the PADARSER framework, the Publicly Available Data Approach to the Realistic Synthetic EHR. The PADARSER framework uses health incidence statistics, clinical practice guidelines and medical coding dictionaries to create digital ontologies by hand [28]. Figure 2.2 visualizes the PADARSER framework in detail. See Figure 1 and Figure 2 in the Appendix for the two official Synthea modules regarding pulmonary cancer.

Patient privacy and data confidentiality are essential objective during SGD in the clinical context. As the PADARSER framework assumes that real EHR data is unavailable, it circumvents one of the major challenges of the field. However, this framework also embodies one major disadvantage, namely scalability. The man-made ontologies are not scalable across geographics, as the expert knowledge may be influenced by environmental factors. For example, the Synthea ontology for the most common cause of death in the United States is based on expert knowledge and health statistics gathered in Massachusetts [28]. An ontology created based on a patient population in Massachusetts will most likely not provide meaningful insight for a Hospital in Germany. Therefore, a system which learns rules from a specific subpopulation of standardized EHRs may effectively distribute the benefits of digital ontologies and the subsequent SDG.

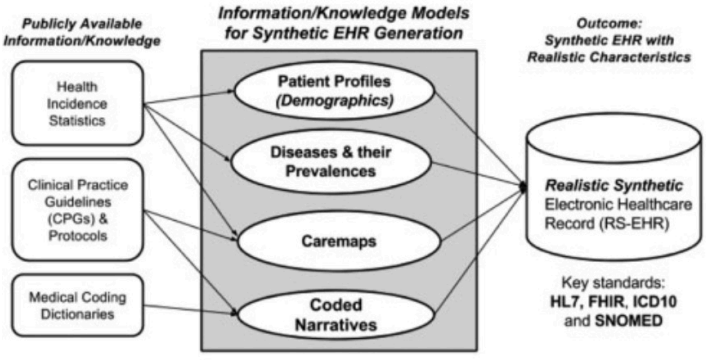


Figure 2.2: The PADASER framework [28]. It defines the conceptual process of creating Synthea graphs by hand.

Essentially, the current research aims to address the gap in a larger pipeline between two existing tools in digital healthcare. More precisely, by building up on a standardized EHR format, the developed intelligent system is scalable across patients, diseases and institutions. By modelling the learnt rules into a digital ontology with the same format as the Synthea modules, one can create a pipeline which synthesizes artificial EHR from real and standardized EHR. A schematic overview is depicted in Figure 2.3. Unstructured clinical source data is

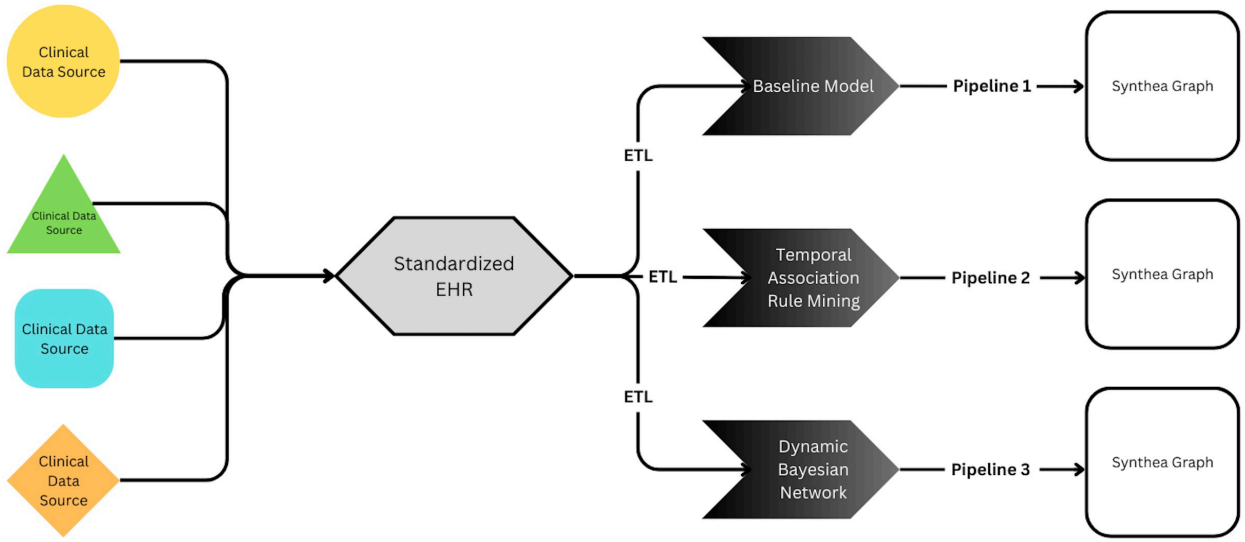


Figure 2.3: Visualization of the larger pipeline the thesis project is embedded in. The unstructured hospital data is mapped to a standardized EHR to enable reusable analytical data pipelines.

gathered in a standardized EHR. Then, an analytical pipeline extracts information from the standardized patient data. Finally, the learnt temporal rules are modelled into a Synthea graph. Note that Figure 2.3 depicts three distinct analysis pipelines which differ in the computational model used. All three approaches will be introduced in the following sections.

2.2 Pulmonary Cancer

The project will be performed on EHR of real patients registered for treatment of pulmonary cancer at the UKE. Therefore, some key points regarding pulmonary cancer and its causes, prevalence and treatment guidelines will be addressed. Pulmonary, or lung, cancer represents the most important cause of cancer death worldwide. More precisely, it is the most common cause of cancer death in men and the second most common cause of cancer death in females [29]. It has been the most common cancer in the world for several decades already and accounted for around 18% of all cancer deaths in 2018 and 2020 [30, 31]. Whereas lung cancer can be traced back to several causes like ionizing radiation, environmental air pollution, infection and obesity, the dominant driving factor is clearly direct tobacco smoking, followed by indirect second-hand smoke (SHS) [29]. Smoking accounts for more than 80% of lung cancer cases in the western world, thereby making it the leading preventable cause of death worldwide [32].

The European Society for Medical Oncology (ESMO) continuously develops and publishes Clinical Practice Guidelines (CPG), which serve as the consensus of expert recommendation and provide a specific treatment structure per type of cancer. Non-small cell lung cancer (NSCLC) is hereby recommended to be treated with surgery, systemic therapy, adjuvant chemotherapy, primary radiotherapy, radio-frequency ablation and postoperative radiotherapy in early-stage

NSCLC (Stages I and II) [33]. Locally advanced NSCLC (Stage III) will be resected if a Multidisciplinary Team (MDT) evaluates a complete resection of the tumor to be possible, otherwise systemic treatment is provided. As these treatments represent a consensus regarding lung cancer treatment across Europe most of them should be rediscovered in the data gathered at the UKE, which will be used to learn disease and treatment progressions through models of computational intelligence.

2.3 Computational Intelligence for Rule Learning

Rule extraction and rule learning are fundamental techniques in computational intelligence that aim to uncover meaningful information from complex data sets. These techniques are widely used in various domains such as data mining, machine learning, and pattern recognition. Rule extraction involves the identification of relevant patterns and relationships within data, while rule learning is the process of constructing a set of rules that describe the data in a compact and interpretable manner. If one uses the original Synthea module in Figure 2 as guidance, it becomes evident that an intelligent model for the task at hand has to be capable of rule extraction *and* rule learning. In the following, two groups of models with promising qualities for addressing the task will be investigated with a particular emphasis on learning from time series data. The two groups of models were identified to possess different mathematical complexity and dissimilar conceptual approaches to rule learning, which makes a comparison interesting with respect to the research objectives. Note, however, that this section serves the purpose of introducing the reader conceptually to the broad approaches, whereas chapter 4 discusses specific algorithms implemented from a methodological perspective.

2.3.1 Temporal Association Rule Mining (TARM)

Data mining describes the technique of selecting, processing and integrating data and retrieving useful information from it. Association Rule mining (ARM) hereby is a method of computational intelligence which serves as analytical tool to categorize and summarize the relationships among data by identifying correlations and patterns in large relational databases [34]. The association rules are hereby used to find relationships between objects which are frequently observed together and are expressed in if/then statements that uncover dependencies in otherwise unrelated data sets [34]. ARM techniques have been around the scientific literature of computational intelligence for some decades already [35] and have repeatedly supported their usability in uncovering dependencies from observational data in finance [36, 37], telecommunication [38, 39], retail marketing [40], but also healthcare specifically [41, 42].

Temporal Association Rule Mining (TARM) extends this idea by identifying relationships between entities on time series data. In other words, the uncovered if/then relationships by TARM can be interpreted as "if A is observed at time t, then B will be observed at the next time step with a probability of P". Besides TARM models arguably implying more meaningful and interesting rules than standard ARM, the literature and code documentation of TARM is scarce and relatively inaccessible due to a lack of standard terminology [43]. That being said, one prior research was identified which developed a general methodology for mining temporal association rules on clinical and administrative hospital data [44]. The lack of visibility of

TARM research may be counterproductive for the development of such models [43], however, by taking a closer look at the intuitive and well-documented ARM methodologies, one can get a better grasp of the underlying idea.

Besides having widespread applicability, the first and most prevalent use case of ARM is the basket analysis [45], which informs a retailer about products that are frequently purchased together and thereby can improve marketing and placement of associated products to increase profit [46]. Transferring this approach to medical diagnosis, one could for example identify frequent associations between a medication and a certain symptomology. This has been done before in the context of predicting drug response in cancer [41] and for determining factors which contribute to heart disease [42]. Taking this idea one step further, by forcing the associations between entities to be learned across time steps, temporal association rule mining would be able to uncover relationships between conditions and treatment procedures with a chronological component. Eventually, these chronological relationships can be compared to actual medical treatment guidelines to validate or falsify adherence to these.

Thus, a temporal association rule mining approach is implemented in this study as the scientific literature recognizes it as a robust, computationally efficient and mathematically transparent approach. Besides only computing rather basic statistical measures, this approach serves as a benchmark regarding robust computing to more complex, yet also more fragile, computational models like Dynamic Bayesian Networks.

2.3.2 Dynamic Bayesian Networks (DBN)

Generally, Bayesian Networks (BN) are graphical representations of Joint Probability Distributions (JPD) in the form of a directed graph [47]. A Dynamic Bayesian Network (DBN) is a probabilistic graphical model that extends the concept of Bayesian networks to capture temporal dependencies and dynamic behavior in time series data [48]. Interestingly, Bayesian approaches are not widely used in the medical field [47], however, their theoretical capabilities seem to match the requirements for transparent learning from patient data. For example, reasoning from effect to cause is a special capability of Bayesian modeling which, in combination with the inherent graph representation of the causal structure, can facilitate diagnostic applications and improve decision-making support [47, 49]. In addition, the convenience of inducing expert knowledge makes them interesting for real-world applications in the medical field [47]. However, it is necessary to note that knowledge of Bayesian statistics is required to understand the model itself, the approach is computationally expensive and evaluation or performance indicators for DBN structure learning are not consistent across studies and researchers [50].

The graph of a DBN can be created in two ways. Firstly, the graphical structure can be built one node at a time with pre-initialized distributions set for each node, the so-called prior distribution in Bayesian Statistics. Secondly, the network structure can be learned in a purely data-driven approach, which is required for the proposed pipeline of this research project [see Figure 2.3]. However, the exact learning of a DBN from data, known as structural learning, is an NP-hard problem [51], meaning it is computationally extremely complex and may be infeasible for larger feature spaces [52]. Dynamic Bayesian networks have been applied to problem statements in healthcare before [53, 54], however, these applications predominantly focus on the prediction of a medical outcome variable rather than unsupervised graph structure learning.

In summary, Dynamic Bayesian Networks can provide a great upside for modelling disease progressions, if applicable to the data. The ability to learn cause-effect relationships is in demand for medical use cases, but a good performance is strongly dependent on the data set. Therefore, in this study, a DBN structure learning approach is implemented to investigate its robustness using real-world medical data.

Chapter 3

Data

Multiple data formats are relevant to this research. First, there is the unstructured source data called VONKO. VONKO holds real patient data from lung cancer patients in Germany. This data set was mapped into the standardized EHR format OMOP. After three temporal rule learning approaches extract and analyze data directly from the OMOP data structure, each output was transformed into a Synthea graph. Thus, the three data formats of VONKO, OMOP and Synthea will be introduced chronologically in the following.

3.1 VONKO Data on Pulmonary Cancer

The VONKO dataset is the unstructured source data which is used for analysis after mapping it into OMOP. This data was gathered by the state cancer registry in the federal state of Schleswig-Holstein in Germany [55]. Thus, it holds all information about lung cancer patients health care institutions need to report to the state. It is comprised of seven data tables which themselves hold information about the tumor, operations, radiotherapy, systemic therapy, disease process, distant metastases of the primary tumor and distant metastases in the disease process. Whereas the tumor and the distant metastases table are concerned with summarizing additional medical observations of a patient, the other five tables hold required information regarding the official reporting of lung cancer patients to the state.

3.1.1 TNM Classification

The TNM classification was developed by the American Cancer Society [56] and is a vocabulary which standardizes the medical tumor staging of a patient. The TNM cancer staging vocabulary was also used in the VONKO data set and will therefore be introduced in the following.

Firstly, the TNM tumor classification is divided into clinical and pathological staging. The clinical staging of tumor diagnosis depends on physical exams like blood tests, x-rays and CT-scans and constitutes key insights for determining preliminary treatment methods. In comparison, the pathological staging, also called the surgical stage, refers to a more precise staging classification as it incorporates the clinical staging with potential surgery results. Secondly, the TNM system is composed of three main variables to define the cancer staging. These describe the original or primary tumor (T), whether the cancer has spread to nearby lymph nodes (N) and whether the cancer has metastasized (M) to distant parts of the body. Thirdly, the stage

of the tumor is defined by Roman numerals from I (1) to IV (4), where lower stages are less advanced tumors [56]. Therefore, the pulmonary tumor of each patient in the VONKO data set is classified by some combination of diagnosis type ('c' for clinical or 'p' for pathological), tumor type ('T', 'M' or 'N') and a stage (I-IV).

3.2 OMOP Common Data Model

Electronic Health Records (EHR) have revolutionized the healthcare industry by enabling the storage and management of vast amounts of patient data. However, the heterogeneity and complexity of EHR data pose significant challenges for data integration, analysis, and research. To address these challenges, the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) has emerged as a widely adopted data standard for harmonizing and organizing EHR. A concise overview is shown in Figure 3.1.

The OMOP CDM provides a standardized framework for transforming and structuring EHR data into a consistent format, facilitating data interoperability and enabling large-scale observational research across disparate healthcare databases. The CDM is designed to accommodate diverse data types and represents the entire healthcare continuum, from diagnosis and procedures to medications and patient demographics. Whereas the OMOP CDM also holds information about the health system and economics of a medical observation, this research is focused on the standardized clinical data [see Figure 3.1]. Further, the structure of the standardized clinical data revolves around the following domains:

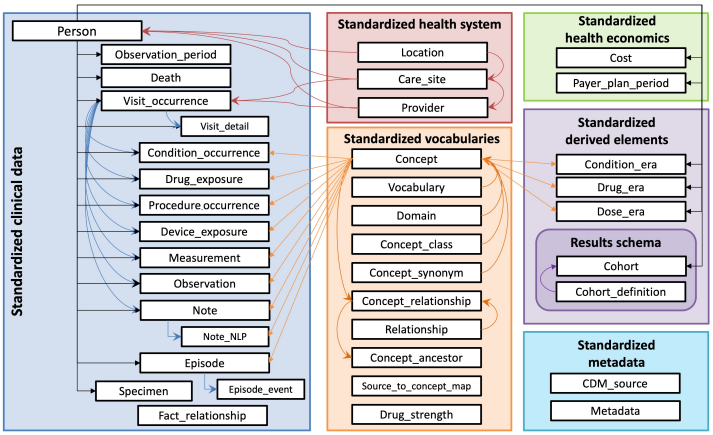


Figure 3.1: The database architecture of the OMOP CMD [23].

1. **Person**: This domain captures information about individuals such as demographic characteristics, gender, and birthdate. It serves as the foundation for linking records across other domains.
2. **Visit/Observation Period**: The Visit/Observation Period domain records information related to patient encounters, including admission and discharge dates, as well as the duration of observation.
3. **Condition/Observation**: This domain captures information related to medical conditions, symptoms, and observations made during patient encounters. It includes diagnoses, symptoms, laboratory test results, and other clinical observations.
4. **Drug Exposure**: The Drug Exposure domain captures information about medications prescribed or administered to patients, including drug names, dosage, and duration of exposure.
5. **Procedure Occurrence**: This domain records details of medical procedures performed on patients, such as surgeries, treatments, and interventions.

6. **Measurement**: The Measurement domain encompasses various clinical measurements taken during patient encounters, including vital signs, laboratory test results, and other quantitative measurements.
7. **Device Exposure**: This domain captures information about medical devices used in patient care, including implantable devices, imaging studies, and diagnostic procedures.
8. **Death**: The Death domain captures information related to patient mortality, including date, cause, and location of death.

All information that was gathered in the unstructured VONKO dataset was mapped into the OMOP cdm. However, in the context of this study information from only two OMOP domains will be extracted for analysis, namely the *condition* domain and the *procedure* domain. Thus, the output of each data analyses pipeline proposed in chapter 4 is focused on the rules that can be learned between and within conditions and procedures of the patients. Nevertheless, the proposed pipelines can be easily modified to process two different or more than two OMOP domains for analysis.

Standardized Vocabularies

All information regarding the OMOP cdm and its conventions is summarized in the book of OHDSI [57]. Within this book, the chapter on standardized vocabularies is central to understanding OMOP as well as this research project. Essentially, whereas the OMOP data structure standardizes the data format, the vocabulary within OMOP standardizes the content.

This being said, there is an extensive amount of different vocabulary sets to categorize medical drugs, conditions, procedures, devices, observations and measurements into unique integers or keys [24]. The usage of these vocabularies may depend on the demographics, prevalence or language of the concept names in the vocabulary. Thus, healthcare institutions across the globe use vastly different vocabularies to encode their medical observation on-site. However, the idea of OMOP is to define a single *standard* vocabulary in the data format and enable the mapping process from some vocabulary to the standardized vocabulary.

The standardized vocabulary used in this project was SNOMED CT [58]. Therefore, all steps of the analytical process described in chapter 4 is performed on SNOMED codes. Only after extracting the codes from OMOP, transforming them into a time series and loading them into one of the models of computational intelligence, the SNOMED codes are mapped to free text. This eventually results in a graph with English descriptions of the medical observation, while the whole analytical process is performed on unique vocabulary keys.

3.2.1 ETL process for OMOP analysis

To enable time series data analysis from the OMOP cdm, an individual ETL process is required for each analysis method. All proposed analysis methods with their corresponding ETL process are presented thoroughly in chapter 4, however, some parts of the ETL processes are the same across analysis methods. These similarities are the following:

- *The general structure of the time series*: For every patient, a time series will be constructed from the observational data. To achieve this, the conditions as well as the procedure observations are extracted from the appropriate OMOP domains, grouped by patient identifier and ordered by the corresponding time stamp of the clinical observation.

Therefore, the input data for all three models are N time series, where N is the amount of patients, with one or multiple clinical observations per discrete time steps, respectively. Note that the time series vary in length depending on the patient's hospital history.

- *Start and Stop nodes*: To every individual patient time series a start and a stop observation will be attached, which can be compared conceptually to start and end tokens defining a sentence in NLP. The start observation is always at t=0 and the end node always extends the original length of the time series by t+1 time steps. By doing so, the computational models are able to learn which procedures or conditions occur at the initial hospital encounter and which occur at the end. In addition, it improves the structure and interpretability of the resulting graph. Also, the Synthea data synthetization tool requires modules to have a start node and without its inclusion in the computational process, the start node would have meaningless connections in the Synthea graph.
- *Mapping of standardized vocabulary*: As explained in section 3.2, the OMOP cdm holds information about clinical methods, diagnoses and concepts by standardized vocabulary. The standardized vocabulary represents every possible clinical concept as unique integer. Therefore, integer values are extracted from the OMOP cdm in all ETL processes and used for analysis. However, after learning and constructing the graph the standardized vocabularies are mapped to free text to facilitate the interpretability of the resulting graph. This is done by identifying the necessary vocabulary packages and downloading them from Athena [59] before implementing a mapping function which transforms the *concept_id* of the standardized vocabulary to the English description of the medical concept.

3.3 Synthea

After the VONKO source data and the OMOP cdm, the third and last data format relevant for this project comes from Synthea. Figure 3.2 depicts one of the official Synthea modules on pulmonary cancer. These official Synthea modules are created in a cumbersome process by hand, however, they are utilized for two important use cases. First, the 'Module Builder' developed by Synthea allows for coherent visualization of interpretable, medical guidelines in a graph structure [60]. Second, these graph structures can be used to sample synthetic patient EHR. As the sampling of EHR is not in the scope of this research, the following paragraphs will explain the data format of the Synthea modules with regard to visualization in the Module builder [60].

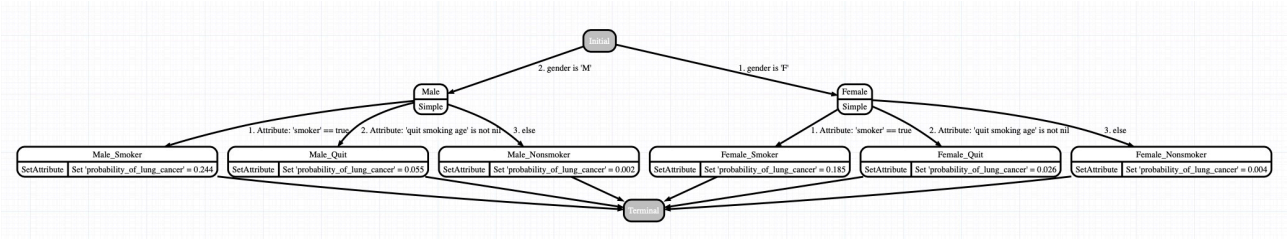


Figure 3.2: Official Synthea module on pulmonary cancer [61].

Synthea modules are defined in a JSON file. The specific syntax for node and edge types are predefined by Synthea [60] and can take on a range of values. More precisely, the graph

structure can be defined with 6 different edge types and a total of 30 node types [62]. Each node or edge type has unique properties specifying the sampling process of the synthetic data generation process. However, this project is focused on visualizing the learned temporal rules in a coherent format, not on leveraging the full complexity of Synthea graphs for sampling EHR. Thus, the choice of node and edge types is dependent on displaying distributed and probabilistic edges between nodes. Nevertheless, the final output of each developed pipeline is a valid Synthea module and could be used for sampling.

For the just outlined objective, one edge type and three node types were selected. Namely, all edges in the final Synthea graphs were defined with the edge type 'Distributed Transition' [62]. This edge type is the only type proposed by Synthea which allowed for distributed probabilistic edges from one antecedent node to multiple consequent nodes. Any other edge type would have resulted in graphs which display no distribution or no probabilities. A noteworthy constraint to this edge type, however, is that the distributed transitions from any antecedent node have to sum to a probability of 1. Figure 3.3 helps in understanding this issue. On the left side, you can see the straightforward case of a set of rules where the amount of consequents is ≥ 1. Either, the probabilities sum to 1 or they are proportionally scaled to a sum of 1 during post-processing. However, the shortcoming of this edge type is visible on the right side of Figure 3.3. Assume the model has learned a single rule from variable X to Y. Regardless of the learned weight of that rule, Synthea forces a weight display of 1.0 for that edge.

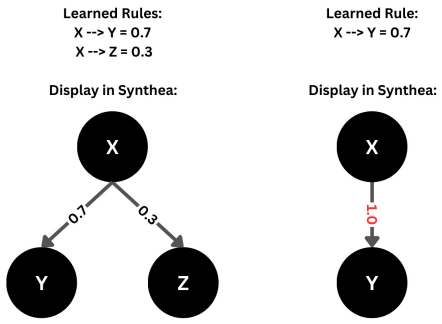


Figure 3.3: Shortcoming in Synthea display of graphs. Transforming the learned probabilistic rules into the Synthea graph structure can lead to a distortion of values.

The three node or 'state' types used in this project were 'Initial', 'Terminal' and 'Encounter' [62]. The 'Initial' state is necessary for the graph structure to be a valid Synthea module [62]. As explained in the previous subsection, start and stop observations were attached to each patient time series. These eventually constitute the 'Initial' and 'Terminal' node of the final Synthea graph structure. Every other node, so every clinical observation that is involved in the rule learning process, was defined as 'Encounter' node in the Synthea module. The 'Encounter' state type was chosen mainly because of two reasons. Firstly, the medical observations used in the temporal rule learning process are actually healthcare encounters of the patient. Secondly, the 'Encounter' node type in Synthea is the only node type which updates the synthetic patient EHR during the sampling process. In other words, by using this node type the graph is conceptually coherent with the medical observations and still holds the potential to allow for data synthetization eventually.

Chapter 4

Methods

In this study, the aim is to learn and visualize temporal rules between medical entities from standardized EHR. To investigate and evaluate this scientific problem three pipelines were implemented. Each pipeline extracts data from the OMOP common data model, transforms the data into a model-specific time series and loads it into one of the models of computational intelligence. Finally, the learnt rules are post-processed and transformed into a Synthea graph. Pipeline 1 serves as manually implemented baseline, which computes the single most basic if-then rule possible. In Pipeline 2, a Temporal Association Rule Mining algorithm learns sequential rules. In Pipeline 3, a Dynamic Bayesian Network structure is learnt from the data. All three models will be explained individually, as well as in the context of their respective pipeline. The three pipelines differ in the required data transformation, computational model and output, but the general structure for each of them corresponds to Figure 2.3.

For all three pipelines, consider N independent realizations of time series $\mathbf{x}_{n,t} \in \mathcal{C}^D$, where D is the set of all variables in the data. A time series $\mathbf{x}_n$, meaning a patient's hospital encounter history, can vary along two dimensions. These dimensions are the length of the time series $t \in \{0, \cdots, T\}$ and the size of the set of observations within a time step $d = \{i, \cdots, j\}$ with $d \in D$. Thus T defines the amount of hospital encounters and d defines the reported medical entities per hospital encounter. Therefore, a distinct medical observation is referred to in the form $\mathbf{x}_{n,t,d}$ in the following sections. For example, $\mathbf{x}_{2,3,i}$ and $\mathbf{x}_{2,3,j}$ refer to two distinct observations that were made for the second patient in the data set at his/her third hospital encounter. Finally, to ensure coherent notation across models, the *autoregressive order* of a model is denoted by the variable p. The autoregressive order refers to the number of past time points used to predict the current value in the time series model.

4.1 Pipeline 1: Baseline Transition Matrix

The most basic mathematical approach to compute if-then rules is calculating the conditional probability between two variables of interest A and B. Equation 4.2 depicts the standard formula for the conditional probability of two variables.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{4.1}$$

The aim of the baseline is to set a reference value for more complex probabilistic models.

Thus, the aim of Pipeline 1 is to calculate all conditional probabilities between observations in D with an autoregressive order of $p = 1$. A matrix of size $D \times D$ that summarizes all these conditional probabilities is also called a Markov Chain transition matrix. For this reason, the following section introduces Markov Chains methodologically before a manually developed algorithm is described, which computes all conditional probabilities on time series data.

4.1.1 Markov Chain

A Markov Chain is a basic stochastic model composed of states within a finite state space and transition probabilities. In the context of this research, the transition probabilities between states define probabilistic rules from some state $\mathbf{x}_{n,t,i}$ to any other state $\mathbf{x}_{n,t+1,j}$ or itself $\mathbf{x}_{n,t+1,i}$ with $i, j \in D$. The transition probabilities between states is therefore defined as the conditional probability of:

$$P(\mathbf{x}_{n,t+1,j}|\mathbf{x}_{n,t,i}) \tag{4.2}$$

Thus, the described Markov Chain is a discrete-time Markov Chain with the Markov Property, which defines that the conditional probability of moving to the next state only depends on the present state and not on the previous states [63]. Further, because the autoregressive order equals $p = 1$, the described transition matrix is a Markov Chain transition matrix of first order [63]. Such a Markov Chain is summarized in a $D \times D$ transition matrix, which contains conditional transition distributions for every state $\mathbf{x}_{n,t,i}$ to any other state $\mathbf{x}_{n,t+1,j}$ or itself $\mathbf{x}_{n,t+1,i}$ with $i, j \in D$. The arithmetic mean is computed for all transition probabilities between any two states across all instances of time series $\mathbf{x}_n$ with $n \in \{1, \cdots, N\}$, resulting in one averaged transition matrix per patient population.

Table 4.1: A structural example of the derived transition matrix.

	START	X_i	X_j	...	D	STOP
START	0.0	0.0	0.0	...	0.0	0.0
X_i	0.2	0.7	0.1	...	0.4	0.0
X_j	0.5	0.3	0.1	...	0.2	0.0
...	...	...	...	...	...	...
D	0.3	0.0	0.8	...	0.4	0.0
STOP	0.0	0.0	0.0	...	0.0	0.0

In the following explanations, assume X_i and X_j are any two medical observations within the state space D. Table 4.1 depicts a simplified version of the transition matrices obtained in this research. Whereas the structure of all transition matrices resemble Table 4.1, it varies in the size of D depending on the data sample. This is highlighted by the row and column of dots. The transition matrix is read from column to row. By additionally introducing graph terminology, one can further ease the reading of the transition matrix. For example, the if-then rule $X_i \rightarrow X_j$ can also be defined as an *antecedent* $\rightarrow$ *consequent* relationship. This observation illustrates that while columns depict the antecedents of an if-then rule, the rows depict the consequent. Therefore, the column of a variable defines all outgoing conditional probabilities from an antecedent which sum to 0 or 1. If nothing chronologically follows from

the antecedent, the column sums to 0. If at least one observation follows from any antecedent $X_i \in D$, the transition probabilities for that antecedent sum to 1. As an example, the rule $X_i \rightarrow X_j$ is denoted by a conditional probability of 0.3. in Table 4.1.

Additional important observations can be made about the transition matrix shown in Table 4.1. As mentioned in chapter 3, a 'START' and 'STOP' observation is attached to each time series and included in the computation process. This results in them being the first and last row and column in every transition matrix. To be precise, as D was defined to be all medical variables in the data, the computed transition matrices are actually of size $(D+2) \times (D+2)$. Also, the row of the 'START' observation as well as the column of the 'STOP' observation always sum to 0. This is due to the fact that the 'START' observation of a time series is always at $\mathbf{x}_{n,t=0,d}$, so no prior observation can be used to predict the start node. Vice versa, because the 'STOP' node is always at $\mathbf{x}_{n,t=T,d}$, nothing can follow chronologically from it.

The transition matrix can be viewed as a baseline model, as it comprises all conditional probabilities from any condition or treatment observation to itself or any other symptom or treatment. In the remaining two pipelines of this experiment, the used computational models are more complex due to two reasons. Firstly, Pipeline 2 and Pipeline 3 are more complex in terms of calculating rules with higher autoregressive order, meaning $p \geq 1$. Secondly, they perform mathematically more sophisticated operations between any two observations $X_i, X_j \in D$ than calculating the conditional probability. However, the algorithm to extract, transform, load and analyze the medical observations for this pipeline was designed and implemented manually. The details will be explained in the following.

4.1.2 ETL Process

As a first step, the relevant OMOP columns were extracted. Because this research is focused on conditions and treatments, the relevant OMOP tables were *condition_occurrence* and *procedure_occurrence*. More precisely, from each of the two tables three columns were extracted, namely the *person_id*, the *condition_start_date* or *procedure_start_date* and the associated *concept_id*.

After extracting these six columns, the data was restructured to three columns named *patient*, *observation* and *date*. Hence, this step resulted in a data structure which does not explicitly discriminate between conditions and procedures. Subsequently, the data was sorted by personal identifier and within the personal identifier by date. Now, the data was split into chunks by unique personal identifier and saved in one list, resulting in a list of patient sequences. The observation dates were turned into integers and observations with the same date were aggregated into one integer time step. As the patient data is in sequence format now, the original dates as well as the personal identifier were dropped. Additionally, every patient sequence was extended by the previously described start and end observation [see subsection 3.2.1]. The final data structure is thus a list of lists of lists. The outer list holds all patient sequences, the second level of lists defines each individual patient sequence and the lowest level of lists is a time step within an individual patient sequence. The hierarchy of this data structure is thus identical to the prior introduced mathematical notation $\mathbf{x}_{n,t,d} \in D$. Building up on this, the algorithm designed to calculate all conditional probabilities will be presented in the following section.

4.1.3 Transition Matrix Computation Algorithm

To compute the transition matrix from the proposed time series representation, the occurrences of item associations across time steps has to be counted algorithmically. The pseudo code of this process is shown below in Algorithm 1.

Algorithm 1 Counting Item Associations

```
Initialize empty list for association tuples
for all n ∈ {0, ··· , N} do:
    for all t ∈ {0, ··· , T − 1} do:
        current = x_{n,t}
        following = x_{n,t+1}
        for all d ∈ current do:
            from_state = current[d]
            for all d ∈ following do:
                to_state = following[d]
                Save tuple (from_state → to_state)
            end for
        end for
    end for
end for
```

The resulting ordered tuples of items are thus a frequency measure of how often some antecedent at time step t occurs with any consequent at $t+1$. These counts are performed across individual patient time sequences, which eventually will allow for averaged transition probabilities across patients. Finally, these frequency counts are turned into conditional probabilities by summing over all item association frequencies per antecedent and dividing each item association by the sum of its antecedent. In this way, the algorithm results in conditional probabilities of the form:

$$P(\text{consequent}|\text{antecedent}) = \frac{P(\text{antecedent} \cap \text{consequent})}{P(\text{antecedent})} \tag{4.3}$$

where the consequent state of an antecedent state at t is always at $t+1$ and the conditional probabilities per antecedent sum to 1. By performing the aforementioned computations per state as potential antecedent, the transition matrix was filled with conditional probabilities from each state in the state space to each state in the state space across time with a lag of 1. Item associations which were not present in the data were represented with a 0% transition probability of the consequent following the antecedent. In simple terms, the resulting transition matrix answers the question of "given an antecedent state X_i, what is the conditional probability of observing consequent state X_j in the next time step?".

Consider Figure 4.1 as a schematic example of any time series $\mathbf{x}_n \in N$. The x-axis defines the discrete time steps, and the y-axis displays $d \in D$. Assume that $D = \{X, Y, Z, A, B\}$, which are the medical interventions that were observed for this patient. As an example, the counting algorithm proposed in Algorithm 1 would result in an item association count of $Y \rightarrow A = 1$. This is true because the ordered tuple $\{Y, A\}$ with an autoregressive order of $p = 1$ occurs

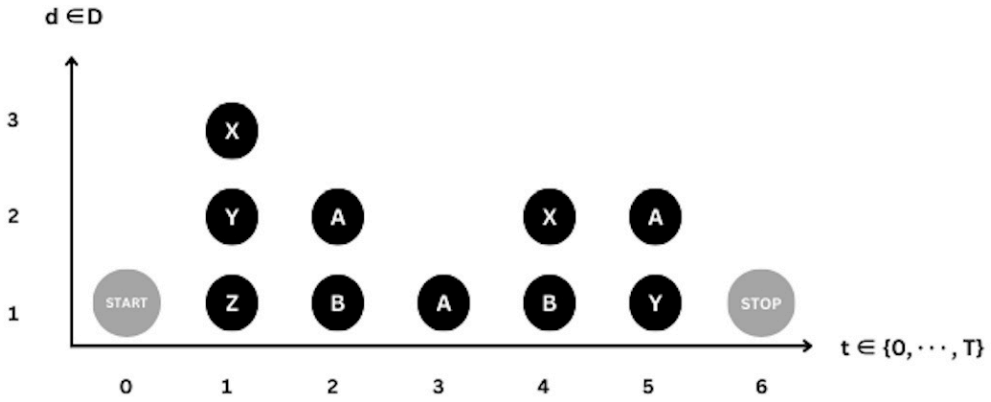


Figure 4.1: A schematic example of any time series $\mathbf{x}_n$, which is a patients hospital encounter history. The letters are placeholders for some medical observation $X_i \in D$. For example, 'X' may be a lung cancer diagnosis and 'A' chemotherapy.

exactly one time in Figure 4.1. In the rule $Y \rightarrow A$, Y is the antecedent and A is the consequent. With this knowledge, Equation 4.3 can be applied to this example.

$$Y \rightarrow A = P(A|Y) = \frac{P(Y \cap A)}{P(Y)} = \frac{1}{2} = 0.5 \tag{4.4}$$

The example in Equation 4.4 displays methodically how Algorithm 1 calculates conditional probabilities from a discrete time series. As a second example, the only remaining item association with antecedent Y in this time series would be $Y \rightarrow \text{STOP} = 0.5$.

Algorithmic Extension

So far, the proposed algorithm calculates all conditional probabilities between medical observations on a discrete time series with a lag of 1. However, this approach is rather simplistic with respect to the overarching goal of learning meaningful rules. In other words, a conditional probability can entail statistical biases. For example, a high conditional probability of the consequent given the antecedent may not be characterized by high likelihood of events following each other, but rather by a high base probability of the consequent. The base probability in this research is defined as the probability of some observation in the state space $\mathbf{x}_{n,t} \in D$ occurring during any time step across patients. Therefore, an algorithmic extension was implemented to account for this bias. This extension subtracts the base probability of the consequent from each conditional probability. The mathematical notation is depicted below.

$$\text{threshold} \geq P(\text{consequent}|\text{antecedent}) - P(\text{consequent}) \tag{4.5}$$

To elaborate on this extension, again refer to Figure 4.1. The temporal rule $X \rightarrow A$ is analyzed exemplary. First, we calculate the conditional probability similar to Equation 4.4.

$$X \rightarrow A = P(A|X) = \frac{P(X \cap A)}{P(X)} = \frac{2}{2} = 1 \tag{4.6}$$

Equation 4.6 shows that the consequent A follows the antecedent X with 100% probability. Now, the algorithmic extension performs the following calculation:

$$\begin{aligned} P(\text{consequent}) = P(A) &= \frac{3}{5}, \\ P(A|X) - P(A) &= 1 - \frac{3}{5} = \frac{2}{5} = 0.4 \end{aligned} \tag{4.7}$$

Equation 4.7 therefore post-processes antecedent consequent connections such that the base probability of each consequent is accounted for. Conceptually, the aim is to reduce bias that emerges through frequently occurring observations. Because A occurs in $\frac{3}{5}$ of time steps, the high conditional probability of $X \rightarrow A$ calculated in Equation 4.6 might not be due to the dependency between the two variables. For example, the rule $B \rightarrow A$ would similarly result in a conditional probability of $P(A|B) = 1$. Without the algorithmic extension, the model would be 100% certain that observation A always follows after observation $X \vee B$.

In addition, this algorithmic extensions introduces a hyperparameter to the algorithm. The threshold in Equation 4.5 is a filter for potential graph edges, where the corrected conditional probability of the consequent given the antecedent has to be above some percentage. For example, a threshold of 0.1 means that only conditional probabilities which are larger than 10% *after* accounting for the base probability of the consequent will be included in the graph. The link $X \rightarrow A$ discussed in Equation 4.6 would, for example, pass this threshold and be used to build the graph.

For this, the links are first rescaled to a probability of 1 per antecedent. Subsequently, the final antecedent consequent links are used to initialize and direct the edges of the graph. The scaled conditional probabilities are used as edge weights. Finally, a script was written to automatically transform the learned if-then rules into a JSON file which constitutes a valid Synthea module for visualization and sampling.

4.2 Pipeline 2: Temporal Association Rule Mining

Temporal Association Rule Models (TARM) are models which learn sequential association rules in large databases. To explain the concept of sequential association rules, one first has to distinguish it from the concept of sequential association patterns. Sequential association pattern mining is the task of finding sequences of events that appear frequently in a sequence database [64]. These association patterns are simply successions of events which are observed frequently across sequences. On the other hand, the task of sequential rule mining incorporates probabilities and provides indications that if some event(s) occur, other event(s) are likely to occur with a certain confidence [64].

Several different rule mining algorithms have been proposed [65, 66], however, in this research `CMRules` was applied to the time series data and will be elaborated on in the following [67]. The algorithm was selected for this research, as it adds another level of complexity to the experiment. Whereas in Pipeline 1 rules were learnt with an autoregressive order of $p = 1$, the autoregressive order of this pipeline is $1 \leq p \leq T$ for any hospital history $\mathbf{x}_n$. In addition, rule mining is described as a robust approach in the scientific literature in terms of varying use cases and data sizes. Hence, its capabilities are deemed suitable for the proposed research objectives.

4.2.1 CMRules

`CMRules` mines sequential rules which are common to several sequences. To achieve this, it first prunes the search space by identifying items which occur jointly in many sequences by mining association rules. Only thereafter the mined association rules are evaluated with the time constraint and become *temporal* or *sequential* association rules. For this reason, it is necessary to first define original association rule mining on a transactions database and subsequently define temporal association rules with regards to how `CMRules` mines these on a sequential database.

Association Rule Mining

Association rule mining is a common knowledge discovery technique for uncovering associations between items or sets of items in a transaction database [45, 64]. The term transaction database is used for this data structure, because association rule mining has its origin in the retail sector [40]. However, the concept can be transferred to applications in the health sector. Whereas a transaction originally defines the set of bought items in a store, in this project a 'transaction' is defined as the set of all medical observations of any patient. A transaction database is thus a simplification of the prior time series notation of a hospital history $\mathbf{x}_{n,t}$ to $\mathbf{x}_n$. It holds the same observations, however, the transaction database does not specify a time order within a patient history. That being said, a transaction database TD is formally defined as a set of transactions $T = \{x_1, x_1, \cdots, x_n\}$ and a set of items $D = \{i_1, i_2, \cdots, i_n\}$, where $x_1, x_2, \cdots, x_n \subseteq D$. The objective of association rule mining is to learn all rules $X \rightarrow Y$, such that

$$(X, Y \subseteq D) \wedge (X \cap Y = \emptyset) \tag{4.8}$$

meaning the item sets X, Y are subsets of the item set D which do not overlap. The

association rules of the form $X \rightarrow Y$ are also filtered in the computation process by some minimum level of interestingness. This level of interestingness is defined by two statistical metrics in the `CMRules` algorithm, namely by support and confidence of a rule. The support of an item set $X \subseteq D$ is denoted as $sup(X)$ and mathematically defined as the number of transactions that contain X. The support of an item set can easily be extended to define the support of a rule, which is defined below in Equation 4.9.

$$sup(X \rightarrow Y) = \frac{sup(X \cup Y)}{|T|} \tag{4.9}$$

The concept of confidence, however, is the support of either item set X or item set Y divided by the support of item sets that contain the antecedent item set X. The formula is depicted in Equation 4.10.

$$conf(X \rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)} \tag{4.10}$$

Temporal Association Rule Mining

Whereas association rules are mined from transaction databases, sequential association rule mining is performed on sequential databases [64]. A sequence database is a generalization of a transaction database, where the occurrence of items contains additional information on the time of occurrence [68]. A sequential database SD is hereby defined as a set of sequences $\mathbf{x}_n = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ and a set of items $D = \{i_1, i_2, \cdots, i_d\}$, where every sequence $\mathbf{x}_n$ is an ordered list of transactions $\mathbf{x}_{n,t} = \{\mathbf{x}_{n,1}, \mathbf{x}_{n,2}, \cdots, \mathbf{x}_{n,T}\}$ with $\mathbf{x}_{n,t} \subseteq D$. Thus, a sequential database SD is of similar format as the list of time series introduced for Pipeline 1. A single sequence is visualized in Figure 4.1 and an SD can be thought of as a table containing multiple instances thereof. The definition of a sequential rule is similar to the association rule defined in Equation 4.8. However, it is extended by the condition that all items of X occur in some transactions of the sequence before items of the set Y occur in some transactions of the same sequence. In addition, the concepts of support and confidence of a rule are extended to define sequential support in Equation 4.11 and sequential confidence in Equation 4.12.

$$seqSup(X \rightarrow Y) = \frac{sup(X \square Y)}{|S|} \tag{4.11}$$

$$seqConf(X \rightarrow Y) = \frac{sup(X \square Y)}{sup(X)} \tag{4.12}$$

The notation $sup(X \square Y)$ defines the number of transactions where all items of X occur before all items of Y. Noteworthy, there is no time ordering *inside* item sets of sequential rules. Time ordering is only present between the sets X, Y such that $X \rightarrow Y$. Now that the terminology of transaction and sequence databases, as well as association rules and sequential rules, is defined, the `CMRules` algorithm will be explained in detail [64]. Throughout the explanation, Figure 4.2 can be referenced as an intuitive visualization of the CMRules algorithm.

`CMRules` builds up on one central observation of the relationship between association rules and sequential rules. This observation is summarized in the fact that if one ignores the time information of a sequential database SD, a transaction database SD' is obtained. And for

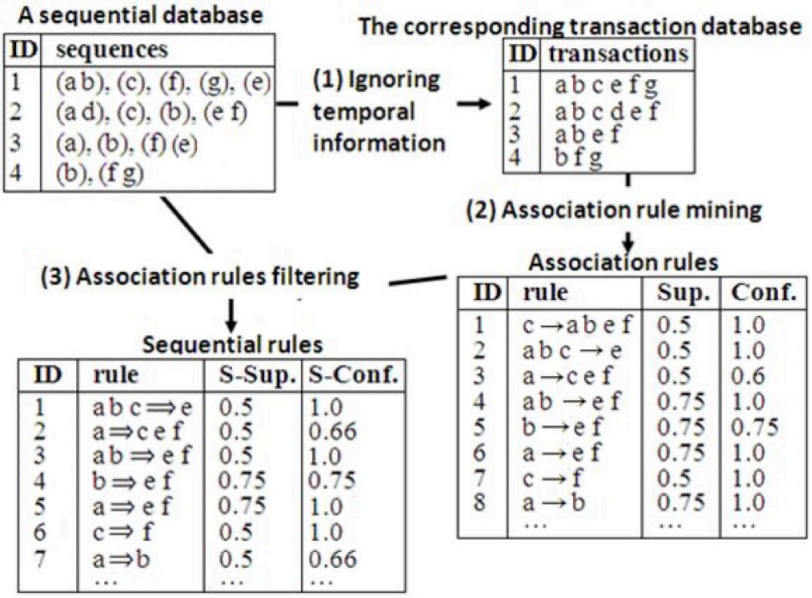


Figure 4.2: Schematic visualization of the `CMRules` algorithm [67]. It concisely shows how only after applying traditional Association Rule Mining chronological information is used to filter out temporal association rules.

each sequential database SD and its corresponding transaction database SD', each sequential rule $r : X \rightarrow Y$ of S has a corresponding association rule $r' : X \rightarrow Y$ of S'. Importantly, as $sup(X \square Y)$ is always lower or equal to $sup(X \rightarrow Y)$ [64] the relations depicted in Equation 4.13 always hold for any sequential rule and its corresponding association rule.

$$sup(r') \geq seqSup(r) \wedge conf(r') \geq seqConf(r) \tag{4.13}$$

This observation allows for the functioning of `CMRules` to discover sequential rules by first mining association rules. `CMRules` has three inputs, a sequential database and the thresholds of *minSeqSup* and *minSeqConf*, defining the minimal sequential support and minimal sequential confidence of rules which should be included in the output. As can be seen in Figure 4.2, the first step is to transform the input sequential database into a transaction database. Subsequently, an association rule mining algorithm computes all association rules from the transaction database with $minSup = minSeqSup$ and $minConf = minSeqConf$. Because Equation 4.13 holds, the set of association rules which are mined with $minSup = minSeqSup$ and $minConf = minSeqConf$ will contain all sequential rules. Finally, as displayed in step 3 of Figure 4.2, `CMRules` computes all sequential rules to the corresponding association rules identified in step 2. Afterwards, the sequential rules which do not meet the *minSeqSup* and *minSeqConf* thresholds are eliminated, resulting in the set of all sequential rules [64].

Finally, the `CMRules` algorithm has to be differentiated methodologically from the previous approach described in Pipeline 1. First, `CMRules` computes rules of the form $X \rightarrow Y$ across the whole time series. Whereas Pipeline 1 results in rules with an autoregressive order of $p = 1$, Pipeline 2 learns rules with an autoregressive order of $1 \leq p \leq T$. Note that `CMRules` does not define the autoregressive order of the learnt rules explicitly. This is due to the fact that rules are learnt between sets of variables, and each set X, Y in $X \rightarrow Y$ is not restricted to a time

step. For example, a valid TARM rule $X \rightarrow Y$ could hold all observations of X in timesteps t_1, t_2 and t_3, whereas the observations of Y are distributed across t_4, t_6 and t_7. Thus, a rule between sets of variables does not have an autoregressive order. However, because it is not feasible to automatically transform the learnt rules between sets of items into Synthea graphs, only rules with a set size of 1 were post-filtered. These theoretically have an autoregressive order of $1 \leq p \leq T$, but practically are not part of the `CMRules` output.

As mentioned above, rules had to be filtered during post-processing such that no set X, Y in $X \rightarrow Y$ is > 1. The ultimate goal of each pipeline is a Synthea graph, however, fitting rules with multiple antecedents or multiple consequent into a graph is a very complex task. The problem can be explained best by a simple example. Consider the rule $\{A, B\} \rightarrow \{C\}$ with $seqConf(A, B \rightarrow C) = 0.6$. The rule has two antecedents, one consequent and one probability associated with it. As the Synthea graph structure holds one variable per node, the probability of such a rule would need to be distributed across two edges. Splitting the $seqConf(A, B \rightarrow C) = 0.6$ into edge weights $seqConf(A \rightarrow C) = 0.3$ and $seqConf(B \rightarrow C) = 0.3$ would mathematically not represent the same rule. To achieve this, it is necessary to allow for nodes in the graph which hold more than one variable. However, this would significantly increase the complexity of the graph, counteract the interpretability and is not in the scope of this research project. Rather, this can be thought of as a potential extension in future work.

Also noteworthy, `CMRules` does not use a sliding-window approach which distinguishes it from the other two computational models in this research. As a result, `CMRules` is a computationally efficient algorithm which can easily be used on large databases. Due to its two-step procedure shown in Figure 4.2, it is possible to mine association rules *and* sequential rules with one algorithm. This can provide additional insight into the data. In essence, the computational efficiency paired with the relatively basic mathematical operations, makes `CMRules` arguably robust and interpretable. Nevertheless, by transforming the rules into a graph a large share of information that `CMRules` is able to capture gets lost.

4.2.2 ETL Process

As described in the previous section, to apply `CMRules` the data needs to be in the format of a sequence database. Thus, this ETL process was concerned with the extraction of relevant information from the OMOP cdm, the transformation of that information into a sequence database and the subsequent training of the `CMRules` algorithm. The ETL process resulted in a sequential database structurally similar to the one shown on the top left of Figure 4.2. However, the set of sequences $\mathbf{x}_n = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ was defined as the temporal hospital history of each patient and the set of items $D = \{i_1, i_2, \cdots, i_d\}$ was denoted as all medical treatments or conditions across patients.

Again, the extracted information from the data in OMOP format was *condition_occurrence* and *procedure_occurrence*. Similar to the ETL in Pipeline 1, the 6 extracted columns are merged into a data structure containing personal identifiers, observations (containing procedures and conditions) and the corresponding time stamp of an observation. After sorting these by personal identifier and within personal identifier by time, a time-ordered list of medical observations per patient was derived. In order to load these patient sequences into the `CMRules` algorithm, duplicates within a discrete time step of any sequence had to be removed and the observations had to be transformed into lexicographical order [65]. Finally, these sequences were automatically

written into a TXT file where all sequences are listed as integers. A new sequence was denoted as '-2' and a new time step was denoted as '-1' in this list of observations. This text file was then used as input for `CMRules`.

After training the model, post-processing of the output was performed to transform the sequential rules into a valid and interpretable Synthea graph. For this, the rules $X \rightarrow Y$ are filtered such that X and Y only contain one observation respectively. Then, the confidence of each rule with the same antecedent is scaled to a probability of one, so the final graph depicts distributed transition probabilities from some antecedent X so all its consequent Y. Antecedents and consequent were subsequently interpreted as nodes and rules as edges of a graph. Lastly, the learnt rules of `CMRules` were automatically written into a JSON file which defines a valid Synthea module.

4.3 Pipeline 3: Dynamic Bayesian Network

Dynamic Bayesian Networks constitute the standard approach to modeling discrete-time temporal dynamics in directed graphical models. In comparison to the previous two approaches, the DBN structure learning approach therefore implicitly results in a graph structure and is not retrospectively engineered from the learned rules. A recently developed and promising approach to the issue of DBN structure learning from high dimensional data is called `DYNOTEARS` [69]. `DYNOTEARS` is an extension of the `NOTEARS` algorithm which addresses the issue of structure learning of static Bayesian Networks. Both algorithms were developed by QuantumBlack Lab of McKinsey & Company, which potentially explains the usage of algorithmic methodology which is common in econometrics. However, transferring this sophisticated methodology to the realm of EHR analyses may prove beneficial and accelerate the development of patient data analyses. Therefore, `DYNOTEARS` will be explained in the following section before the ETL process from OMOP data to `DYNOTEARS` is presented.

4.3.1 DYNOTEARS

`DYNOTEARS` is a score-based optimization approach to learning DBNs from high-dimensional data. The authors propose that `DYNOTEARS` is able to learn time series of any order and without any assumptions about the underlying graph topologies [69]. To achieve this, `DYNOTEARS` is building upon an approach extensively used in the field of econometrics, namely structural vector autoregressive models (SVARM) [70, 71]. Therefore, SVARM models will be introduced first before the actual `DYNOTEARS` algorithm is presented.

Structural Vector Autoregressive Models (SVARM)

In the `DYNOTEARS` algorithm, the data is structured using a Structural Vector Autoregressive Model (SVAR). Generally, SVAR is a class of statistical models used to analyze multivariate time series data in macroeconomics. The SVAR model used in `DYNOTEARS` follows a standard approach implemented and explained in multiple scientific studies [70, 71, 72]. More precisely, it models temporal data in the form shown in Equation 4.14. Whereas the matrix $\mathbf{W}$ represents the *intra*-slice dependencies, the matrix $\mathbf{A}_i$ with $i \in \{1, \cdots, p\}$ represent the *inter*-slice dependencies of the DBN visualized in Figure 4.3. This means the DBN structure is learned assuming variables influence each other within a time step and across time. Also, the variable $\mathbf{z}$ in Equation 4.14 represents a vector of error variables. The utility of error variables will be discussed in detail in the following subsection.

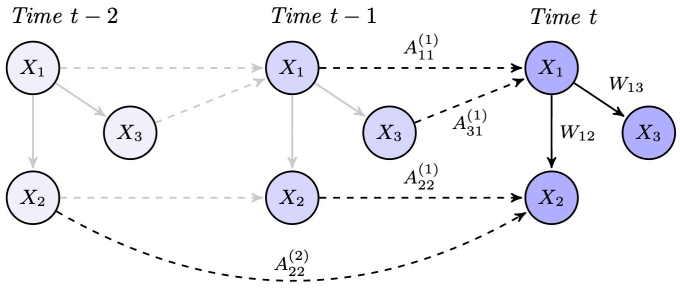


Figure 4.3: Visualization of intra (solid lines) and inter (dashed lines) slice dependencies.

$$\mathbf{x}_{n,t}^{\intercal} = \mathbf{x}_{n,t}^{\intercal}\mathbf{W} + \mathbf{x}_{n,t-1}^{\intercal}\mathbf{A}_1 + \cdots + \mathbf{x}_{n,t-p}^{\intercal}\mathbf{A}_p + \mathbf{z}_{n,t}^{\intercal} \tag{4.14}$$

As aforementioned, a BN is a directed acyclical graph. In the case of DBNs, the acyclicity constraint is only concerned with the intra-slice edges of the graph, meaning with the adjacency matrix $\mathbf{W}$. This is due to the fact that the edges of $\mathbf{A}$ only go forward in time and can thus not create any cycles. In other words, if the edges within a time-step of the graph do not contain cycles, the graph structure consisting of $\mathbf{W}$ and $\mathbf{A}$ is assumed to parameterize a DBN [69]. Secondly, it is assumed that the network structure is constant across time and is of the form shown in Equation 4.15. The effective sample size hereby is $m = N(T+1-p)$. In Equation 4.15, $\mathbf{X}$ is an $m \times D$ matrix whose rows are $\mathbf{x}_{n,t}^\intercal$ and the matrices $Y_1, \cdots, Y_p$ are time-lagged versions of $\mathbf{X}$.

$$\mathbf{X} = \mathbf{X}\mathbf{W} + \mathbf{Y}_1\mathbf{A}_1 + \cdots + \mathbf{Y}_p\mathbf{A}_p \tag{4.15}$$

The structural equation modelling (SEM) in Equation 4.15 can be further simplified by defining $\mathbf{Y} = [\mathbf{Y}|\cdots|\mathbf{Y}_p]$ to be the $m \times pD$ matrix of time-lagged data and $\mathbf{A}^\intercal = [\mathbf{A}^\intercal|\cdots|\mathbf{A}_p]^\intercal$ the $pD \times p$ matrix of inter-slice weights. This concise form of the SEM can be seen in Equation 4.16

$$\mathbf{X} = \mathbf{X}\mathbf{W} + \mathbf{Y}\mathbf{A} + \mathbf{Z} \tag{4.16}$$

Given the data in $\mathbf{X}$ and $\mathbf{Y}$, the goal is to estimate the weighted adjacency matrices $\mathbf{W}$ and $\mathbf{A}$ which constitute the causal structure of the DBN. This problem is formulated as an optimization problem of minimizing the least-squares loss while holding the acyclicity constraint for $\mathbf{W}$ [69], which is defined as:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{A}} \mathcal{L}(\mathbf{W},\mathbf{A}) \text{ s.t.} \mathbf{W} \text{ is acyclical,} \\ &\text{where } \mathcal{L}(\mathbf{W},\mathbf{A}) = \frac{1}{2n}||\mathbf{X} - \mathbf{X}\mathbf{W} - \mathbf{Y}\mathbf{A}||_F^2 \end{aligned} \tag{4.17}$$

The optimization problem defined in Equation 4.17 is modified by regularization terms of l_1 loss to introduce sparsity to the weighted adjacency matrices $\mathbf{W}$ and $\mathbf{A}$. The l_1 regularization term is a hyperparameter of `DYNOTEARS` and can be defined for each matrix in the terms $\lambda_\mathbf{W}$ and $\lambda_\mathbf{A}$, respectively. Considering this, the regularized optimization problem becomes:

$$\begin{aligned} &\min_{\mathbf{W},\mathbf{A}} f(\mathbf{W},\mathbf{A}) \text{ s.t.} \mathbf{W} \text{ is acyclical,} \\ &\text{where } f(\mathbf{W},\mathbf{A}) = \mathcal{L}(\mathbf{W},\mathbf{A}) + \lambda_\mathbf{W}||\mathbf{W}||_1 + \lambda_\mathbf{A}||\mathbf{A}||_1 \end{aligned} \tag{4.18}$$

The major difficulty in the regularized optimization problem statement of Equation 4.18 is the acyclicity of $\mathbf{W}$. This issue is solved by relying on the acyclicity constraint developed for the predecessor algorithm `NOTEARS` which is used to learn static BN structures [73, 74]. Score-based structure learning of DBNs usually is defined as a combinatorial problem. Each potential DAG in the search space is evaluated and ranked based on a discrete score associated with its fit to the observed data. However, `DYNOTEARS` reformulates this issue to a continuous optimization problem. Thereby structure learning is not a NP-hard problem anymore, which makes `DYNOTEARS` a scalable approach to real-world structure learning problems.

This is achieved by reformulating the acyclicity constraint of the graph s.t. the connection between trace of matrix power and number of cycles in the graph is leveraged [73]. To be precise, the trace (the sum of its diagonal elements) of a matrix power W^k (the matrix multiplied with itself k times) denoted as $tr(W^k)$ corresponds to the amount of cycles in a graph. Therefore the objective function $h(\mathbf{W})$ of matrix $\mathbf{W} \in R^{D \times D}$ defines a DAG, if and only if:

$$h(\mathbf{W}) = tr(e^{W \circ W}) - D = 0 \tag{4.19}$$

where $W \circ W$ is the Hadamard product of the intra-slice matrix with itself. Reformulating the acyclicity constraint of $\mathbf{W}$ makes `DYNOTEARS` computationally efficient and applicable to high dimensional real-world data. Essentially, the result of applying structural vector autoregression to the time series data is a so-called structure model, which is simply a graph with directed edges. Only by introducing the acyclicity constraint on $\mathbf{W}$ and $\mathbf{A}$, the structure model becomes a DBN.

4.3.2 Cause-Effect Relationships

Importantly, the authors of the used `CausalNex` library [75] propose the learnt graph structure should be interpreted as causal structure and the edges as cause-effect relationships [76]. Dynamic Bayesian Networks belong to the class of Structural Causal Models (SCMs) [76] and the proposed 'causal dependencies' constitute a major reason why the `DYNOTEARS` model was implemented in this project. Namely, cause-effect relationships introduce another level of complexity to the research objective of learning meaningful rules. Especially in the context of analysing the medical intervention history of patients, cause-effect relationships are arguably more meaningful than statistical if-then associations. Thus, benchmarking the performance of a causal model against non-causal models is worthwhile for learning disease progressions. By contrasting the DBN structure learning approach quantitatively and qualitatively to the prior two models in Pipeline 1 and Pipeline 2, these assumptions are evaluated.

Generally, establishing causality is a challenging task conceptually and mathematically. However, the following sections illustrate how `DYNOTEARS` uncovers cause-effect relationships and how they are distinct from rules in Pipeline 1 and Pipeline 2. To achieve this, the issue of bias in statistical dependencies is revisited, the concept of causality is outlined and the methodology responsible for uncovering cause-effect relationships is explained.

In the previous two pipelines, the learnt rules were based on statistical relationships between two variables. As pointed out before, these statistical relationships have biases. A common example of statistical bias is the confounding variable and an intuitive example is shown in Figure 4.4. Assume after training some model, the learnt rule suggests "if someone is carrying a lighter, then the person is more likely to have lung cancer". However, this statistical relationship is explained by a confounding variable, namely that the person is a smoker. Therefore, the statistical relationship between the explanatory variable and the response variable is biased by the external, or *exogenous*, confounding variable. Whereas this is simply unexplained variance in a statistical model, interestingly, structural causal models mathematically account for the error caused by exogenous variables.

A structural causal model is composed of three sets, namely (1) a set of endogenous variables, (2) a set of exogenous variables and (3) a set of structural functions, one per endogenous variable as a function of other variables. Evidently, a precise distinction between *exogenous*

and *endogenous* variables is crucial. Referring to Figure 4.4, the confounding variable 'Is a smoker' is exogenous because it is external to the model. Precisely, an exogenous variable is not influenced by any other variable in the model [77]. Vice versa, an endogenous variable is influenced by other variables in the model. Every endogenous variable is a consequent of at least one exogenous variable. For example, 'Lung cancer' is an endogenous variable in the graphical causal model (GCM) of Figure 4.4 and 'Carrying a lighter' is endogenous if and only if the edge 'Is a smoker'→ 'Carrying a lighter' is present.

Part (3) in defining structural causal models is concerned with a set of structural modelling equations as defined in Equation 4.14. Some variable $\mathbf{x}_{m,t}$ is therefore modelled by all other variables within the same time step through $\mathbf{W}$ and all prior variables within the autoregressive order of $\mathbf{A}_p$. By extending on the previous example, the following paragraphs walk through the methodology of `DYNOTEARS` to identify cause-effect relationships from observational data.

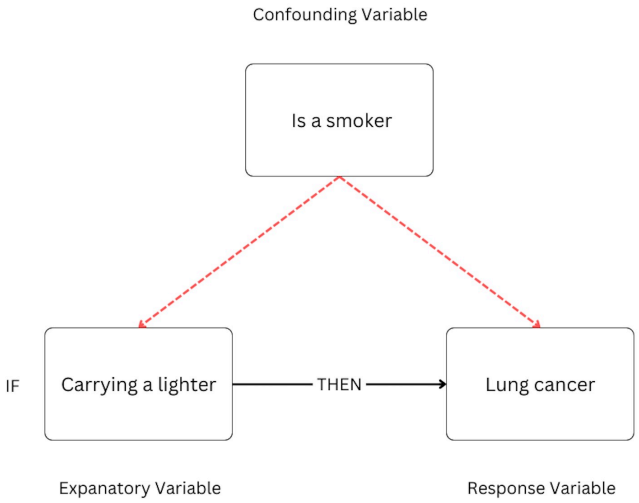


Figure 4.4: Example of a confounding variable in a IF-THEN rule.

Assume we have three variables X, Y, Z within the data that are observed in the following chronological order:

1. X = Is a Smoker
2. Y = Carrying a Lighter
3. W = Lung Cancer

First, the variables within the model must be put into *causal order*. Within a time series, this is simply their chronological order $X \rightarrow Y \rightarrow W$. Put simply, a potential cause-effect relationship can only go forward in time. Then, the model is defined by the following structural equations which are of the same form as the general definition in Equation 4.16:

$$\begin{aligned} X_t &= \alpha_1 X + \mathbf{z}_{Xt} \\ Y_t &= \beta_1 X + \beta_2 Y + \mathbf{z}_{Yt} \\ W_t &= \gamma_1 X + \gamma_2 Y + \gamma_3 W + \mathbf{z}_{Wt} \end{aligned} \tag{4.20}$$

It can be seen that the dynamic of X is influenced by the presence of itself (from other instances of the time series), the dynamic of Y is influenced by itself and the presence of X while W is influenced by the presence of itself, Y and X. These influences are endogenous. However, also exogenous influences, so potential confounding variables, are mathematically represented in each equation in the form of the *error variable* $\mathbf{z}$.

The error variable within each structural equation is used to introduce an exogenous, unanticipated event or influence that perturbs the variable's behavior. These deliberate manipulations are called *structural shocks* [72]. The effect a shock on some variable X has on all other variables in the system is subsequently measured to understand its impact. For example, if $\mathbf{z}_{Xt}$

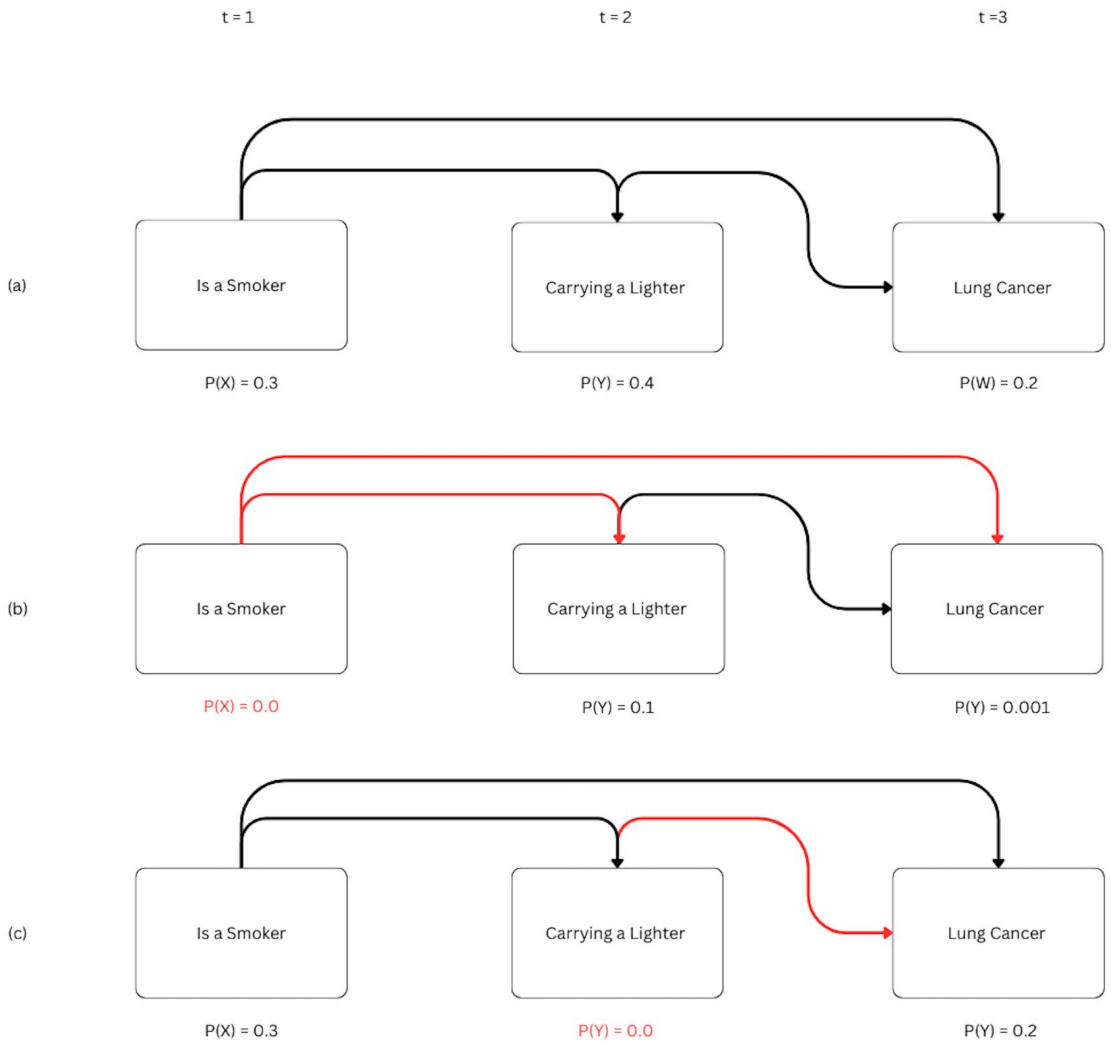


Figure 4.5: Example of structural shocks. Sub-figure (a) displays the dynamics of the observational data. In (b) a shock is administered to variable X. In (c) the effects of a structural shock on Y is shown.

is manipulated in Equation 4.20 we distort the likelihood that any person is a smoker. Subsequently, the impact this distortion has on whether the person is carrying a lighter or whether the person has lung cancer is measured. If the magnitude of change in the following variables of the causal order is significant, X is said to cause Y or W [72].

Figure 4.5 visualizes the effects of structural shocks. Assume Figure 4.5 (a) displays the dynamics of the actual data. It can be seen that 30% of the data identified as smokers, 40% of individuals at $t = 2$ were observed to carry a lighter and 20% have lung cancer. Note that these probabilistic values are for explanatory purpose only. Now, a structural shock to variable X is administered in Figure 4.5 (b). Basically, the effect of having no smokers in the data is tested. While the changes in variable X are gradual in the actual SVAR model, this example illustrates the method more clearly. The change in X will have an influence on all endogenous variables

in the causal order following X. If our data represents the true population, the probability of observing variable W will change because being a smoker has a causal effect on having lung cancer. In a second step, variable Y is shocked while all other are held constant in Figure 4.5 (c). This shock will not influence the presence of W (Lung cancer).

After the shocks are administered, the magnitude of effects throughout the system is measured. In the example, it will become evident that a change in variable X (Smoker) will have a significantly higher effect on the presence of W (Lung Cancer) than variable Y (Carrying a Lighter). In this way, X is estimated to cause W, rather than Y. Depending on the hyperparamter threshold λ, only the most significant cause-effect relationships are extracted. Whereas this approach should not be viewed as establishing true causality from observational data, it is a sophisticated method for *estimating* cause-effect relationships within the models framework. A confounding variable will not be detected and excluded from the models in Pipeline 1 and Pipeline 2. However, assuming the right hyperparameter threshold and representative data, `DYNOTEARS` will account for these unwanted influences and identify more meaningful temporal rules than the previous two models.

Besides estimating cause-effect rules, `DYNOTEARS` can also be differentiated from the models in the previous two pipelines on other methodological aspects. For example, `DYNOTEARS` is the only algorithm that explicitly outputs the autoregressive order p associated with every rule. Also, it computes intra and inter slice edges between single observations. In comparison, the baseline computes intra-slice edges between single observations and TARM only computes inter-slice edges between sets of observations. Another useful aspect of the algorithm can be found in the whitelist and blacklist which can be defined. Whereas a whitelist of edges computes a DBN only on those specified edges, a blacklist of edges defines directed edges that should not be included in the final graph. Especially in the medical context, expert human intervention through black and whitelists can greatly improve the final results. Finally, a quite obvious advantage of the DBN structure learning compared to the first two models is that it directly results in a directed acyclical graph (DAG). While information gets lost during the process of turning a set of rules into a graph in Pipeline 1 and 2, this issue is not present in Pipeline 3.

4.3.3 ETL Process

In order to perform DBN structure learning from any OMOP database, the relevant concept_id columns have to be extracted. For this study, the focus was on learning dependencies of conditions and procedures. Therefore, the relevant columns from the OMOP cdm were identified as '*condition_concept_id*' from the condition table and '*procedure_concept_id*' from the procedure table. Similarly as in the previous pipelines, the concept_ids were extracted with the corresponding personal identifier and the date of the clinical encounter.

After sorting the observations by personal identifier and by date, the conditions and procedures were dummy encoded in a $T \times D$ matrix where D is defined as all clinical observations across patients and T is defined as time steps $T = t_1, t_2, \cdots, t_{max}$, where t_{max} can vary across patients. Now assuming N is defined as the total sample size for analyses, the input for the `DYNOTEARS` algorithm is a list of N matrices with size $T \times D$ respectively, where observations are dummy encoded columns and rows are ordered ascending regarding time.

Chapter 5

Experimental Setup

This research is exploratory in the sense that, to the author's knowledge, there is no previous study on the subject of comparing temporal rule learning approaches from standardized EHR with the goal of visualizing them in graphs. That being said, the evaluation of the final graphs will be approached from a quantitative and a qualitative perspective. The two evaluation approaches will be discussed below.

5.1 Quantitative Experiments

The quantitative graph evaluation method is concerned with the structure of each graph. The aim is to obtain an indication under which experimental setting the respective models have the capacity to learn an interpretable graph from the medical time series. The experimental settings are manipulated on two dimensions.

The first dimension hereby is concerned with varying the size of data samples. By testing each model pipeline from chapter 4 on different sample sizes, one can derive evidence for or against the robustness of a model to varying sizes of data and its dimensionality. Especially for the real-world use case of learning temporal rules from standardized EHR, a model which exhibits robust performance independent of the data size is preferred. Thus, five sample sizes will be compared. An overview is given in Table 5.1. All data samples were derived with a random seed of 0 to enable reproducibility. Note that Dataset 1 is the complete VONKO data introduced in chapter 3. Whereas the sample size in Table 5.1 is defined as the amount of patients and time series, the average sequence length is the arithmetic mean of all time series per sample. Lastly, the dimensionality of the dataset is defined as the amount of unique medical

Table 5.1: Description of the data samples used for experiments.

Referred Name	Sample Size N	Average Sequence Length	Dimensionality
Dataset 1	11641	5.487	726
Dataset 2	5000	5.513	512
Dataset 3	500	6.130	200
Dataset 4	50	5.720	68
Dataset 5	10	6.400	26

observations across patients per sample.

The second dimension of experimental manipulation is introduced by hyperparameter tuning. Each model pipeline has at least one hyperparameter which defines the granularity level on which rules are learned. This experimental manipulation also aims to provide an indication of the robustness of each pipeline. In addition, it can be seen as an investigation of the trade-off between graph complexity and accuracy. For example, a highly complex graph may be very accurate and represent all important dependencies in the data but suffer from less accessibility and transparency. Vice versa, a graph with very low complexity has higher accessibility and transparency, but it may not display all important rules that represent the data. Hence, this experimental manipulation aims to investigate the relationship between hyperparameter granularity and the resulting graph complexity with regards to accessibility and accuracy. To achieve this, three levels of granularity are defined for each hyperparameter and model. The settings for Pipeline 1 are shown in Table 5.2, for Pipeline 2 in Table 5.3 and for Pipeline 3 in Table 5.4. Note that all hyperparameters and their respective utility are explained in chapter 4.

In summary, three hyperparameter settings are tested on five datasets per pipeline. This results in 45 graphs in total. The following section introduces graph complexity measures which are used as evaluation method for these experimental settings.

Table 5.2: Enumerated hyperparameter (HP) setting for the MM model.

Setting	Threshold
1	0%
2	15%
3	30%

Table 5.3: Enumerated hyperparameter (HP) setting for the TARM model.

Setting	*MinSup*	*MinConf*
1	5%	1%
2	10%	5%
3	20%	10%

Table 5.4: Enumerated hyperparameter (HP) setting for the DBN model.

Setting	λ **W**	λ **A**
1	0.01	0.005
2	0.02	0.01
3	0.04	0.02

5.1.1 Evaluation

Each graph structure is evaluated based on six graph complexity measures. These include the number of nodes, number of edges, the graph density, the average clustering of the graph, the

amount of strongly connected components and the flow hierarchy. For the purpose of later understanding and reasonable interpretation, each of the measures will be shortly introduced here. Whereas the number of nodes and edges is straightforward, the remaining graph complexity measures require individual definitions, which are listed below:

- *Graph density*: Measure of how connected a graph is, representing the ratio of actual edges to possible edges. It indicates the proportion of connections present in the graph and can range from 0 (sparse) to 1 (dense), with higher values indicating a higher level of connectivity.
- *Average Clustering*: Measures the extent to which nodes in a graph tend to cluster together. It quantifies the level of local interconnectedness within the graph. It can range from 0 to 1, where higher values indicate a higher degree of clustering in the graph.
- *Amount strongly connected components*: A strongly connected component in a directed graph refers to a subgraph where there is a directed path between every pair of nodes in the component. The relevant metric in this study is the amount of strongly connected components.
- *Flow Hierarchy*: The flow hierarchy of a graph measures the relative influence or importance of nodes in directing the flow of information within the graph. It indicates the extent to which certain nodes have a higher influence on the flow compared to others, with values ranging from 0 (equal influence) to 1 (hierarchical influence).

In addition to these individual graph matrices, the intersection between all graphs will be calculated as a measure of direct comparison. The *intersection* of two graphs is defined as the set of nodes and edges which are present in both graphs. The intersection between two graphs A and B will be represented as the percentage of overlapping nodes and edges, respectively. Therefore, the intersection can be viewed as a measure of how similar the learned rules are within models across experimental settings as well as across models. As the three models themselves vary in computational complexity, the measure of intersection can be used to identify if and to what extent models which are mathematically more sophisticated learn differing rules compared to simple probabilistic models.

5.2 Qualitative Evaluation

After running the above mentioned quantitative experiments, the second step of this research is to investigate the graphs qualitatively. For this, one graph per model pipeline was selected. Whereas the quantitative evaluation is used to assess the computational models from a technical perspective, the qualitative evaluation using expert knowledge is concerned with the medical relevance of this project. These three graphs will be used to gather expert opinions of two physicians by means of a questionnaire.

As it is infeasible to gather the expert opinion across results of all experimental settings, one graph per model pipeline was pre-selected by the researcher. To facilitate the comparability between models, all three graphs in the questionnaire were selected to have around the same graph complexity identified in the quantitative experiments. The aim of this was to keep the graph complexity as constant as possible, while the distinguishing factor of the resulting graph is the model of computational intelligence. The model of computational intelligence can subsequently be argued to have learnt the temporal rules of real world medical data better or worse

in relation to each other. Good or bad is hereby defined on three metrics of expert knowledge, namely the interpretability of the graph, how coherent it is with the domain knowledge of the expert and how many 'unreasonable' edges are identified. Details on the structure of the questionnaire are presented in the following.

5.2.1 Questionnaire

For each graph, the experts were presented with four statements regarding their structure and medical relevance. The questionnaire assesses the expert's agreement with each of the statements on a Likert Scale from 1 ('I strongly disagree') to 5 ('I strongly agree'). The physicians were advised to answer each question according to their subjective best medical knowledge. The same four statements were answered for each of the three graphs, resulting in 12 qualitative indications of medical relevance per expert. The questionnaire was handed to two physicians at the UKE and the exact statements of the questionnaire are listed below. Note that question 2 accompanies a definition of graph terminology to ensure the subsequent statement is interpreted correctly.

1. The graph is interpretable.
2. Definition: A path in a directed graph is defined as a sequence of nodes one gets when following the directed edges.
   Statement: The paths in the graph can be interpreted as chronological sequence of hospital encounters which are coherent with my medical knowledge about lung cancer.
3. Some direct edges depict reasonable relationships regarding cause-effect in lung cancer treatment.
4. Some edges within the graph are absurd or display unreasonable connections.

In the last part of the questionnaire, three multiple-choice questions had to be answered. Whereas the previous Likert Scale questions measure the individual graphs qualitatively, the following questions were aimed to let the experts compare the graph and pick a choice. Only one graphs could be selected for each of the following multiple-choice questions:

1. The most interpretable graph was:
2. The graph with the most meaningful rules according to my medical knowledge was:
3. The graph with the most unreasonable connections according to my medical knowledge was:

The questionnaire was handed to the physicians online via Google Forms with and instruction manual and the three graphs. There was no time limit and the answers were gathered anonymously, however, the answers can be grouped per person. This is done to potentially outline individual response trends.

Finally, as only one graph per model pipeline was included in the questionnaire, the following list depicts information on HP setting and dataset used for computing each graph. Whereas the graphs from Pipeline 2 and Pipeline 3 were computed on the complete dataset, note that Graph 1 from Pipeline 1 was computed on the smallest dataset. This was the only practical solution as the baseline model otherwise computes graphs which are too large to visualize and interpret. All three graphs are shown in section B of the Appendix.

1. Graph 1: MM on the smallest dataset (5) with HP 3
2. Graph 2: TARM on the complete dataset (1) with HP 2
3. Graph 3: DBN on the complete dataset (1) with HP 2

Chapter 6

Results

As proposed in chapter 5, the graphs were evaluated quantitatively and qualitatively. First, the qualitative results will be presented. The quantitative results are divided into tables describing the individual graph complexity and tables depicting the intersection of graphs. Thus, the resulting graphs are evaluated individually as well as across hyperparameters, data samples and models. Finally, the qualitative results present the expert opinion on the graphs.

6.1 Quantitative Results

The first three tables, namely Table 6.1, Table 6.2 and Table 6.3, depict individual graph complexity measures grouped by model and hyperparameter (HP) setting on the 5 datasets. For an overview of HP setting, datasets and complexity measures see chapter 5.

Table 6.1: Resulting graph complexity measures for the MM pipeline per data sample and hyperparameter (HP) setting.

HP	Data	Nodes	Edges	Density	Avg. Clustering	Components	Flow Hierachy
1	1	725	3458	0.006	0.268	453	0.421
	2	513	2203	0.008	0.270	367	0.559
	3	201	714	0.017	0.187	150	0.613
	4	70	200	0.041	0.117	58	0.700
	5	28	75	0.099	0.224	23	0.706
2	1	685	1028	0.002	0.005	684	0.993
	2	488	718	0.003	0.004	488	0.995
	3	194	262	0.006	0.018	193	0.984
	4	69	92	0.019	0.021	69	0.967
	5	27	32	0.045	0.000	27	1.000
3	1	566	633	0.001	0.000	566	1.000
	2	405	454	0.002	0.000	405	0.997
	3	167	172	0.006	0.000	167	1.000
	4	64	61	0.015	0.000	64	1.000
	5	21	22	0.052	0.000	21	1.000

Table 6.2: Resulting graph complexity measures for the TARM pipeline per data sample and hyperparameter setting.

HP	Data	Nodes	Edges	Density	Avg. Clustering	Components	Flow Hierachy
1	1	19	60	0.175	0.292	16	0.883
	2	19	62	0.181	0.328	17	0.919
	3	21	89	0.211	0.358	14	0.730
	4	20	81	0.213	0.354	15	0.802
	5	28	153	0.202	0.428	22	0.777
2	1	14	30	0.164	0.156	14	1.000
	2	14	30	0.164	0.142	14	1.000
	3	15	41	0.195	0.270	15	0.951
	4	14	36	0.197	0.222	13	0.944
	5	28	153	0.202	0.428	22	0.777
3	1	7	11	0.261	0.109	7	1.000
	2	7	11	0.261	0.109	7	1.000
	3	10	19	0.211	0.135	10	1.000
	4	8	14	0.250	0.141	8	1.000
	5	12	27	0.204	0.212	12	1.000

Table 6.1 depicts the complexity information regarding all graphs from Pipeline 1. The first dominant observation is the very large amount of nodes and edges Pipeline 1 produces. In addition, the observed graph density is low. Also striking is the vanishing average clustering coefficient of the graphs for hyperparameter settings 2 and 3. The amount of strongly connected components, however, can be observed to be relatively large, as the values approach the amount of nodes in almost all graphs. Finally, the flow hierarchy is small in the first hyperparameter setting, but almost always 1 for settings 2 and 3.

Table 6.2 describes the graphs from Pipeline 2. Importantly, the amount of nodes and edges is significantly less compared to Pipeline 1. The graphs are smaller but have a higher density and average clustering. The amount of strongly connected components is also high, as almost every node in each graph is considered to be a strongly connected component. However, the flow hierarchy is observed to be high across hyperparameter settings and datasets for the results of Pipeline 2.

Finally, Table 6.3 depicts the individual graph complexity measures for Pipeline 3. The amount of nodes and edges is similar to Pipeline 2, however, for the smallest dataset (Data 5) Pipeline 3 computes only about half the amount of edges to be relevant compared to Pipeline 2. The density and average clustering of the graphs is similar to Pipeline 2, and thereby significantly higher than in Pipeline 1. The amount of strongly connected components is less, as for example HP 3 on dataset 4 the amount of nodes relative to the amount of strongly connected components is almost half [see Table 6.3]. Also eye-catching is the low flow hierarchy across all hyperparameters and datasets of Pipeline 3 compared to both, Pipeline 1 and 2.

The following 5 tables summarize the intersection between any two graphs that were computed on the same dataset. As a reminder, the intersection is the percentage of identical direct edges. An intersection of '1.00' therefore means that the two compared graphs have identical

Table 6.3: Resulting graph complexity measures for the DBN pipeline per data sample and hyperparameter setting.

HP	Data	Nodes	Edges	Density	Avg. Clustering	Components	Flow Hierarchy
1	1	21	55	0.130	0.246	14	0.400
	2	20	53	0.139	0.260	13	0.396
	3	20	55	0.144	0.214	13	0.454
	4	26	67	0.103	0.233	17	0.462
	5	26	82	0.126	0.332	20	0.634
2	1	15	30	0.142	0.161	15	0.733
	2	15	29	0.138	0.145	12	0.551
	3	15	34	0.161	0.194	11	0.441
	4	17	39	0.143	0.245	9	0.307
	5	26	74	0.113	0.299	20	0.260
3	1	11	16	0.145	0.068	11	0.680
	2	11	17	0.154	0.068	11	0.647
	3	10	14	0.155	0.000	10	0.642
	4	8	16	0.285	0.111	7	0.562
	5	11	25	0.227	0.276	7	0.280

directed edges, disregarding the edge weight. In other words, any cell in the adjacency matrix shows how much the graph specified in the row overlaps with the graph specified in the column. In this way, the following tables represent a measure of graph similarity across models and hyperparameters. Note that the naming of rows and columns of each adjacency matrix follows the syntax 'Model_Hyperparameter'.

The first observation, which holds for all intersection tables, is that they are adjacency matrices. As commonly observed in adjacency matrices, the diagonal is 1. This is due to the reason that any graph is compared to itself in these cells. Also noteworthy, within each model (e.g. MM_1 - 3), the lower triangular matrix is always equal to 1. This is correct, as the hyperparameters from 1-3, and thereby the granularity of the rules, increases. One would expect a set of granular rules to be a subset of the rules the same model computes with less granular

Table 6.4: Intersection between all graphs on dataset 1 (N= 11641).

	MM_1	MM_2	MM_3	TARM_1	TARM_2	TARM_3	DBN_1	DBN_2	DBN_3
MM_1	1.00	0.30	0.18	0.01	0.00	0.00	0.01	0.01	0.00
MM_2	1.00	1.00	0.62	0.01	0.00	0.00	0.00	0.00	0.00
MM_3	1.00	1.00	1.00	0.01	0.00	0.00	0.00	0.00	0.00
TARM_1	0.63	0.10	0.08	1.00	0.50	0.18	0.43	0.23	0.10
TARM_2	0.53	0.07	0.03	1.00	1.00	0.37	0.37	0.27	0.17
TARM_3	0.36	0.00	0.00	1.00	1.00	1.00	0.55	0.45	0.36
DBN_1	0.82	0.05	0.02	0.47	0.20	0.11	1.00	0.55	0.29
DBN_2	0.80	0.07	0.03	0.47	0.27	0.17	1.00	1.00	0.53
DBN_3	0.75	0.06	0.00	0.38	0.31	0.25	1.00	1.00	1.00

Table 6.5: Intersection between all graphs on dataset 2 (N = 5000).

	MM_1	MM_2	MM_3	TARM_1	TARM_2	TARM_3	DBN_1	DBN_2	DBN_3
MM_1	1.00	0.33	0.21	0.02	0.01	0.00	0.02	0.01	0.01
MM_2	1.00	1.00	0.63	0.01	0.00	0.00	0.00	0.00	0.00
MM_3	1.00	1.00	1.00	0.01	0.00	0.00	0.00	0.00	0.00
TARM_1	0.66	0.10	0.08	1.00	0.48	0.18	0.42	0.23	0.10
TARM_2	0.53	0.07	0.03	1.00	1.00	0.37	0.37	0.30	0.13
TARM_3	0.36	0.00	0.00	1.00	1.00	1.00	0.55	0.45	0.27
DBN_1	0.81	0.06	0.02	0.49	0.21	0.11	1.00	0.55	0.32
DBN_2	0.79	0.07	0.03	0.48	0.31	0.17	1.00	1.00	0.59
DBN_3	0.82	0.06	0.00	0.35	0.24	0.18	1.00	1.00	1.00

thresholded rules. That being said, the most prominent observations for each individual table will be outlined in the following paragraphs.

Table 6.4 summarizes the intersections between any two graphs that were computed on the full dataset. Firstly, it can be seen that all DBN rules have a larger overlap (75-82%) with the MM_1 compared to the TARM (36-63%). Secondly, MM with the HP settings 2 and 3 barely have any overlap with all other 6 graphs (0.00-0.10%). On the other hand, practically no rules learned by the baseline model are present in TARM and DBN across all hyperparameters (0.00-0.02%), which can be seen on the upper right-hand of the matrix. The largest overlap between models is found between MM_1 + DBN_3 with 82% similar rules.

Table 6.5 depicts the adjacency matrix comparing all graphs trained on dataset 2. The similarity measures display resemblant patterns as described in Table 6.4. Thus, the main observation is that resemblance patterns between graphs on the full dataset and on the largest subsample ($N = 5000$) only differ selectively in a few percentage points.

The percentage of overlapping edges for all models on the $N = 500$ subsample is shown in Table 6.6. Again, the baseline model with HP settings 2 and 3 is observed to have very low similarity with all other graphs. However, on this dataset similarity between TARM and DBN graphs increases. This can be seen in the rows of DBN to the columns of TARM, which increased around 10% for any combination from the previous dataset. In simple words, more

Table 6.6: Intersection between all graphs on dataset 3 (N= 500).

	MM_1	MM_2	MM_3	TARM_1	TARM_2	TARM_3	DBN_1	DBN_2	DBN_3
MM_1	1.00	0.37	0.24	0.07	0.03	0.01	0.05	0.03	0.01
MM_2	1.00	1.00	0.66	0.02	0.02	0.00	0.01	0.01	0.00
MM_3	1.00	1.00	1.00	0.03	0.02	0.00	0.01	0.01	0.00
TARM_1	0.56	0.07	0.06	1.00	0.46	0.21	0.40	0.22	0.09
TARM_2	0.59	0.10	0.07	1.00	1.00	0.46	0.46	0.29	0.12
TARM_3	0.47	0.00	0.00	1.00	1.00	1.00	0.47	0.32	0.21
DBN_1	0.67	0.05	0.02	0.65	0.35	0.16	1.00	0.62	0.25
DBN_2	0.62	0.06	0.03	0.59	0.35	0.18	1.00	1.00	0.41
DBN_3	0.57	0.07	0.00	0.57	0.36	0.29	1.00	1.00	1.00

Table 6.7: Intersection between all graphs on dataset 4 (N= 50).

	MM_1	MM_2	MM_3	TARM_1	TARM_2	TARM_3	DBN_1	DBN_2	DBN_3
MM_1	1.00	0.46	0.30	0.22	0.21	0.02	0.23	0.14	0.06
MM_2	1.00	1.00	0.66	0.13	0.12	0.00	0.15	0.09	0.04
MM_3	1.00	1.00	1.00	0.11	0.10	0.00	0.05	0.00	0.00
TARM_1	0.54	0.15	0.09	1.00	0.68	0.17	0.38	0.27	0.12
TARM_2	0.27	0.07	0.04	0.36	1.00	0.09	0.19	0.14	0.04
TARM_3	0.29	0.00	0.00	1.00	1.00	1.00	0.43	0.29	0.21
DBN_1	0.67	0.21	0.04	0.46	0.43	0.09	1.00	0.58	0.24
DBN_2	0.74	0.21	0.00	0.56	0.54	0.10	1.00	1.00	0.41
DBN_3	0.75	0.25	0.00	0.62	0.38	0.19	1.00	1.00	1.00

rules of the DBN graphs are also present in the TARM graphs for this smaller dataset.

Table 6.7 presents the similarity measure of all graphs on dataset 4. It can be seen that now the graphs learned with the baseline model are more similar to the TARM and DBN graphs (0.00-0.23%) compared to the previous dataset 3 (0.00-0.07%). Besides, the largest overlap between models is still found in the DBN rules compared to the MM rules with HP setting 1 (75%).

The similarity measures for graphs of the smallest dataset of $N = 10$ are shown in Table 6.8. Generally, it can be observed that the percentages of overlapping edges are higher on average. Interestingly, the baseline model with HP 2 and 3, which resulted in very different graphs on the previous datasets, here learns 100% of rules that are found in the TARM graphs with HP 1 and 2. Whereas on larger datasets, almost no intersection was present, on the smallest dataset all rules of the baseline model are present in TARM graphs. This observation also entails that, interestingly, the highest intersection between models is not between DBN and MM anymore.

Summarizing the observations from the previous 5 tables, it can be said that the HP settings 2 and 3 for the baseline MM pipeline did result in very dissimilar graphs compared to all others. Only on the smallest dataset ($N = 10$), these experimental settings produce comparable results. Secondly, the highest overlap between models on the same dataset was always found in the percentage of DBN rules compared to the baseline MM_1. Note that the HP setting for MM_1

Table 6.8: Intersection between all graphs on dataset 5 (N= 10).

	MM_1	MM_2	MM_3	TARM_1	TARM_2	TARM_3	DBN_1	DBN_2	DBN_3
MM_1	1.00	0.43	0.29	0.95	0.95	0.15	0.53	0.52	0.23
MM_2	1.00	1.00	0.69	1.00	1.00	0.06	0.50	0.47	0.03
MM_3	1.00	1.00	1.00	1.00	1.00	0.09	0.41	0.41	0.00
TARM_1	0.46	0.21	0.14	1.00	1.00	0.18	0.45	0.41	0.12
TARM_2	0.46	0.21	0.14	1.00	1.00	0.18	0.45	0.41	0.12
TARM_3	0.41	0.07	0.07	1.00	1.00	1.00	0.48	0.41	0.15
DBN_1	0.49	0.20	0.11	0.84	0.84	0.16	1.00	0.88	0.30
DBN_2	0.53	0.20	0.12	0.84	0.84	0.15	0.97	1.00	0.34
DBN_3	0.68	0.04	0.00	0.76	0.76	0.16	1.00	1.00	1.00

is 0, so practically no filter is applied to the rule learning process. Also noteworthy, the smaller the dataset, the higher the general resemblance across models and hyperparameters. Finally, it is important to outline that the similarity between TARM and DBN graphs for large datasets (≥ 500) is only around 50-60%.

Finally, whereas the previous tables depict the intersection of all graphs *per dataset*, all graph intersections *per model* are shown in section C of the Appendix. Here, only the main observations will be highlighted. The overview of graph intersections per model provides an insight into how robust the individual rule learning approaches are across dataset sizes and hyperparameters.

The most fundamental observation trends are two-fold. Firstly, (1) the larger the dataset, the larger the graph (e.g. more rules are learnt). Secondly, (2) the less granular the hyperparameter setting, the larger the resulting graph. These two trends can be seen especially in Table 1 (Baseline in Pipeline 1) and Table 3 (DBN in Pipeline 3). This is represented in the fact that the model on the full dataset with the least granular HP setting computes the graph which all other graphs have the largest intersections with. Put simply, these graphs are the largest and the others are subsets of them. In Table 2 (TARM in Pipeline 2), however, a strikingly different trend can be observed. Whereas observation trend (2) still holds, observation (1) is reversed. In other words, for the TARM model in Pipeline 2, a smaller dataset resulted in larger graphs.

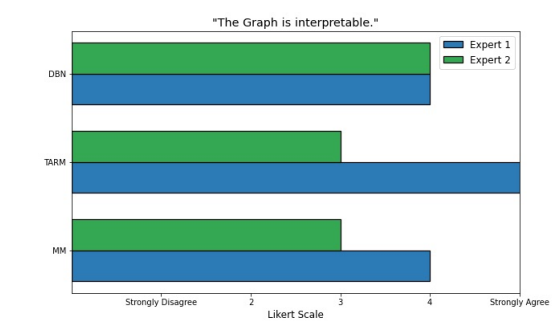
6.2 Qualitative Results

The qualitative results are summarized in Figure 6.1 and Table 6.9. In Figure 6.1 the answers of the two physicians are contrasted for each graph and statement, where the statement is provided as a subfigure title. For example, Figure 6.1a shows that both experts rated the DBN graph as 4 on the statement 'The Graph is interpretable', where 5 indicates strong agreement. Table 6.9, however, displays the expert's picks for the most interpretable graph, the graph with the most meaningful rules and the graph with the most unreasonable rules.

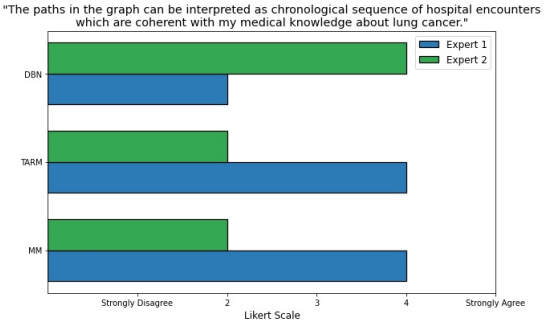
The first observation for Figure 6.1 is that the subjective opinions vary consistently between experts. Only one graph was rated the same (see above). Also noteworthy, Expert 1 generally provided more positive answers. Figure 6.1a, Figure 6.1b and Figure 6.1c are positive statements and Expert 1 indicates equal or greater agreement in 77% of the cases. In contrast, Figure 6.1d is a negative statement and Expert 1 agrees less in $\frac{2}{3}$ of the cases. The largest difference in opinion is visible in Figure 6.1c, which assesses whether the graph can be said to depict cause-effect relationships. Whereas Expert 1 indicates maximal agreement for the TARM and DBN models to display cause-effect relationships, Expert 2 disagrees.

Table 6.9: Expert picks for most interpretable graph, most meaningful rules and most unreasonable rules.

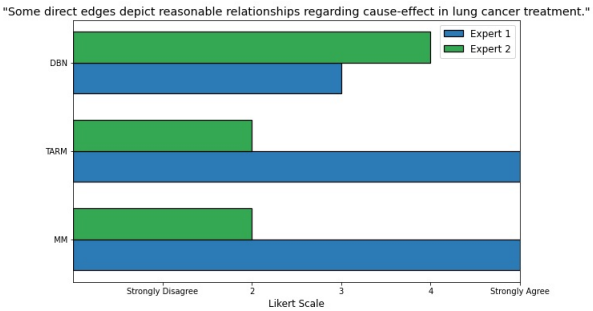
	Expert 1	Expert 2
The most interpretable graph was:	TARM	DBN
The graph with the most meaningful rules was:	TARM	DBN
The graph with the most unreasonable rules was:	DBN	MM



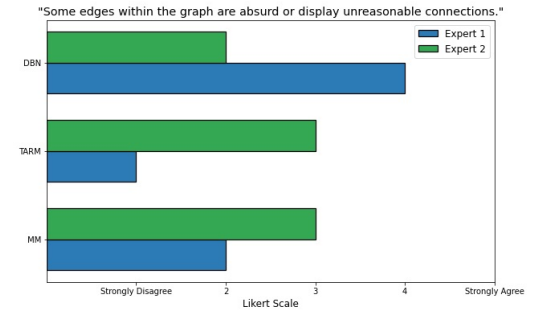
(a) Statement 1



(b) Statement 2



(c) Statement 3



(d) Statement 4

Figure 6.1: The expert answers summarized by statement and graph.

The overall subjective preference, however, is concisely represented in Table 6.9. Expert 1 chooses the TARM graph from Pipeline 2 for both positively coined phrases, whereas Expert 2 chooses the DBN graph from Pipeline 3. The choice for the negatively coined phrase was the DBN graph for Expert 1 and the MM (Baseline) graph for Expert 2.

Chapter 7

Discussion

In the following, the three types of results are discussed separately. Following the same structure as chapter 6, first the individual graph complexity measures are discussed. Subsequently, the graph intersections are discussed as a measure of similarity. To outline specific trends, first the graph similarity within datasets is addressed before the graph similarity within models is discussed. Lastly, the qualitative expert opinion will be elaborated on before the project is embedded into the current scientific literature. The conclusion concisely integrates all insights across results and finally, a critical perspective on the project is taken to highlight potential further research objectives.

7.1 Graph Complexity

Firstly, the graph complexity measures regarding Pipeline 1, so the baseline model (MM), will be discussed. It was observed that the amount of nodes and edges, so the sizes of all graphs across HPs and datasets, were significantly higher compared to the latter two models. This needs to be attributed to the fact that the baseline model is not *learning* a set of rules which describes the data in a compact manner. Rather, all conditional probabilities with an autoregressive order of $p = 1$ are calculated and also represented in the graph. Actually, this resulted in graphs which are too large for Synthea to display. This lack of rule reduction in the baseline model also made it infeasible to qualitatively evaluate graphs from Pipeline 1 for larger datasets (≥ 500). This is a direct result of the graph complexity growing linearly with the dimensionality of the dataset [see Table 5.1].

Whereas the baseline model in Pipeline 1 only *extracts* rules, Pipeline 2 (TARM) *learns* a set of rules. As aforementioned, rule extraction involves the identification of relevant patterns and relationships within data, while rule learning is the process of constructing a set of rules that describe the data in a compact and interpretable manner. This claim is supported by the fact that the size of the graph was observed to stay consistent across dataset sizes and dimensionality. Whereas this holds for datasets 1-4, quite uncommon results were observed for the smallest dataset ($N = 10$). Namely, the amount of edges (i.e. the learnt rules) were observed to grow disproportionately in Pipeline 2. With a closer look at the methodology of the `CMRules` algorithm, however, the explanation for this behavior becomes obvious. Rules across the whole sequence are filtered by the two hyperparameter thresholds *support* and *confidence*. These probabilistic thresholds, especially for HP settings 1 and 2 [see Table 5.3], are met very easily

on a dataset of $N = 10$. For example, in a dataset with $N = 10$ and a $MinSup = 10\%$ every single rule that is in the dataset is learnt. This is due to the fact that $MinSup = \frac{\text{Count(Rule)}}{N}$. In a dataset of $N = 10$, any rule which occurs in any time series has a support of $\frac{1}{10} = 0.1 = 10\%$. Thus, one can argue that `CMRules` is not robust with regards to small datasets.

Lastly, also Pipeline 3 *learns* a set of compact rules which describe the data. For dataset 1-4 both, Pipeline 2 and Pipeline 3, learn graphs of similar complexity. However, whereas `CMRules` in Pipeline 2 does not display robust performance on smaller datasets, `DYNOTEARS` in Pipeline 3 does so. Another important observation, however, is the significantly lower flow hierarchy of all graphs from Pipeline 3. As a reminder, a flow hierarchy of 1 indicates a hierarchical structure of the graph. Arguably, for a graph representing chronological disease progressions, a hierarchical graph is preferred. However, the explanation why Pipeline 3 produces less hierarchically structured graphs is found in two reasons, where both reasons directly depend on the graph visualization used. More precisely, because the Synthea Module Builder does not explicitly visualize a time axis, the complexity of DBN rules can not be accounted for appropriately and create circles in the graph. The first scenario in which a cycle may be constructed is due to intra-slice edges. Intra-slice edges, which are rules within a time step, disturb the chronological order of nodes during display. Second, it is possible for the DBN to learn a rule from a variable X to itself at a later point in time. This will also result in a cycle in Synthea graphs, because any variable is only represented once without temporal information. These two methodological obstacles create cycles in the Synthea graphs of Pipeline 3, which in turn decreases their flow hierarchy.

7.2 Graph Intersection

This section places the measured graph intersections into perspective. The first subsection answers the question: "How similar are the learnt rules between models on the same dataset?". In contrast, the second subsection addresses the question: "How similar are the learnt rules within each model on decreasing sample sizes?".

7.2.1 Within Datasets

The first major observation is that on almost all datasets, the largest overlap of rules is found between either any TARM or DBN model and MM_1, meaning the baseline model with HP setting 1. The explanation for this ties in to the aforementioned lack of rule reduction in the baseline model. The HP setting 1 does not filter rules, rather it calculates and extensively represents all conditional probabilities with autoregressive order $p = 1$. Considering this, it is not surprising that a majority of TARM and DBN rules overlap with MM_1. Actually, all rules of TARM and DBN which do not overlap with MM_1 necessarily are rules with autoregressive order > 1.

Another important observation regarding the algorithmic extension introduced to the baseline model in chapter 4 needs elaboration. The algorithmic extension was implemented with the purpose of mathematically relativising the influence of prevalent observations. The aim was to reduce the statistical bias of conditional probabilities. However, the percentage of overlap strongly suggests that this algorithmic extension computes rules which are very dissimilar to

all other models. This suggests that the extension actually introduced a new bias or distorted the learnt probability distribution to a counterproductive magnitude. Nevertheless, it should be noted that the algorithmic extension is not present in HP setting 1 (Model MM_1) and the bias only holds for MM_2 and MM_3. Hence, a valid baseline model was still present in this study.

The third and last observation worth discussing is the low graph similarity between TARM and DBN models. Besides on dataset 5 ($N = 10$), the maximal overlap between TARM and DBN graphs across hyperparameters is around 50-60%. If one combines this knowledge with the prior observation that the individual graph complexities of both models are similar, the logical consequence is that the two models learn qualitatively different rules to some extent. Both models, however, also approach temporal rule learning with different methodologies as outlined in chapter 4. The fact that the rules qualitatively differ on the same dataset and across hyperparameters, can thus be interpreted as evidence for their respective methodology. Going one step further, the main differences methodologically between the two computational models are that (1) the DBN learns intra-slice edges and (2) the DBN estimates cause-effect relationships through structural shocks. Nonetheless, the hyperparameter threshold for intra-slice edges was intentionally set to a lower granularity, so only extremely prominent intra-slice edges are included in the graphs of Pipeline 3. Taken together, the results of this study suggest that the estimation of cause-effect relationships through structural shocks produces qualitatively different rules compared to simple probabilistic temporal association rule mining.

7.2.2 Within Models

The similarity measure between graphs of the same model and with decreasing sample sizes resulted in one striking observation. For Pipeline 1 (MM) and Pipeline 3 (DBN), two intuitive observations hold. These are: (1) The larger the dataset, the larger the graph and (2) the less granular the HP setting, the larger the graph. Put together, the model learnt on the largest dataset with the less granular HP settings resulted in the largest graph (e.g. the largest amount of rules were learnt). Because the decreasing subsamples are randomly drawn from the same population, these general trends are expected from a robust model. In other words, the set of rules learnt on the largest dataset represent the population of lung cancer patients to the best of our knowledge. If the same model is trained on randomly drawn data subsets, one expects a robust model to learn a subset of the rules. If a model learns significantly more rules on subsamples of data, the model is representing another underlying probability distribution and is therefore not robust to varying dataset sizes. This is exactly what was observed for Pipeline 2 (TARM).

The explanation for this model behavior from a methodological perspective ties directly to the prior mentioned example. More precisely, the rule filtering thresholds of support and confidence are not functional with decreasing sample sizes. Whereas one could argue that the algorithm would need hyperparameter tuning depending on the dataset, it is necessary to note that the other two models perform the desired behavior without hyperparameter tuning. These observations are very relevant to research objective Q2. While the objective was to identify models which are able to produce robust results on small sample sizes, it was found that temporal association rule mining should not be used on small datasets. In addition, the baseline model was found to be not restrictive enough for large data and thus is not suitable

for learning graphs from large data. The Dynamic Bayesian Network, however, was identified as robust in performance and computational efficiency for small and large datasets.

7.3 Expert Opinion

A fundamental insight from the questionnaire was that the more complex models seemingly justified their computational complexity. One Expert choose the graph of Pipeline 2 (TARM) and one Expert choose Pipeline 3 (DBN) as the most interpretable and meaningful graph. Nevertheless, the experts displayed quite opposing opinions on the DBN graph. Whereas it held the most meaningful information for one, it was the least meaningful according to the second physician. A possible explanation for this can be attributed to the uncommon overall structure which the DBN graph displays compared to the remaining graphs. Mainly, the DBN graph differed in two ways. Firstly, the graph contained circles for the aforementioned reasons. The resulting decreased flow hierarchy counteracts the thought of chronological observations and can arguably confuse an interpreting person. Secondly, the start node was not present in the DBN graph. This distorts the structure and interpretation of the graph and seems like a mistake without proper methodological explanation.

The explanation for this structural flaw of the missing start node, however, is actually a supporting argument for the methodology of `DYNOTEARS` in Pipeline 3. Recall that `DYNOTEARS` estimates cause $\rightarrow$ effect relationships by measuring the effects of structural shocks to the dynamic system. These shocks are also applied to the start node and the effect that the presence or absence of the start node has on the dynamical system is measured. Because the start node is not identified to cause any of the first observations in the chronological order, no edge from 'START' $\rightarrow X_{t=1}$ is included in the DBN. As the start node was manually added during pre-processing of the time-series, it actually should not cause any medical observation. Thus, the exclusion of the start node supports the claim that Pipeline 3 computes the most meaningful rules. However, not understanding and negatively interpreting the missing start node is comprehensible if one assumes no methodological knowledge of the interpreting person.

As a clarifying remark, one can explain why only the start node was missing and not both, the start and the stop node, as both were added manually during pre-processing. The answer to this can be again found in the methodology of structural shocks. As it was explained in chapter 4, the variables are in causal order (e.g. chronological order), so a shock to some variable $X_{t=1}$ can only influence variables $Y_{t>1}$ at a later point in time. Because the stop node is always at $X_{t=T}$, meaning the last instance of each time-series, the shocks on the stop node can not have any effect on the dynamic system. In simpler words, the stop node can not be interpreted as the cause of any other variable, because it is always the last observation. However, it can be interpreted as an effect, because certain medical observations potentially are estimated to cause lung cancer hospital histories to end.

In summary, the graphs of Pipeline 2 (TARM) and Pipeline 3 (DBN) were best accepted by the experts to represent official medical treatment guidelines. In the optional feedback of the questionnaire, one physician referred to the official medical guidelines for lung carcinoma in Germany [78], in which he was able to identify parts of each graph. This observation can be interpreted as support for the final research question of this study. However, this interpretation needs further differentiation. Learning official medical treatment guidelines from the data can

be seen as a methodological success of the computational models. Nevertheless, the fact that not all direct edges are interpreted as meaningful and in line with medical guidelines can be explained in two ways. Either, the computational model learns irrelevant dependencies that should not be included in such a chronological visualization of patient histories. The second option, however, could be that the computational model uncovers chronological dependencies in the hospital which do not follow the official medical guidelines. Actually, one utility factor of this project from the perspective of the UKE was to develop a data-driven solution to uncover whether patients have been treated according to the medical guidelines or not. Therefore, the interpretation of the relevancy of rules within the graph requires cautious assessment and can provide an interesting starting point for further research.

7.4 Relation to the Scientific Literature

In this section, the question of how this thesis relates to the rest of the scientific literature is answered. Whereas to the author's knowledge no prior study investigated the problem statement of temporal rule learning from standardized EHR with the goal of creating Synthea graphs, the methodologically closest studies will be outlined below.

Mainly the differences fall into one or multiple of the following categories. Either, (1) temporal rule learning is applied to different data than EHRs [69, 79, 80], (2) the model used is not interpretable (e.g. Deep Learning approaches) [81, 82, 83, 84], (3) the goal of the model is predictive rather than descriptive [85, 86] or (4) rules are learnt without regards to the time axis [87]. One study, in particular, was found to have a similar objective in modelling patient data using Dynamic Bayesian Networks [88]. However, this approach did not build upon a specific standardized EHR format and had the direct goal of synthetic data generation without using Synthea. Nevertheless, this study provides some support for the accuracy of modelling patient data using Dynamic Bayesian networks [88].

7.5 Conclusion

Relating back to the research objective, it can be said that it is possible to extract information from standardized EHR, transform the data into time-series format and learn temporal rules which can be interpreted from a medical professional without prior knowledge of the underlying computational model. Actually, given the right framework, different models may be suitable for certain situations. Whereas the TARM is computationally very efficient on large data and still reasonably interpretable, it is a suitable choice for large datasets (≥ 10000). For very small datasets (< 50), the baseline transition matrix can be useful to provide an extensive overview. However, the only model which displayed robust performance for varying data set sizes was the `DYNOTEARS` in Pipeline 3. In line with what the authors propose [69], `DYNOTEARS` also displayed computational efficacy for the complex task of Dynamic Bayesian Network structural learning on large data ($N = 11641$).

Concluding, the baseline model in Pipeline 1 should be viewed as such, a model which holds information about all connections summarized in the transition matrix. This information is valuable and can be used to gather information on specific dependencies of interest directly from the transition matrix. Visualizing the transition matrix, however, only makes sense for very

small datasets. On the other hand, temporal association rule mining is only suitable for large datasets ($\geq$ 10000), as the risk for inadequate hyperparameter tuning and distorted results on small data is too high. Finally, while 'meaningfulness' of learnt rules is not quantifiable without a ground truth, this study provided some evidence that the use of a Structural Vector Autoregressive Model (SVARM) for Dynamic Bayesian Network structure learning can lead to a set of cause-effect relationships with less statistical bias compared to basic probabilistic models. In combination with `DYNOTEARS`'s real-life applicability due to computational efficacy, the approach described in Pipeline 3 should be preferred in further research.

7.5.1 Future Work

To finalize the scientific work, a critical perspective is taken on this research project. Because no prior study in the field is addressing the same research objective, the project should be seen as explorative. Hence, the results of this study should be viewed as basis for future research and used to optimize temporal rule learning from standardized EHR with the purpose of visualizing disease progressions. That being said, the following sections summarise the opportunities for improvement identified by the researcher during the course of this project.

Firstly, it needs to be stressed that the mapping of standardized vocabularies in all analytical pipelines is error prone regardless of the computational model used. The successful mapping from clinical observations to standardized vocabularies and back to the correct English free text greatly determines the quality of the resulting graph. Basically, if the wrong concept is mapped into the OMOP database or the wrong concept is mapped out after learning the rules, the best computational model will not learn a meaningful graph structure. In this project, the mapping has been optimized in several feedback loops with physicians and required some manual quality assurance. Therefore, the mapping of standardized vocabularies in the proposed analytical pipelines should not be fully automated in further research, but rather be subject to deliberate processing.

The second point of improvement poses a trade-off for future research objectives. Either, one focuses on visualization of disease progression and chooses another tool than Synthea for the repeatedly highlighted reasons. If, however, the goal remains to optimize graph visualization within the framework of Synthea, the following improvements represent the next steps which should be undertaken. (1) According to the expert's feedback, it would be helpful to distinguish the treatments and procedures as such within the Synthea graph. The most convenient solution would be to algorithmically determine different Synthea node types depending on the OMOP domain the data was extracted from. (2) The analytical pipeline, disregarding the model used, could be extended to give the user the opportunity to select desired OMOP domains for temporal rule learning. For example, the user could select to learn and visualize the temporal dependencies between the medication and condition domain. Basically, an extension which allows for more combinatorial variability between different OMOP domains than just conditions and treatments is not too far away methodologically and holds great practical benefit for end users.

Apart from these two major points, a few smaller suggestions for future research building up on this study can be made. For example, the three models have been identified to each posses strengths in different scenarios and therefore they should only be used in these. It would also be interesting to extend the baseline transition matrix to second, third or fourth order to

evaluate temporal dependencies with a higher autoregressive order. The TARM model pipeline suffered from rigid post-filtering of rules, which can be overcome by allowing for more flexible nodes (e.g. nodes which hold information on combinations of observations). The DBN model would greatly benefit from a visualization on a temporal axis, highlighting clear interactions along time steps. Finally, although the groundwork for this has been demonstrated in this study, it is crucial to leverage the full complexity of Synthea graphs (e.g. node and edge types) with the explicit goal of synthetic data generation in subsequent studies.

Bibliography

[1] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A Review and Comparative Study on Probabilistic Object Detection in Autonomous Driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 9961–9980, Aug. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9525313/

[2] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, "Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming," 2019. [Online]. Available: https://arxiv.org/abs/1907.07484

[3] L. Cao, "AI in Finance: Challenges, Techniques, and Opportunities," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1–38, Mar. 2023. [Online]. Available: https://dl.acm.org/doi/10.1145/3502289

[4] B. G. Buchanan, "Artificial intelligence in finance," Zenodo, Tech. Rep., Mar. 2019. [Online]. Available: https://zenodo.org/record/2626454

[5] C. Milana and A. Ashta, "Artificial intelligence techniques in finance and financial markets: A survey of the literature," *Strategic Change*, vol. 30, no. 3, pp. 189–209, May 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/jsc.2403

[6] B. Vlačić, L. Corbo, S. Costa E Silva, and M. Dabić, "The evolving role of artificial intelligence in marketing: A review and research agenda," *Journal of Business Research*, vol. 128, pp. 187–203, May 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0148296321000643

[7] S. Verma, R. Sharma, S. Deb, and D. Maitra, "Artificial intelligence in marketing: Systematic review and future research direction," *International Journal of Information Management Data Insights*, vol. 1, no. 1, p. 100002, Apr. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2667096820300021

[8] J. Sterne, *Artificial intelligence for marketing: practical applications*. Hoboken, New Jersey: Wiley, 2017.

[9] S. Graumann, I. Bertschek, T. Weber, M. Ebert, and J. Ohnemus, "Monitoring-Report Wirtschaft DIGITAL 2017-Kompakt," ZEW-Gutachten und Forschungsberichte, Tech. Rep., 2017.

[10] S. Ellahham, N. Ellahham, and M. C. E. Simsekler, "Application of Artificial Intelligence in the Health Care Safety Context: Opportunities and Challenges," *American Journal*

of Medical Quality, vol. 35, no. 4, pp. 341–348, Jul. 2020. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1062860619878515

[11] K. Abouelmehdi, A. Beni-Hssane, H. Khaloufi, and M. Saadi, "Big data security and privacy in healthcare: A Review," *Procedia Computer Science*, vol. 113, pp. 73–80, 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050917317015

[12] R. C. Barrows and P. D. Clayton, "Privacy, Confidentiality, and Electronic Medical Records," *Journal of the American Medical Informatics Association*, vol. 3, no. 2, pp. 139–148, Mar. 1996. [Online]. Available: https://academic.oup.com/jamia/article-lookup/doi/10.1136/jamia.1996.96236282

[13] Peter the Great Saint Petersburg Polytechnic University, O. Y. Iliashenko, E. L. Lukyanchenko, and Peter the Great Saint Petersburg Polytechnic University, "Possibilities of using computer vision for data analytics in medicine," *Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, vol. 22, no. 2, pp. 224–232, May 2022. [Online]. Available: https://mmi.sgu.ru/ru/articles/vozmozhnosti-primeneniya-kompyuternogo-zreniya-dlya-analitiki-dannyh-v-medicine

[14] A. Esteva, K. Chou, S. Yeung, N. Naik, A. Madani, A. Mottaghi, Y. Liu, E. Topol, J. Dean, and R. Socher, "Deep learning-enabled medical computer vision," *npj Digital Medicine*, vol. 4, no. 1, p. 5, Jan. 2021. [Online]. Available: https://www.nature.com/articles/s41746-020-00376-2

[15] J. Olveres, G. González, F. Torres, J. C. Moreno-Tagle, E. Carbajal-Degante, A. Valencia-Rodríguez, N. Méndez-Sánchez, and B. Escalante-Ramírez, "What is new in computer vision and artificial intelligence in medical image analysis applications," *Quantitative Imaging in Medicine and Surgery*, vol. 11, no. 8, pp. 3830–3853, Aug. 2021. [Online]. Available: https://qims.amegroups.com/article/view/70834/html

[16] N. Ahmadi, Y. Peng, M. Wolfien, M. Zoch, and M. Sedlmayr, "OMOP CDM Can Facilitate Data-Driven Studies for Cancer Prediction: A Systematic Review," *International Journal of Molecular Sciences*, vol. 23, no. 19, p. 11834, Oct. 2022. [Online]. Available: https://www.mdpi.com/1422-0067/23/19/11834

[17] European Parliament and Council of the European Union, "Regulation EU 2016/679 of the European Parliament and of the Council," May 2016. [Online]. Available: https://data.europa.eu/eli/reg/2016/679/oj

[18] G. Bologna, "A Rule Extraction Technique Applied to Ensembles of Neural Networks, Random Forests, and Gradient-Boosted Trees," *Algorithms*, vol. 14, no. 12, p. 339, 2021, publisher: MDPI.

[19] "FHIR official Webpage." [Online]. Available: https://hl7.org/fhir/index.html

[20] M. Ayaz, M. F. Pasha, M. Y. Alzahrani, R. Budiarto, and D. Stiawan, "The Fast Health Interoperability Resources (FHIR) standard: systematic literature review of implementations, applications, challenges and opportunities," *JMIR medical informatics*, vol. 9, no. 7, p. e21929, 2021, publisher: JMIR Publications Toronto, Canada.

[21] S. Maxhelaku and A. Kika, "Improving interoperability in healthcare using Hl7 Fhir," in *Proceedings of the 47th International Academic Conference*, 2019.

[22] "Tech industry looks to improve healthcare through cloud technology. Information Technology Industry Council (ITI)." [Online]. Available: https://www.itic.org/news-events/news-releases/tech-industry-looks-to-improve-healthcare-through-cloud-technology

[23] "Data Standardization – OHDSI." [Online]. Available: https://www.ohdsi.org/data-standardization/

[24] "Athena." [Online]. Available: https://athena.ohdsi.org/search-terms/start

[25] G. Hripcsak, J. D. Duke, N. H. Shah, C. G. Reich, V. Huser, M. J. Schuemie, M. A. Suchard, R. W. Park, I. C. K. Wong, P. R. Rijnbeek, J. van der Lei, N. Pratt, Nor&#233, G. N. N, Y.-C. Li, P. E. Stang, D. Madigan, and P. B. Ryan, "Observational Health Data Sciences and Informatics (OHDSI): Opportunities for Observational Researchers," *MEDINFO 2015: eHealth-enabled Health*, pp. 574–578, 2015. [Online]. Available: https://ebooks.iospress.nl/doi/10.3233/978-1-61499-564-7-574

[26] M. Garza, G. Del Fiol, J. Tenenbaum, A. Walden, and M. N. Zozus, "Evaluating common data models for use with a longitudinal community registry," *Journal of Biomedical Informatics*, vol. 64, pp. 333–341, Dec. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1532046416301538

[27] "Welcome to OMOPonFHIR Github Repository," Mar. 2023, original-date: 2021-09-23T00:41:56Z. [Online]. Available: https://github.com/omoponfhir/omoponfhir-main-r4-sql

[28] J. Walonoski, M. Kramer, J. Nichols, A. Quina, C. Moesel, D. Hall, C. Duffett, K. Dube, T. Gallagher, and S. McLachlan, "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record," *Journal of the American Medical Informatics Association*, vol. 25, no. 3, pp. 230–238, 2018, publisher: Oxford University Press.

[29] M.-H. Tao, "Epidemiology of lung cancer," in *Lung Cancer and Imaging*, A. El-Baz and J. S. Suri, Eds. IOP Publishing, Dec. 2019, pp. 4–1–4–15. [Online]. Available: https://iopscience.iop.org/book/edit/978-0-7503-2540-0/chapter/bk978-0-7503-2540-0ch4

[30] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: A Cancer Journal for Clinicians*, vol. 68, no. 6, pp. 394–424, Nov. 2018. [Online]. Available: http://doi.wiley.com/10.3322/caac.21492

[31] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries," *CA: A Cancer Journal for Clinicians*, vol. 71, no. 3, pp. 209–249, May 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.3322/caac.21660

[32] K. C. Thandra, A. Barsouk, K. Saginala, J. S. Aluru, and A. Barsouk, "Epidemiology of lung cancer," *Contemporary Oncology/Współczesna Onkologia*, vol. 25, no. 1, pp. 45–52, 2021. [Online]. Available: https://www.termedia.pl/Epidemiology-of-lung-cancer,3,43345,0,1.html

[33] ESMO, "NSCLC Guidlines." [Online]. Available: http://interactiveguidelines.esmo.org/esmo-web-app/gl_toc/index.php?GL_id=46

[34] T. A. Kumbhare and S. V. Chobe, "An overview of association rule mining algorithms," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 927–930, 2014, publisher: Citeseer.

[35] B. Liu, W. Hsu, Y. Ma, and others, "Integrating classification and association rule mining." in *Kdd*, vol. 98, 1998, pp. 80–86.

[36] G. Ho, W. Ip, C. Wu, and Y. Tse, "Using a fuzzy association rule mining approach to identify the financial data association," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9054–9063, Aug. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417412002916

[37] Y. Zhang, Wuhan-Gongcheng-Daxue, Hua zhong ke ji da xue, Institute of Electrical and Electronics Engineers, and IEEE Industrial Electronics Society, Eds., *2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications: PACIIA 2009 ; Wuhan, China, 28 - 29 November 2009.* Piscataway, NJ: IEEE, 2009.

[38] T. Li and X. Li, "Novel alarm correlation analysis system based on association rules mining in telecommunication networks," *Information Sciences*, vol. 180, no. 16, pp. 2960–2978, Aug. 2010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025510001696

[39] C.-F. Tsai and M.-Y. Chen, "Variable selection by association rules for customer churn prediction of multimedia on demand," *Expert Systems with Applications*, vol. 37, no. 3, pp. 2006–2015, Mar. 2010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417409006459

[40] A. N. Paidi, "Data mining: Future trends and applications," *International Journal of Modern Engineering Research*, vol. 2, no. 6, pp. 4657–4663, 2012, publisher: Citeseer.

[41] K. Vougas, T. Sakellaropoulos, A. Kotsinas, G.-R. P. Foukas, A. Ntargaras, F. Koinis, A. Polyzos, V. Myrianthopoulos, H. Zhou, S. Narang, V. Georgoulias, L. Alexopoulos, I. Aifantis, P. A. Townsend, P. Sfikakis, R. Fitzgerald, D. Thanos, J. Bartek, R. Petty, A. Tsirigos, and V. G. Gorgoulis, "Machine learning and data mining frameworks for predicting drug response in cancer: An overview and a novel in silico screening process based on association rule mining," *Pharmacology & Therapeutics*, vol. 203, p. 107395, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016372581930138X

[42] J. Nahar, T. Imam, K. S. Tickle, and Y.-P. P. Chen, "Association rule mining to detect factors which contribute to heart disease in males and females," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1086–1093, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741741200989X

[43] A. Segura-Delgado, M. J. Gacto, R. Alcalá, and J. Alcalá-Fdez, "Temporal association rule mining: An overview considering the time variable as an integral or implied component," *WIREs Data Mining and Knowledge Discovery*, vol. 10, no. 4, Jul. 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/widm.1367

[44] S. Concaro, L. Sacchi, C. Cerra, P. Fratino, and R. Bellazzi, "Mining Healthcare Data with Temporal Association Rules: Improvements and Assessment for a Practical Use," in *Artificial Intelligence in Medicine*, C. Combi, Y. Shahar, and A. Abu-Hanna, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, vol. 5651, pp. 16–25. [Online]. Available: http://link.springer.com/10.1007/978-3-642-02976-9_3

[45] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93*. Washington, D.C., United States: ACM Press, 1993, pp. 207–216. [Online]. Available: http://portal.acm.org/citation.cfm?doid=170035.170072

[46] S. Gupta and R. Mamtora, "A survey on association rule mining in market basket analysis," *International Journal of Information and Computation Technology*, vol. 4, no. 4, pp. 409–414, 2014.

[47] P. Arora, D. Boyne, J. J. Slater, A. Gupta, D. R. Brenner, and M. J. Druzdzel, "Bayesian Networks for Risk Prediction Using Real-World Data: A Tool for Precision Medicine," *Value in Health*, vol. 22, no. 4, pp. 439–445, Apr. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1098301519300579

[48] G. Trabelsi, P. Leray, M. Ben Ayed, and A. M. Alimi, "Benchmarking dynamic Bayesian network structure learning algorithms," in *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)*. Hammamet: IEEE, Apr. 2013, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/6552549/

[49] J. W. Coid, S. Ullrich, C. Kallis, M. Freestone, R. Gonzalez, L. Bui, A. Igoumenou, A. Constantinou, N. Fenton, W. Marsh, M. Yang, B. DeStavola, J. Hu, J. Shaw, M. Doyle, L. Archer-Power, M. Davoren, B. Osumili, P. McCrone, K. Barrett, D. Hindle, and P. Bebbington, "Improving risk management for violence in mental health services: a multimethods approach," *Programme Grants for Applied Research*, vol. 4, no. 16, pp. 1–408, Nov. 2016. [Online]. Available: https://www.journalslibrary.nihr.ac.uk/pgfar/pgfar04160/

[50] G. Trabelsi, "New structure learning algorithms and evaluation methods for large dynamic Bayesian networks," phdthesis, Université de Nantes ; Ecole Nationale d'Ingénieurs de Sfax, Dec. 2013. [Online]. Available: https://theses.hal.science/tel-00996061

[51] D. M. Chickering, C. Meek, and D. Heckerman, "Large-Sample Learning of Bayesian Networks is NP-Hard," 2012. [Online]. Available: https://arxiv.org/abs/1212.2468

[52] C. P. De Campos, Z. Zeng, and Q. Ji, "Structure learning of Bayesian networks using constraints," in *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal Quebec Canada: ACM, Jun. 2009, pp. 113–120. [Online]. Available: https://dl.acm.org/doi/10.1145/1553374.1553389

[53] M. A. Van Gerven, B. G. Taal, and P. J. Lucas, "Dynamic Bayesian networks as prognostic models for clinical patient management," *Journal of Biomedical Informatics*, vol. 41, no. 4, pp. 515–529, Aug. 2008. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1532046408000154

[54] K. Orphanou, A. Stassopoulou, and E. Keravnou, "DBN-Extended: A Dynamic Bayesian Network Model Extended With Temporal Abstractions for Coronary Heart Disease Prognosis," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 3, pp. 944–952, May 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7080845/

[55] Schleswig-Holstein, "State Cancer Registry." [Online]. Available: https://www.krebsregister-sh.de

[56] American Cancer Society, "Cancer Staging." [Online]. Available: https://www.cancer.org/cancer/diagnosis-staging/staging.html

[57] OHDSI, "The Book of OHDSI." [Online]. Available: https://ohdsi.github.io/TheBookOfOhdsi/

[58] BfArM, "SNOMED CT." [Online]. Available: https://www.bfarm.de/DE/Kodiersysteme/Terminologien/SNOMED-CT/_node.html#:~:text=SNOMED%20CT%20ist%20die%20derzeit,beim%20elektronischen%20Austausch%20von%20Gesundheitsdaten.

[59] OHDSI, "Athena." [Online]. Available: https://athena.ohdsi.org/search-terms/start

[60] MITRE, "Synthea Module Builder." [Online]. Available: https://syntheticheath.github.io/module-builder/#example_module

[61] ——, "Synthea Github Modules." [Online]. Available: https://github.com/synthetichealth/synthea/blob/master/src/main/resources/modules/lung_cancer/lung_cancer_probabilities.json

[62] ——, "Synthea Wiki." [Online]. Available: https://github.com/synthetichealth/synthea/wiki/

[63] J. R. Norris, *Markov chains*, 1st ed. Cambridge, UK: Cambridge University Press, 1998, oCLC: 817914077.

[64] P. Fournier-Viger, R. Nkambou, and V. S.-M. Tseng, "RuleGrowth: mining sequential rules common to several sequences by pattern-growth," in *Proceedings of the 2011 ACM Symposium on Applied Computing*. TaiChung Taiwan: ACM, Mar. 2011, pp. 956–961. [Online]. Available: https://dl.acm.org/doi/10.1145/1982185.1982394

[65] "SPMF: A Java Open-Source Data Mining Library." [Online]. Available: http://www.philippe-fournier-viger.com/spmf/index.php?link=documentation.php

[66] P. Fournier-Viger, J. C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, "The SPMF Open-Source Data Mining Library Version 2," in *Machine Learning and Knowledge Discovery in Databases*, B. Berendt, B. Bringmann, Fromont, G. Garriga, P. Miettinen, N. Tatti, and V. Tresp, Eds. Cham: Springer International Publishing, 2016, vol. 9853, pp. 36–40. [Online]. Available: https://link.springer.com/10.1007/978-3-319-46131-1_8

[67] P. Fournier-Viger, U. Faghihi, R. Nkambou, and E. M. Nguifo, "CMRules: Mining sequential rules common to several sequences," *Knowledge-Based Systems*, vol. 25, no. 1, pp. 63–76, Feb. 2012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0950705111001456

[68] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the Eleventh International Conference on Data Engineering*. Taipei, Taiwan: IEEE Comput. Soc. Press, 1995, pp. 3–14. [Online]. Available: http://ieeexplore.ieee.org/document/380415/

[69] R. Pamfil, N. Sriwattanaworachai, N. Desai, P. Pilgerstorfer, P. Beaumont, K. Georgatzis, and B. Aragam, "DYNOTEARS: Structure Learning from Time-Series Data," Apr. 2020. [Online]. Available: https://arxiv.org/pdf/2002.00498.pdf

[70] N. R. Swanson and C. W. J. Granger, "Impulse Response Functions Based on a Causal Approach to Residual Orthogonalization in Vector Autoregressions," *Journal of the American Statistical Association*, vol. 92, no. 437, pp. 357–367, Mar. 1997. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01621459.1997.10473634

[71] S. Demiralp and K. D. Hoover, "Searching for the Causal Structure of a Vector Autoregression*," *Oxford Bulletin of Economics and Statistics*, vol. 65, no. s1, pp. 745–767, Dec. 2003. [Online]. Available: http://doi.wiley.com/10.1046/j.0305-9049.2003.00087.x

[72] L. Kilian, "Structural vector autoregressions," *CEPR Discussion Paper No. DP8515*, 2011. [Online]. Available: http://mayoral.iae-csic.org/timeseries_insead/kilian_var.pdf

[73] X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing, "DAGs with NO TEARS: Continuous Optimization for Structure Learning," 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/e347c51419ffb23ca3fd5050202f9c3d-Paper.pdf

[74] S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett, and Neural Information Processing Systems Foundation, Eds., *Advances in neural information processing systems 31: 32nd Conference on Neural Information Processing Systems (NeurIPS 2018): Montréal, Canada, 3-8 December 2018*. Red Hook, NY: Curran Associates, Inc, 2019.

[75] "CausalNex Library." [Online]. Available: https://github.com/quantumblacklabs/causalnex/tree/develop

[76] "Documentation Causalnex." [Online]. Available: https://causalnex.readthedocs.io/en/latest/04_user_guide/04_user_guide.html#

[77] J. Pearl, M. Glymour, and N. P. Jewell, *Causal inference in statistics: a primer*. Chichester, West Sussex, UK: John Wiley & Sons Ltd, 2016.

[78] *Prävention, Diagnostik, Therapie und Nachsorge des Lungenkarzinoms*, ser. Leitlinien Programm Onkologie, 2022, no. 2.1. [Online]. Available: https://register.awmf.org/assets/guidelines/020-007OLl_S3_Praevention-Diagnostik-Therapie-Nachsorge-Lungenkarzinom_2022-12.pdf

[79] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning Graphs From Data: A Signal Representation Perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8700665/

[80] I. Ebert-Uphoff and Y. Deng, "Causal Discovery for Climate Research Using Graphical Models," *Journal of Climate*, vol. 25, no. 17, pp. 5648–5665, Sep. 2012. [Online]. Available: http://journals.ametsoc.org/doi/10.1175/JCLI-D-11-00387.1

[81] S. Cao, W. Lu, and Q. Xu, "Deep Neural Networks for Learning Graph Representations," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Feb. 2016. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10179

[82] Y. Liu, Y. Zheng, D. Zhang, H. Chen, H. Peng, and S. Pan, "Towards Unsupervised Deep Graph Structure Learning," in *Proceedings of the ACM Web Conference 2022*. Virtual Event, Lyon France: ACM, Apr. 2022, pp. 1392–1403. [Online]. Available: https://dl.acm.org/doi/10.1145/3485447.3512186

[83] A. Narayan and P. H. O'N Roe, "Learning Graph Dynamics using Deep Neural Networks," *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 433–438, 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896318300788

[84] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep EHR: A Survey of Recent Advances in Deep Learning Techniques for Electronic Health Record (EHR) Analysis," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, Sep. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8086133/

[85] Y. Wang, P. Wu, Y. Liu, C. Weng, and D. Zeng, "Learning Optimal Individualized Treatment Rules from Electronic Health Record Data," in *2016 IEEE International Conference on Healthcare Informatics (ICHI)*. Chicago, IL, USA: IEEE, Oct. 2016, pp. 65–71. [Online]. Available: http://ieeexplore.ieee.org/document/7776329/

[86] L. Jamian, L. Wheless, L. J. Crofford, and A. Barnado, "Rule-based and machine learning algorithms identify patients with systemic sclerosis accurately in the electronic health record," *Arthritis Research & Therapy*, vol. 21, no. 1, p. 305, Dec. 2019. [Online]. Available: https://arthritis-research.biomedcentral.com/articles/10.1186/s13075-019-2092-7

[87] M. Julia Flores, A. E. Nicholson, A. Brunskill, K. B. Korb, and S. Mascaro, "Incorporating expert knowledge when learning Bayesian network structure: A medical case study," *Artificial Intelligence in Medicine*, vol. 53, no. 3, pp. 181–204, Nov. 2011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0933365711001084

[88] J. De Benedetti, N. Oues, Z. Wang, P. Myles, and A. Tucker, "Practical Lessons from Generating Synthetic Healthcare Data with Bayesian Networks," in *ECML PKDD 2020 Workshops*. Cham: Springer International Publishing, 2020, vol. 1323, pp. 38–47. [Online]. Available: https://link.springer.com/10.1007/978-3-030-65965-3_3

Appendices

A Synthea Modules

Figure 1 and Figure 2 display the two official Synthea modules on Lung Cancer that are available online [61]. The goal of this study is to generate such a Synthea Graph in a data-driven fashion.

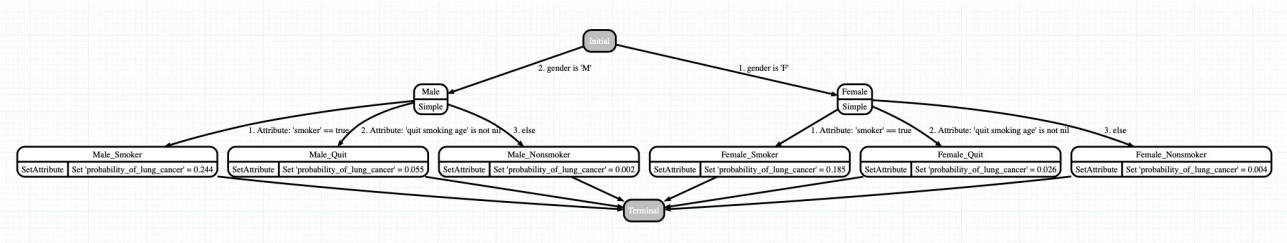


Figure 1: Synthea module for statistics on pulmonary cancer.

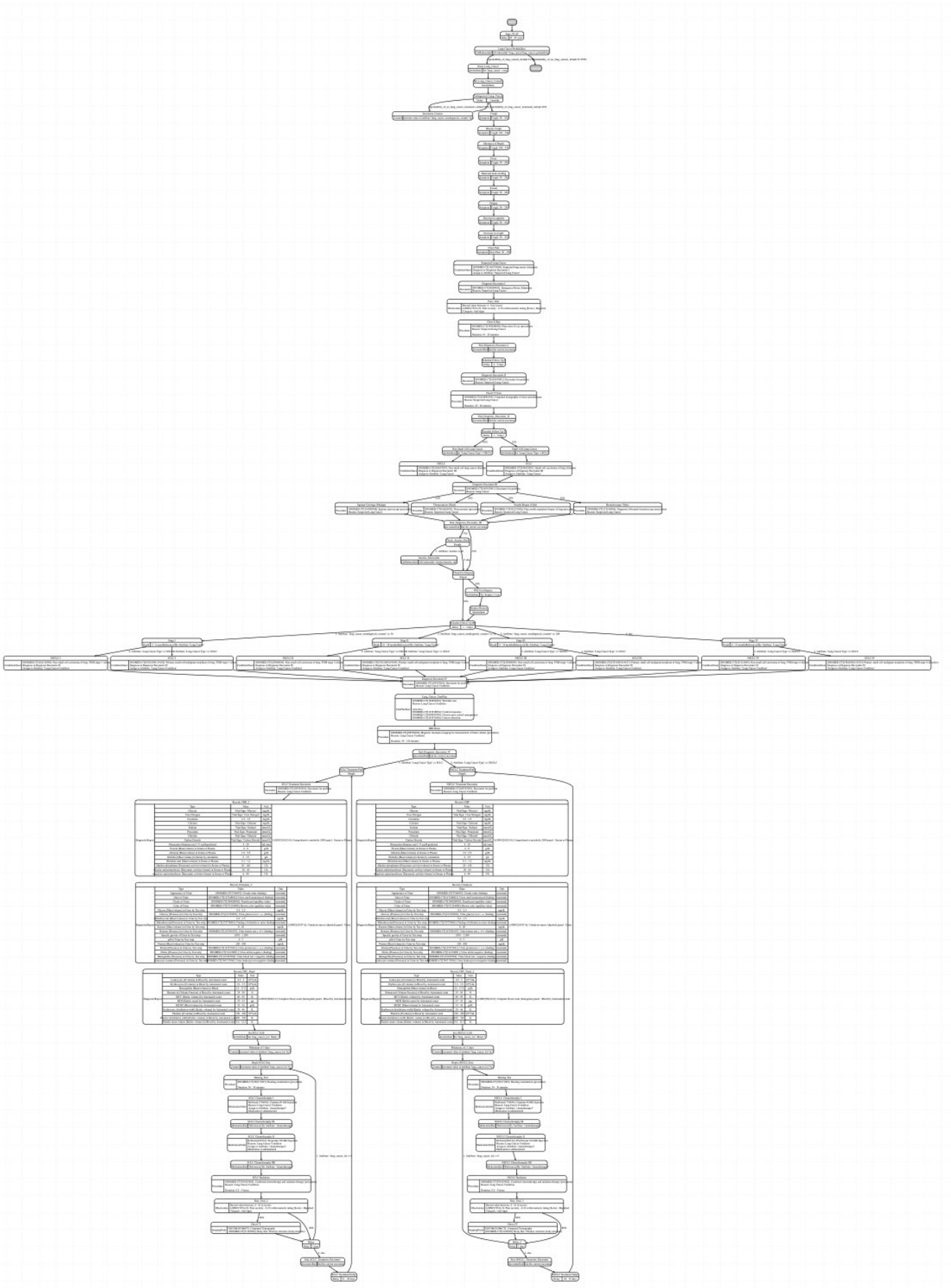


Figure 2: Synthea module for pulmonary cancer diagnosis and treatment.

B Questionnaire Graphs

In this section, 3 out of the 45 generated graphs are shown. These three graphs were included in the questionnaire that was handed to physicians during the study to investigate the graphs qualitatively from medical experts without domain knowledge of Artificial Intelligence. Figure 3 was generated with Pipeline 1, Figure 4 was generated with Pipeline 2 and Figure 5 was generated with Pipeline 3.

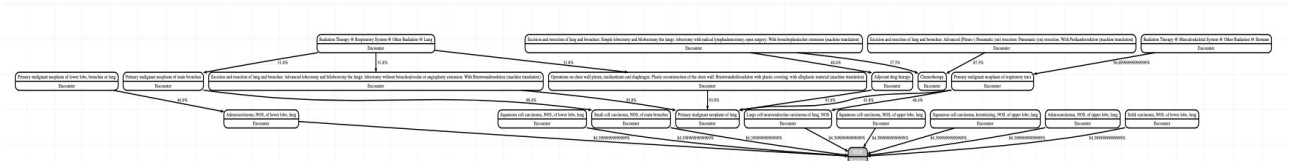


Figure 3: Graph from Pipeline 1 included in the Expert Questionnaire.

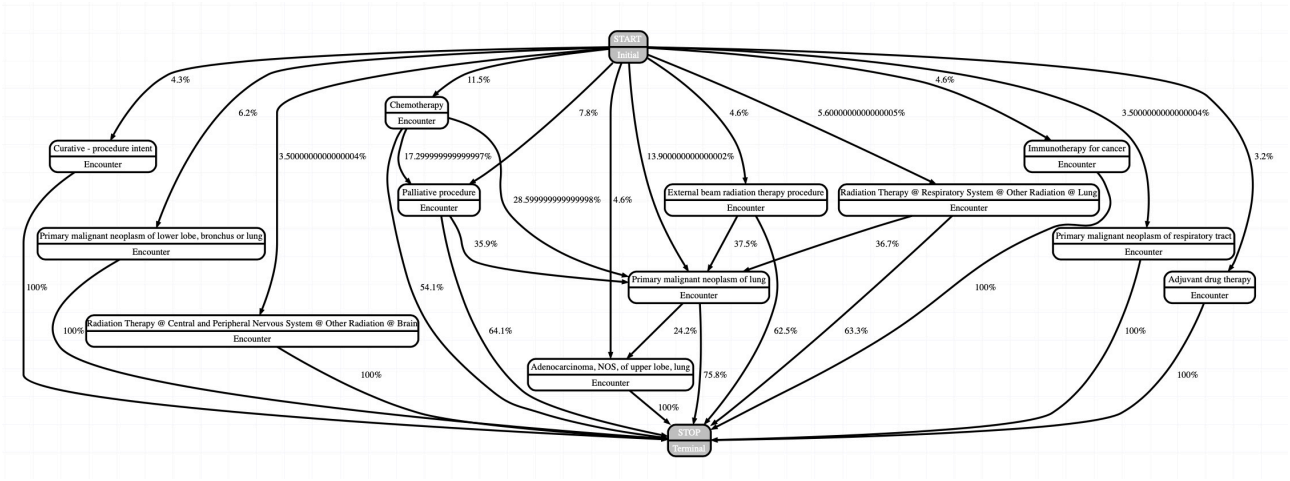


Figure 4: Graph from Pipeline 2 included in the Expert Questionnaire.

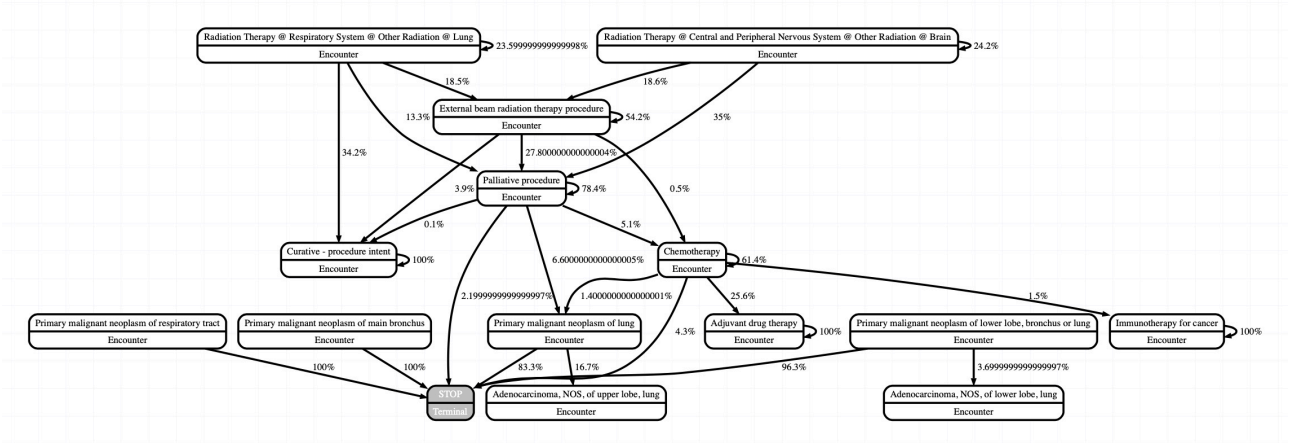


Figure 5: Graph from Pipeline 3 included in the Expert Questionnaire.

C Graph Intersection within Pipeline

Table 1, Table 2 and Table 3 provide an exhaustive overview of the graph intersections within models. More precisely, it shows the percentage of overlapping direct edges that the resulting graphs within each pipeline have across hyperparameters and datasets.

Table 1: Intersection of edges between all graphs within Pipeline 1 (Baseline). The row and column names follow the syntax 'dataset_hyperparamter'.

	10_1	50_1	500_1	5000_1	full_1	10_2	50_2	500_2	5000_2	full_2	10_3	50_3	500_3	5000_3	full_3
10_1	1.00	0.39	0.65	0.89	0.93	0.43	0.12	0.16	0.21	0.21	0.29	0.08	0.12	0.12	0.12
50_1	0.14	1.00	0.69	0.88	0.96	0.07	0.46	0.20	0.25	0.29	0.04	0.30	0.14	0.17	0.21
500_1	0.07	0.19	1.00	0.82	0.96	0.03	0.08	0.37	0.21	0.23	0.02	0.05	0.24	0.15	0.15
5000_1	0.03	0.08	0.26	1.00	0.96	0.01	0.04	0.09	0.33	0.28	0.01	0.02	0.07	0.21	0.18
full_1	0.02	0.06	0.20	0.61	1.00	0.01	0.03	0.07	0.21	0.30	0.01	0.02	0.05	0.13	0.18
10_2	1.00	0.44	0.72	0.97	1.00	1.00	0.28	0.38	0.50	0.50	0.69	0.19	0.28	0.28	0.28
50_2	0.10	1.00	0.62	0.85	0.96	0.10	1.00	0.43	0.53	0.62	0.08	0.66	0.32	0.36	0.46
500_2	0.05	0.16	1.00	0.78	0.98	0.05	0.15	1.00	0.55	0.60	0.04	0.13	0.66	0.40	0.40
5000_2	0.02	0.07	0.21	1.00	0.99	0.02	0.07	0.20	1.00	0.82	0.02	0.06	0.16	0.63	0.54
full_2	0.02	0.06	0.16	0.59	1.00	0.02	0.06	0.15	0.57	1.00	0.01	0.05	0.12	0.42	0.62
10_3	1.00	0.41	0.64	0.95	1.00	1.00	0.32	0.45	0.68	0.64	1.00	0.27	0.41	0.41	0.41
50_3	0.10	1.00	0.59	0.80	0.97	0.10	1.00	0.56	0.67	0.80	0.10	1.00	0.43	0.54	0.69
500_3	0.05	0.17	1.00	0.85	0.98	0.05	0.17	1.00	0.67	0.71	0.05	0.15	1.00	0.59	0.60
5000_3	0.02	0.07	0.23	1.00	1.00	0.02	0.07	0.23	1.00	0.94	0.02	0.07	0.22	1.00	0.83
full_3	0.01	0.07	0.17	0.61	1.00	0.01	0.07	0.17	0.61	1.00	0.01	0.07	0.16	0.60	1.00

Table 2: Intersection of edges between all graphs within Pipeline 2 (TARM). The row and column names follow the syntax 'dataset_hyperparamter'.

	10_1	50_1	500_1	5000_1	full_1	10_2	50_2	500_2	5000_2	full_2	10_3	50_3	500_3	5000_3	full_3
10_1	1.00	0.36	0.42	0.35	0.34	1.00	1.00	0.27	0.20	0.20	0.18	0.09	0.12	0.07	0.07
50_1	0.68	1.0	0.84	0.70	0.67	0.68	0.68	0.49	0.37	0.37	0.31	0.17	0.23	0.14	0.14
500_1	0.73	0.76	1.00	0.70	0.67	0.73	0.73	0.46	0.34	0.34	0.30	0.16	0.21	0.12	0.12
5000_1	0.85	0.92	1.00	1.00	0.95	0.85	0.85	0.66	0.48	0.48	0.44	0.23	0.31	0.18	0.18
full_1	0.87	0.90	1.00	0.98	1.00	0.87	0.87	0.68	0.50	0.50	0.45	0.23	0.32	0.18	0.18
10_2	1.00	0.36	0.42	0.35	0.34	1.00	1.00	0.27	0.20	0.20	0.18	0.09	0.12	0.07	0.07
50_2	1.00	0.36	0.42	0.35	0.34	1.00	1.00	0.27	0.20	0.20	0.18	0.09	0.12	0.07	0.07
500_2	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	0.73	0.73	0.63	0.34	0.46	0.27	0.27
5000_2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.67	0.47	0.63	0.37	0.37
full_2	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00	0.70	0.47	0.63	0.37	0.37
10_3	1.00	0.93	1.00	1.00	1.00	1.00	1.00	0.96	0.74	0.78	1.00	0.48	0.67	0.41	0.41
50_3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.93	1.00	1.00	0.64	0.64
500_3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.95	0.74	1.00	0.58	0.58
5000_3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.82	1.00	1.00	1.00
full_3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.82	1.00	1.00	1.00

Table 3: Intersection of edges between all graphs within Pipeline 3 (DBN). The row and column names follow the syntax 'dataset_hyperparamter'.

	10_1	50_1	500_1	5000_1	full_1	10_2	50_2	500_2	5000_2	full_2	10_3	50_3	500_3	5000_3	full_3
10_1	1.00	0.41	0.45	0.44	0.44	0.88	0.33	0.32	0.29	0.30	0.30	0.12	0.13	0.16	0.16
50_1	0.51	1.00	0.67	0.63	0.66	0.46	0.58	0.48	0.40	0.42	0.21	0.24	0.21	0.25	0.24
500_1	0.67	0.82	1.00	0.91	0.93	0.60	0.64	0.62	0.53	0.55	0.29	0.27	0.25	0.31	0.29
5000_1	0.68	0.79	0.94	1.00	1.00	0.62	0.62	0.64	0.55	0.57	0.30	0.28	0.26	0.32	0.30
full_1	0.65	0.80	0.93	0.96	1.00	0.60	0.62	0.62	0.53	0.55	0.29	0.27	0.25	0.31	0.29
10_2	0.97	0.42	0.45	0.45	0.45	1.00	0.34	0.34	0.31	0.32	0.34	0.14	0.14	0.16	0.16
50_2	0.69	1.00	0.90	0.85	0.87	0.64	1.00	0.69	0.62	0.62	0.31	0.41	0.33	0.41	0.38
500_2	0.76	0.94	1.00	1.00	1.00	0.74	0.79	1.00	0.85	0.88	0.44	0.41	0.41	0.50	0.47
5000_2	0.83	0.93	1.00	1.00	1.00	0.79	0.83	1.00	1.00	0.97	0.48	0.48	0.48	0.59	0.55
full_2	0.83	0.93	1.00	1.00	1.00	0.80	0.80	1.00	0.93	1.00	0.47	0.43	0.47	0.57	0.53
10_3	1.00	0.56	0.64	0.64	0.64	1.00	0.48	0.60	0.56	0.56	1.00	0.28	0.32	0.32	0.32
50_3	0.62	1.00	0.94	0.94	0.94	0.62	1.00	0.88	0.88	0.81	0.44	1.00	0.62	0.69	0.69
500_3	0.79	1.00	1.00	1.00	1.00	0.71	0.93	1.00	1.00	1.00	0.57	0.71	1.0	0.93	1.00
5000_3	0.76	1.00	1.00	1.00	1.00	0.71	0.94	1.00	1.00	1.00	0.47	0.65	0.76	1.00	0.88
full_3	0.81	1.00	1.00	1.00	1.00	0.75	0.94	1.00	1.00	1.00	0.50	0.69	0.88	0.94	1.00