An Application of Bayesian Network in Cognitive Behavioral Therapy

Benjamin Andersen

Supervisor: Fazle Rabbi
Co-supervisor: Bjørnar Tessem

University of Bergen
Faculty of Social Sciences
Department of Information Science and Media Studies
Master Thesis
30.11.2021

"I've written some great things. That's a gift, but there's consequences. Yeah, you get this great work, but you suffer. You really, really suffer."

- Frank Ocean

Abstract

Mental health has received increased focus in recent years, with a larger emphasis on treatment and acceptance. However, evidence-based psychological interventions are of poor availability and have room for improvement. The amount of data being gathered across applications and practices provide opportunities for deeper analysis through machine learning based technologies. By applying Bayesian networks (BNs) in a cognitive behavioral therapy for adults with ADHD, this research analyzes historic self-report data to predict the behavior of future participants at an early stage of the online intervention. Bayesian networks represent probabilistic models that describe the joint probability distribution through an acyclic graph. The contribution of this thesis is an artifact with the purpose of serving as a decision making support tool. Methods of Design Science Research was applied to achieve this, in a development cycle with three main iterations.

Using Bayesian networks for analyzing behavioral patterns yield positive results with its predictive capabilities when dealing with uncertainty. Domain experts from the internet-delivered intervention provided useful feedback and insight that contributed to the novelty and research scope of this thesis. Future work should update the model when a larger population sample is available, and focus on implementing the artifact in a more user-centered desktop application.

Acknowledgements

First and foremost, I would like to thank my supervisor, Fazle Rabbi, for supporting and encouraging my research. I would also like to thank my co-supervisor, Bjørnar Tessem, for your shared knowledge and insight.

I would like to thank Dr. Robin Kenter for patiently answering my questions regarding the online intervention, as well as the rest of the domain experts that contributed to this research.

A special thank you to Daniel Ostnes for our online study sessions during the whole thesis period, only disrupted by our Pokémon runs, and Sunniva Blom Stolt-Nielsen for proofreading this thesis and constantly having my back. You both somehow kept me motivated and sane throughout this thesis, and your friendships are highly valued.

I would also like to thank my roommate Jostein Bakke Helle for ensuring that I actually ate something while sitting in my office and slamming on my keyboard. I owe you some cooking.

Finally, a big thank you to my friends and family for always supporting me, motivating me, and for believing in me even when I do not.

Benjamin Andersen

Contents

List of Tables

List of Figures

1 Introduction

Intervention delivered over the internet is promising, however, the availability of evidence-based psychological interventions is limited (*An Internet-delivered Intervention for Coping With ADHD in Adulthood (MyADHD)*, n.d.). This thesis focus on an internet delivered intervention for adults with attention deficit hyperactivity disorder (ADHD) that builds on principles of cognitive behavioral therapy. ADHD is a neurodevelopmental disorder that can be characterized by symptoms of inattention and/or hyperactivity that are persistent throughout the affected person's daily functioning, with an estimated prevalence of 2-3% in adulthood. Methods including psychoeducation to increase the understanding of the disorder involve cognitive approaches to restructure the maladaptive beliefs and dysfunctional thoughts that reinforce emotional maladjustment. This area still has room for improvement, as studies show that lack of sustained adherence propose a challenge in self-guided internet interventions (*An Internet-delivered Intervention for Coping With ADHD in Adulthood (MyADHD)*, n.d.). Bayesian networks provide a means for analyzing patterns to uncover new properties that previously has been unknown to the human eye (Friedman, Linial, Nachman, & Pe'er, 2000). By handling uncertainty through accurate predictions, Bayesian networks can be updated as new evidence come to light to make decision making more information based (BayesFusion, 2020). Independence assumptions is a powerful tool in Bayesian networks, especially when handling large amount of numbers. What may seem impossible due to an explosion of values become manageable as the required amount of numbers drop drastically, which can turn the ocean of variables from problems to opportunities and resilience. An interesting fact contemplate when using Bayesian network, is that for exact algorithms, the feature that condition performance is topology (Charniak, 1991). The working principle of a Bayesian network is easily explainable as it relies on dependencies and conditional independencies. According to new regulations from the *General Data Protection Regulation* (GDPR), it is required that decision support systems used in the healthcare sector to be explainable (Goodman & Flaxman, 2017). This proposes an advantage to Bayesian networks over more complex methods in this sector, as machine learning based predictions models is expected to play a major role in aiding the decision making done by heatlcare experts (Marcos, Juarez, Lenz, Nalepa, & Nowaczyk, 2020).

1.1 Research Questions

The following research questions helped establish the scope of this thesis:

- **RQ1:** What are the strengths and limitations of a Bayesian Network?
- **RQ2:** How can Bayesian networks be utilized as a decision making tool in cognitive behavioral therapy?
- **RQ3:** How can Bayesian network theory be applied for predicting the dropout of internet

based cognitive behavior treatment program, and how can we measure the accuracy of such applications?

1.2 Contribution

This thesis explore applying Bayesian networks in cognitive behavioral therapy to predict participant behavior. The main contribution is the artifact that the Bayesian networks presents. The artifact aims to help solve the problem of participants dropping out of the treatment program by improving today's practice with a supplementary decision-making tool. A literature review covering the principles and disciplines of Bayesian networks is also among the contributions of this thesis.

1.3 Thesis Outline

The following is an outline of the thesis:

Chapter 2: Background presents background on the Indernet-delivered intervention for adults with ADHD and a literature review of Bayesian network principles and disciplines.

Chapter 3: Methodology and Methods provides a description of the methodology and methods that were used.

Chapter 4: The Dataset describes the dataset, dataset properties, and data processing.

Chapter 5: Network Development presents the development stages through the three iterations.

Chapter 6: Results presents the artifact results.

Chapter 7: Discussion reviews the research approach, results, limitations, and answers the research questions.

Chapter 8: Conclusion and Future Work presents a conclusion to the research with a summary and recommendations for future work.

2 Background

This chapter presents background information and theoretical topics related to this research. An overview of the internet-delivered intervention for adults with ADHD is first described, before different data mining methods are presented. The rest of the chapter covers a thorough literature review on Bayesian networks to explain the most important principles and disciplines of this genre of performing data computation and analysis.

2.1 Internet-delivered intervention for adults with ADHD

Attention deficit hyperactivity disorder (ADHD) is a common neurodevelopmental disorder that have an estimated prevalence of 2-3%. Inattention and/or hyperactivity are the most characterized symptoms that are persistent across various situations during a person's lifespan. Associated challenges of ADHD in adulthood comes with severe consequences on the affected person's daily life functioning. Even though this is a widespread concern, evidence-based psychological interventions is of poor availability (*An Internet-delivered Intervention for Coping With ADHD in Adulthood (MyADHD)*, n.d.).

2.1.1 Program Content

The main goals of the online treatment program are to help participants achieve better functioning in daily life, reduce inattention, offer strategies that will lead to stress reduction, and improve quality of life. It builds on principles of goal management training (GMT), cognitive behavioral therapy (CBT), and dialectical behavioral treatment (DBT). The program consists of seven training modules that are accessed weekly, which was developed in a co-design effort with end-users, clinicians, health-and it-researchers by implementing the Person-based approach (Yardley, Morrison, Bradbury, & Muller, 2015). Examining the efficacy of a self-guided internet-delivered intervention for coping with ADHD by conducting a randomized controlled trial, enables assessment of the effects on various symptoms, including inattention and quality of life, through a post-treatment phase and a 3 months follow-up. Depression, stress, and anxiety are classified as secondary outcomes and are tightly related symptoms of people struggling with ADHD. The study also investigates the effects of individual adaptation on adherence and outcome measures as a result of the intervention (*An Internet-delivered Intervention for Coping With ADHD in Adulthood (MyADHD)*, n.d.).

Cognitive behavioral therapy for adults with ADHD focus on behavioral interventions that target the practice of compensatory skills and cognitive intervention targeting negative thoughts, avoidance, and procrastination. Strategies for dealing with these problems include organization, prioritization, problem solving, and stress management. In addition to this, DBT strategies include impulse control, self-regulation, self-esteem, self-respect, and emotional regulation (*MyADHD - Digital Training for Adults With ADHD*, n.d.). In order to tackle the clinical

outcomes, each participant have to answer the following self-report scales, covering various psychological deficits that are either directly or indirectly associated with ADHD:

- **The Adult ADHD Self-Rating Scale (ASRS)**: A questionnaire that includes all of the 18 symptoms of ADHD, split into two subscales regarding problems with inattention and hyperactivity.
- **Adult ADHD Quality of Life Measure (AAQoL)**: Used to assess health related quality of life among adults with ADHD.
- **The Perceived Stress Scale (PSS)**: A measurement of a person's stress.
- **The Patient Health Questionnaire (PHQ-9)**: A questionnaire that measures the depression severity of a person.
- **General Anxiety Disorder (GAD-7)**: A questionnaire to map a person's mental health state that focus on anxiety.
- **Perceived Deficits Questionnaire (PDQ-5)**: Used to assess subjective cognitive dysfunction in people with depression.
- **The Self-Compassion Scale (SCS)**: Examines different components of self-compassion, such as emotions, thoughts, and behavior.

2.1.2 Study Design

The flowchart in Figure 1 displays how the study is conducted. By first recruiting to the online open access module, an anonymous online survey is used for inclusion criteria where participants meeting the inclusion criteria will book a time slot for a screening over phone. Those who does not meet the survey's inclusion criteria will not be eligible to participate in the study. When the phone screening is completed, participants that meet the inclusion criteria in this stage will gain access to the training program, while those who do not will have their access to the intervention declined. When accepted into the program, step 3 is to sign a digital informed consent form, before starting a Pre-intervention and outcome measure. Module 1-8 begins after the Pre Mapping phase is completed, where participants receive daily homework assignments every week. Step 6 in the program is a self-report post-measurements in a secure online platform, marking the end of training. The last step is then self-report follow-up measurements in a secure online platform, which will be issued 3 months later and is the end of the study (*An Internet-delivered Intervention for Coping With ADHD in Adulthood (MyADHD)*, n.d.).

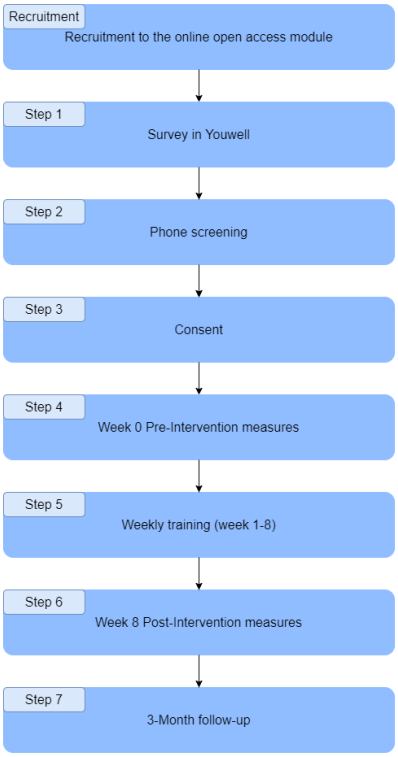

Figure 1: Study Flowchart

The following inclusion / exclusion criteria are used:

- **Criteria for inclusion**:
  - Adults with a self-reported diagnosis of ADHD (date, venue, and diagnosis physician).
  - Access to and ability to use a computer, smartphone, and the Internet.
  - Current problems with organizing daily activity and 17 points of more on at least one of the ASRS subscales.
  - Participant are by investigators considered able to follow through the training protocol and take part in measures taken during the study time frame.
  - Speaks, writes, and reads Norwegian.
- **Exclusion criteria are**:
  - Current self-reported diagnosis of severe psychiatric illness such as borderline or antisocial personality disorder, bipolar disorder, ongoing substance abuse, and / or suicidal ideation assessed with item 9 on the MADRS.
  - Participants who are taking prescribed ADHD medication have to be stable on the medication at least four weeks before the study and during the study.

2.2 Data Mining

Data mining can be defined as an analysis of data sets with the goal of finding suspected relationships, and to summarize the data in an understandable and useful way (Hand, Mannila, & Smyth, 2001). Initially, data mining was not well received due to terms like “data snooping”, “fishing”, and “data dredging”, which are techniques to extract conclusions from data without a strong statistical backing. In more recent times it has become a more solid scientific method with limitless practical applications. It is typical for data mining to take data sets in the form of tables as input, and provide clusters, graphs, equations, rules, tree structures, patterns, and more as the desired output (van der Aalst, 2011)[p. 59]. The overall aim of data mining techniques and algorithms is to understand reality based on historical data.

2.2.1 Machine Learning

Applications of Machine Learning (ML) are endless with the data available today. Many people think that ML can only be integrated by large companies with extensive research teams, but this is far from the case. Application areas range from medical diagnosis and treatment, to social network analytics, twitter sentiment analysis, etc. Machine Learning techniques are typically used to extract hidden patterns from data, but there is still a need for a data engineer to work on how the gathered information should be presented. The topic of ML is a research field at the intersection of statistics, artificial intelligence, and computer science. It is known as both statistical learning and predictive analytics. In the past, Machine Learning applications have been used with success to find planets, understand stars, analyze DNA sequences, discover new particles, and provide personalized cancer treatments (Mueller & Guido, 1997).

Supervised Learning

Detecting faces in image detection was a problem that provided a long lasting headache for researchers and developers. The root of the problem was that computers perceived pixels very differently from how humans perceive a face, so coming up with sets of rules for what constitutes a face in a digital image was challenging. With Machine Learning floating to the surface, one could simply present a program with a large data set of facial images, and the algorithm would determine the needed characteristics on its own (Mueller & Guido, 1997). This is the most successful kind of Machine Learning algorithm, and is known as *supervised learning*. It consists of automating decision-making processes by generalizing from known examples provided in large data sets. The user provides the system with sample data sets as input and specifies desired outputs, then the algorithm will find its own way of delivering the desired output given an input. At the bottom line, if implemented correctly, the Machine Learning algorithm will be able to create an output for an input it has never seen before without any human supervision.

Markov Chains

With randomly varying coupled condition affected by external disturbances, one can make use of a Markov Chain. Contrary to Bayesian networks, a Markov Chain represents an undirected graph. A Markov Chain is discrete-time and homogeneous, and takes values in a finite set and its transition probability matrix (Shen, Huo, Cao, & Huang, 2018).

Neural Networks

In Neural networks, one defines a recurrent network architecture before analyzing the hidden neuron activity with the goal of discovering states and transitions for resulting grammar. There is usually a layer of input neurons, one or more layers of hidden or internal neurons, and a final layer of output neurons. Neurons are split into layers consisting of the outputs of the neurons in one layer, which will feed forward into all the neurons of the next layer. Activation flows forward from the input neurons until the output neurons are activated in a pattern. Backwards propagation of the difference between actual and desired outputs is how Neural networks are trained. The given difference is referred to as the learning error of the Neural network (Cook & Wolf, 1998).

The inexactness of implementing a Neural network approach is that one can not direct the network to produce a machine just for a given stream. Even with a perfect sample input, it will also model behaviour that is not present in that stream (Cook & Wolf, 1998).

Bayesian Networks

Bayesian networks are considered as graphical modeled networks. They have some restrictions as they are basically static methods, where all parameters are probabilities. It is a powerful version of data filtering, consisting of variables and a set of directed edges between the variables. These variables each have a finite set of states, which is mutually exclusive. Together these variables, referred to as nodes, and the directed edges form what is called DAGs, directed acyclic graphs (Richardson & Jensen, 1997). In a DAG it is impossible to end up back at the same node by traversing the edges, which is illustrated in Figure 2.

2.3 Background on Bayesian Networks

Probabilistic graphical models are usually used for probabilistic inferences: (1) asking queries to the model, and (2) receiving answers in the form of probability values (Moreira, 2015). A representation of the dependence structure between multiple interacting quantities can be visualized through Bayesian networks. Its capability of estimating confidence in network features and handling noise are some key advantages of using BN, resulting in the ability to focus on interactions with strong signals in the data (Friedman et al., 2000). Another advantage of Bayesian networks is its ability to analyze expression patterns. One area where we can find them useful is

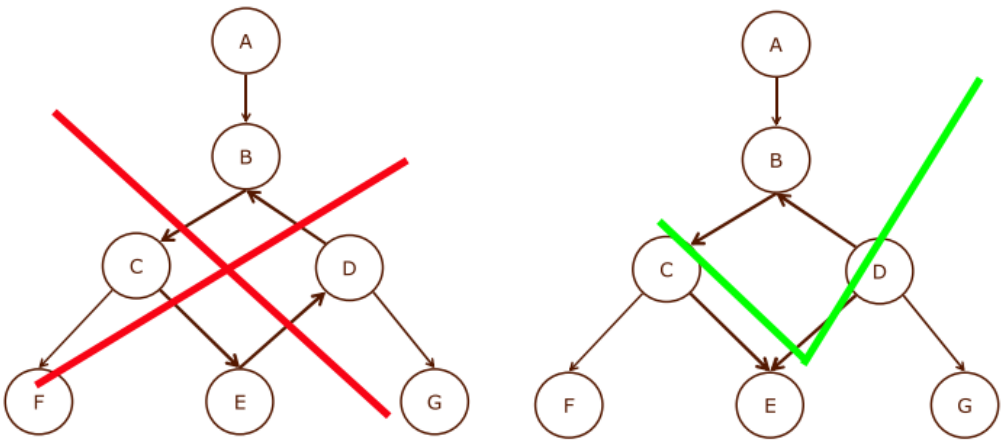

Figure 2: The variables and directed edges form a directed acyclic graph (Richardson & Jensen, 1997).

for describing processes that are composed of locally interacting components, where each value directly depends on the values of components from a relatively small set. Being well understood and mathematically defined in terms of probabilities and conditional independence statements, the potential for implementing statistical foundations for learning BN from algorithms and observations is promising (Friedman et al., 2000).

Three Groups of Bayesian Networks

Several algorithms emerge from applying Bayes theorem (Section 2.3.7) with different approaches combined with diverging orders of updating. The algorithms for reasoning within and constructing Bayesian networks can essentially be divided into three groups, namely **Graph reduction**, **message passing**, and **stochastic simulation**. Increased computational tractability of probabilistic reasoning is allowed through explicit representation of independences. Even though probabilistic inference is very efficient in singly connected Bayesian networks, exact algorithms for multiple connected networks are unfortunately liable to exponential complexity with respect to the increasing number of nodes in the network (BayesFusion, 2020). The problem has been shown to be *NP-hard* (Section 2.3.16) in general (G. F. Cooper, 1990).

2.3.1 Probability

Understanding the principles and meaning of probability is rather important for a decision modeler, as probability is used to quantify uncertainty in decision theoretic and decision analytic methods. Especially *three* fundamental interpretations of probability can be used to explain it further (BayesFusion, 2020):

Frequentist interpretation

Probability through the frequentist interpretation can be viewed by studying an event in an infinite number of trials, and is defined as the limiting frequency of occurrence. This can be further explained by considering the chance of rolling *one* from a single dice roll. The probability is proportional to hitting one in an infinite number of dice rolls.

Propensity interpretation

Physical and objective properties of an object or the process that generates an event is the determining factor of probability elicited through the propensity interpretation. Consider the dice example with this interpretation. The probability of rolling one in a single dice roll is determined by the physical properties of the dice, such as its pointed sides and its six symmetric sides.

Subjectivist interpretation

The perspectives of frequency and propensity surrounding chance are known as objectivist interpretations, as probability by assumption is an objective property of the physical world. The subjectivist interpretation of probability, also known as Bayesian interpretation, perceives chance from a different perspective, namely that it is subjective to personal measure of the belief in the occurrence of an event.

The interpretations mentioned above are theoretical and can therefore be subject to philosophical discussions and controversies. Probability can be explained through other definitions, but they do impose serious implications on decision analysis and its practice. *Most real world decision problems are impractical with the objectivist view*, as dealing with a process that is, or at least can be imagined as, repetitive in nature is a necessity to make a meaningful measure of uncertainty through probability. While dice rolls provide a process that falls under this category, modeling probability for uncertainty related to nuclear war inflict difficulties - as no nuclear wars have been present in the past, and even more so it being hard to imagine the repetition of such events. It is not easy to make use of physical considerations to present a viable argument for the complexity of the circumstances leading up to events like this. Decision analysis embraces the subjectivist interpretation as it provides a meaningful tool for managing such problems.

Believing that the probability of rolling one in a single dice roll is 0.2 is just as legitimate as believing that it is 0.17 (1/6) as long as the axioms of probability is not violated, being that the sum of probabilities for an event is required to be equal 1.0. Tracing this back to the subjectivist interpretation of probability as a measurement of personal belief, one can understand why measuring the uncertainty of a nuclear is just as legitimate. As it may seem that this holds too much freedom at first, the true advantage comes with a rule for updating probability through

evidence, namely Bayes theorem. When the degree of belief is updated through the application of Bayes theorem existing *limits theorems* proves that the degree of belief will converge to the limiting frequency. This will happen without regard to the value of the initial degree of belief unless extreme cases with values of exactly zero or one occur. The importance of reasonable prior probabilities lies in faster convergence rates, as these theorems gives guarantees in the infinity.

Prior probabilities can be based on both experts and extracted from databases. The combination of frequency data and expert assessment is a natural product of the subjectivist interpretation when there is need for accurate results. The process of calculating the degree of belief is known as *probability assessment* and there are several decision analytic methods ready for implementation, some of which will be covered in this research.

2.3.2 Utility

Preference is an important factor when working with real world applications, as a decision maker's preferences will often contribute to a products field of use. Decision theory introduces *utility* as a measurement of preference - a function on the set of real numbers that map a decisions process' attributes of possible outcomes. Utility is being conditioned up to a linear transformation that implicates that it has neither a significant zero point, nor a significant scale. Adding a constant and multiplying the utility by a non negative number results in an invariant preference over decision alternatives (BayesFusion, 2020).

Utility is, like the subjectivist interpretation of probability, subjective by assumption. When facing the same choice, even with a common set of beliefs, various decision makers may end up with separate results due to a different set of preference structure and utility functions. It is therefore essential that a utility function for any given decision problem is obtained from a relevant decision maker - a process primarily known as *utility elicitation* (BayesFusion, 2020).

Variables that measure utility are always *continuous*, which means that they are able to make an assumption from a continuous interval of any values. A common mistake is to regard them as *discrete* variables taking a finite number of values, such as in graphical models where the variables normally have discrete parents. This distinction is more evident when dealing *multi-attribute utility* (MAU) variables - where a function is specified by the combination of the parent nodes, known as utility nodes (BayesFusion, 2020).

2.3.3 Usability of Bayesian Networks

Classical analysis tools based on clustering algorithms have proven to be useful for discovering variables with similar functions and attributes. To reveal structural regulation processes is more tricky, especially with data that usually contains noise. Classical analysis tools only give a partial picture, unable to reflect over key events, which is not satisfactory to construct detailed

models that can deliver sound statistical significance (Friedman et al., 2000). Such detailed models, based on statistical properties of dependence and conditional independence in data, can be enabled by implementing Bayesian networks.

Bayesian networks provide an opportunity for reasoning under uncertainty, which is enabled through the use of probabilities. Conditional probability distributions describe all interdependencies in the model, making it possible to reason against the causal direction. Bayesian networks enable a consistent combination of information from various sources at the same time. Well calculated probabilities makes estimation of certainties for non-observable sets of variables and values, or values that are not cost effective to measure, a possibility. These values are referred to as hypothesis variables. By entering evidence in information variables that influence and/or depend on the hypothesis variable, Bayesian networks makes it possible to obtain these estimates. To each variable A with parents $B_1, B_2 \dots B_n$, the probability table (Equation 1) is attached (Richardson & Jensen, 1997).

$$P(A|B_1, B_2 \dots B_n) \tag{1}$$

In a superficial description of how a Bayesian network is built, the compilation starts with the creation of a moral graph where edges are added between all pairs of nodes having a common child. The next step is to remove all directions, so that one can triangulate the moral graph, and add edges until there is a chord of more than three nodes in all cycles. Next step is to identify the cliques of the triangulated graph, before organizing them into a junction tree for visualization and certain estimates (Richardson & Jensen, 1997).

Bayesian networks have an advantage over Markov Chains (Section 2.2.1) with the different ability to uncertainty. With the cycle free and directed structure of Bayesian networks, each task in the business process can either be *present* or *absent*. Given the uncertainty of which tasks that have already been performed, BN enables the performance of special analysis to further compute the probabilities of tasks occurring or not (Pearl, 2009).

2.3.4 Decision Support Systems

Probabilistic *Decision Support Systems* uses practically invaluable methods of decision theory and probability theory that are theoretically sound. When implemented correctly, they can assist in solving problems concerning classification, prediction, and diagnosis by modeling any real world decision problem. The area of use is vast, as decision support systems are able to arrive at intelligent solutions by combining the aspect of gathering, managing, and processing information with frequency data and expert opinions. This can be achieved by representing the problem structure through graphical models. Dedicated user interfaces can then be equipped to allow various desired observations and results to be entered in order to display the probability

distribution according to the most likely events (BayesFusion, 2020).

2.3.5 Support for diagnosis

A fusion of observations such as risk factors like test results, risk factors with symptoms, and patient or equipment history can be performed with Bayesian networks. Performing a combination of both predictive and diagnostic inference makes way for diagnosis as one of the most successful applications for such graphical models. A model can represent diverse system components, possible faulty behaviors such as symptoms, in addition to diagnostic test results. The essential part is the capability of capturing how plausible system defects can manifest themselves by test results, symptoms, and error messages. In practice, the system in question can range from devices such as an airplane or a car, to a natural system like a human body. The produced results can be viewed in a ranked list of likely defects along with a ranked list of the most cost effective and informative tests (BayesFusion, 2020).

2.3.6 Learning in Bayesian Networks

There is more than one way to define the structure and the numerical parameters represented in a Bayesian network, as they can be obtained either from an expert or learned from data. The structure of the graphical model is merely a representation of independences inferred from the data, with the numbers representing the *joint probability distributions* (Section 2.3.11). This allows both the structure and the numerical probabilities in a Bayesian network to be elicited from a combination of measurements, expert knowledge, and objective frequency data. It is common practice to classify the construction of Bayesian networks by two main approaches (Koller & Friedman, 2009):

- Construct the network *by hand*, where an expert is used to estimate *the conditional probability tables*.
- Use statistical models that will automatically *learn* these probabilities.

Expert Assessment

In many situations, the network will be so large that it will be nearly impossible for an expert to take on the assignment of assessing the probabilities to the random variables. The distribution of data may also vary over time, making it impossible for an expert assessment (Moreira, 2015).

Statistical Models

Statistical models offer a mechanism that will deal with the *probability distribution* by automatically learning a model. Depending on the situation that is being modeled, one can either have a fully observed dataset, an incomplete dataset, or a partially observed dataset. If one is dealing

with a complete event log, *Maximum Likelihood Estimation* (Section 2.3.12) can be used to simply count how many times each of the possible assignments of X and Y appear in your training data. To deal with incomplete logs, the network can be trained using *EM Clustering* (Section 2.3.13) in order to find an approximate probability distribution for task occurrence (Moreira, 2015).

2.3.7 Bayes Theorem

Bayes Theorem, proposed by Rev. Thomas Bayes, have been widely acknowledged and are still highly relevant to this day (Bayes, 1958). In terms of a number of independent *causes*, $A_i, i = 1,2,...,n_A$, that can cause one *effect* B, Bayes' Theorem can be stated as (D'Agostini, 1994):

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_{l=1}^{n_A} P(B|A_l)P(A_l)} \tag{2}$$

where it is assumed that we already know the *initial probability* of $P(A_i)$ along with the *conditional probability* of the i-th cause that produces $P(B|A_i)$. It is apparent from Equation 2 that $P(A_i|B)$ is dependant on the initial probability of the causes. Bayes rule increases the knowledge of $P(A_i)$ as the number of observations increase. A uniform distribution will be the start point when there is no a priori prejudice on $P(A_i)$. The final distribution is also depending $P(B|A_i)$, and is calculated manually or estimated with Monte Carlo methods. One thing to note is that these probabilities do not update by the observations (D'Agostini, 1994).

2.3.8 Evidence

Entering evidence as more observations are made is one of the basic operations that can be made on a probabilistic model, and is feasible through the implementation of Bayes theorem as introduced in Section 2.3.7. A graphical representation is allowed to be adjusted to a new situation in light of more available information. The result is a system that can be subsequently queried with regard to new posterior probability distributions (BayesFusion, 2020).

Virtual Evidence

Systems that model real world problems will often encounter observations of variables that are normally unobservable, e.g., when determining whether a disease is present or not. *Virtual evidence* is a term used to characterize the practice of entering evidence as a shortcut for such variables. This can be accomplished by modeling these variables next to other observable variables that might provide information about the unobservable variables. Even though one typically are unable to determine whether a disease is present or not with absolute certainty, a medical test can be modeled next to it. Since a test result is easily observable, it will provide

further evidence that either points to or against the given disease. This construct is in some practices used to modify the prior probability distribution representing a variable, but it is worth mentioning that this variable can not have any parents for this to work (BayesFusion, 2020).

Probability distribution over possible states is allowed to be to be entered as uncertain observations due to virtual evidence, making it similar to entering evidence. The main difference is that the probability distribution over all of a node's states is entered instead of observing a sate of a node.

2.3.9 Naïve Bayesian

A Naïve Bayes classifier will make an assumption called *class conditional independence*:

"The effect of the value of a predictor x on a given class c is independent of the values of other predictors."(Sayad, n.d.)

For the posterior probability $P(c|x)$:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{3}$$

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times ... \times P(x_n|c) \times P(c) \tag{4}$$

With respect to:

- $P(c|x)$ referring to the posterior probability of *class c* given *predictor x*.
- $P(c)$ being the prior probability of *class*.
- $P(x|c)$ being the likelihood, i.e., the probability of *predictor* given *class*.
- $P(x)$ referring to the prior probability of the *predictor*.

2.3.10 Discrete and continuous variables

It is important to distinct between fundamental properties when dealing with vast amounts of variables. One of these properties is the variable domain - the set of values the variables can assume. Even though there is no restriction in how many potential domains there can be for a set of variables, they can still be divided into two basic classes: **discrete variables** and **continuous variables** (BayesFusion, 2020).

Discrete variables

Discrete variables follow a finite set of conditions by taking values from a predefined finite set of states. Another characteristic is that these sets of states are usually small. An example that

most people can relate to is when ordering food, where one often specify how spicy the food should be. This variable often have three values: *Mild*, *Medium*, and *Spicy*. Another example are boolean variables that will assume between the values *True* and *False*. A numerical example is when items in a questionnaire are filled out on a scale from 1-5 (Likert scale).

Continuous variables

While discrete variables can assume values from a finite set of states, continuous variables take values from an infinite number of values. An example of a continuous variable can be *Future market value of Bitcoin*, assuming any monetary value between *zero* and *$100K*. Another example might be *Body temperature*, assuming any value between *30* and *45 degrees Celsius*.

The majority of Bayesian network algorithms are designed for discrete variables. To best exploit these algorithms, most Bayesian network models include discrete variables or *conceptually continuous variables*. Conceptually continuous variables are continuous variables that have been discretized for the purpose of reasoning (BayesFusion, 2020).

Even though the distinction between discrete and continuous variables is concise, the contrast between discrete and continuous quantities is indistinct. It is possible to represent numerous quantities as both discrete and continuous. Characteristics of discrete variables are that they are often sufficient for the purpose of reasoning and convenient approximations of real world quantities. With this in mind, *Body temperature* can be continuous variables but it may also be discretized as [*Low, Normal, Fever, High Fever*]. Three to five point approximations have historically proven to achieve good results in most cases through experience in decision analytic modeling (BayesFusion, 2020).

2.3.11 Joint Probability Distribution

The full joint distribution in Bayesian network, with X being the list of variables, is given by (Russel & Norvig, 2009):

$$Pr_c(X_1,...,X_n) = \prod_{i=1}^{n} Pr(X_i|Parents(X_i)) \tag{5}$$

The full joint distribution (Equation 5) is also the basis for computing classical exact inferences on Bayesian networks. In this formula, e figure as the list of observed variables where Y represents the remaining unobserved variables in the Bayesian network. For the query X, we get the inference given by:

$$Pr_c(X|e) = \alpha Pr_c(X,e) = \alpha \sum_{y \in Y} Pr_c(X,e,y) \tag{6}$$

Where

$$\alpha = \frac{1}{\sum_{x \in X} Pr_c(X = x, e)} \tag{7}$$

The given summation is for all possible y, which in Bayesian networks translates to all the possible combinations of values of the unobserved variables y. The α parameter represents the normalization factor for the distribution $Pr(X|e)$ (Russel & Norvig, 2009).

As introduced in Section 2.3.9, Bayesian networks are based on the Naïve Bayes rule, and needs to normalize the final probabilities by factor α (Equation 7) (Moreira, 2015).

Origin of Bayesian Networks

As described in Section 2.3.1, probability can be viewed as subjective and this perception is often referred to as the *Bayesian approach*, which is why Bayesian networks are sometimes called *belief networks*. The name Bayesian descends from this connection between the subjective representation of the joint probability distribution and the fact that it can be updated in the light of new evidence through the use of Bayes theorem (BayesFusion, 2020).

Representation of the joint probability distribution over n binary variables

A representation of the probability of every combination of states is required in order to represent the joint probability distribution straightforward with regards to n binary variables. Using n binary variables as an example, that would require $2^n - 1$ such combinations. Consider a network with four tables, containing a total of 30 numbers. This would require a total number of independent parameters to be equal to 15, as $2^4 - 1 = 15$. Knowing that the sum of all probabilities has to be 1.0 in every distribution, this results in half of the variables being implied by other parameters (BayesFusion, 2020).

Independence and arcs

There is a general rule with regards to the joint probability distribution that a missing arc follow each independence between a pair of variables. If there is no arc directly connecting two nodes, then conversely, a set of variables making them conditionally independent exists in the joint probability distribution. As a general principle, accessible and efficient representations of joint probability distributions is achieved through simplifications of the graphical model using independencies (BayesFusion, 2020).

Example - Comparing joint probability distribution with atoms in the world

Figure 3 illustrates a Bayesian network demonstrating the value of using joint probability distributions with an extreme example involving diesel locomotives. Various problems that are

encountered when modeling diagnosis of the locomotives, test results, symptoms, and their possible causes are included.

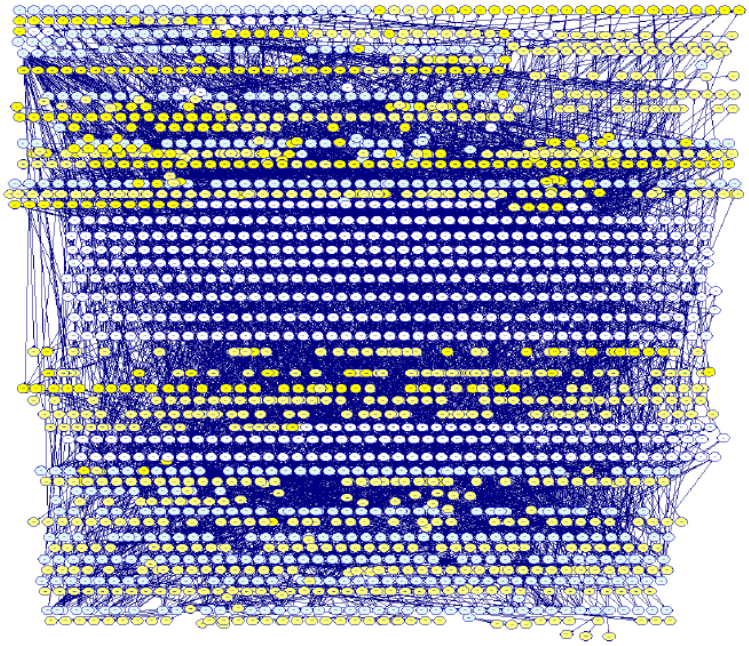

Figure 3: Comparing joint probability distribution over locomotive model to atoms in the world (BayesFusion, 2020).

The network pictured above contains 2 127 nodes, i.e., 2 127 variables are modeled through the joint probability distribution. To represent this distribution manually one would need $2^{127}-1$ numbers, translating to around 10^{632} numbers given that each variables are binary. The number of atoms in the universe is 10^{82}, in other words 550 orders of magnitude smaller to put the size of this number in perspective. In comparison, this model was represented by the use of only 6 433 independent variables thanks to the joint probability distribution. It is not uncommon to encounter models of similar size as presented in Figure 3, making representations of joint probability distribution practical.

2.3.12 Maximum Likelihood Estimation

The maximum likelihood estimation is a statistical method to estimate the mean and the variance of the probability distribution by only knowing a partial sample of the dataset (Bishop, 2006). It assumes that the data follows a Gaussian probability distribution, and can be used in Bayesian networks when you have complete event logs.

Gaussian Probability Distribution

A random vector $X = [X_1, X_2 ... X_n]$ can belong to a multivariate Gaussian distribution if one of the these statements are true:

- "Any linear combination $Y = a_1X_1 + a_2X_2 + ... + a_nX_n, \in \mathbb{R}$ is a univariate distribution."

- "There exists a random vector $\mathbf{Z} = [Z_1, ..., Z_M]$ with components that are independent and standard normal distributed, a vector $\mu = [\mu_1, ..., \mu_N]$ and N-by-M matrix $\mathbf{A}$ such that $X = AZ + \mu$."
- "There exists a vector μ and a symmetric, positive semi-definite matrix Γ such that the characteristic function of X can be written $\phi_x(t) \equiv \langle e^{it^T X} \rangle = e^{i\mu^T - \frac{1}{2}t^T \Gamma t}$." (Ahrendt, 2005)

The Maximum Likelihood Function

The likelihood function is given by:

$$L(\theta : D) = \prod_{m=1}^{M} Pr(x[m], [m] : \theta) \tag{8}$$

A full joint probability distribution $Pr(x|m|, y|m| : \theta)$ can be specified in Bayesian networks. Equation 8 is converted by the chain rule:

$$L(\theta : D) = \prod_{m} Pr(x[m] : \theta_X) Pr(y[m]|x[m] : \theta_{Y|X}) \tag{9}$$

The likelihood function from Equation 8 can be decomposed into two separate terms. Each of the separate terms represent a *local likelihood function*, predicting how well a variable can predict its parents:

$$L(\theta : D) = \left(\prod_{m} Pr(x[m] : \theta_X) \right) \left(\prod_{m} Pr(y[m]|x[m] : \theta_{Y|X}) \right) \tag{10}$$

If there is N random variables, the function will have N terms (Moreira, 2015).

The maximum likelihood has significant limitations regarding variance of the distribution, where it systematically underestimates it. This *bias* is due to the problem of overfitting that is encountered with polynomial fitting (Bishop, 2006).

2.3.13 Expected Maximization Clustering

The Expexted Maximization (EM) clustering algorithm is used in Bayesian networks when there are event logs present that are not complete. The expectation of the log likelihood is maximized in this algorithm according to the *Wishart distribution* (Kersten, Lee, & Ainsworth, 2005). The Wishart distribution is a complex probability density function. This is given in a gamma function, modeling a complex covariance matrix. The Wishart distribution plays an important role in the clustering algorithm, when the alternating phases gets derivated. The EM

clustering algorithm calculates the expected log likelihood by using updated posterior probabilities and assuming a priori probabilities (Kersten et al., 2005).

2.3.14 Representation of Cycles

Direct representations of business process diagrams, by capturing direct dependencies between tasks, is one of the advantages of Bayesian networks. However, they do not allow an explicit representation of cycles. This is because BNs are directed acyclic graphs. Many instances of the same node that are intractable to perform inferences needs to be created for a Bayesian network to represent such a cycle. The reason is that the inference problem is NP-Complete. This can be done by implementing an heuristic choosing the most probable transition between nodes (Moreira, 2015).

2.3.15 Mutual Exclusion Problem

Mutual exclusion is another structure that Bayesian networks is unable to represent directly. For two events to be mutually exclusive, it has be impossible for them to occur at the same time. To fix this problem when working with Bayesian networks, new edges has to be manually added to the network.

Example

A Bayesian network represent a business process where node A is the task that starts the process, and nodes B and C represents the end of the process. In the semantics of the business process it is thus required that the nodes B and C become mutually exclusive, because the process flow can only end in one of them. This is a problem as Bayesian networks cannot represent this structure, because all nodes depend on each other. For this to happen, an edge needs to be added between $B->C$, creating a new dependency between those nodes. The conditional probability table of the nodes needs to be manually configured so that when node B is *true*, the probability of node C occurring is *zero*. The same has to be done the other way. The result of this operation is that the probability that node C will occur when nothing is observed will be *changed* from what is was before the edge was manually created. The reason for this is that the extra edge will change the configurations of the conditional probability tables - ***changing the final probability values*** (Moreira, 2015).

2.3.16 Computational Complexity (NP-hard)

As introduced in Section 2.3, probabilistic inference can in worst case be NP-hard (G. F. Cooper, 1990; Dagum & Luby, 1993), meaning that models dealing with probability can easily reach both size and complexity that is excessive. The complexity of said models derive from two sources: (1) connectivity of the directed graphs that model the problem structure, and (2) ex-

ponential growth in the number of parents' conditional probability tables. The best practice in order to avoid such complexity is to carefully consider the number of parents of a node, as the size of the conditional probability table of the node in question will grow exponentially along with the number of nodes. The following example provides a brief visualization of how fast adding a few more parents can end up exhausting the computer memory: while 10 binary parents will result in $2^{10+1} = 2\,048$ parameters, adding one more leads to $2^{11+1} = 4\,096$ parameters. Adding as many as 20 parents will result in $2^{20+1} = 2\,097\,152$ parameters. Therefore, it is recommended to slow down the process of adding new parents to a node when the number surpass 15 or so. It is also worth mentioning that this number becomes even smaller if the node in question has a high number of states (BayesFusion, 2020).

2.3.17 Bayesian Updating

Observations (e.g., symptoms and test results) are often saved to databases and then stored as variables. This has high value as the impact of observing such variables can represent a subset of a model and be used to perform Bayesian inference. This impact can be measured towards the probability distribution over the remaining variables in the graphical model, and give information about its significance in the problem that is being modeled. Numerical parameters captured in a model like this is the basis of *Bayesian updating*. This is often referred to as belief updating or even probabilistic inference, despite the latter being somewhat less precise. The structure of the model can be explained in more detail as an explicit statement of domain independences. A more efficient algorithm for Bayesian updating is often achieved through a robust network structure. All algorithms for performing Bayesian updating are based on Bayes theorem (BayesFusion, 2020).

Bayesian updating is computationally complex and some algorithms are in worst case NP-hard (G. F. Cooper, 1990). Graphs consisting of tens or hundreds of variables are fortunately manageable due to various efficient algorithms. Pearl developed an algorithm for the joint probability distribution in a BN dealing with observations of one or more variables through a message passing strategy (Pearl, 1986). A productive way to transform a Bayesian network into a corresponding tree that utilizes various mathematical properties to perform probabilistic inference was introduced by Dawid (1992); Jensen (1990); Lauritzen and Spiegelhalter (1988). Each node in the graphical structure are corresponding to a subset of variables in the original graph.

Various approximate algorithms for stochastic sampling have been introduced, although Bayesian updating with approximations is also proven to be NP-hard in worst case (Dagum & Luby, 1993). Some of the most recognized approximate algorithms are *probabilistic logic sampling* (Henrion, 1988), *backward sampling* (Fung & Del Favero, 1994), *likelihood sampling* (Fung & Chang, 1990; Shachter & Peot, 1990), and *adaptive importance sampling* (Cheng & Druzdzel, 2000). The best stochastic sampling today is plausibly *evidence pre-propagation importance*

sampling, or exclusively referred to as EPIS (Yuan & Druzdzel, 2012).

2.4 Related Work

The following section provides an example where a Bayesian network was applied in process mining and provided the better results when benchmarked against Markov Chains. Another study is then investigated where healthcare analytics is used to determine an effective diagnostic model for ADHD in students by combining behavioral symptoms and physical symptoms. Lastly, an example of a research that discriminates ADHD children based on a proposed Deep Bayesian network is described.

2.4.1 An Experiment Using Bayesian Networks for Process Mining

Moreira (2015) proposed a new way of performing process mining by implementing Bayesian networks, to better take into account the probability of a task in a business process being present or absent. To compute these probabilities, one can use mechanisms such as Maximum Likelihood and EM clustering. Moreira's team only worked with complete logs, which means maximum likelihood was sufficient. Their goal was to define and test the structure of a Bayesian network made for a Loan Application Case study. The study suggested that Bayesian networks have much the same performance as Markov Chains, in their case with a 1.27% lower error percentage. This tells us that Bayesian networks make up good models for accurate event sequence predictions and compare well against alternative approaches like Markov Chains. However, this case study only worked with complete logs, where it is not necessary to estimate the probability tables through the usage of EM Clustering (Moreira, 2015).

Approach

Relationships between nodes from the events were first extracted by a Java program to return a Bayesian network that was readable by the SamIam toolkit. SamIam created a graph in a matrix which again was converted by another Java program into a network file recognized by SamIam. To eliminate cycles into an acyclic directed graph, the Bayesian network was altered in order to add mutually exclusive relationships between the nodes. To finally test the application, a MatLab program was created to receive the SamIam's network file and return a Bayesian network structure to compute full joint probability distributions and marginal probabilities. A Java program validated the model from a test set, and a Markov Chain was also made from the same log of events to compare their model with other literature (Moreira, 2015).

2.4.2 Using Healthcare Analytics to Determine an Effective Diagnostic Model for ADHD in Students

In a review of the effectiveness of common screening tools in relation to the Diagnostic and Statistical Manual of Mental Disorders (DSM) for a ADHD classifier, Mitchnick, Kumar, Fraser, et al. (2016) explored ADHD in an attempt to confirm the implications of interoperability of datasets and shared awareness of diagnostic algorithms. Behavioral symptoms like hyperactivity, inattention, and impulsiveness was together with physical symptoms such as fatigue, stress, and reduced brain region size analyzed to identify the strength of the relationship (correlation coefficient) between patient data from screening tool studies and the adult ADHD DSM-V classifier. The highest correlation coefficient was found when a combined method of the Adult ADHD Self Reporting Scale (ASRS), MRI, and Continuous Performance Tests (CPTs) was used. The study further goes on to propose a research design where Bayesian networks or Neural networks take those inputs for patient data in order to run analysis on the data collected from these algorithms to further define the influence of the relationship between the classification terms and the identifiers (Mitchnick et al., 2016).

2.4.3 Discrimination of ADHD children based on Deep Bayesian Network

Hao, He, and Yin (2015) proposed a method of using a Deep Bayesian network to retrieve information between different brain areas to discriminate children with ADHD. The Deep Bayesian network proposed in this research is a combination of Deep Belief networks and Bayesian networks. The model was used to classify fMRI ADHD image data, and was found to compute relationships among brodmann brain areas more effectively when Support Vector Machine (SVM) was used as classifier in the model. The results were compared to other contributions in the ADHD-200 competition and improved the prediction accuracies in three datasets that were tested (Hao et al., 2015).

3 Methodology and Methods

This chapter presents the approach of developing Bayesian networks to be able to make accurate predictions based on historic data and relationships found in hidden patterns by the various algorithms that are implemented. Both the methodology and methods that are artifact specific are covered.

3.1 Methodology

By applying design science as a research methodology, the objective is to develop an accurate solution that assist domain experts in the decision making process that today is mainly based on expert knowledge and experience. Specific design guidelines were followed, and included the use of some algorithmic techniques.

3.1.1 Desk Research

The desk research phase of this master thesis included an extensive literature review on Bayesian networks, background information about the internet-delivered intervention for adults with ADHD, getting to know various data management tools, and familiarizing with important health-related concerns and properties. The dataset for the treatment program was received early March 2021. On account of this, the desk research was an important part of establishing the necessary groundwork to meet the research inquiry.

3.1.2 Design Science Research

Researchers in the field of Information Science (IS) have to strive to obtain "further knowledge that aids in the productive application of information technology to human organizations and their management" (ISR, 2002). It is also key to gain "knowledge concerning both the management of information technology and the use of information technology for managerial and organizational purposes", to deliver meaningful research contribution to the field (Zmud, 1997). March and Smith (1995) found two paradigms working around each other, vital for this purpose - behavioral science and design science. The goal of behavioral science is *truth*, while the goal of design science is *utility* (Hevner et al., 2004). The principles of design science was implemented throughout this thesis.

Artifacts in the field of IT have broadly been defined as *constructs*, *models*, *methods*, and *instantiations*:

- **Constructs** - Refer to vocabulary and symbols.
- **Models** - Being abstractions and representations.
- **Methods** - When describing algorithms and practices.

- **Instantiations** - When the results are implemented and prototype systems.

These different practices needs to be assessed and evaluated with respect to the utility that is provided for the class of problems addressed. The contributions of both behavioral science and design science in the line of research will often be assessed as they are applied and add to the current knowledge base in a given business. Hevner et al. (2004) states: "A justified theory that is not useful for the environment contributes as little to the IS literature as an artifact that solves a nonexistent problem".

The design science paradigm comes from engineering and the sciences of the artificial (Simon, 1996). In design science, the focus is on the creation and evaluation of IT artifacts that is intended to solve problems in an organization. An emphasis is put on artifacts that are represented in a structured form. This can be a complete software, advanced mathematics, or formal logic, but can also include artifacts like informal natural language descriptions. Mathematical artifacts opens the door for a varied approach of quantitative evaluation methods, including analytical simulation, optimization proofs, and quantitative comparisons with similar artifacts (Hevner et al., 2004). To really understand and appreciate the importance of design science when working with information systems, it is important to apprehend that design is both a *process* and a *product*. In other words, "Design science describes the world as acted upon (process) and the world as sensed (artifacts)" (Walls, Widmeyer, & El Sawy, 1992).

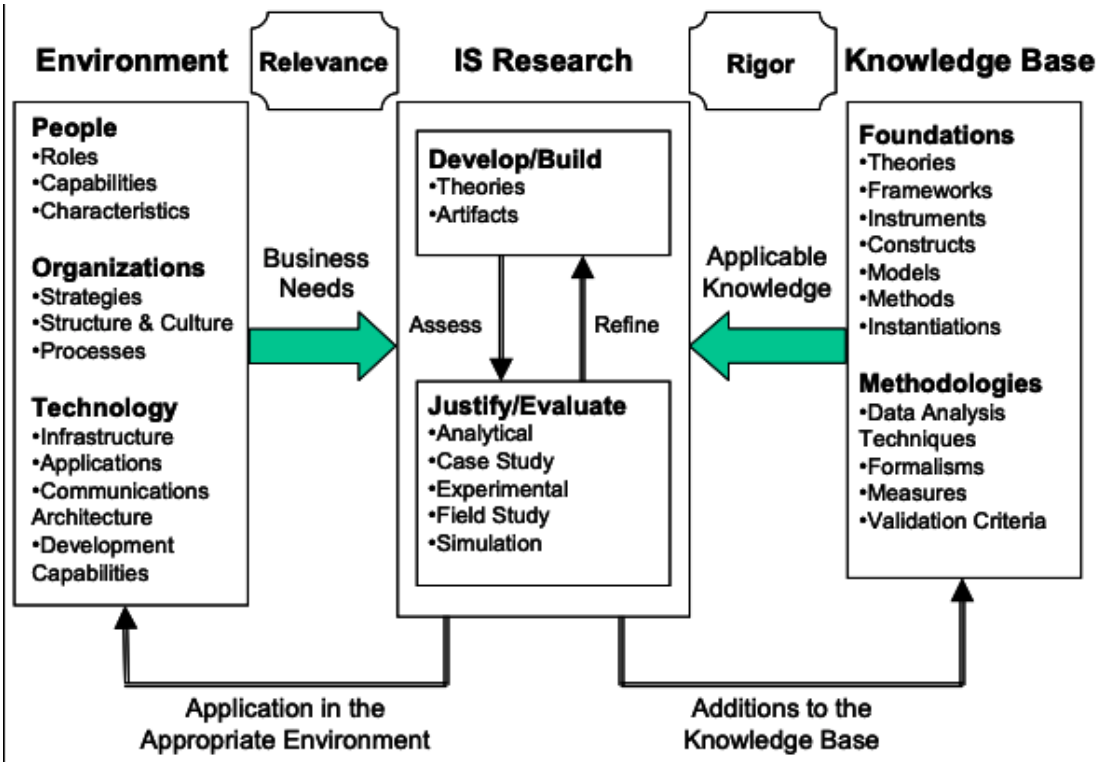

Figure 4: Design Science Research Model Hevner et al. (2004).

Figure 4 shows how two of the main factors in design science research, relevance and rigor, are linked together. The relevance that the given research provides to organizations should be

considered, as this may utilized by professionals to solve practical problems. Rigor is imperative to counteract conclusions that are not supported by the research, and is vital for research to be considered valid and reliable, and can contribute to increased knowledge in the domain area (Dresch, Lacerda, & Antunes, 2015).

March and Smith (1995) identified two design processes and four design artifacts that are produced by design science research . *Build* and *evaluate* are the two processes, while the artifacts are the four mentioned previously in this section - constructs, models, methods, and instantiations.

The *problem space* is defined by Simon (1996) as what ride the phenomena of interest. In the field of information science, this relates to people, business organizations, and current or future planned technologies (Silver, Markus, & Beath, 1995). Methodologies are typically used to evaluate the quality and effectiveness of artifacts through computational and mathematical methods, even though empirical techniques tend to be employed as well (Hevner et al., 2004).

Design science also addresses what is called *wicked problems* (Brooks & Kugler, 1987; Brooks Jr, 1996; Rittel & Webber, 1984). This is problems that concerns the following (Hevner et al., 2004):

- Constraints and requirements that are unstable due to poorly defined context.
- Complex interactions between components linking to both the solution and the problem at hand.
- Constant flexibility to change the design processes and the design artifacts in question.
- Being overly dependent of human social interactions to produce effective results.
- Being critically dependent of human cognitive abilities, like creativity, to produce effective results.

The first point on the list of wicked problem was relevant up to a certain point, but was sorted out when the research scope was fine tuned with the help of domain experts. The next complex interactions and constant flexibility to change was highly relevant in this thesis. The last two points on the list of wicked problems did not present any challenges in this research.

3.1.3 Design Science Guidelines

It is important that the guidelines presented in the following section are followed to some degree in any design research. For this reason, they are both adaptive and process oriented (Hevner et al., 2004).

1 Design as an Artifact

An IT artifact must enable the implementation of its application in a suitable domain. For that to be possible, the artifact needs to be described effectively. By definition, a dedicated IT artifact addressing important organizational problems should be the result of a design science research in the field of information systems (Hevner et al., 2004). The IT artifact has been referred to as the "core subject matter" in the respective field (Orlikowski & Iacono, 2001). Theories of instantiations and their representations are key concepts, as those theories are meant to serve as an explanation of both how the artifacts are created and adapted accordingly to changing environments and technologies (Weber, 1987, 2003).

As described in section 3.1.2, Hevner et al. (2004) include not only instantiations, but also models, methods, and constructs applied throughout the design cycle. They stand out from other literature by excluding elements and people of organizations in the definition. How artifacts evolve over time is also not under consideration. The reason is that models, methods, constructs, and instantiations are seen as equally crucial and are required for the creation of IT artifacts. Especially since the constructed results in design science research are rarely finished products to be put directly in practice. The feasibility of design product and the design process are demonstrated through the instantiation principle. This is especially relevant since the artifact delevoped in this thesis do not identify with that of a finished product, but rather to uncover the usefulness and possibilities of such a decision making support tool that can be utilized more effectively when more data is available and if implemented in an application.

The identification of needed capabilities that are yet to be developed, in order for information systems to thrive further, is seen as a critical nature of design science research (Markus, Majchrzak, & Gasser, 2002).

2 Problem Relevance

A problem can be stripped down to the difference between a set goal state and the system's current state. Problem solving is therefore important in design science. It can be defined as a *search process* (guideline 6) where actions are used to either reduce or eliminate the previous stated differences (Simon, 1996).

The key concept of research in the field of IS is to obtain understanding and knowledge to push the development and implementation of technology based solutions forward and solve business problems. This is done in design science by constructing artifacts that aim to change the occurring phenomena. To overcome predicted acceptance problems associated with new artifacts, a combination of organization based artifacts, technology based artifacts, and people based artifacts are needed to address the problems correctly (Hevner et al., 2004). The Bayesian networks developed in this thesis mainly falls under technology based artifacts.

3 Design Evaluation

When designing an artifact, it is important to put an emphasis on evaluation throughout the whole design cycle. Design is an iterative process where evaluation needs to start in early stages, not just on the finished product. Efficacy, utility, and quality altogether have to be demonstrated through rigorously executed evaluation methods. The evaluation results must reflect the requirements set by the business environment, including technical infrastructure (Hevner et al., 2004).

Evaluation of artifacts in design science requires data gathering, analysis, and definition of performance metrics. An evaluation could include metrics of consistency, completeness, accuracy, functionality, reliability, performance, usability, and other attributes that er case relevant (Hevner et al., 2004). Mathematical evaluation is possible when analytical metrics are appropriate, where distributed database design algorithms could be evaluated through average response time or expected operating cost (Johansson, March, & Naumann, 2003). In this master thesis, where Bayesian networks was applied in an online intervention in cognitive behavioral therapy, metrics as accuracy, AUC value, calibration curve, and sensitivity analysis was especially crucial for validation purposes (Section 3.2.4). The reason is the amount of data that will be evaluated, and the importance of being able to trust the given results. As the design process is an iterative process, the results of the evaluation will provide information approximate to how close the artifact is to an end product. The completeness of an artifact relies on the satisfaction of the constraints and requirements that was set early, and possibly changed underway, in the design process. Three major iterations was completed in this thesis, and evaluation through validation after each iteration was important to identify how robust the artifact was, what changes needed to be done, and how close the artifact was to completed. Hevner et al. (2004) contributed with an overview of design evaluation methods (Hevner et al., 2004):

1. **Observational**
    - *Case Study:* Study artifact in depth in business environment.
    - *Field Study:* Monitor use of artifact in multiple prospects.
2. **Analytical**
    - *Static Analysis:* Examine structure of artifact for static qualities (e.g., complexity).
    - *Architecture Analysis:* Study fit of artifact into technical IS architecture.
    - *Optimization:* Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior.
    - *Dynamic Analysis:* Study artifact in use for dynamic qualities (e.g., performance).

3. **Experimental**

    - *Controlled Experiment:* Study artifact in controlled environment for qualities (e.g., usability).
    - *Simulation:* Execute artifact with artificial data.

4. **Testing**

    - *Functional (Black Box) Testing:* Execute artifact interfaces to discover failures and identity defects.
    - *Structural (White Box) Testing:* Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation.

5. **Descriptive**

    - *Informed Argument:* Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifact's utility.
    - *Scenarios:* Construct detailed scenarios around the artifact to demonstrate its utility.

Among these various styles of evaluating an artifact, it is important to identify the appropriate methods for each specific artifact. Descriptive methods of evaluation is not as important when dealing with mathematical algorithms, as it is more suitable for innovative artifacts where other evaluation methods are not feasible. In this thesis, principles of static analysis, dynamic analysis, and optimization was highly optimal, along with with white box testing. This was due to the need of accurate and reliable predictions delivered from the Bayesian networks when predicting participant behavior, and the complexity that such networks can provide. There is also a measurement of style implemented in the field of design, where sufficient degrees of freedom remain to the developer (Norman, 2013). This has been defined in information systems as *machine beauty* (Gelernter, 1998). This can help vary the design process in a creative way, while still following given constraints and requirements, and add value to participating designers and the project as a whole.

4 Research Contributions

It is important to provide clear contributions when developing a new artifact. The work will often be assessed by what its new contributions are. There are three different potential types of research contributions in design science research. They are based on generality, novelty, and significance, where at least one of these contributions should be delivered in a given artifact (Hevner et al., 2004):

The Design Artifact - The contribution is the artifact itself, where it provides a solution to previous unsolved problems. This can both be by applying existing knowledge with new ideas

and ways, or by extending the current knowledge base in the field. Examples of contribution artifacts can be design tools, prototype systems, or system development methodologies. In this thesis, the design artifact provides a solution through a decision making support tool based on existing knowledge that is currently available.

Foundations - Development of products that will extend or improve already existing foundations in the knowledge base of design science are contributions that also have importance. Examples can be found in ontologies, design algorithms, modeling formalism, problem and solution representations, and innovative information systems. Even this is a new artifact that is not currently being used by the domain experts in Helse Bergen, the artifact is meant to both extend and improve existing foundations in the treatment program by making it more rigorous.

Methodologies - The last type of research contribution is the creative use and development of evaluation methods and metrics. Metrics in evaluation deliver a crucial part of information during the evaluation phase.

Implementability and representational fidelity are key metrics for assessing research contribution. Research must contribute with solutions to unsolved problems in the business environment. The presented artifact need to be implementable, representing the business and technology environments that are used (Hevner et al., 2004). This thesis focus on presenting an algorithm in the form of a Bayesian network that can be used as a decision making support tool. This makes both *the design artifact* and *foundations* the main contributions, as it aims to extend and improve existing working principles in the given sector.

5 Research Rigor

Rigor is about the way the research is conducted, and is derived from the effective use of the current knowledge base throughout the project. It includes the use of theoretical foundations as well as research methodologies. Importance lies in the selection of appropriate development techniques when working on an artifact or a theory (Hevner et al., 2004). An overemphasis on rigor can often lessen the relevance of the work, and it is both possible and necessary that the research paradigms are both relevant and rigorous (Applegate, 1999).

Performance metrics are usually important when assessing artifacts. Formal mathematics and algorithmic approaches heavily rely on evaluation criteria that match their performance and effectiveness through how appropriate the metrics are. This is why it is important to do thorough research before applying the first and best method of approach, both when it comes to development and evaluation along the way. Performance metrics was especially important to assess the research rigor in this thesis. The various validation methods are described in detail in Section 3.2.4.

6 Design as a Search Process

Design in all its essence can be viewed as a search process where the aim is to discover an effective solution to a given problem. The search for the most optimal solution is iterative, and the best solution will often be intractable in real world IS problems. Problem solving can be seen as making use of available means in the search for sought after ends, simultaneously as one satisfies the laws set by the environment (Simon, 1996). A problem will often be simplified to represent a subset of those means, ends, and laws to at least get a starting point. As several iterations are made, progress will be seen as the problem gets further expanded to more realistic terms. This will render the artifact more relevant and valuable. Means, ends, and laws in the field of information systems can often be represented by tools of mathematics and operations search (Hevner et al., 2004). This was important part in this thesis, as the algorithm was both defined and evaluated through validation of certain metrics, where accuracy, sensitivity, and complexity are important aspects.

The search for all possible means, ends, and laws will often be computationally infeasible. In such cases, the search must shift over to look for satisfactory solutions. In these cases one does not specify all possible solutions. It is important to understand *why* an artifact works, especially why the branching conditions are the way that they are in a Bayesian network. Most importantly is to establish that it *does* work, and to distinguish in which environments it works, even if the *why* in which it works is still to some degree unknown. The pros of this thinking is that the researchers are able to take advantage of the artifact as is to improve practice and produce context for further research to understand more about its underlying abilities (Hevner et al., 2004).

7 Communication of Research

There is a need for research to be presented to technology-oriented audiences, as well as management-oriented audiences. The prior needs to be enabled to implement the given artifact within the organization, and to take advantage of its benefits (Hevner et al., 2004). It also provides a growth in the knowledge base that will allow for further extension and evaluation, making it important to provide an understanding of the processes that the artifact was constructed on. For this reason, a thorough literature review on the working principles of Bayesian networks is among the main contributions of this thesis, along with the artifact itself.

Management-oriented audiences do not need the same detailed descriptions as mentioned above, but do need details that enables them to determine if organizational resources should be used to construct or purchase the artifact in question. It is also important for them to consider if it can be used purposefully in their specific organization. With this in mind, they need to know the knowledge required to effectively put in use the product, and the effectiveness of the solution approach that the artifact supplies. With that being said, it may serve its purpose to deliver some

advanced details to illustrate how it works and enable managers to appreciate the nature of the artifact (Hevner et al., 2004). Following this, the scope of this master thesis is set to focus on aspects of advanced details, easily comprehensible examples, and artifact performance.

3.1.4 Algorithmic Technique

As this research focus on the development and implementation of a Bayesian network in cognitive behavioral therapy for adults with ADHD, it involves the use of some algorithmic techniques. Bayesian networks, as described in Section 2.3, are represented by DAGs, which can fall under graph traversal. Mathematical optimization is also relevant, as some of the BN techniques (Sections 2.3.12 and 2.3.13) rely on the maximization of a function. Another key algorithmic technique is learning, being that Bayesian networks has its roots in Machine Learning.

Data Mining Techniques

There are several techniques used in the field of data mining. Some of the most regular are (*Data Mining Techniques*, n.d.):

- Classification
- Clustering
- Regression
- Outer
- Sequential Patterns
- Prediction
- Association Rules

Bayesian networks are probabilistic graphical models. This makes it fall under the **prediction** technique. The prediction technique uses a combination of some of the other data mining techniques, making **classification** and **clustering** techniques that will also be of importance throughout this research.

3.2 Methods that are artifact specific

This section will focus on methods that are artifact specific, and will cover the various structure learning algorithms used to learn the structure of the networks based on historic data, which will make up the prior probabilities of the resulting models.

3.2.1 Structure Learning Algorithms

There are five structure learning algorithms that was implemented at various stages of this thesis. These have different approaches and will be described throughout this section.

General properties and obstacles

There are some general properties of structure learning algorithms that needs to be covered before diving into the characteristics of each algorithm that will be considered and tested in this research. There are three major obstacles that needs to be tested before running each algorithm with GeNIe, the selected tool for constructing Bayesian networks (BayesFusion, 2020):

- **Discrete and Continuous Variables:** All of the six algorithms that will be covered in this section are capable of learning the structure of the graphical model when all variables are categorical. The PC algorithm is even able to perform structural learning when all variables are both continuous and have a joint probability distribution that is multivariate normal. The limitation arises when dealing with a mixture of discrete and continuous variables. When this is the case, all continuous variables needs to be discretized and represented as discrete.
- **Missing Values:** The structure of a model can not be learned through implementation of these algorithms if the data contains missing values. The only exception is the Naive Bayes algorithm, but this alternative does not actually learn the model structure as it merely creates it based on strong independence assumptions.
- **Constant Values:** Data variables that contain the exact same value across all of the columns in the data are collectively known as *constant values*. They are generally useless in a model's learning process, and none of the following structure learning algorithms allow for constant variables. The reason is that a variable x can not be a predictor for any other variable in the dataset when it takes the same value across each column. Variable x will still take same value no matter what values the other variables take. It is possible to enhance the model after the structure is learned if there is a situation where one wants to include variable x in the model. This can be done by then adding x and make a judgement of how the variable and its parameters is connected to the rest of the model. It should be stated that there is no basis for judgement about the relationship between x and the remaining variables.

Bayesian Search

One of the earliest and most popular algorithms used for structure learning is called the *Bayesian Search* algorithm. It uses the log likelihood function, guided by a scoring heuristic, and basically follows a hill climbing procedure with random starts. The Bayesian Search algorithm was first introduced by (G. F. Cooper & Herskovits, 1992) and was later refined by (Heckerman, Geiger, & Chickering, 1995).

The algorithm produces a Bayesian network achieving the highest score that, given the structure, is proportional to the probability of the data. Assuming that the same prior probability is

assigned to any structure, this score will also be proportional to the probability of the structure given the input data. In GeNIe, all of the parameters can be influenced by expert knowledge through a text box covering settings that is produced by the algorithm. It is good practice to investigate the theoretical limits of what the acyclic directed graph can identify based on the data. (BayesFusion, 2020). The Bayesian Search algorithm is also the basis for three of the following structure learning algorithms.

Greedy Thick Thinning

One of the algorithms that is based on the Bayesian Search approach is the *Greedy Thick Thinning* algorithm (Cheng, Bell, & Liu, 1997). This approach is split into several phases, starting with a thickening and a thinning phase. The algorithm starts with an empty graph, repeatedly adding the arc that will increase the marginal likelihood $P(D|S)$ maximally. This is the thickening phase, and it will be repeated until adding an arc no longer results in a positive increase. Also, no cycle will be created at this point. When this is done, arcs will be repeatedly removed until no positive increase will occur due to arc deletion (which is the thinning phase). Having the characteristic of being very fast, the Greedy Thick Thinning structure learning algorithm is an approximate approach that gives quite good results (BayesFusion, 2020).

Naive Bayes

The Naive Bayes algorithm does not actually learn the structure of the directed acyclic graph as it is rather fixed by assumption. The reason why it is being included in the category of structure learning algorithms is because the two following algorithms (*TAN and ABN*) uses a Naive Bayes structure, and the fact that it creates a Bayesian network. It is a naive method where the class variable is the sole parent of every remaining feature variables. This means that the nodes in the rest of the network have no other connections between them than its shared parent. The algorithm is prone to inaccuracies when the features are not independent conditional on the class variable, as the Naive Bayes structure makes this assumption (BayesFusion, 2020).

Tree Augmented Naive Bayes

The *Tree Augmented Naive Bayes* (TAN) algorithm is both described and thoroughly evaluated by (Friedman, Geiger, & Goldszmidt, 1997). It is a semi naive structure learning approach, and is also based on the Bayesian Search algorithm. Performing structural learning with TAN starts off with a Naive Bayes structure and accounts for possible dependencies between the feature variables by adding connections between them. This is conditional on the class variable, and the algorithm establishes that for every feature variable one is limited to only one additional parent. The algorithm is prone to inaccuracies when the features are not independent conditional on the class variable, as the Naive Bayes structure makes this assumption. The Tree Augmented Naive will produce a Bayesian network where the class variable is the parent of all other feature

variables, as well as additional connections between those. The result of this approach is a structure with the maximum score, which is a similarity between those algorithms that are based on the Bayesian Search (BayesFusion, 2020).

Augmented Naive Bayes

Both the Tree Augmented Naive Bayes (TAN) and the *Augmentented Naive Bayes* (ABN) structure learning algorithms are described and reviewed by (Friedman et al., 1997). Most of the information about this approach can be traced back to the previous paragraph, as the two mostly share the same principles. The main difference is that where the TAN algorithm sets the limit on the number of parents to 2, the ABN algorithm have no limitation on the number of additional added connections when entering each of the feature variables. The only parent limitation one will encounter is when *Max Parent Count* is set as parameter for the structure learning, which can be manually set by own preferences. Despite being a simple approach, the Augmented Naive Bayes algorithm has proven to perform reliably better than Naive Bayes (BayesFusion, 2020).

3.2.2 Analytics

Data often needs to be processed before a structure learning algorithm can be applied to raw data, and there are some areas that needs special attention to manage this effectively.

Missing Values

As previously mentioned, none of the algorithms can learn the structure of a model when the data contains missing values (except for the Naive Bayes algorithm). There are primarily two ways of dealing with missing values in order to run a structure learning algorithm: (1) delete the rows or columns in the data with missing values, or (2) replace them with something. When replacing the missing values, one can either replace it with a specific value, or choose to replace it with an average of the selected data.

Discretization

Discretization is method for dealing with continuous variables. This is done in order to present the data as discrete, which offers a better way of performing prediction analysis. The continuous variables are divided into a set amount of categories to represent different clusters based on data or domain specific standards. Three to five discretizised categories has proven to be effective in the past (BayesFusion, 2020).

Merging States

Two or more states might denote the same value as a result of an error in the data collection or encoding, e.g., *woman* and *female* are likely referring to the same value. This needs to be corrected in order to yield a result that is as accurate as possible. Values can either be merged manually, or through a functionality made available in GeNIe.

Knowledge Editor

As described in Section 2.3.6, expert knowledge can be a valuable tool when combined with frequency data in the learning phase. This is typically done through three actions:

- **Force arcs:** arcs that are manually forced will guaranteed appear in the learned network structure.
- **Forbid arcs:** arcs that are manually forbidden will guaranteed be absent in the learned network structure.
- **Temporal tier:** variables can be assigned to temporal ties.

Forbidden arcs can be viewed as a way of expressing expert knowledge that is so certain that it should not be overridden by data.

The temporal tier is used to specify the temporal order among variables. This means that no arcs will be constructed from variables that occur later in time, i.e., in higher temporal tiers, to variables in previous temporal tiers. It is beneficial to view the structure of a Bayesian network in terms of causation: Arcs should be forbidden to go from variables in later temporal ties to earlier tiers knowing that causality never work backwards (BayesFusion, 2020).

3.2.3 Learning Parameters

When frequency data is available and the structure of a model is learned, the next step is to learn the parameters of that network. This can be done by matching the network to a data file.

EM clustering algorithm

The first step is to go through a mapping phase between the variables defined in the network and the variables defined in the data set. The EM algorithm (Dempster, Laird, & Rubin, 1977; Lauritzen, 1995) is used in order to learn the probability distributions. As covered in Section 2.3.13, Expected Maximization Clustering is capable of learning parameters from data where there are missing values.

3.2.4 Validation

Validation method is important when using an algorithmic technique, and focus on specific metrics that are used to evaluate the quality of a model. Validation of the results is a crucial element of the structure learning. Some consideration should be put into what validation method is best suitable, as they have different strengths and weaknesses depending on the situation and data used in the process. Following comes an overview of some of the available validation methods (BayesFusion, 2020):

- **Test Only:** This is the simplest evaluation method available, where one test the model directly on the data file. When the graphical model has been learned from a different dataset and the goal is to test it on data it has never seen, or when it has been elicited based on expert knowledge, Test Only is a suitable option.
- **K-Fold Crossvalidation:** A more typical situation than the one mentioned above is when one wants to do both the learning and the evaluation of the model on the same dataset. This calls for a method known as *cross validation*, splitting the data into two subsets: training and testing. The most powerful method of cross validation is known as K-fold cross validation. This approach divides the dataset into K folds of equal size, before traning the network on $K-1$ folds, and finally tests it on the last Kth fold. This operation is then repeated K number of times, each time with a different fold of the dataset being designated for testing. There are various ways of controlling this approach. One can manually sett the number of parts selected by adjusting *fold count*. *Folding seed* is another, which allows to set up random assignment of records to different parts. This is a way to of assuring that the evaluation process is repeatable, as long as it is set to anything different than zero. An actual random number seed from the system clock will be picked when zero folding seed is selecting, making it truly random.
- **Leave One Out (LOO):** Following comes an extreme version of K-fold cross validation, namely the Leave One Out method. What differentiates it from the K-fold cross validation is that K amounts to the number of records (n) in the dataset. This further leads to the network being trained on $n-1$ records before it is tested on the *one* remaining record. The operation is then respectively repeated n number of times. It is advised to use the LOO method whenever it is reasonable with respect to computation time, as it has shown to be the most efficient method of evaluation. The only disadvantage that comes with implementing the LOO method is that one might suffer from long computation time when dealing with very large number of records in the dataset.

The second important element, next to validation method, is the selection of *Class nodes*. This refers to the nodes that the model will aim to predict, and there has to be at least one class node selected. When the validation process is is finished, the following metrics will as a result be

available: *Accuracy*, *Confusion Matrix*, *ROC Curve*, and *Calibration Curve*.

Accuracy

This demonstrates the accuracy that the graphical model achieved through validation. The class node that is most probable over all other states is chosen for each record throughout this process. The results yield both the sensitivity and specificity of the model, which can be valuable tools of further analysis.

Confusion Matrix

A confusion matrix is a good supplementary to the accuracy of the model, as it specifically demonstrates the number of records that have been classified correctly and incorrectly. Presents the model's guess in the rows of the matrix and indicates the actual state of affairs along the columns. Off diagonal cells show the classes that are incorrectly identified, while the diagonal, marked with bold numbers, demonstrates the numbers of correctly identified instances.

Receiver Operating Characteristic (ROC) Curve

The states of each of the class variables are presented on the *Receiver Operating Characteristic* (ROC) curves, and there are as many ROC curves for each class nodes as there are states. With roots from Information Theory, it is an exceptional way to express the quality of a model that is independent of the classification decision. The ROC curve is able to present the possible accuracy ranges, and gives insight into what has to be sacrificed in point on the curve in order to improve another point. The theoretical limits of accuracy on one plot of the model is presented, making the ROC curve effective when choosing a criterion that is appropriate for the application at hand.

Area Under the ROC Curve (AUC) is displayed above the ROC curve, and is a simple but imperfect way to use one number to express the quality of the graphical model. An AUC value of 0.5 suggests in general that the model in question has no discriminatory ability, i.e., ability to diagnose patients with and without a condition or disease. Acceptable values are considered to be somewhere between 0.7 and 0.8, while 0.8 to 0.9 is considered to be excellent. A model with more than 0.9 is considered to perform outstanding (Mandrekar, 2010). The ROC curve will be above the diagonal line when the implemented classifier achieves good results. It should be mentioned that the curve is drawn based on a finite number of points, which is based on the same dataset that was provided for the verification phase. This means that the curve can be rugged when the number of points is small. It is something that often occur when the data file is small, which happens to be the case in this research. It might provide insight to overlook these points on the curve, as they show the probability threshold value needed to achieve that specific point. ROC curves are both useful and fundamental measures of the performance of a model

(BayesFusion, 2020).

Calibration Curve

Another important performance measure from the validation stage is *calibration*, which can be viewed from a calibration curve. Since the output of a probabilistic model is a probability that can serve as a useful tool in decision making, this probability should ideally be as accurate as possible. A calibration curve compares how the output probability of a model measures up to observed frequency data. The x-axis displays the probability p for an event happening produced by the model, while the y-axis displays the actual observed frequencies in the data for the corresponding probabilities. There is a diagonal line that represents the ideal calibration curve, depicting a scenario where every probability p are corresponding to the observed data. The values of probability are grouped in a way that can give sufficiently amounts of data records in order to estimate an actual frequency for the y-axis. This can be done through either *Binning* or *Moving average*. Binning divides the interval into $[0...1]$ equal sized bins, where changing the number of bins changes the plot as well. Moving average always takes the neighboring k output probabilities on the x-axis and displays class frequency in a sliding window on the y-axis. Changing the window size results in a replaced plot in the same way as with binning (BayesFusion, 2020).

Validation for multiple target nodes

There might be times where one work with multiple class nodes, i.e., when there are more than one problem present simultaneously. The accuracy will then be computed and presented in separation for each of the nodes. An overall accuracy of the model can be found by combining the accuracies of interest. The confusion matrix requires that one of the class nodes is selected, and there is as many confusion matrices as there are class nodes. A state of one of the class nodes is required to be selected for both the ROC curve and the calibration curve (BayesFusion, 2020).

3.2.5 Sensitivity Analysis

In addition to the previously mentioned validation methods, *Sensitivity Analysis* (Castillo, Gutiérrez, & Hadi, 1997) is a valuable technique used to validate the probabilistic parameters of a Bayesian network. Examining the effect of small changes in numerical parameters enables observation of changes in posterior probabilities to identify which parameters has the highest effect on the output of the model. The output of a Bayesian network, in form of reasoning results, are more affected by highly sensitive parameters. Being aware of those variables allows for a directed allocation of effort to achieve desired accurate results.

GeNIe, the chosen tool for this research, uses an algorithm proposed by Kjærulff and Van

Der Gaag (2013) that enables simple sensitivity analysis of a Bayesian network. Target nodes over the numerical parameters in a BN are used to efficiently calculate a complete set of derivatives of their probability distributions. Knowledge about the importance of precision that these numerical parameters hold when calculating posterior probabilities can be achieved through these derivatives. Substantial changes in the value of a variable make little difference when the derivative is small. However, if the derivative for a parameter p is large, substantial changes in the targets posterior probabilities may result from even a small change in p.

The result of the sensitivity analysis is displayed with a visualization of the nodes in the Bayesian network colored in red and grey. The nodes that are important for calculation of the posterior probability distribution with the current network structure are marked red, where the transparency of the color is an indication of sensitivity. Nodes colored in gray means that their parameters are not used in this calculation, and have its sensitivity qualitatively determined as it is deemed to be zero. The results from the sensitivity analysis algorithm are context dependent. This is important to understand since the set of observations made in the network and the current set target influence the value of the calculated derivatives. These will be recalculated if further observations are performed, which can result in a recolored graph (BayesFusion, 2020).

Tornado Diagram

A *tornado diagram* is a useful way to further analyse the results from the sensitivity analysis. For a selected state of the target node, it shows the most sensitive parameters sorted from most to the least sensitive. The precise location in the model is also available for each parameter, as well as the range of changes in the target state as the parameter changes. The exact numerical sensitivities for each bar can be accessed by hovering over any of the bars. Following is a brief explanation of these parameters (BayesFusion, 2020):

- **Target value range:** Displays the minimum and maximum values for the selected target outcome's posterior probability.
- **Parameter range:** Displays the minimum and maximum parameter value.
- **Current parameter value:** Displays the nominal value of the probability in the conditional probability table of the node. The states of the conditioning variables uniquely identifies this probability.
- **Derivative:** This is the value of the first derivative from the posterior probability T related to the target node's selected state. It is a measurement over the parameter p in question. The following linear functional form represents the posterior probability:

$$T = (a * p + b)/(c * p + d) \tag{11}$$

The derivative is the basic measure of sensitivity and is together with the target posterior range obtained through four coefficients (a, b, c, and d) that the sensitivity analysis algorithm calculates. The equation for calculating the derivative is:

$$D = (a * d - b * c)/(c * p + d)^2 \tag{12}$$

The sign of the derivative is constant for all values of p, meaning that the function is either monotonic or constant. This is because the denominator is always positive. How much the posterior probability will change given that p is modified in its entire range can be calculated by substituting 0 and 1 for p. This range is defined by:

$$p_1 = b/d \tag{13}$$

$$p_2 = (a+b)/(c+d) \tag{14}$$

Which value is the minimum and maximum is determined by the sign of $a * d - b * c$.

- **Coeffs:** This lists the calculated values of a, b, c, and d used to represent the posterior probability.

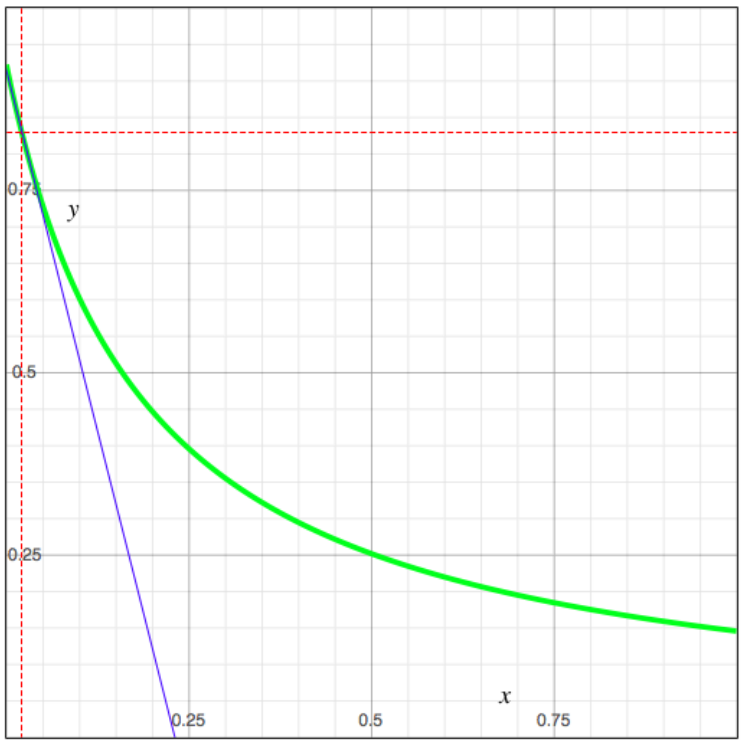

Figure 5: Calculation of sensitivity analysis, from BayesFusion (2020).

Figure 5 illustrates how the sensitivity is calculated by the algorithm. x corresponds to p (value of the selected parameter) in the equations above, where y stands for T (posterior probability of the target node's selected state). The posterior probability, displayed with the green line, is a function of the selected parameter's value. The blue line displays the derivative, and is the tangent to the green line at current value of two parameters.

There are few standard procedures when it comes to sensitivity analysis. As it is beneficial to identify and pay attention to the most important parameters in a probabilistic model, sensitivity analysis serve as a good first step in that process (BayesFusion, 2020).

3.3 Technology

This section presents an overview of the technologies that were used in this thesis.

3.3.1 Excel

Excel is a spreadsheet that is developed by Microsoft, featuring graphing tools, calculating, pivot tables, and macro programming with Visual Basic (*Microsoft Excel*, n.d.). It is made for Windows, macOS, Android and iOS and supports real-time query technologies. Excel can be used to manipulate data, and was used for some of the data processing in this thesis.

3.3.2 Pandas

Pandas is an open source data analysis and manipulation tool, and is built on top of the Python programming language (*Pandas*, n.d.). It is fast, flexible, and offers data structures and operations to manipulate numerical tables and time series, and was used for data processing in this thesis.

3.3.3 GeNIe

GeNIe Modeler is a graphical user interface to SMILE Engine that allows for interactive model building and learning (*GeNIe Modeler: Complete Modeling Freedom*, n.d.). It was made under the criterion that it should allow for a complete modeling freedom, and is intended to be able to model whatever the domain demands. In this research, GeNIe was used for Bayesian network development and evaluation.

3.3.4 Diagrams.net

Diagrams.net, formerly known as draw.io, is a free online diagram software that can be used to create flowcharts, network diagrams, UML diagrams, ER diagrams, database schemes, and more (*Diagrams*, n.d.). This was used in this research to create flowcharts of workflows.

4 The Dataset

This chapter presents an overview of the dataset that was supplied by Helse Bergen to be used in this thesis. An introduction to the data properties is provided, where especially the self-report scales are explained as they offered key features. The data processing is then described, including how the data was prepared, calculation of scoring results, discretization into categories, and splitting into different datasets.

4.1 Ethical Concerns and Consent

The dataset used for predicting participant behaviour was anonymized by Helse Bergen who provided the data and approved it to be used in this research. Participants are listed by anonymous IDs which only purpose was to exclude those who were used as test participants by the domain experts working on the program. Participants who were eligible to participate in the study had to sign a consent to participate, which was digitally signed in the YouWell portal with BankID (*An Internet-delivered Intervention for Coping With ADHD in Adulthood (MyADHD)*, n.d.). Involved researchers in the project signed non-disclosure agreements for confidential disclosure. The purpose of the research and its use of participant data is to identify relationships and hidden patterns that enable predictions to be made in order to assist the decision making process and further aid in making the life of adults struggling with ADHD easier.

4.2 Dataset Properties

This section will provide insight to the various dataset properties that was used in this thesis. The data could with advantage be richer in terms of volume, as only 109 participants have currently been through the program, and where some never finished. However, there is a considerable amount of properties available on each participant. Not all data points were relevant for the scope of this project, and more focus is put on the identified properties as a result. The questionnaires that were used to map participants at the start and at the end of the treatment program will also be explained to provide better understanding. The data was provided in an Excel format (.xlsx) making it easy to handle as most of it was already structured, and participant IDs are matching throughout all datasets.

4.2.1 Mapping of ADHD participants: Pre/Post

At the start of the treatment program, before being assigned to any training modules, the participants had to go through a mapping phase. This includes answering several questionnaires where each participant assess how they relate to various questions and statements based on a Likert-scale. The pre-mapping data has 115 data rows an feature every participant that started the treatment program, while the post-mapping data only contains 65 data rows. The post-mapping

activity was conducted at the end of the treatment program, albeit before the follow-up activity, and suffers from the loss of participants dropping out mid-program.

The following subsections will present an explanation of the different properties that are represented in the pre/post-mapping data. The datasets includes the same properties, the only difference being that the tests scores are from different stages in the program (and have a mismatched number of participants).

Demographics

The data includes four different properties that covers the demographics of the different participants, namely *age*, *gender*, *education*, and *occupation*. This was regarded relevant as Bayesian networks are good at identifying hidden knowledge in branching conditions, and investigating the potential impact of demographics was part of the research scope.

ADHD General

There were also answers from six questions regarding participant background following the demographics. These were answered by the participants in free-text, and included year of ADHD diagnosis, the clinic that diagnosed them, what medication they are on, and how often they take it. These were excluded as it was considered that they did not provide contributions to the research scope.

The Adult ADHD Self-Rating Scale (ASRS)

The *Adult ADHD Self-Rating Scale* (ASRS) is a questionnaire that includes all of the 18 symptoms of ADHD that is included in the diagnostic manual DSM-5 (Association et al., 2013). The self-report scale has 18 items, and is divided into the following two subscales: one scale that contain 9 questions regarding problems with *Inattention*, and one scale that contain 9 questions measuring problems with *Hyperactivity*. The ASRS uses a 5-point Likert scale with options "Never" (0), "Rarely" (1), "Sometimes" (2), "Often" (3), or "Very Often" (4). This gives the full-scale ASRS a total of 72 points, and 36 points on each of the subscales for inattention and hyperactivity. Test-retest reliability of the ASRS has been proven to be 0.88 (Kim, Lee, & Joung, 2013). Both subscales have the following cut-offs: a score of 0-16 means unlikely to have ADHD, 17-23 means that the subject is likely to have ADHD, and 24-36 means highly likely to have ADHD (*MyADHD - Digital Training for Adults With ADHD*, n.d.).

ADHD Quality of Life Measure (AAQoL)

The *ADHD Quality of Life Measure* (AAQoL) has 29 items designed to assess health-related quality of life (HRQL) among adults with ADHD during the past two weeks (Gjervan & Nordahl, 2010). AAQoL uses a 5-point Likert scale with options "Not at all / Never" (1), "Rarely /

A Little " (2), "Sometimes" (3), "A lot / Often" (4), and "Extremely / Very Often" (5). It gives a total score based on all 29 items, and also have the following 4 subscales: *Life Productivity* (**11 items**, including getting things done on time, completing projects or tasks, remembering important things, and balancing multiple projects), *Psychological Health* (**6 items**, including feeling anxios, overwhelmed, and fatigued), *Life Outlook* (**7 items**, including perceptions that energy is well spent, people enjoy spending time with you, you can successfully manage your life, and you are as productuve as you would like to be), and *Relationships* (**5 items**, including tension, annoyance, and frustration in relationships). Both total and subscale scores are computed by in the following three-step procedure: (1) all scores except from the seven items in the Life Outlook subscale are first reversed, before (2) transforming all item scores to a 0-100 point scale (1 = 0; 2 = 25; 3 = 50; 4 = 75; 5 = 100), and then (3) summing the item scores before dividing by the item count. It is indicated by the scoring algorithm that the total score can be computed with up to three missing items, and each of the subscale scores with up to one missing item (*An Internet-delivered Intervention for Coping With ADHD in Adulthood (MyADHD)*, n.d.).

The Perceived Stress Scale (PSS)

The *Perceived Stress Scale* (PSS) is a 14 items questionnaire meant to measure a person's stress, as well as how uncontrollable respondents think about their lives during the past month (Cohen, Kamarck, & Mermelstein, 1983). The PSS uses a 5-point Likert scale with options "Never" (0), "Almost Never" (1), "Sometimes" (2), "Fairly Often" (3), "Very Often" (4), and results in a Chronbach's alpha of 0.89 (Roberti, Harrington, & Storch, 2006), which potray the internal reliability of the test (*MyADHD - Digital Training for Adults With ADHD*, n.d.). The PSS-14 has 7 positive weighted questions and 7 negative weighted questions. In order to calculate the total score, the positively weighted items first have to be reversed before summing the results from all items together.

The Patient Health Questionnaire (PHQ-9)

The *Patient Health Questionnaire-9* (PHQ9) (Kroenke, Spitzer, & Williams, 2001) is a measure of depression severity. The PHQ-9 uses a 4-point Likert scale with options "Not at all" (0), "Several days" (1), "More than half the days" (2), and "Nearly every day" (3). The self-report tool has 9 items that maps giving a total of 27 points. Ranging from increasing severity, the scores are divided into the following categories: 0-4, 5-9, 10-14, 15-19, and 20 or greater. A total score of 5 on the PHQ-9 indicates mild depression, 10 represents moderate, 15 represent moderately severe, and 20 represent severe depression. Both validity and reliability of the PHQ-9 have indicated that the tool has solidified psychometric properties, and internal consistency has been proven to be high. It received Cronbach alpha scores of 0.86 and 0.89 in a study that involved two different patient populations (*MyADHD - Digital Training for Adults With ADHD*, n.d.).

General Anxiety Disorder (GAD-7)

The General Anxiety Disorder (GAD-7) questionnaire (Spitzer, Kroenke, Williams, & Löwe, 2006) is a self-report tool designed to map the a person's mental health state during the last two weeks. The questionnaire consists of seven items aiming to assess anxiety in particular, being one of the most common mental disorders. The items covers the patient's nervousness, feeling anxious or on edge, uncontrolled worrying, having trouble relaxing, having trouble sitting still due to being restless, feeling afraid, and being easily annoyed or irritable. The response type consists of a 4-point Likert scale with options "Not at all" (0), "Several days" (1), "More than half of the days" (2), or "Nearly every day" (3) giving a scale from 0 to 21 points. Mild, moderate, and severe anxiety is identified through cut-off points of 5, 10, and 15 points, respectively (Williams, 2014).

Perceived Deficits Questionnaire (PDQ-5)

The *Perceived Deficits Questionnaire* was first introduced as a 20 item self-rated tool to assess subjective cognitive dysfunction in people with depression. The 5 item version used in this research (PDQ-5) uses a 5-point Likert scale with options "Never the past 7 days" (1), "Rarely (once or twice)" (2), "Sometimes (3 or 5 times)" (3), "Often (around once a day)" (4), and "Very often (more than once a day)" (5). This gives the PDQ-5 a total score of 25 points. It was originally developed as a scale that was intended for patients with multiple sclerosis, but has in later time been adapted and validated as a measure for patients with major depressive order (*Perceived Deficits Questionnaire*, n.d.).

The Self-Compassion Scale (SCS)

The *Self-Compassion Scale* (SCS) (Neff, 2016) examines different components of self-compassion, such as emotions, thoughts, and behavior. It was introduced as a 26 item scale (*Self-Compassion*, n.d.), but this research used a 12 item version that measure how people respond to feelings of inadequacy or suffering with self-kindness, self-judgement, common humanity, isolation, mindfulness, and over-identification. The questionnaire uses a 5-point Likert scale with options "Almost never" (1), "A little" (2), "Some" (3), "A lot" (4), and "Almost always" (*MyADHD - Digital Training for Adults With ADHD*, n.d.). Unfortunately, the data provided was too incomplete to provide any value and was for that reason excluded from further analysis.

4.2.2 ASRS data

The ASRS questionnaire was included in both the Pre Mapping data and the Post Mapping data that were conducted at the start and at the end of the treatment program (before a 3 month followup was scheduled). The datasets provided for this research featured various versions of ASRS scoring results, mainly data from the Pre/Post Mapping that were structured in different

ways. As the best structured versions were identified and used for further cleaning and analysis, the rest were mainly sidelined due to it containing the same content. It was later identified that one of the semi-structured datasets included scoring results from ASRS questionnaires that were handed out between modules, depicting how the participants were feeling as the treatment program progressed. Including only results from 9 of the ASRS questions, these were only labeled with a date. The participants started the treatment program on different dates and answered various amounts of these between-modules ASRS questionnaires, some even answered none of these. However, it was included in later experiments in an attempt to find a pattern, and turned out to be of great significance to the results.

4.2.3 Activity Data

Activity data was also included in the provided datasets, containing several tabs of different data properties including *Activity*, *Logins*, *Module Count*, *Module Activity*, *Notifications*, and *Randomized Reminders*. This was mainly a combination of structured, semi-structured, and unstructured data, where some of the data consisted of over 40 000 data rows. Due to the considerable size of this data, as well as the uncertainty in contribution value and extent of data processing needed to be able to include this in the development of any Bayesian networks, it was excluded from this research as there were not enough time or resources available to complete this in a feasible manner.

4.3 Data Processing

The given data needed both cleaning and calculations before any experiments could be conducted to construct and test out the usability of any Bayesian networks. This process was primarily performed in four steps:

1. Cleaning up and identifying usefulness
2. Calculating scoring results
3. Discretization into categories
4. Splitting into different datasets

The following subsections provide a review of how the various processes were performed in more detail.

4.3.1 Cleaning up and identifying usefulness

In order for data to be entered used to construct any Bayesian networks, properties showing usefulness in form of potential contribution needed to be identified and cleaned properly. The

provided data were both structured and semi-structured, and included some degree of duplication. Excel was primarily used in this phase for its versatility and readability. It was important to get to know the data properly, its correlation, and most importantly its completeness. This was simplified through an overview of which participant IDs that were featured in the various datasets. The overview was useful as it differentiated completed modules among participants, i.e., that the amount of participants that completed the Post Mapping phase were almost halved from the Pre Mapping phase. This also provided information about which participants dropout out during the treatment program.

The data included 114 data rows of different participants that completed the Pre Mapping phase and 63 completing the Post Mapping phase. One participant (ID: 72) was featured two times with both different scoring results and demographics, and both were removed from further analysis conformation that this was a test user from the domain expert working on-site with the program. Another participant (ID: 844) was later removed due to only completing the ASRS questionnaire. It was later discovered that this was another test user. After thorough analysis of the scoring results aiming to identify its importance and potential further use, two more participants (IDs: 1212, 2272) were identified as test users based on irregularities in the scoring results. It was later confirmed that these were also test users, where someone working on the program received a user ID to perform tests as the program was fine tuned and analysed. This makes a total of **109** real participants completing the Pre Mapping phase, and **63** participants completing the Post Mapping phase. It was discovered from this that **46** participants dropped out of the treatment program at some stage. After a meeting with several of the domain experts working on this program, it was of keen interest to them to gain knowledge of dropout rates. The usefulness in knowing this information ahead of time was in order to identify and facilitate a more tailored treatment in an attempt to get this number down and facilitate improvement.

Features

The Bayesian networks were constructed in three main iterations with changes to the selected property features. Tables 1, 2, and 3 display an overview of this to show the changes that were made to achieve more usable and accurate results.

Table 1: Pre meeting features

age_class
gender
education
occupation
pre_kartlegging_time_class
post_kartlegging_time_class
pre_ASRS_score_class
post_ASRS_score_class
pre_GAD-7_score_class
post_GAD-7_score_class
pre_PHQ-9_score_class
post_PHQ-9_score_class
pre_AAQoL_Life_Productivity_score_class
pre_AAQoL_Psychological_Health_score_class
pre_AAQoL_Life_Outlook_score_class
pre_AAQoL_Relationships_score_class
pre_AAQoL_score_class
post_AAQoL_Life_Productivity_score_class
post_AAQoL_Psychological_Health_score_class
post_AAQoL_Life_Outlook_score_class
post_AAQoL_Relationships_score_class
post_AAQoL_score_class
post_AAQoL_score_class
post_PSS-14_score_class
pre_PDQ-5_score_class
post_PDQ-5_score_class

Table 1 includes all data features that were used in the structure learning process in the first iteration that was conducted before consulting with the domain experts.

Table 2: Post meeting features

age_class
gender
education
occupation
pre_kartlegging_time_class
Dropout
ASRS_score_class
GAD-7_score_class
PSS-14_score_class
PHQ-9_score_class
AAQoL_Life_Productivity_score_class
AAQoL_Psychological_Health_score_class
AAQoL_Life_Outlook_score_class
AAQoL_Relationships_score_class
AAQoL_score_class
PDQ-5_score_class

The features that were used in the second iteration, followed directly after meeting with the domain experts, are displayed in Table 2. There is no mention of Pre / Post concerning the various self-report scale properties, as data from the Post Mapping phase were dropped in the development after the consulting with the domain experts.

Table 3: Dropout with ASRS Weekly Modules

Dropout
ASRS_Inactivity_score_class
AASRS_Hyperactivity_score_class
ASRS_week1
ASRS_week2
ASRS_week3
ASRS_week4
GAD-7_score_class
PSS-14_score_class
PHQ-9_score_class
AAQoL_Life_Productivity_score_class
AAQoL_Psychological_Health_score_class
AAQoL_Life_Outlook_score_class
AAQoL_Relationships_score_class
PDQ-5_score_class

Table 3 show the features that were used in the third iteration of this research, where all properties are from the Pre Mapping phase of the treatment program.

The reasoning behind the feature selection in the three iterations was due to various implications and findings that are covered in the remaining sections of this chapter.

4.3.2 Calculating scoring results

Calculation of scoring results was briefly mentioned in Section 4.2.1. This section will provide a more thorough explanation of how the various self-report scales were calculated. An error in the first and second iteration of Bayesian networks was that some of these were wrongly calculated, as simply adding the scores together was not the right procedure to get representative result. This was corrected in the third iteration, and are described in detail below. One of the reasons is that some questions are weighted positive while others are weighted negative, and therefore a need arose to reverse some of these. Low scores display a positive result, while high scoring results depict negative (except from the AAQoL, where a high score is viewed as a positive result and a low score negative).

The Adult ADHD Self-Rating Scale (ASRS) - *Inattention & Hyperactivity + Week 1-4*

The ASRS scale features 18 items that covers the 18 different symptoms for ADHD. In the first iteration, the scores was summed together to a total ASRS score. From using the one full ASRS score of a total 72 points in the first and second iteration, only the two separate subscales were used in iteration three. Since they depict different symptoms of ADHD, this could provide helpful insight in discovering patterns between scoring results and the various participants' progress and behaviour. They both had a total score of 36 points each, and the scores were calculated straight forward since the questions were not weighted differently.

The weekly ASRS questionnaires that were voluntarily answered consisted of 9 questions, and was calculated similarly as above.

AAQoL - *Life Productivity, Psychological Health, Life Outlook, and Relationships*

The AAQoL scale features 29 items that covers health-related quality of life among adults with ADHD during the past two weeks. In the first and second iteration, both a total AAQoL score and the four subscales were including in the Bayesian networks that was constructed. They were all calculated like explained with the ASRS scores above, by summing each item score together to a total score, which was the one used (after being categorized). It was later discovered that this approached was flawed, and the complete process was re-done after further in-depth research was conducted on the self-report tool. One major development was the discovery of weighted questions - which meant that all subscores except Life Outlook had to be reversed before any further calculations could be performed. As stated in the start of this section, a high scoring result in the AAQoL is viewed as a positive result while a low score is negative. For this to be accurate, Life Productivity, Psychological Health, and Relationships had to be

reversed due to the items being weighted negative, while the Life Outlook items being weighted positive. Before reversing scores, each subscale needed to include the associated items. The first approach used question 1-11 in Life Productivity, 12-17 in Pshychological Health, 18-24 in Pshychological Health, and 25-29 in Relationships. After researching the AAQoL in depth, it was discovered that this approach was incorrect and that a specific approach was needed to map the various AAQoL items to their related (Gjervan & Nordahl, 2010). A complete mapping is displayed in table 4.

Table 4: Mapping of AAQoL Subscale Items

AAQoL Subscale	**Item #**										
Life Productivity	1	2	3	4	5	11	22	23	24	25	26
Psychological Health	6	7	8	13	20	21	-	-	-	-	-
Life Outlook	14	15	16	17	27	28	29	-	-	-	-
Relationship	9	10	12	18	19	-	-	-	-	-	-

After being reversed properly, each scale had to be transformed to a 0-100 point scale. With the Likert scale having 5 options, this translates to the following scheme: 1=0; 2=25; 3=50; 4=75; 5=100. This was separately calculated for each subscale using the Pandas library with Python.

When the subscales were mapped correctly, reversed where needed, and transformed accordingly, the final step was to sum the item scores together before dividing them by item account, e.g., Life Productivity which had 11 questions was then divided by 11. The result of this was used as the final scoring result on each of the subscales. A total AAQoL was also calculated when doing this, but was not used in the third iteration. Being that every participant symptom and behavior was tracked through the various subscales, it was deemed that the total score would only confuse the results and partly count the results double.

Perceived Stress Scale (PSS)

The PSS self-report tool measures a subjects' stress and how they think about their lives recently. This scale was also flawed from the first and second iteration of calculating scores and constructing Bayesian networks, as it was calculated by summing together all item scores. Just as the AAQoL, some of the items in the PSS had to be reversed before any scores could be correctly calculated. Remember from earlier that low scores is viewed as a negative result while a high score is viewed as a negative score in all scales except from the AAQoL. In this questionnaire, there were 7 positive weighted questions and 7 negative weighted questions, which meant that the positive weighted items first had to be reversed in order to correctly sum the results to achieve a total score. The mapping of the questions can be viewed in table 5.

Table 5: Mapping of PSS-14 Items

Method	**Item #**						
Reversed scores	4	5	6	7	9	10	13
Calculated regularly	1	2	3	8	11	12	14

Patient Health Questionnaire (PHQ-9)

The PHQ-9, which measures the severity of a person's depression, was calculated the same way as in the first iterations - by summing together the scores of all 9 items.

General Anxiety Disorder (GAD-7)

Calculation of the questionnaire that focus on anxiety while mapping a person's mental health state during the last two weeks, the GAD-7, did not change in the last iteration and was performed by summing together the scores of the 7 items from the Likert scale.

Perceived Deficits Questionnaire (PDQ-5)

The PDQ-5, which aims to assess subjective cognitive dysfunction in people with depression, was calculated by summing together the item scores from the 5 in the questionnaire and was calculated the same way in all iterations.

Time spent - *Pre Mapping*

The datasets of the mapping of ADHD participants included time of first activity and time completed. This was used to calculate a total score for *Time Spent* as it was interesting to see if this could have any impact on any branching relationships and patterns - especially since lack of concentration is common amongst ADHD participants.

4.3.3 Discretization into categories

With total scores calulated, the data processing phase was closer to being ready to build networks. The data was not represented as continues variables, and could be give any real contribution through the implementation of Bayesian networks. The next step was discretization into categories. The distinction between continuous and discrete variables is crisp, but the contrast between continuous and discrete quantities is rather vague, as many quantities often can be represented as both continuous and discrete (BayesFusion, 2020). When variables are presented as discrete, it is usually to provide convenient approximations of real world quantities that provides a sufficient purpose for reasoning. Continuous variables that are represented by discrete approximations between three to five points perform very well in most cases from experience in decision analytic modeling.

Just as the calculation of scoring results, this process went through several iterations as well. The approach of the first and second iterations focused on implementing a mathematical calculation of the point system into five point categories, before being slightly modified to better fit the dataset. The Pandas library in Python was used for finding more suitable categories in the third and final iteration. The goal was to find categories that naturally separated the participants throughout the categories to easier locate any patterns between groups.

First set of iterations

Scores were calculated for both the Pre Mapping phase and the Post Mapping phase, and categories ended up being different in some of the cases within the same self-report tool. Following is a short overview of the process.

ASRS:

The ASRS was answered with a 5 point Likert scale with points: {0, 1, 2, 3 , 4}. Assuming that one would answer the exact same on each question, this could be transferred to a total of: 0 - 18 - 36 - 54 - 72. With a max score of 72, the test subject has 73 possible outcomes when the unlikely score of 0 is included. The score range of 72 divided by the point scale of 5 is 14.6, meaning that there categories would not be divided equally. When this was the case, which was more often than not, two options were considered: (1) placing the biggest category in the middle of the scale where the largest bulk of participants often were placed, or (2) placing the biggest category in the first or last category where there often were very few to none participants, in an attempt to catch outliers and anomalies. This process ended with the following ASRS categories after the first and second iterations:

Pre Mapping: 0-14; 15-28; 29-42; 43-56; 57-72.
Post Mapping: 0-9; 10-27; 28-44; 45-61; 62-72.

As the scoring results usually varied a lot from the Pre to Post Mapping phase, this resulted in different categories after the model had been adapted to better fit the data.

AAQoL - Total:

The AAQoL was answered with a 5 point Likert scale with points: {1, 2, 3, 4, 5}, and with the same assumption as with the ASRS this could be transferred to a total of: 29 - 58 - 87 - 116 - 145. These categories could also not be divided equally, and as this was the largest scale by far, it needed some adjusting to better fit the data to be able to provide any meaningful contribution. The Pre Mapping included 56 as the lowest total score with 117 as the highest. The Post Mapping was quite similar in this condition, with 57 as the lowest and 122 as the highest score. This resulted in the following categories:

Pre/Post: < 60; 60-76; 77-93; 94-110; > 110.

In this case there was little meaning in following a strict mathematical approach, as very few

scored close to the minimum and maximum score, which led the categories to be clustered in a smaller point range.

AAQoL - Life Productivity:

The Life Productivity subscale was answered with points: {1, 2, 3, 4, 5} –> 11 - 22 - 33 - 44 - 55. This provided the opportunity for evenly divided categories. The Pre Mapping had a lowest score of 15 and a highest score of 51, while Post Mapping had 15 and 46 respectively. The following categories was used in in the first two iteration of building Bayesian networks:

Pre/Post: 11-19; 20-28; 29-37; 38-46; 47-55.

The highest category (47-55) in the Post Mapping dataset had no hits due to the highest score being 46. This was considered when new categories were implemented in the third iteration.

AAQoL - Psychological Health:

The Psychological Health was also answered with points: {1, 2, 3, 4 , 5} –> 6 - 12 - 18 - 24 - 30. Pre Mapping included a lowest score by the subjects of 10, with 27 the highest. These numbers were 6 (lowest) and 26 (highest) from the Post Mapping phase. These categories were used:

Pre/Post: 6-10; 11-15; 16-20; 21-25; 26-30.

AAQoL - Life Outlook:

Life Outlook had the following point scale: {1, 2, 3, 4, 5} –> 7 - 14 - 21 - 28 - 35. Pre Mapping showed a low score of 10 and 34 as high, while 11 and 33 was the case from the Post Mapping study. It resulted in these initial categories:

Pre/Post: 7-11; 12-17; 18-23; 24-29; 30-35.

AAQoL - Relationships:

The Relationships subscale had the point scale: {1, 2, 3, 4, 5} –> 5 - 10 - 15 - 20 - 25. Low point in Pre Mapping was 9, with 24 as the highest result. 10 was the lowest and 23 the highest score from Post Mapping. The resulting categories was the following:

Pre/Post: < 11; 11-14; 15-18; 19-22; 23-25.

PSS:

The PSS-14 self-report tool was answered with points: {0, 1, 2, 3, 4} –> 0 - 14 - 28 - 42 - 56. The following categories was initially tested out: 0-7; 8-21; 22-34; 35-48; 49-56. The result of this turned out to be a model where neither the first nor the last category had any hits. This was then changed and the following was used in the first and second iterations:

Pre Mapping: < 24; 24-29; 30-35; 36-41; > 41.

Post Mapping: < 20; 20-26; 27-34; 35-41; > 41.

PHQ-9:

It was discovered during data processing that the PHQ-9 questionnaire had an inconsistency error from when the Pre Mapping was conducted, to how it was scored when the Post Mapping phase was completed. The Pre Mapping had a point scoring on a 4-point Likert scale from: {1, 2, 3, 4} –> 9 - 18 - 27 - 36. The Post Mapping point scoring was answered with the following: {0, 1, 2, 3} –> 0 - 9 - 18 - 27. The categories used in the first and second iterations showed noticeable differences due to this. This was reported back to the domain experts working on the treatment program, and was corrected in the third iteration. This was especially important with regards to differentiating the severity of depression from 4.2.2 on the PHQ-9. The following categories were used in early stages:

Pre Mapping: 9-12; 13-17; 18-22; 23-26; > 26.
Post Mapping: 0-4; 5-9; 10-14; 15-19; > 19.

GAD-7:

The GAD-7 questionnaire, much like the PHQ-9, also used a 4-point Likert scale. The same inconsistency error was found with scores from det Pre Mapping being: {1, 2, 3, 4} (1-28 scale). The questionnaire from the Post Mapping with the same items used the following: {0, 1, 2, 3} (0-21 scale). The following categories were used:

Pre Mapping: 7-10; 11-18; 19-24; 25-28.
Post Mapping: 0-3; 4-7; 8-11; 12-15; > 15.

The difference in scoring from Pre to Post Mapping was corrected for iteration three.

PDQ-5:

The GAD-7 used a 5-point Likert scale with the following point system: {1, 2, 3, 4, 5} –> 5 - 10 - 15 - 20 - 25. The categories used in this phase were quite similar, but the first category in the Pre Mapping dataset was adjusted some to better fit the data. This resulted in the following categories:

Pre Mapping: 5-9; 10-13; 14-17; 18-21; 22-25.
Post Mapping: 5-8; 9-12; 13-17; 18-21; 22-25.

Iteration 3: Using Pandas for discretization

The implementation of Pandas was done to enhance the precision of the scoring categories that the subjects of the treatment program was grouped in to better capture patterns and tendencies. This new approach better captured this by finding categories that provided a natural spread of participants across categories. This often meant that the three middle categories had the largest bulk of subjects, while only a few were grouped in the first and last categories - marking themselves as anomalies of some degree.

Some of the surveys did not need to have their scoring categories calculated, as further research revealed that they had predefined cut-offs that help to separate the severeness of various symptoms and illness, e.g., the ASRS subscales, which are divided into three categories depicting "unlikely to have "ADHD", "likely to have ADHD", and "highly likely to have ADHD". The two other surveys that had predefined cut-off points were the PHQ-9 and the GAD-7. An overview of the different categories will be displayed below.

ASRS:

As introduced in Section 4.2.1, the ASRS have the following cut-offs: a score of 0-16 means unlikely to have ADHD, 17-23 means that the subject is likely to have ADHD, and 24-36 means highly likely to have ADHD (*An Internet-delivered Intervention for Coping With ADHD in Adulthood (MyADHD)*, n.d.). This would mean only three categories to differ the scoring results for both Hyperactivity and Inattention in the ASRS. It was decided to split the last category (24-36) in two to better map tendencies for those who fall on the highest scores stretching up to 36 from the people on the low end of this. The new categories for the ASRS was as follows:

Hyperactivity / Inattention + Week 1-4: 0-16; 17-23; 24-29; 30-36.

An overview of how the participant spread were amongst the various categories are displayed in Table 6.

Table 6: Participant Spread: ASRS Categories

Category	Hyperactivity Spread	Hyperactivity %	Inattention Spread	Inattention %
1	4	4%	17	16%
2	16	15%	41	38 %
3	62	57%	35	32%
4	27	25%	16	15%

AAQoL:

There was a considerable change in the calculation of the AAQoL scores from the first two iterations to iteration three, as questions were previously not allocated to the correct subscales, and the scores had not been reversed and transformed into a 0-100 point scale and divided on item count earlier. The items (question number) are displayed under the correct subscales in Table 4.

The AAQoL was divided into five categories in the 0-100 scale, with the following scoring categories for all subscales:

AAQoL: 0-25; 26-41; 42-58; 59-74; 75-100

An overview of the participant spread throughout the subscales of the AAQoL categories are presented in Table 7.

Table 7: Participant Spread: AAQoL Categories

Category	Life Productivity	Psychological Health	Life Outlook	Relationships
1	2 (2%)	3 (3%)	8 (7%)	8 (7%)
2	5 (5%)	10 (9%)	19 (17%)	14 (13%)
3	33 (30%)	49 (45%)	51 (47%)	40 (37%)
4	52 (48%)	26 (24%)	24 (22%)	35 (32%)
5	17 (16%)	21 (19%)	7 (6%)	12 (11%)

PSS:

After reversing the positively weighted questions (4, 5, 6, 7, 9, 10, 13) in the PSS-14, new categories were calculated through the use of the Pandas library:

PSS: 0-19; 20-26; 27-34; 35-41; 42-56.

The participant spread in the PSS-14 categories are shown in Table 8.

Table 8: Participant Spread: PSS-14 Categories

Category	Spread Count	Spread Percentage
1	6	6%
2	19	17%
3	49	45%
4	31	28%
5	4	4%

PHQ-9:

Before the new categories were set, the scores from the Pre Mapping were adjusted to be equal to the scoring system from the Post Mapping. The categories remained the same as from the Post Mapping in the first and second iterations since this matched the cut-offs. These categories are meant to represent where a person lies with regards to mild depression (score: 5), moderate depression (score: 10), moderately severe depression (15) and severe depression (20). This means the following categories were used:

PHQ-9: 0-4; 5-9; 10-14; 15-19; 20-27.

The participant spread throughout the categories from the PHQ-9 self-report tool are presented in Table 9.

Table 9: Participant Spread: PHQ-9 Categories

Category	Spread Count	Spread Percentage
1	8	7%
2	35	32%
3	44	40%
4	18	17%
5	4	4%

GAD-7:

As previously stated, the GAD-7 also had predefined cut-offs with regards to a person's scoring result. These were scores of 5, 10, and 15 - to map mild, moderate, and severe anxiety. These cut-offs were decided to be the borderlines of the various categories. Just as with the ASRS, the category with the highest scores was divided into two categories to be able to differentiate the people with the most extreme results. This resulted in the following categories:

GAD-7: 0-14; 5-9; 10-14; 15-17; 18-21.

An overview of how the participant spread were distributed amongst the categories used for the GAD-7 are displayed in Table 10.

Table 10: Participant Spread: GAD-7 Categories

Category	Spread Count	Spread Percentage
1	14	13%
2	49	45%
3	28	26%
4	11	10%
5	7	6%

PDQ-5:

There was no need to calculate new categories for the PDQ-5 self-report tool. This means that the following categories were used for the rest of the experiments:

PDQ-5: 5-9; 10-13; 14-17; 18-21; 22-25.

Table 11: Participant Spread: PDQ-5 Categories

Category	Spread Count	Spread Percentage
1	0	0%
2	19	17%
3	52	48%
4	23	21%
5	15	14%

The participant spread throughout the categories from the PDQ-5 self-report tool are presented in Table 11.

4.3.4 Splitting into different datasets

The first iteration of Bayesian networks were made with a dataset consisting of the Pre Mapping and Post Mapping (without modules from week 1 through week 4). The results were not that accurate when measured through different metrics discussed in Chapter 3, but still showed great potential in mapping participant behaviour based on previous results. When this was showed to the domain expert working directly on the treatment program and its participants, it received positive feedback in having a good potential. The networks shown at this time showed predictions for the Post Mapping phase when the scores from the Pre Mapping was used as evidence in the models. It was raised a desire to implement a dropout rate in future models, as this was pressed as the most important property to monitor due to the number of participants never completing the program. Unstructured data of weekly ASRS questionnaires were noticed and labeled correctly to also contribute to more accurate models. This lead to splitting the data into three datasets: (1) Pre/Post Mapping with old categories, no dropout rate, and without weekly modules, (2) Pre Mapping with dropout, and (3) Pre Mapping with new categories, dropout, and weekly modules.

The dataset with Pre Mapping data, new categories, and dropout (3) was chosen to be used in the last experiments to meet with the requirements that the domain experts deemed as important. This was the data that would simulate what data would be available during the course of the treatment program within the time frame where measures could be acted upon, in order to hopefully prevent red-listed participants from dropping out of the program. The Post Mapping data was excluded in these experiments because the tests were conducted after those participants had already chosen to quit the program.

5 Network Development

This chapter will focus on how the development of Bayesian networks were planned out and executed. This will make it more clear how the data affected both the structure of the networks as well as the development approach. Several types and versions of Bayesian networks were constructed and tested (Section 3.2.1) during the entire development process, and consisted of three main iterations:

- **Initial Testing** - Exploring usefulness and demonstration purposes.
- **Post Expert Meeting** - Adjusting to meet case specific preferences.
- **New Categories, Scoring Calculations, and Weekly ASRS** - Correct calculations, more accurate categories, and the addition of weekly ASRS scores.

5.1 Iteration 1: Initial Testing

The main goal in the first iteration was to get a thorough understanding of Bayesian networks, its ground principles and its foundations. When working with medical data, and specifically a treatment program for people with ADHD with no prior knowledge to the field, it can pose a challenge not only to effectively find important connections and context, but also to identify actual needs as well. Due to this, the main focus was becoming comfortable with constructing various networks based on needs, to be able to enlighten the domain experts with the usefulness of implementing Bayesian networks as a decision making tool when historical data is available. This would enable said experts to provide case specific context and feedback to further build on.

The structure learning algorithms mentioned in Section 3.2.1 were constructed based on the data being used at that time. Two of them were selected to be presented as examples to the domain experts: Bayesian Search and Greedy Thick Thinning. Both of these networks were constructed with forced arcs, as the algorithms did not effectively identify enough accurate patterns at the given time.

5.1.1 Bayesian Search

As described in Section 3.2.1, Bayesian Search is one of the earliest and most popular algorithms for structure learning, and uses the log likelihood function guided by a scoring heuristic. Data from both Pre Mapping and Post Mapping were used to demonstrate how results from the initial tests can be used to get an indication to how a specific person will progress towards the end of the program. The data exclusively consisted of 63 data rows, as only properties that were available for all of the participants were included - namely those who had completed both the Pre Mapping and the Post Mapping surveys.

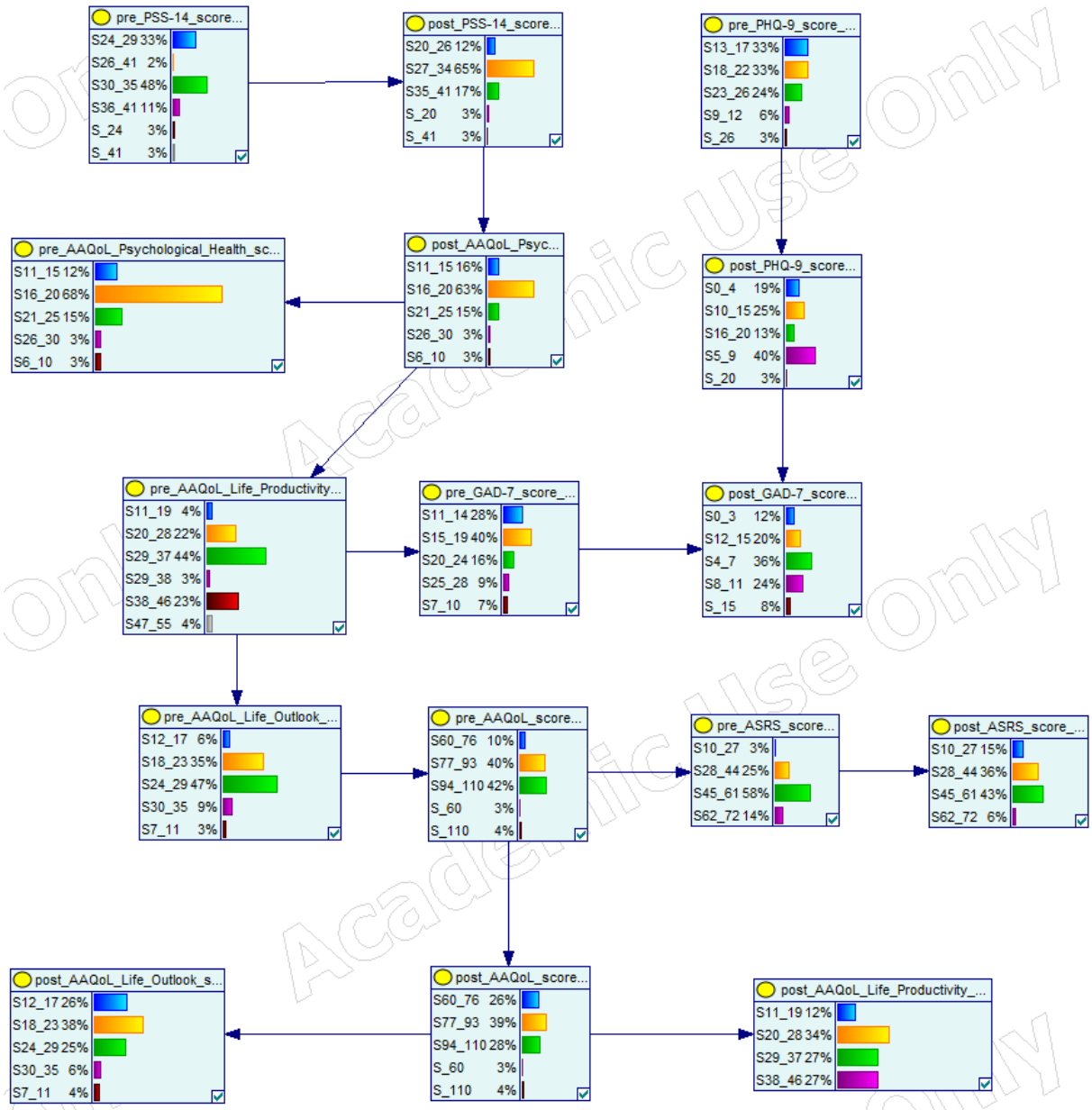

Figure 6: First Iteration Bayesian Search

Figure 6 shows how the structure of the Bayesian network that was created through Bayesian Search. It consists of 15 arcs between 16 nodes in total; 8 of these arcs were forced, meaning that the structure learning algorithm identified 7 arcs. The forced arcs were put between the correlated Pre/Post test, e.g., from Pre Mapping ASRS to Post Mapping ASRS etc. The nodes are from most of the various surveys each person had to take in both Pre Mapping and Post Mapping: ASRS, AAQoL, AAQoL Life Productivity, AAQoL Psychological Health, AAQoL Life Outlook, PSS-14, PHQ-9, and GAD-7. This means that the structure learning algorithm did not find any patterns that included either the AAQoL Relationships subscale or the PDQ-5. Further, no structural patterns could be found to connect any of the demographic properties, such as age, gender, education, or occupation, to the rest of the model. The same can be said about time spent on the tests. Max parent count was set to 8, the sample size to 50, and the search went through 20 iterations before a sufficient structure was found. The seed was set to 0, meaning that the seeding is random and not repeatable. Elapsed time was 5.031 seconds before the Bayesian Search concluded with the given structure.

The different categories can be seen under each node title in Figure 6, with prior probabilities of any given person scoring within that category when no evidence is present. This alone can give insight to tendencies for adults diagnosed with ADHD, but as mentioned in Chapter 2 (Sections 2.3.8 and 2.3.1), the true advantage of Bayesian networks appear when updating probability through evidence. This will be discussed in more detail in Chapter 6.

5.1.2 Greedy Thick Thinning

The Greedy Thick Thinning is based on the Bayesian Search approach, but differs as it is split into several phases - including a thickening and a thinning phase. This structure learning algorithm was applied to the same dataset as the Bayesian Search network and produced a different network structure as a result.

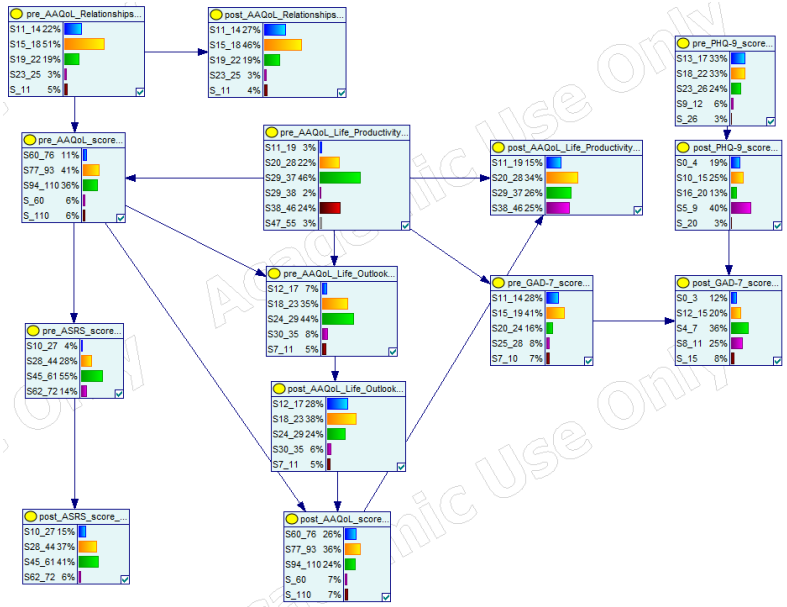

Figure 7: First Iteration Greedy Thick Thinning

It consists of 16 arcs between 14 nodes in total. The algorithm identified 5 arcs while the remaining 11 arcs were forced before applying Greedy Thick Thinning, which was done the same way as with the Bayesian Search. Out of the 10 different scales (including both total AAQoL and subscales), no connections were found that linked the AAQoL Psychological Health, PSS-14, or the GAD-5 to the rest of the model. Max parent count was set to 8, and elapsed time before finding the structure was 0.047 seconds. One can clearly notice that the structure is quite different even though the same data was used for both networks. A walk through of how these networks were used to demonstrate results and usefulness when talking to domain experts can be viewed in Chapter 6 which describes the results of this thesis.

5.2 Iteration 2: Post Expert Meeting

It was revealed during the expert meeting that the participants who did not complete the Post Mapping surveys included every enrolled subject that dropped out of the treatment program somewhere between Pre Mapping and Post Mapping. It was of keen interest to include these in future experiments and attempt to predict whether a person is likely to drop out or not. This was pressed as the biggest concern, as close to half (44%) of the people who enrolled in the program did not complete it. When further work was done to meet this request, the Post Mapping data was primarily used to identify which of the participants completed the program. The scoring results was not used when developing the networks, as those who dropped out had already done it at this point.

The first step when constructing networks with the various structure learning algorithms was to try it without any background knowledge or forced arcs, to see what patterns the algorithms would find on its own. The next step was to recreate this with any expert knowledge or edu-

cated guesses to see if the results improved or decreased. The second iteration proposed some difficulties, as structural patterns to the dropout property needed to be manually added through forced arcs in most of the networks.

5.2.1 Tree Augmented Naive Bayes

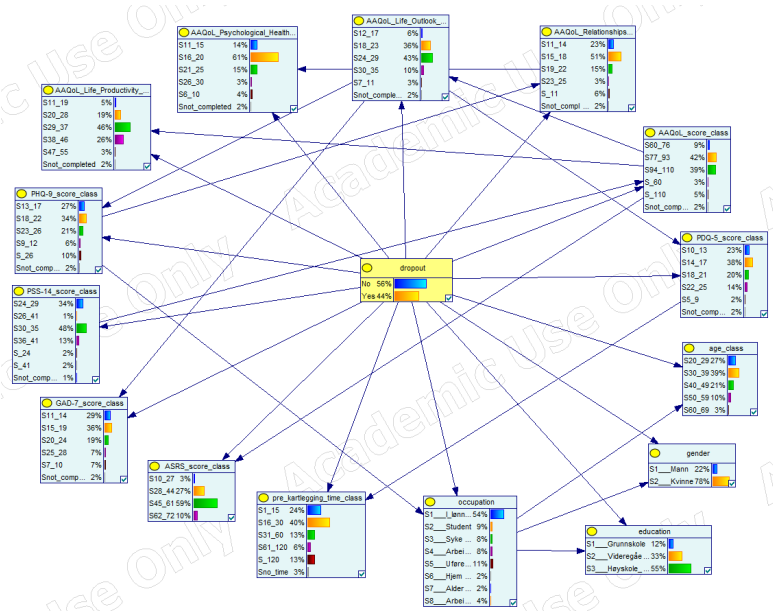

Figure 8: Post Expert Meeting - Tree Augmented Naive Bayes with Dropout

Tree Augmented Naive Bayes (Figure 8) was the structure learning algorithm that stood out the most during the second iteration. As mentioned in Section 3.2.1, TAN is conditional on one single class variable, and the best results for predicting dropout among participants at this stage was achieved by choosing Dropout as the class variable. When practising this, the class variable automatically receives an arc to each child node in the network, and the structure learning algorithm then tries to find any other patterns between the child nodes. The network in Figure 8 has 29 arcs in total. Every survey that that was described in Section 4.2.1 was included in this network. At this stage, the ASRS was still represented by one total score, just as the AAQoL still included a total score in addition to the subscales. Time spent, and demographic properties such as age, gender, education, and occupation, was also included in this network. This adds up to 15 child nodes in total, meaning that out of the 29 arcs in total, 14 of them were patterns found by the algorithm. Total elapsed time was merely 0.031 seconds before the structure was completed. The network was based on 112 data rows in total, which from Section 4.3.1 is three more than the amount of participants completing the Pre Mapping phase. The reason being that the three last test participants was still to be discovered and excluded.

5.3 Iteration 3: New Categories, Scoring Calculations, and Weekly ASRS

This subsection covers the third and final iteration of network development in this thesis. Considerable changes were made in the datasets used between the first and the last iterations. The effect of those changes will mainly be covered in Chapter 6, but the changes that were made substantially improved the quality of the results. The most noticeable addition was the inclusion of weekly ASRS result scores that were taken throughout the program, backed up by scoring calculation and categories that were that could be used with confidence with regards to credibility. From the semi-structured data, it was discovered that this was conducted in two week intervals, meaning that "ASRS Week 1" was answered in the second week, "ASRS Week 2" in the fourth week, "ASRS Week 3" om the sixth week, and finally "ASRS Week 4" in the eight week. Total scores for the ASRS and AAQoL were not used at this stage, as it was decided that the subscale scores covered this information. One can also see from Table 3 that in addition to removing Post Mapping data, demographic data and time spent were excluded from further network development as no relations were found to connect it to any of the models.

5.3.1 Naive Bayes

The Naive Bayes algorithm uses a fixed assumption instead of actually learning the structure of a Bayesian network. Just as the TAN and ABN, it relies on just one class variable with the rest becoming child nodes to the class variable. The Bayesian network shown in Figure 9 use the Inattention subscale of the ASRS as its class variable, as this proved to generate the most accurate results through testing out all options.

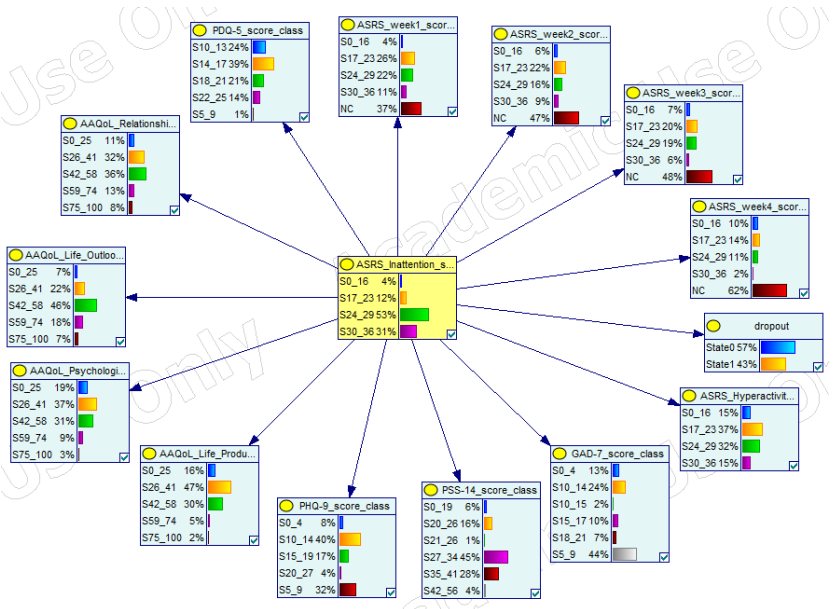

Figure 9: Weekly ASRS Included - Naive Bayes

The network consists of 15 nodes in total, which means that there are 14 arcs because the class variable has a relationship to each child node in the directed acyclic graph. The dataset used for

structure learning had 109 rows, and at this point all test participants had been removed. The algorithm is relatively fast, and was completed after an elapsed time of 0.046 seconds.

5.3.2 Augmented Naive Bayes

The Bayesian network learned from Augmented Naive Bayes that performed the best included more complexity than the regular Naive Bayes, but not as much as the Tree Augmented Naive Bayes produced. After choosing a class variable and adding background knowledge, like temporal tiers for feature various feature variables, the structure learning algorithm produced 20 arcs between the 15 nodes. That means that 6 relationships were identified by the algorithm, which can be seen in Figure 10. The relationships found by the algorithm can be located between child nodes, e.g., the arc going from ASRS Week 2 to Dropout. This relationship means that the posterior probability of a person dropping out is highly affected by evidence being entered from the ASRS Week 2.

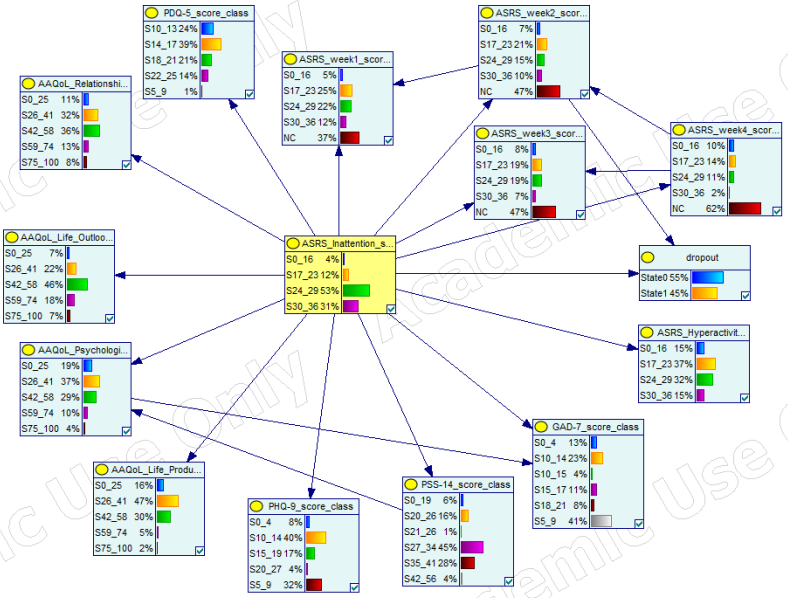

Figure 10: Weekly ASRS Included - Augmented Naive Bayes

Default search settings were used when running this learning algorithm, which were the case through most of the iterations. Max parent count was set to 8, but there were not any risk of this being challenged as the actual max parent count after structure learning was 2. The search used 20 iterations from a sample size of 50, and seed 0 which is used for random execution. The seed count is only specified when one wants the ability to reproduce the exact same structure learning result. With regards to computation time, this was one of the slower learning processes as elapsed time was 5.078 seconds. When assessing the time spent, it is important to consider the size of the dataset used for structure leaning (109 rows), as this can easily scale up with large datasets.

5.3.3 Tree Augmented Naive Bayes

Just as the Naive Bayes and Augmented Naive Bayes, the class variable that performed the best was the ASRS Inattention subscale scores, and became the parent node to the rest of the nodes in the network. Since it receives an arc to every child node, this means that every property (data column) automatically gets included in the final network structure. This version, which was the network learned from the Tree Augmented Naive Bayes that achieved the best result, included 15 nodes. It ended up having one of the most complex structures, as there were 27 arcs (edges) in total. The highest edge count of any node except the parent node was 5, which occurred for the GAD-7 (3 out and 2 in) and the ASRS Week 4 (3 out and 2 in). The network structure can be viewed in Figure 11.

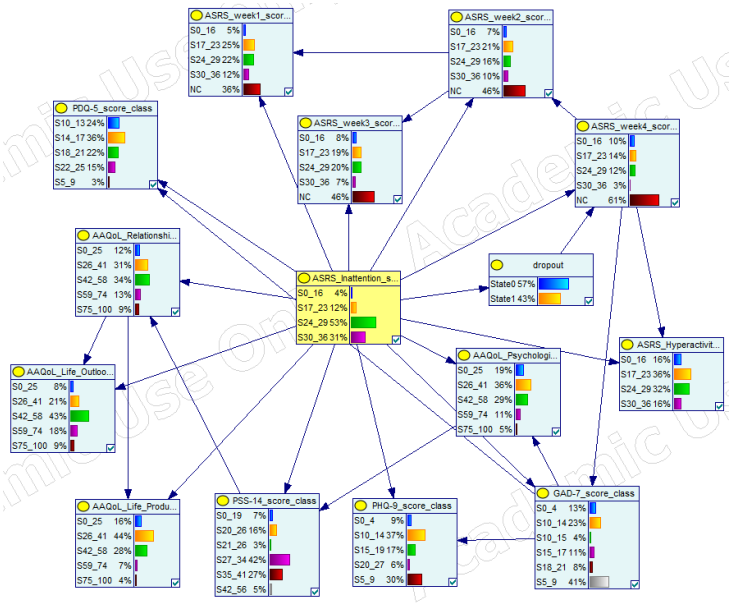

Figure 11: Weekly ASRS Included - Tree Augmented Naive Bayes

Even though the network was one of the more complex with regards to structure, it was extremely fast and was completed with an elapsed time of only 0.016 seconds. This was based on 109 data rows like the other networks. Temporal tiers were of great use to tell the structure learning algorithm which events occurred at a later stage in the process, especially for the ASRS Week 1-4. This helped both with learning the structure and to achieve more accurate results, which is described more in dept in Chapter 6.

5.3.4 Bayesian Search and Greedy Thick Thinning

The last two structure learning algorithms that went through testing were the Bayesian Search and the Greedy Thick Thinning algorithm. This ended up being quite unique as they produced the same structures, even though they were developed through two different approaches. These algorithms do not rely on one single class variable that becomes parent to the rest of the features that are included in the final network structure. This means that some of the features might fall

outside the network if no relationships are found to include them. This happened in every iteration with various results. The special thing about the third iteration was that both these algorithms produced a structure that included Dropout with only five nodes in total. This was the four weekly ASRS scoring results and the dropout rate. Another structure included 8 of the other features, but was not used further as it did not include the Dropout - this was also exactly the same through both Bayesian Search and Greedy Thick Thinning. The rest were either just two nodes, or single nodes with no patterns found. Figure 12 displays how this structure looks. Only the structure containing the Dropout node can be viewed as bar chart, while the rest are are displayed as icons.

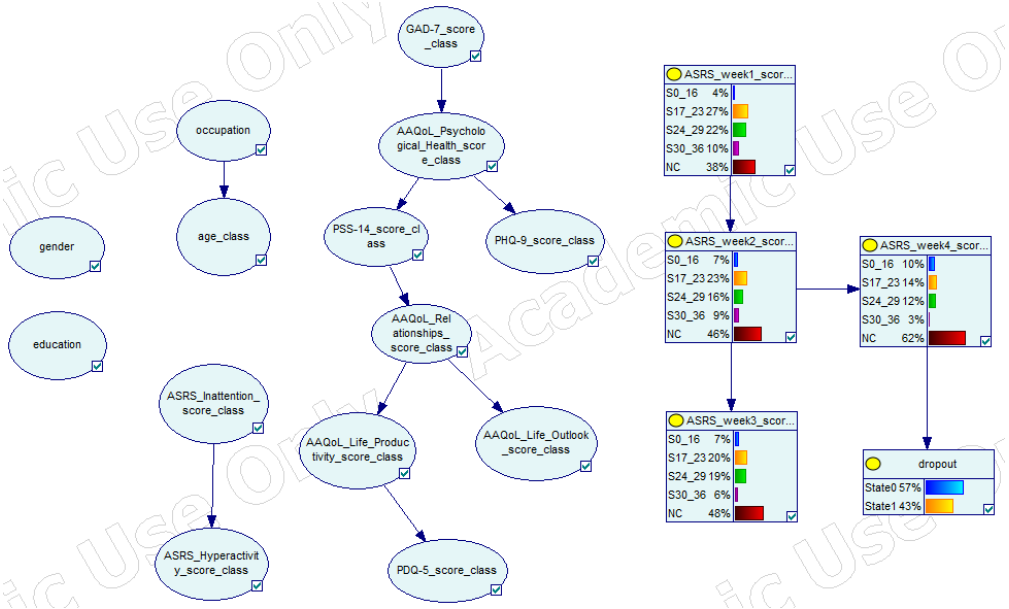

Figure 12: Weekly ASRS Included - Bayesian Search / Greedy Thick Thinning

The Bayesian Search was the slowest algorithm to finish, taking 0.359 seconds. Elapsed time for Greedy Thick Thinning was much faster with only 0.031 seconds. Even though they were both relatively fast, computation time should be taken into consideration when working with big data and large datasets. Max parent count were set to 8 for both algorithms, but this was never close to being challenged. Only Bayesian Search specifies sample size and number of iterations, which were 50 and 20 respectively. The number of nodes assigned to temporal tiers was 14 out of 19 total. To represent this, five temporal tiers were used to represent five different stages that the data came from (Pre Mapping, ASRS Week 1, ASRS Week 2, ASRS Week 3, and ASRS Week 4). Figure 13 exclusively displays the structure including the Dropout rate, which was the relevant one for this research. Not only was the same network structure produced through Bayesian Search and Greedy Thick Thinning, but the result was also identical as those of another network structure on one particular point. A further explanation of this can be found in the Chapter 6.

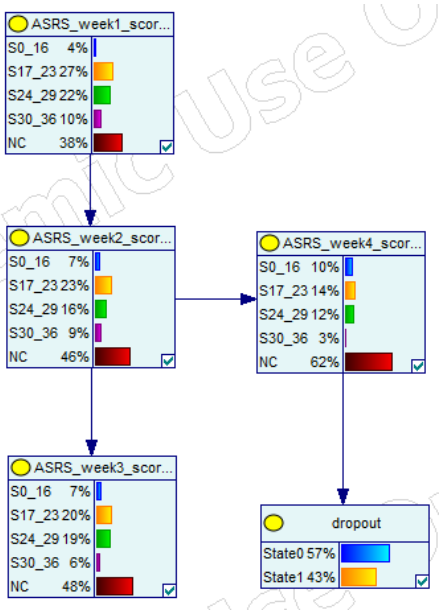

Figure 13: Weekly ASRS Included - Bayesian Search / Greedy Thick Thinning dropout structure

This chapter has provided an overview of how the network development evolved as this research went from its first iteration to the final iteration. It has also served a purpose in showcasing how the development was conducted and given an insight into various aspects that affected the decision making process. Some hints to the results were included for demonstration purposes to effectively achieve this, and the next chapter builds on this to present a complete picture of the findings of this thesis.

6 Results

This chapter will cover the results of the research. The key takeaways from each iteration will be examined to provide insight to how this was used and how it contributed to consequent changes, further development, before landing on the final results. The results was validated against area specific standards, and as to what they meant for the treatment program that is the focus of this thesis.

6.1 Results from Iteration 1: Initial Testing

The results with regards to actual scoring metrics were not too important during the initial testing, as the focus was to present a demonstration to the domain experts that would demonstrate the potential of implementing Bayesian networks. To successfully achieve this, it was important to provide an understandable overview of its area of use and how it can provide valuable contributions to aid the decision making process. This process is often completed based off of expertise knowledge and the experience of the domain experts in question. The feedback and appeals towards wanted results gained from this meeting was the most valuable results in this process.

6.1.1 Usage Demonstration

As mentioned in Section 5.1, it was decided to present two different Bayesian networks during the expert meeting, to express that there is a lot of flexibility and scalability when working with this kind of technology. Actual scoring metrics was not a priority in this phase. The reason for this is that further insight and understanding of the treatment program and its consequential features and properties were needed in order to be able to produce this.

Greedy Thick Thinning

One of the Bayesian networks that was demonstrated was learned through the Greedy Thick Thinning structure learning algorithm. It displays prior probabilities that maps out where a randomly given person would score when no evidence is inserted, both in the Pre Mapping and the Post Mapping phase. These probabilities can be seen in Figure 7 in Chapter 5. For further demonstration, a participant was picked at random whose scoring results were available. These results were used to insert evidence in the nodes from the Pre Mapping phase, which would then update the model to deliver posterior probabilities pointing to how this participant would score in the Post Mapping phase at the end of the treatment program.

Figure 14 shows how the model from Figure 7 updated when knowledge input from the Pre Mapping phase was added. The bars displaying 100% probabilities from inside the bar charts represent the inserted evidence that was known to be true. The remaining probabilities are from

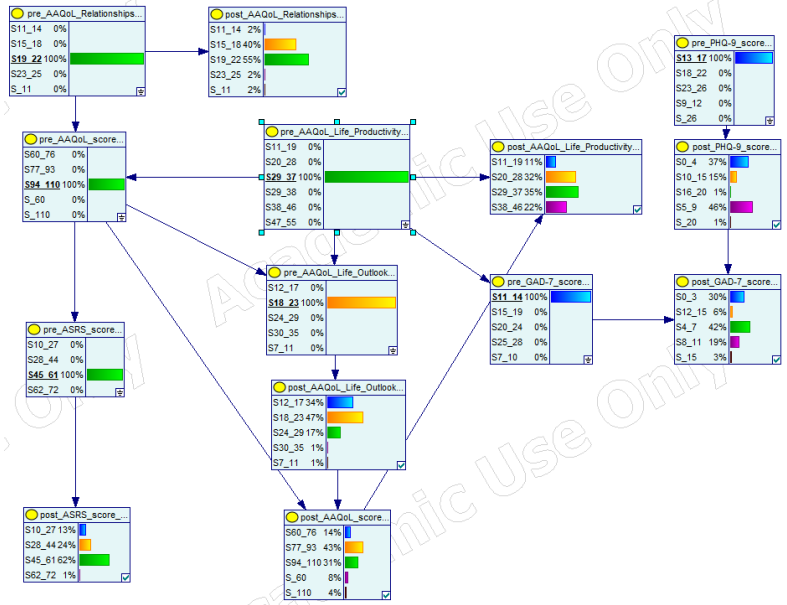

Figure 14: Greedy Thick Thinning: Posterior Probabilities After Inserting Evidence

nodes that were tied to the Post Mapping phase of the treatment program. Since a participant whose scoring results were available was chosen, the updated model could now be put up to a comparison between the actual results from that participant and the predicted output from the model. When this model was created, it had forced arcs between Pre/Post activities inserted through the knowledge editor, but the structure learning model also found some hidden patterns in the model. As mentioned in Section 5.1.2, these relationships were identified between:

- Patient Health Questionnaire (PHQ-9) *Post Mapping* and Generalized Anxiety Disorder (GAD-7) *Post Mapping*
- ADHD Quality of Life Measure (AAQoL) subsale Life Productivity *Pre Mapping* and GAD-7 *Pre Mapping*
- AAQoL *Pre Mapping* and AAQoL Life Productivity *Pre Mapping*
- AAQoL *Post Mapping* and AAQoL Life Productivity *Post Mapping*
- AAQoL *Pre Mapping* and AAQoL Life Outlook *Pre Mapping*
- AAQoL *Post Mapping* and AAQoL Life Outlook *Post Mapping*
- AAQoL *Pre Mapping* and AAQoL Relationships *Pre Mapping*
- AAQoL *Pre Mapping* and ASRS *Pre Mapping*
- AAQoL Life Productivity *Pre Mapping* and AAQoL Life Outlook *Pre Mapping*

All those relationships were involved in producing a model that can predict patterns based on historic data, and to be updated through evidence insertion. The relationships between the total

AAQoL scale and the various AAQoL subscales were expected as they came from the same self-report tool, but there were also some that might not have been as expected. Especially the pattern between the GAD-7, which focus is dealing with anxiety, and the AAQoL Life Productivity was an interesting connection. The relationship between depression (PHQ-9) and anxiety (GAD-7) was also interesting, but expected to some degree.

When tested with a random participant for demonstration purposes, it was found that 5 out of 7 metrics were correctly predicted inside the highest predicted score category. One of the metrics that was not in the highest predicted score category, the AAQoL Post Mapping, was in the second highest. Even though the model predicted the person to score inside of the 77-93 score category with a 43% probability, it also showed that there were a 31% chance that the given person would score inside of the 94-110 score category. Both of these categories are still a good indication to where on the scale this specific person would be.

The last prediction that did not match the real score was on the PHQ-9 scale, where the actual score of this person was located in the third most predicted category. It should be mentioned that the score was only one scoring point away from falling into the highest predicted category of 5-9, as the confirmed score on the PHQ-9 was 10.

Bayesian Search

The other Bayesian network that was developed for demonstration purposes was learned through the Bayesian Search structure learning algorithm. It was made with the same approach as the Greedy Thick Thinning network, and it was found that this model correctly predicted 5 out of 8 metrics inside its highest predicted score category.

6.1.2 Validation

Even though validation through metrics such as accuracy were not too important in this iteration, it was still of interest to examine how these results scored at this point. This could then be used to contemplate what and how big changes were needed in order to elevate the results to a satisfactory level.

Accuracy: Greedy Thick Thinning

This model used both Pre Mapping and Post Mapping data to learn the structure and validate the network. The intended use of a model like this was to have a model showing indications to where the average person would score, but more importantly to be used as a support tool where one could insert evidence underway in the treatment program. This would be done by inserting what is known from the Pre Mapping phase, as shown in the demonstration, and access how the various participants would score (within high probability) towards the end to be able to make tailored decisions. Due to this fact, only the nodes concerning the Post Mapping scores were

used as target class variables when validating prediction accuracy. The total Accuracy for all 7 concerned nodes were 0.44, or 44%, where 195 out of 441 outcomes were predicted correctly. A correct prediction is when the algorithm predicts the exact correct scoring category out of the the 5 categories.

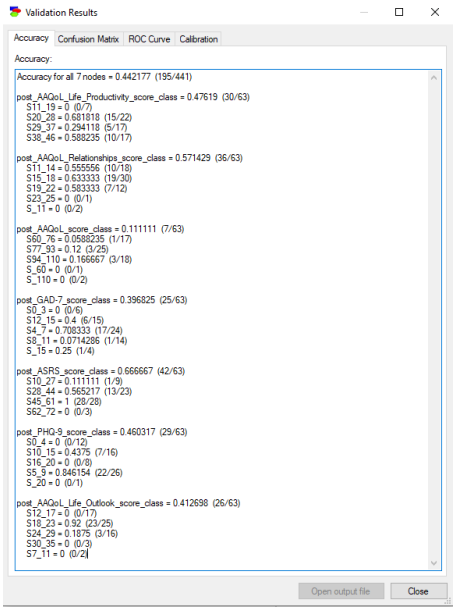

Figure 15: Validation Greedy Thick Thinning: Accuracy Post Mapping Nodes

The area where this specific model struggled the most to predict the outcome was on the total AAQoL scale, where the accuracy was only 0.11. The strongest feature of this model was on the other hand predicting the ASRS scale, where the accuracy was 0.67.

It is important to keep one thing in mind when validating these predictions - the prediction is ranged on a Likert scale (1-5) instead of yes/no. A 50% prediction accuracy is more precise in the first case than with 50/50 cases. This is because the model can still predict the correct outcome with a 35% probability next to a higher predicted outcome with 40% probability. The model will predict wrong in terms of accuracy, but can in reality be very useful as it gives a good indication of the possible outcomes.

Confusion Matrix: Greedy Thick Thinning

The confusion matrix can be used to look more closely into the weaknesses and strong points of the current model. This is a more in dept analysis of the results shown in Figure 15. There is one confusion matrix for each class node, displaying not only how many of the total cases were predicted correctly for every category, but also what the model predicted the outcome to be in those cases were the prediction was incorrect.

Figure 16 displays the confusion matrix of the Post Mapping ASRS scale, which is the most accurate predicted scale in this specific model. It was known from Figure 15 that only 1 out of

		Predicted			
		S10_27	S28_44	S45_61	S62_72
Actual	S10_27	**1**	3	5	0
	S28_44	0	**13**	10	0
	S45_61	0	0	**28**	0
	S62_72	0	0	3	**0**

Figure 16: Validation Greedy Thick Thinning: ASRS Confusion Matrix

9 cases where a person scores in the lowest category of 10-27 points in the ASRS was correctly predicted. The confusion matrix could then provide further information which showed that the Bayesian network predicted the the next lowest category (28-44 points) in three of the cases, and the next highest category (45-61 points) in 5 of the outcomes where the prediction was incorrect. The confusion matrix could further tell that even though the model correctly predicted the outcome in all 28 outcomes where a person scored in the second highest category (45-61), it actually predicted that specific category in 46 out of 63 total outcomes.

Accuracy: Bayesian Search

The validation process was the same for both networks, and Leave One Out was used as the preferred validation method in both cases. The network learned from Bayesian Search had an accuracy on all target class nodes of 0.47, where 239 out of 504 outcomes were predicted correctly. It scored the lowest when predicting the total AAQoL scale, same as the Greedy Thick Thinning, but with a slightly better accuracy of 0.21. This model scored the best when predicting the PSS-14 scale with an accuracy of 0.68.

6.1.3 Key Takeaways

One problem with showing the results through a demonstration like what was done in this early stage of the research, is that the participant picked for demonstration also was a part of the training data, making it vulnerable to overfitting. This did not affect the validation stage, as the Leave One Out was used as the preferred validation method. This is because it is the most accurate validation method, and its weak point of computation time was not a problem when working with small datasets such as in this thesis.

Apart from the previously mentioned flaw, the networks showed promising results for predicting participant patterns. It became evident that the scoring categories should be optimized in order to better reflect the different participant groups, and that better expert knowledge could assist in a network structure with more accurate predictions. The domain experts were excited about how this technology could be used during the treatment program, but it was clear from the feedback that precise predictions about how the participants would score at the end of the program were not too important and of any priorities. They explained that the participants who had missing

Post Mapping data included everyone that had dropped out during the program. It was of keen interest to have something that could provide indications of who is likely to drop out, and it was from this excitement that this became the main focus throughout the rest of the research.

6.2 Results from Iteration 2: Post Expert Meeting

The biggest changes between the first and second iteration were based on the feedback from the domain experts. The participants who did not complete the Post Mapping phase was added to the datasets, all nodes that included data from the Post Mapping phase was removed, and a feature describing if the person dropped out or not was added. The focus was now to determine whether a given person was going to complete the treatment program or not. As introduced in Chapter 5, the Tree Augmented Naive Bayes structure learning algorithm produced the most promising results at this stage. More detailed descriptions of what results were produced from this iteration phase will be covered through the rest of this section.

6.2.1 Validation

This subsection will focus on examining the various algorithms' results by validating them through various metrics, with emphasis on accuracy calculated through the Leave One Out validation method.

Accuracy

As the models created in this iteration focused on the participants that did not complete the treatment program, dropout became the most important feature to validate. This was displayed as a simple Yes or No in the network, differing from most of the other features which had 4-5 discretizised categories.

Table 12: Accuracy: Tree Augmented Naive Bayes

State	Accuracy	Accuracy %	Correctly predicted
Dropout (total)	0.58	58%	64 / 111
No	0.67	67%	42 / 63
Yes	0.46	46%	22 / 48

Table 12 shows the accuracy of the Bayesian network that scored the best in the second iteration of this research. The first row presents the accuracy of dropout in total, including both predictions for people that are going to drop out and the people that will complete the treatment program. The next row is for predictions of participants that are specifically not going to complete the program, which was the most important feature to identify. Being able to tell who will complete the program was not as crucial as identifying the participants that will most likely drop out before they actually do it. Knowing this information can enable measures to be taken

in order possibly make changes that can convince the participants in question to complete the program. The last row in the table shows the model's accuracy when predicting if a person is going to complete the treatment program.

Table 13: Accuracy: All Networks Compared

Algorithm	Accuracy	*CP	*AY	*CPY
Bayesian Search	0.57	63 / 111	**0.00**	**0 / 48**
Greedy Thick Thinning	0.57	63 / 111	**0.00**	**0 / 48**
Naive Bayes	0.52	58 / 111	**0.31**	**15 / 48**
Augmented Naive Bayes	0.52	58 / 111	**0.35**	**17 / 48**
Tree Augmented Naive Bayes	0.58	64 / 111	**0.46**	**22 / 48**

A comparison between accuracy of the created networks can be seen in Table 13. **CP** is an abbreviation for Correctly Predicted, and is related to how many cases were correctly predicted out of the 111 in total. **AY** is short for Accuracy: "Yes", and is used to represent the accuracy where the model predicted participants to drop out. **CPY** stands for Correctly Predicted: "Yes", and is related to the instances where the model correctly predicted a person not to complete the treatment program out of the 111 in total. When the results from the five structure learning algorithms were compared, both Bayesian Search and Greedy Thick Thinning seemed to produce moderately good results at first glance. One can see that both of these networks achieved a total accuracy of 0.57 when predicting Dropout. Further analysis uncovered that these two models actually predicted a person to drop out 0/111 times, meaning that they predicted participants to complete the program every time.

Tree Augmented Naive Bayes did not only achieve the highest total accuracy, but also the highest accuracy when predicting a person to drop out (0.46). Even though this was the best result far, it also made it clear that some changes needed to be made in order for any Bayesian network to provide any meaningful contributions.

Confusion Matrix

The confusion matrix provides meaningful insight that can give a more detailed description of how the network performed during validation. When looking at the network that scored the best results, learned from the TAN structure learning algorithm, it was evident that 2/3 of the correct predictions came from predicting participants that would complete the treatment program. Out of the 48 people that dropped out, 26 were predicted to complete the program by the model (Figure 17). Through the confusion matrix it became clear how flawed results both Bayesian Search and Greedy Thick Thinning had produced, even though it might have looked promising at first.

		Predicted	
		No	Yes
Act.	No	**42**	21
	Yes	26	**22**

Figure 17: Validation Tree Augmented Naive Bayes: Dropout Confusion Matrix

ROC curve

Another way to assess the performance of a of a diagnostic test can be achieved through the receiver operating characteristic (ROC) curve. By looking at the range of possible cut-points for the predictor variable, the AUC score can be used to measure the discrimination ability of a given model.

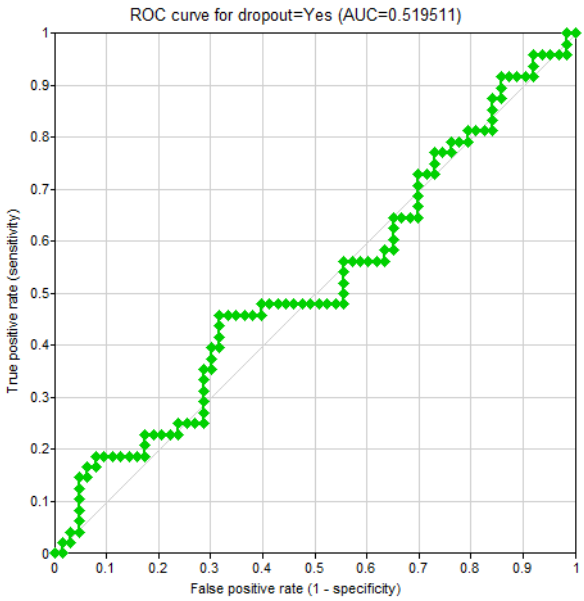

Figure 18: Validation Tree Augmented Naive Bayes: ROC Curve

The curve in Figure 18 follows the diagonal, being slightly above it most of the time. There is one AUC value for both class node outcomes (Yes and No) in total. The AUC value was 0.52 for the Tree Augmented Naive Bayes. This meant that the even the best model from this stage in the development did not live up to what can be regarded as an acceptable result.

6.2.2 Takeaways From Second Iteration

Results from the second iteration showed progress as it improved from the Bayesian networks that were showed to the domain experts. However, the results had still to live up to acceptable standards in order to provide any real contributions. Takeaways from the results displayed in this section can be described as the following: (1) the results gave further indications of the potential impact a tool like this could have, and (2) these results made clear the shortcomings of the models that were created so far. It was apparent that in order to achieve any desired

results when developing new Bayesian networks, it was necessary to take a step back to create categories that better reflected various participant groupings, recalculate the scores, and add new features to locate patterns which could raise the quality of the results. These points were successfully implemented between this iteration and the third and final iteration, as described in Chapter 5.

6.3 Results from Iteration 3: New Categories, Scoring Calculations, and Weekly ASRS

New scoring calculations, discretizised categories, and the addition of weekly ASRS scores greatly improved the final results in this research. New relationships were formed, and features that can be more important than others to monitor were identified. This section will cover what these results were, present various important metrics and measure some of these up to area specific standards.

6.3.1 Validation

Leave One Out was chosen as the preferred validation method, which was the same throughout the research. This is the most accurate method as it represents an extreme version of K-fold cross validation, and produces metrics for accuracy, confusion matrix, ROC curve, and calibration.

Accuracy

Accuracy of the models (except Naive Bayes) developed throughout iteration three produced a significant lift in results with regards to precision compared to the previous iterations. All networks except from the one learned from Naive Bayes had a total accuracy of 0.77 or higher when prediction whether a person will drop out or not (Table 14). When looking at accuracy, specifically for the state where a person drops out of the treatment program, the network learned by the ABN predicted correctly with an accuracy of 0.78. The remaining structure learning algorithms, Bayesian Search, Greedy Thick Thinning, and the TAN, all produced networks with accuracies of 0.98. This means that when predicting the state where a participant drops out, these three models predict the correct outcome 98% of the time.

Table 14: Final Results Accuracy: All Networks Compared

Algorithm	Accuracy	*CP	*AY	*CPY
Bayesian Search	0.77	84 / 109	**0.98**	**45 / 46**
Greedy Thick Thinning	0.77	84 / 109	**0.98**	**45 / 46**
Naive Bayes	0.58	63 / 109	**0.00**	**0 / 46**
Augmented Naive Bayes	0.77	84 / 109	**0.78**	**36 / 46**
Tree Augmented Naive Bayes	0.79	86 / 109	**0.98**	**45 / 46**

It was identified that the addition of the weekly ASRS scores helped to elevate the the results to a higher level. As this was discovered, a new set of Bayesian networks were developed to test how this affected the accuracy. One of the networks that produced the best results were chosen, and five new networks were constructed through the same approach. The Tree Augmented Naive Bayes structure learning algorithm was used, and the following networks were developed:

- A network where none of the ASRS weekly scores were included.
- A network where scores from ASRS Week 1 was included in addition to the rest of the Pre Mapping Data, including dropouts.
- A network where scores from ASRS Week 1 and ASRS Week 2 were included.
- A network where scores from ASRS Week 1-3 were included.
- A network where scores from ASRS Week 1-4 were included.

This provided insight to how the extra information available throughout the treatment program will affect the precision of the model, as it was intended to be used with evidence insertion as more and more evidence become available. An overview of how this affected the network learned from the Tree Augmented Naive Bayes can be viewed in Table 15.

Table 15: Accuracy Tree Augmented Naive Bayes: Weekly ASRS Impact

ASRS Weekly Modules	Accuracy	*CP	*AY	*CPY
No Weeks	0.54	59 / 109	**0.33**	**15 / 46**
Week 1	0.72	79 / 109	**0.65**	**30 / 46**
Week 1-2	0.77	84 / 109	**0.78**	**36 / 46**
Week 1-3	0.77	84 / 111	**0.78**	**36 / 46**
Week 1-4	0.79	86 / 111	**0.98**	**45 / 46**

Confusion Matrix

The confusion matrix (Figure 19) of the network including ASRS Week 1-4 could tell that the algorithm correctly predicted one less participant to complete the treatment program (41 / 63) than in the previous iteration. However, when predicting the case where a person drops out, the model only **missed** 1 out of 46 predictions. This was a significant improvement from previous iterations, as predicting this state previously had been one of the hardest ones to produce an accurate prediction for.

		Predicted	
		No	Yes
Act.	No	**41**	22
	Yes	1	**45**

Figure 19: Final Validation Tree Augmented Naive Bayes: Dropout Confusion Matrix

Receiver Operating Characteristic (ROC) Curve

When assessing the quality of the model through the ROC curve, the focus lies on the area under the ROC curve (AUC). This area is displayed above the ROC curve, where good results move parallel with how much of the curve that is above the diagonal. This can easily be measured through the AUC score, and a comparison between networks developed through the five various structure learning algorithms can be seen in Table 16.

Table 16: ROC Curve AUC Score: All Networks Compared

Algorithm	AUC Score
Bayesian Search	0.61
Greedy Thick Thinning	0.61
Naive Bayes	0.36
Augmented Naive Bayes	0.76
Tree Augmented Naive Bayes	0.71

Naive Bayes did not produce an accurate model, and received an AUC score of 0.36. Both the network learned from Bayesian Search and the Greedy Thick Thinning received AUC scores of 0.61, which is not considered to be very good. Even though these networks achieved exceptional accuracies when prediction the state where a person is dropping out, the AUC score suffers from a model that wrongly predicted 24 / 63 cases when the prediction was that a person will not drop out. Both the network learned from the TAN and the ANB received AUC scores that lived up to acceptable standard, with 0.71 and 0.76 respectively. A further analysis was conducted to see how the impact of the weekly ASRS scores affected the AUC score, much like the one showed with the accuracy of the TAN in Table 15. Another network was chosen for this, and five networks were made with the Augmented Naive Bayes algorithm.

Table 17: ANB ROC Curve AUC Score: Weekly ASRS Impact

ASRS Weekly Modules	AUC Score
No Weeks	0.36
Week 1	0.36
Week 1-2	0.76
Week 1-3	0.76
Week 1-4	0.76

Table 17 shows AUC score comparison of the weekly ASRS impact. This procedure gave similar results as the comparison of weekly ASRS impact on the accuracy of the network made with Tree Augmented Naive Bayes, where the AUC scores received a significant lift when the scoring results from ASRS Week 2 was added to the model during learning. There was no change in the AUC score when adding week 3 and 4, indicating that it is especially important to take special care concerning this phase of the treatment program.

The complete ROC curve for the network learned from the Augmented Naive Bayes can be seen

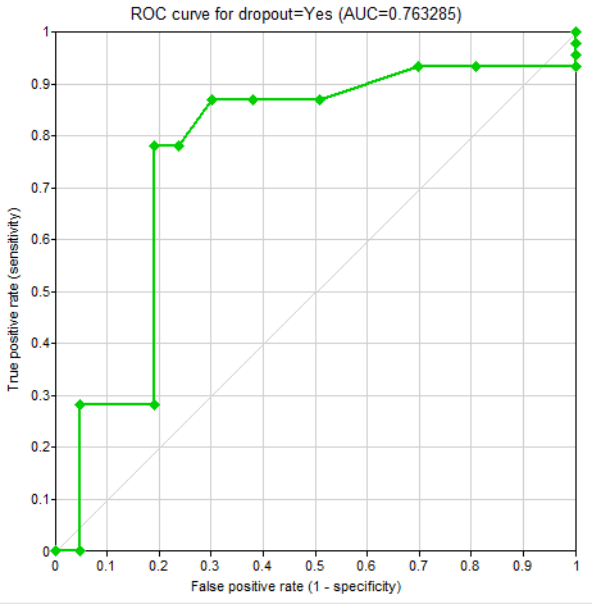

Figure 20: Augmented Naive Bayes: ROC Curve For Dropout=Yes

in Figure 20. It shows how the curve looks for the class node *Dropout* when the outcome is *Yes*. The y-axis depicts the *True Positive Rate* (sensitivity), which are results that are genuinely positive with regards to the data and also received positive results from the model. The x-axis shows the *False Positive Rate* (1 - specificity), and are results that are predicted negative by the model that are actually negative with regards to the data. The figure shows a curve that was mainly above the diagonal line in the curve, and to achieve even better results one should focus on improving the few areas that fell below this line.

Calibration Curve

The calibration curve compares how the output probability of a model measures up to observed frequency data, as described in Section 3.2.4. The x-axis, which is the *classifier probability*, displays the probability of an event happening produced by the model. The y-axis is the *prevalence of Yes*, and displays the actual observed frequencies in the data of a person dropping out (Yes). Figure 21 shows the calibration curve for the Augmented Naive Bayes learned network, and is displayed using a moving average and window size of 3. The diagonal line represents the ideal calibration curve where every probability corresponds to the observed data.

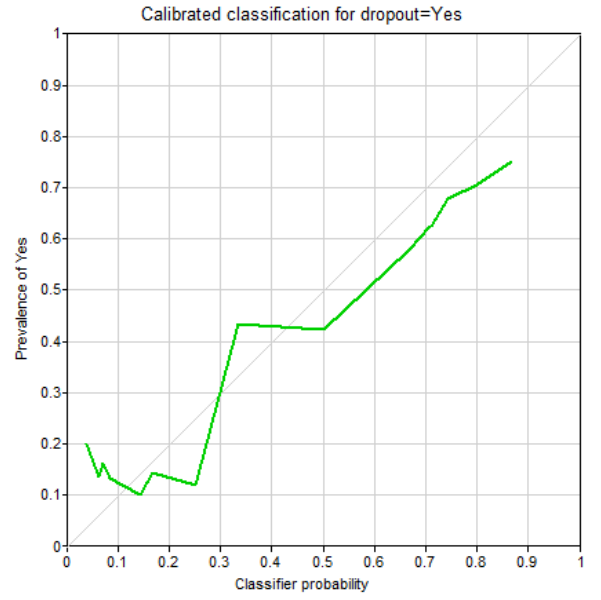

Figure 21: Augmented Naive Bayes: Calibration Curve Classification For Dropout=Yes

The total accuracy of this model was 0.77, and 0.78 when focus was on predicting the Dropout state *Yes*. The AUC score was 0.76, and these metrics are related to the points on the curve that falls off the diagonal line.

6.3.2 Sensitivity Analysis

A sensitivity analysis validates the probabilistic parameters of a Bayesian network by examining the effect of small changes in numerical parameters (Section 3.2.5). The observation of changes in posterior probabilities helps identify which parameters has the highest effect on the model's output. The sensitivity analysis displayed in Figure 22 is from the same network as analyzed above (Augmented Naive Bayes). When target node is set to observe *Dropout*, the sensitivity analysis presents the results by coloring the nodes in which small changes can lead big changes in the target node. ASRS Week 2, ASRS Week 4, and ASRS Inattention was the nodes that was highlighted during the analysis, and had the largest influence on the Dropout node. As the Augmented Naive Bayes always rely on one parent node that have relationships to the remaining child nodes, it was expected that this would be included as being of importance. After seeing the

impact the Weekly ASRS scores had on both accuracy and AUC score (Tables 15 and 17), the ASRS Week 2 was also expected to be highlighted in the sensitivity analysis. This result marked itself as another confirmation about the importance of putting an emphasis on monitoring this specific phase of the treatment program.

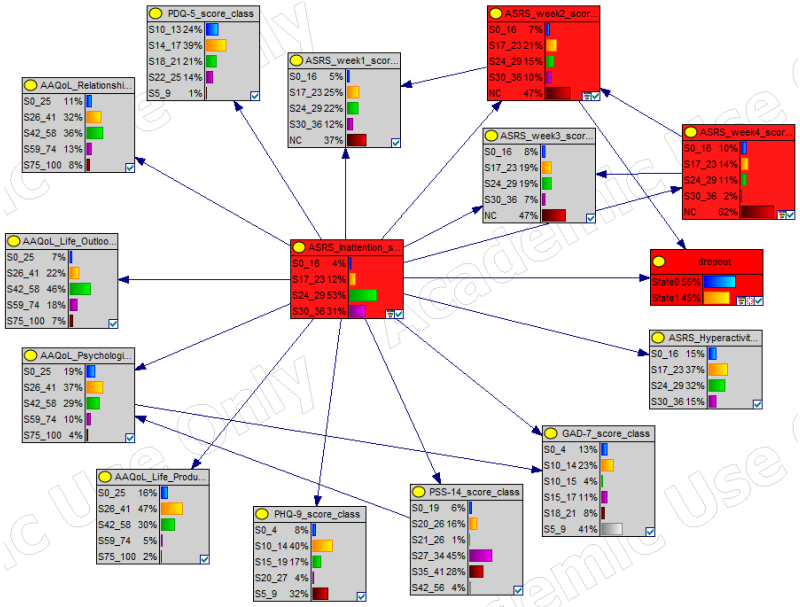

Figure 22: Augmented Naive Bayes: Sensitivity Analysis

Tornado Diagram

A tornado diagram shows the most sensitive parameters sorted from most to least sensitive, where exact numerical sensitivities for each bar can be accessed. The diagram in Figure 23 is a demonstration of the tornado diagram from the ANB network when the target outcome was set to *Dropout - Yes*. The model was most sensitive to the the parameters ASRS Week 2 = NC (Not Completed), ASRS Inattention = 24-29 (scoring category), and ASRS Week 4 = NC when prediction if a participant will drop out. This correlated to colored nodes in the sensitivity analysis, but also further emphasized the important of the weekly ASRS modules. One can further see from the diagram that either ASRS Week 2, ASRS Week 4, or both of them were included in all of the top 10 most sensitive points for the network, where ASRS Week 2 was included in 8 of them.

6.4 Result Takeaways

This chapter has provided a detailed description of the results that were achieved in this thesis. Each iteration described throughout the chapter had its importance in the development to be able to land on the final results that were presented in Section 6.3. The initial results provided knowledge and insight on how to implement Bayesian networks tailored to cognitive behavioral therapy, and made it possible to demonstrate something to the domain experts working on the

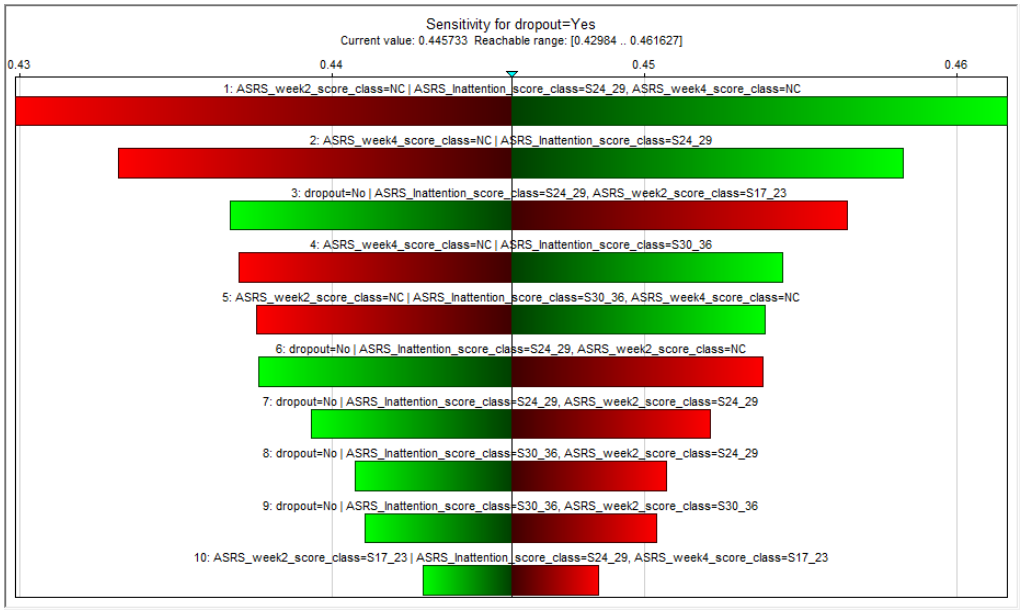

Figure 23: Augmented Naive Bayes: Tornado Diagram

online ADHD treatment program. The feedback and awareness that was supplied by the domain experts was crucial in the next step of development, as they were able to answer questions related to the program and area of expertise. Receiving specific requests directly from them made further development more feasible and goal-driven. Figure 24 displays a flowchart of the development cycle of this research.

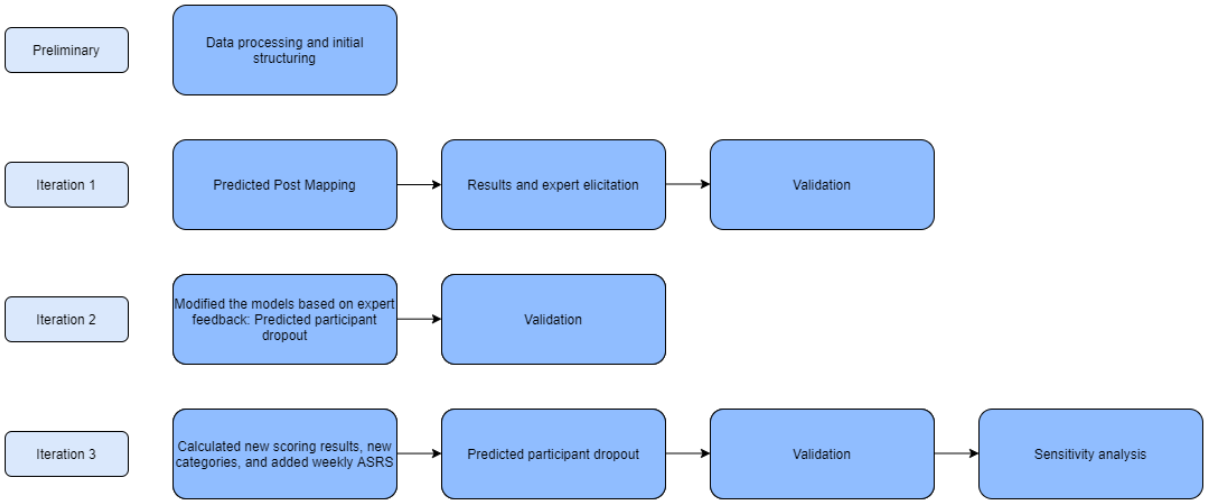

Figure 24: Development Cycle

Development in later stages achieved better results where validation through accuracy, confusion matrix, ROC curve and AUC score, and calibration curve made it possible to evaluate and identify parameters that is important to pay extra attention to during the treatment program. Especially week 2 of the weekly ASRS modules emerged as a critical point of the program and should be monitored with extra care.

7 Discussion

This chapter discusses the research approach, network development and validation, and presents the findings and results in relation to the research questions.

7.1 Research Approach

This research was guided by principles and disciplines of Design Science Research and a number of artifact specific methods as described in Chapter 3. The *desk research* was important to familiarize with management tools, perform data processing to identify usefulness and limitations to the data, and also get acquainted with health-related concerns and properties. Domain experts from Helse Bergen working with the internet-delivered intervention for adults with ADHD helped describe the data properties and expressed what contributions they were looking for based on their needs, which further helped shape both the research questions and research scope. It was important to consolidate the findings in this thesis to concepts of design science and to link it to the applicable literature. An extensive literature review that both covered the online intervention and the principles of a Bayesian network was conducted to accomplish this (Chapter 2).

Design as an artifact describes that an artifact must enable the implementation of its application in a suitable domain (Hevner et al., 2004). The artifact developed in this research was a set of Bayesian networks to be utilized as a decision making support tool in cognitive behavioral therapy. This showed promising results when compared to domain standards with its prediction capabilities. The *problem relevance* was important to assess constantly in order to effectively work towards a set goal. As design is seen as a *search process* in design science, the aim is to discover an effective solution to a given problem. This is an iterative process, which is why the first iteration of this thesis mainly focused on obtaining valuable feedback and knowledge from domain experts to properly identify the problem relevance at hand. Both theoretical foundations and methodologies are key features to establish *research rigor*, where the selection of appropriate development methods are important. The Bayesian networks in this research were learned with *structure learning algorithms*, where various conventions of *analytics* were applied in order to effectively *learn the parameters* in the networks at hand. The knowledge editor in GeNIe was used to assign features to temporal tiers (Section 3.2.2) after discretization was performed on the continuous variables. As designing an artifact is an iterative process, it was important to *evaluate* throughout the whole design cycle. Principles of static analysis, dynamic analysis, optimization, and white box testing were used in this thesis to follow an iterative process. Complexity is an important static quality to consider when working with Bayesian networks, while performance is a dynamic quality that is crucial for the end results. Identifying the optimal properties to achieve this was also key, as these were constantly changing between the iterations (Tables 1, 2, and 3. White box testing was relevant as coverage testing of metrics like accuracy,

AUC value, and calibration were performed at the end of iterations. This was done through *validation*, the primary convention of evaluation of this thesis. The Leave One Out method was used as the preferred validation method, as it presents an extreme version of K-Fold Crossvalidation (Section 3.2.4). The LOO was appropriate to use in all of the experiments in this research as computation time was never an issue due to the small dataset. *Sensitivity analysis* was also one of the implemented methods to evaluate the results which further emphasized the findings by identifying sensitive parameters.

Communication of research states that research needs to be presented to both technology-oriented and management-oriented audiences. Technology-oriented audiences need detailed descriptions to enable implementation and to take advantage of the benefits of the research, while there is a need for stakeholders to understand the contributions for decision making purposes (Hevner et al., 2004). The literature includes both advanced principles of Bayesian networks detailing concepts, strengths, and limitations, and easily comprehensible examples to satisfy this need. This way, management-oriented audiences can understand details that enables them to determine if organizational resources should be used to construct or purchase such an artifact. This is combined with metric performance to underline the potential contributions of the artifact. *Research contributions* has to be clear when a new artifact is developed, where the three potential types of research contributions are based on generality, novelty, and significance. A novelty in this research was the inclusion of semi-structured data and the use of input and feedback from domain experts. It is not uncommon to use surveys or structured entries to study self-report data and reflect on performance, but data processing is required to enable this for semi-structured data. The two main contribution types in this thesis are the design artifact and foundations. 44% of the participants in the intervention program dropped out before completing it, and the artifact itself is a contribution that aims to help solve this problem by identifying as early as possible. The program itself is an already existing study, making foundations a contribution type as Bayesian networks are developed to help extend and improve this practice with an end goal of helping participants struggling with ADHD improve their life quality. Lastly, an extensive literature on the principles and disciplines of Bayesian networks are among the contributions in this thesis.

7.2 Bayesian Network in Cognitive Behavioral Therapy

The most distinguished appeal to Bayesian networks is the way it can be used to handle uncertainty and missing values (BayesFusion, 2020). There were primarily completed three separate iterations in this thesis. Sections 5.1 and 6.1 presented the first iteration where emphasis was on exploring whether Bayesian networks could be implemented to predict participant behavior, and if a support tool like the one presented in this thesis was of any interest to the domain experts working on the program. Bayesian Networks can be developed by two main approaches: (1) Construct the network *by hand*, where an expert is used to estimate *the conditional probability*

tables, or (2) use statistical models that will automatically *learn these probabilities* (Koller & Friedman, 2009). Structure learning algorithms was used to learn the probabilities in this thesis, and expert knowledge helped guide the process. The networks did not produce satisfactory results when validated in this phase as the highest achieved accuracy was 0.44 on predictions, which does not produce any real contributions in terms of correct prediction rate. An accuracy of 0.44 means that the model misses 56% of its predictions. The expert feedback proved highly valuable, as the input emerging from the first iteration was used to assess usability concerns and facilitate further iterative development. After the demonstration described in Section 6.1.1, it was clear that such a support tool could be useful and that there was interest in knowing if the participants are predicted to drop out of the program or not. This was of great importance when redefining the scope of the thesis, as it had been too abstract and broad with regards to predicting participant behavior, specifically *what* to predict, prior to this. As described in Section 2.3.2, preference is an important factor when working with real world applications. Decision makers can help provide a utility function for a given decision problem, known as utility elicitation.

After receiving feedback from the domain experts, and having tailored the research scope, the second iteration consisted of exploring the viability of those preferences. Sections 5.2 and 6.2 illustrated the development and results of the second iteration, including a more thorough validation approach than the first iteration. Changes were made to the applied data based on expert feedback, and the various structure learning algorithms struggled to identify good patterns that included the dropout feature, where the highest total accuracy was 0.58. The accuracy for correctly predicting that a participant would drop out was down at 0.46, with an AUC value of 0.52 which is below acceptable standards. This urged to make changes to the data processing in which the structure learning algorithms use to find patterns, and to further examine some of the unstructured data that was available. As metric values from validation was expected to be affected by the size of the dataset to some degree, the results from this iteration gave indications that a full re-implementations of data and features could yield promising results.

Looking at the improvements that were made during the third iteration from Sections 5.3 and 6.3, results were substantially more promising. The first and second iteration laid the groundwork to identify usefulness, reshape research scope, and highlight areas of improvement. All structure learning algorithms except the Naive Bayes, which do not actually learn the structure of a model (Section 3.2.1), produced accuracies from 0.77-0.79. Three networks had an accuracy of 0.98 when predicting that a participant will drop out, an increase of 0.52 from the previous iteration. The highest scoring AUC value increased with 0.24, from 0.52 to 0.76, which is comfortably above acceptable standards. Some significant contributions and findings were behind this major increase. New and correctly calculated scoring results and new discretizised categories for the self-report scales were instrumental to the improvement. Even more so, it was the inclusion of the weekly ASRS modules that the participants answered during the intervention that helped elevate the results. There were 4 weekly modules included where the

participants answered an additional ASRS questionnaire between the Pre Mapping and Post Mapping phase, where participants answered on various dates. From the semi-structured data, it was discovered that this was conducted in two week intervals, meaning that "ASRS Week 1" was answered in the second week, "ASRS Week 2" in the fourth week, "ASRS Week 3" om the sixth week, and finally "ASRS Week 4" in the eight week. When only ASRS Week 1 and 2 was included in the data, the AUC value for the Augmented Naive Bayes learned network increased by 0.40, from 0.36 to 0.76 (Table 17). This value did not increase further after including the two remaining weekly modules. The Tree Augmented Naive Bayes learned network increased its accuracy on predicting a participant to drop out from 0.33 to 0.65 by including the ASRS Week 1 module, and further increased it to 0.78 after ASRS Week 2 was added. It was not before ASRS Week 4 was also added that this number achieved an accuracy of 0.98. The quality of the calibration curve (Figure 21) solidified these findings. In a real situation, it is not preferable to have to wait until the eight week to have accurate results, but with the size of the data used for learning in mind, perfectly accurate predictions was not expected. Looking at the sensitivity analysis, it is again clear that the ASRS Week 2 module is an important feature to monitor, as this proved to be the most sensitive parameter followed by the ASRS Week 4 module.

7.3 Answering Research Questions

RQ1: What are the strengths and limitations of a Bayesian Network?

The fact that Bayesian networks are based on probability theory can rise opposition concerning its results and the consequences of potential errors introduced when implemented. A counter argument is that the ability to reason with uncertainty by exploiting hidden patterns and provide accurate probabilities is the most obvious benefit of Bayesian networks (Barton, Saloranta, Moe, Eggestad, & Kuikka, 2008). Through the implementation of Bayesian networks in a probabilistic model, we can expect to encounter some of the following strengths and limitations:

Strengths:

- *Likelihood Estimation and EM clustering*: Two nodes that can cause the same state of affairs without any other connection are independent. A converging node will be the result any time there are two two potential causes for that state of affairs (Charniak, 1991). By following a Gaussian probability distribution, this statistical method uses the mean and variance (Bishop, 2006). By only having the knowledge a partial sample of a given data set, the maximum likelihood estimation is able to estimate this. Expected Maximization Clustering is another available approach when constructing BNs, which is more favourable when dealing with incomplete logs due to the precision of likelihood estimation.

- *Handling Noise*: Current data tends to be extremely noisy and can lead to some diffi-

culties. Bayesian networks contributes to eliminate some of these difficulties since they are built on properties of dependence and conditional independence. By choosing BNs over alternative tools based on clustering algorithms, Friedman et al. (2000) were able to analyze gene expression patterns to uncover new properties of a transcriptional program consisting of thousands of genes from the health sector while handling the noise in their data and estimate the confidence in the networks' features.

- *Decision Theory*: Bayesian networks has roots in probability theory and was later extended to handle a close relative in decision theory (Charniak, 1991). When the goal is to discover which action will maximize an expected utility, specifying *decision nodes* indicating available actions and *value nodes* that indicates different outcomes' values, a BN becomes an *influence diagram* (Howard & Matheson, 2005). This can be utilized to automatically choose what to check for next if the current state is not adequate to conclude a diagnosis.

- *Data Analysis*: Most organizations today rely on information systems, leaving stored data which value can be maximized when analyzed efficiently. The challenge is that information is too often stored unstructured, dispersed across tables and sub-systems communicating with each other (van der Aalst, 2011). By finding the structure of this data within accurate approximate values close to the exact values, Bayesian networks demonstrates its usefulness by providing needed information.

- *Versatility*: Bayesian networks have shown to be versatile and can be implemented with other practices. It is possible to perform causal interpretations for Bayesian networks, despite the fact that there might not seem to be any direct connections between probability distribution and causality. Causal nets are also represented by directed acyclic graphs, and can be interpreted as BNs when the *Causal Markov Assumption* is made (Friedman et al., 2000). Bayesian networks can also be implemented in other applications as recommendation systems and desktop applications by incorporating it as an underlying engine performing analysis (L. BayesFusion, 2017).

- *Bayesian networks are explainable*: Bayesian networks are able to compute the probability of events occurring by analyzing the network. By relying on dependencies and conditional independences, the working principle of a BN is therefore easily explainable. Bayesian networks can be advantageous over more complex methods due to this, especially in the healthcare sector. It is required that decision support systems in the healthcare sector are explainable, according to new regulations from the GDPR (Goodman & Flaxman, 2017). Prediction models based on machine learning, like Bayesian networks, are expected to play a major role in aiding the decision making done by healthcare experts (Marcos et al., 2020).

Limitations:

- *Explosion of values*: One potential pitfall of BNs is the amount of numbers in a complete specification of a potential probability distribution (Charniak, 1991). As this may seem like a headache, Bayesian networks rely on built-in independence assumptions to deal with this. Variables that intuitively do not seem independent of one another may yet be so, which drastically lessens needed values.
- *Inconsistent probabilities*: Since BNs are naive probabilistic schemes, inconsistent probabilities can become a potential burden. Fortunately, there is a possibility to specify the required numbers manually. With consistent numbers, the network will ultimately define a unique distribution (Charniak, 1991).
- *Computation time*: Computation time can emerge as a challenge when modeling realistic Bayesian networks. Networks consisting of tens of nodes might pose too long computation time, but networks of several thousands of nodes can deliver acceptable time. The key factor lies on the care taken by developers that performs the implementation as well as the algorithm used, not only the particulars of a given network (Charniak, 1991).
- *NP-hard*: The fact that the computation is generally NP-hard can be one of the most inhibitory constraints of implementing BNs (G. Cooper, 1987). Implementation of multiple connected BNs on probabilistic inference with uninstantiated variables is NP-hard, implying there is an exact algorithm (G. F. Cooper, 1990). This means that any attempt to make a general algorithm to cover all cases will be extremely hard, if not impossible, and an attempt like this should not be of high priority.
- *Mutual exclusion problem*: Unexpected error rates in Bayesian networks can be related to the mutual exclusion problem, something BNs struggle to account for in a desired manner. For two events to be mutually exclusive, it has to be impossible for two events to occur at the same time, and new edges have to be manually added to represent this. This will introduce non-trivial effects the network and end up changing the probability values as all nodes depend on each other (Moreira, 2015).

Several potential limitations can arise when choosing Bayesian networks to model the structure of a process. However, most of them comes with valid solutions. The advantages are many, and the key factor is often the care taken during implementation.

RQ2: How can Bayesian networks be utilized as a decision making tool in cognitive behavioral therapy?

Bayesian networks offers an effective tool of handling uncertainty through accurate predictions (BayesFusion, 2020). There is a need for features in the form of data properties to successfully implement this. It is important to perform utility elicitation to assess any decision problem by

communicating with a relevant decision maker (Section 2.3.2). When important features are identified, there is a need to process the available data, as this often comes as a combination of structured, semi-structured, and unstructured data. The data used in this research was first cleaned, before being calculated into scoring results, and discretizied into categories to avoid continuous variables. When there is a need to label semi-structured or unstructured data, there can be limited availability of experts or this can prove to become too expensive. This can motivate employing additional non-experts to process large datasets in a crowdsourcing effort to produce structured, valuable data to ease this effort (Chen et al., 2016).

A Bayesian network will provide prior probabilities that indicate average outcomes, but the real contribution in terms of uncovering its potential is through evidence evidence insertion (Section 2.3.8). Constantly updating what the model predicts as new evidence comes to light will produce accurate predictions that will help the decision making process and facilitate more knowledge, more rapidly.

RQ3: How can Bayesian network theory be applied for predicting the dropout of internet based cognitive behavior treatment program, and how can we measure the accuracy of such applications?

Out of the 109 participants that entered the online treatment program, as many as 46 of them did not complete its course (Section 4.3.1). This was identified by domain experts as the most crucial feature to predict. With processed data and expert knowledge to augment the process, structure learning algorithms could execute the learning process in order to produce accurate models. Background knowledge, i.e., temporal tiers and class variables, were added before letting the algorithms learn the structure of a model to produce the best results. Validation measured the performance of the model when tested on real life data, with accuracy displaying how precise the predictions were. Further analysis was performed by looking at confusion matrices predicted outcomes and actual outcomes. A sensitivity analysis could also further confirm findings from validation by highlighting the most sensitive parameters in the model. This is decisive information to gain, as it is directly associated with critical factors that determine whether a participant will complete the program or drop out.

7.4 Limitations

There were several limitations to consider in this thesis, some of them being algorithm limitations discussed in Section 7.3. Firstly, the size of the dataset should be mentioned as a clear limitation, as data from only 109 participants was available in the research. This removed any concerns towards computation time, but that could yet arise when more data is available. Second, there is also the possibility that activity data could have provided valuable contributions, but this was excluded due to lack of time and resources in terms of the additional data processing this would have needed. Third, even though results substantially improved after calculating

new scoring categories, to convert from continuous variables to discrete variables, one should not exclude the possibility that more preferable categories exists.

8 Conclusion and Future Work

The main contribution in this research is the Bayesian networks as an artifact, with the goal of assisting domain experts by providing a decision making tool tailored towards cognitive behavioral therapy. A literature review describing in detail what a Bayesian network presents was conducted. This serves an independent contribution to the existing knowledge base. Existing literature does not provide a solution for delivering a decision making support tool in cogitative behavioral therapy like the one presented in this thesis.

This thesis has demonstrated a way of predicting participant behavior in cognitive behavioral therapy by using scoring results to develop Bayesian networks in an internet delivered intervention for adults with ADHD. Domain experts with hands-on experience from the treatment program was included in the process at an early stage to identify the stakeholders' needs. Design science concepts was applied to the analysis of data and Bayesian networks. A novel artifact of relevance for a given problem that also provides value for intended users is essential in the design science methodology. Based on the domain expert elicitation during development, this research could be regarded as novel and a meaningful contribution to the knowledge base. Evidence-based psychological interventions is of poor availability, and the need for robust analytical capabilities is of high demand as there is room for improvement in today's practices. This thesis addresses this concern of evidence-based research by utilizing hidden patterns through various structure learning algorithms. Sensitivity analysis provides further useful insight in identifying the most sensitive parameters in a model. This is beneficial information for stakeholders and decision makers, as it may represent critical instants in the treatment program.

Based on the results from this research, some tangible conclusions can be drawn. Data that is properly processed and analyzed can produce accurate probabilistic models for cognitive behavioral therapy that scale well with new data. Validation showed promising results when metrics were compared to acceptable standards, and research indicate that this will only improve alongside the availability of more data. Today, decision making is mainly executed based on expert knowledge and experience. Ad-hoc feedback in terms of accurate predictions can assist the decision making process by making the process more instantaneous and information based.

Finally, the research shows the promise and use of historical data to predict participant behavior in cognitive behavioral therapy. Predictions from Bayesian networks display accurate results. If this can help psychiatrists and other domain experts adapt to a wider group of participants and produce a more tailored treatment, this could support participants undergoing cognitive behavioral therapy. In this case, adults affected by severe consequences of ADHD can improve their quality of life.

8.1 Future Work

The next step in the development of this artifact would be to extend the Bayesian networks to a more user-focused application. The SMILE engine is a useful tool that could help implement the artifact in a desktop application that is easier to use without complicated details that is not necessary for the end-user. It would also be interesting to use Bayesian networks as a basis for a recommendation system that could be used ad-hoc by the domain experts for rapid tailored suggestions during the treatment program.

It is natural to regard future work as updating the model when more data becomes available. The dataset used in this research was quite limited, and it would be beneficial to use an artifact like this with more historical data for better learning. In addition, it would be interesting to look at the activity data that is available. This could be useful to identify new patterns and possibly improve the probabilistic model through a deeper analysis.

References

Ahrendt, P. (2005). The multivariate gaussian probability distribution. *Technical University of Denmark, Tech. Rep*.

Applegate, L. M. (1999). Rigor and relevance in mis research–introduction. *Mis Quarterly*, *23*(1), 1–2.

Association, A. P., et al. (2013). *Diagnostic and statistical manual of mental disorders (dsm-5®)*. American Psychiatric Pub.

Barton, D., Saloranta, T., Moe, S., Eggestad, H., & Kuikka, S. (2008). Bayesian belief networks as a meta-modelling tool in integrated river basin management—pros and cons in evaluating nutrient abatement decisions under uncertainty in a norwegian river basin. *Ecological economics*, *66*(1), 91–104.

Bayes, F. (1958). An essay towards solving a problem in the doctrine of chances. *Biometrika*, *45*(3-4), 296–315. (reprint)

BayesFusion. (2020). Genie modeler - user manual. *Version 3.0.R2*.

BayesFusion, L. (2017). Genie modeler. *User Manual. Available online: https://support. bayesfusion. com/docs/(accessed on 21 October 2019)*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Brooks, F., & Kugler, H. (1987). *No silver bullet*. April.

Brooks Jr, F. P. (1996). The computer scientist as toolsmith ii. *Communications of the ACM*, *39*(3), 61–68.

Castillo, E., Gutiérrez, J. M., & Hadi, A. S. (1997). Sensitivity analysis in discrete bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *27*(4), 412–423.

Charniak, E. (1991). Bayesian networks without tears. *AI magazine*, *12*(4), 50–50.

Chen, C., Woźniak, P. W., Romanowski, A., Obaid, M., Jaworski, T., Kucharski, J., ... Fjeld, M. (2016). Using crowdsourcing for scientific analysis of industrial tomographic images. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *7*(4), 1–25.

Cheng, J., Bell, D. A., & Liu, W. (1997). An algorithm for bayesian belief network construction from data. In *proceedings of ai & stat'97* (pp. 83–90).

Cheng, J., & Druzdzel, M. J. (2000). Ais-bn: An adaptive importance sampling algorithm for evidential reasoning in large bayesian networks. *Journal of Artificial Intelligence Research*, *13*, 155–188.

Cohen, S., Kamarck, T., & Mermelstein, R. (1983). Perceived stress scale (pss). *J Health Soc Beh*, *24*, 285.

Cook, J. E., & Wolf, A. L. (1998). Discovering models of software processes from event-based data. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, *7*(3), 215–249.

Cooper, G. (1987). *Probabilistic inference using belief networks is np-hard (paper no. smi-87-*

0195). Stanford: Knowledge Systems Laboratory, Stanford University.

Cooper, G. F. (1990). The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, *42*(2-3), 393–405.

Cooper, G. F., & Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine learning*, *9*(4), 309–347.

D'Agostini, G. (1994). *A multidimensional unfolding method based on bayes' theorem* (Tech. Rep.). P00024378.

Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in bayesian belief networks is np-hard. *Artificial intelligence*, *60*(1), 141–153.

Data mining techniques. (n.d.). `https://www.javatpoint.com/data-mining-techniques`. (Accessed: 2021-05-13)

Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and computing*, *2*(1), 25–36.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, *39*(1), 1–22.

Diagrams. (n.d.). `https://www.diagrams.net/`. (Accessed: 2021-11-27)

Dresch, A., Lacerda, D. P., & Antunes, J. A. V. (2015). Design science research. In *Design science research: A method for science and technology advancement* (pp. 67–102). Cham: Springer International Publishing. Retrieved from `https://doi.org/10.1007/978-3-319-07374-3_4` doi: 10.1007/978-3-319-07374-3_4

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, *29*(2), 131–163.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*(3-4), 601–620. doi: 10.1089/106652700750050961

Fung, R., & Chang, K.-C. (1990). Weighing and integrating evidence for stochastic simulation in bayesian networks. In *Machine intelligence and pattern recognition* (Vol. 10, pp. 209–219). Elsevier.

Fung, R., & Del Favero, B. (1994). Backward simulation in bayesian networks. In *Uncertainty proceedings 1994* (pp. 227–234). Elsevier.

Gelernter, D. H. (1998). *Machine beauty: Elegance and the heart of technology*. Perseus Books.

Genie modeler: Complete modeling freedom. (n.d.). `https://www.bayesfusion.com/genie/`. (Accessed: 2021-11-27)

Gjervan, B., & Nordahl, H. M. (2010). The adult adhd quality of life questionnaire (aaqol). *Nordic Psychology*.

Goodman, B., & Flaxman, S. (2017). European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, *38*(3), 50–57.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining. a bradford book*. MIT Press, Cambridge, MA.

Hao, A. J., He, B. L., & Yin, C. H. (2015). Discrimination of adhd children based on deep bayesian network. In *2015 iet international conference on biomedical image and signal processing (icbisp 2015)* (pp. 1–6).

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, *20*(3), 197–243.

Henrion, M. (1988). Propagating uncertainty in bayesian networks by probabilistic logic sampling. In *Machine intelligence and pattern recognition* (Vol. 5, pp. 149–163). Elsevier.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75–105.

Howard, R. A., & Matheson, J. E. (2005). Influence diagrams. *Decision Analysis*, *2*(3), 127–143.

An internet-delivered intervention for coping with adhd in adulthood (myadhd). (n.d.). `https://clinicaltrials.gov/ct2/show/NCT04726813?cond=ADHD&cntry=NO&draw=2&rank=1`. (Accessed: 2021-05-31)

ISR. (2002). Editorial statement and policy. *Information Systems Research*(13:4).

Jensen, F. (1990). Bayesian updating in recursive graphical models by local commutations. *Comput. Stat. Data Anal.*, *4*, 269–282.

Johansson, J. M., March, S. T., & Naumann, J. D. (2003). Modeling network latency and parallel processing in distributed database design. *Decision Sciences*, *34*(4), 677–706.

Kersten, P. R., Lee, J.-S., & Ainsworth, T. L. (2005). Unsupervised classification of polarimetric synthetic aperture radar images using fuzzy clustering and em clustering. *IEEE Transactions on Geoscience and Remote Sensing*, *43*(3), 519–527.

Kim, J.-H., Lee, E.-H., & Joung, Y.-S. (2013). The who adult adhd self-report scale: reliability and validity of the korean version. *Psychiatry investigation*, *10*(1), 41.

Kjærulff, U., & Van Der Gaag, L. C. (2013). Making sensitivity analysis computationally efficient. *arXiv preprint arXiv:1301.3868*.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Kroenke, K., Spitzer, R. L., & Williams, J. B. (2001). The phq-9: validity of a brief depression severity measure. *Journal of general internal medicine*, *16*(9), 606–613.

Lauritzen, S. L. (1995). The em algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, *19*(2), 191–201.

Lauritzen, S. L., & Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, *50*(2), 157–194.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, *5*(9), 1315–1316.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems*, *15*(4), 251–266.

Marcos, M., Juarez, J. M., Lenz, R., Nalepa, G. J., & Nowaczyk. (2020). *Artificial intelligence in medicine: Knowledge representation and transparent and explainable systems*. Springer.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *MIS quarterly*, 179–212.

Microsoft excel. (n.d.). `https://www.microsoft.com/en-us/microsoft-365/excel`. (Accessed: 2021-11-27)

Mitchnick, D., Kumar, V., Fraser, S., et al. (2016). Using healthcare analytics to determine an effective diagnostic model for adhd in students. In *2016 ieee-embs international conference on biomedical and health informatics (bhi)* (pp. 1–4).

Moreira, C. (2015). An experiment on using bayesian networks for process mining. *arXiv preprint arXiv:1503.07341*.

Mueller, & Guido. (1997). *Introduction to ML with Python*.

Myadhd - digital training for adults with adhd. (n.d.). `https://clinicaltrials.gov/ct2/show/NCT04511169?cond=ADHD&cntry=NO&draw=2&rank=4`. (Accessed: 2021-05-31)

Neff, K. D. (2016). The self-compassion scale is a valid and theoretically coherent measure of self-compassion. *Mindfulness*, *7*(1), 264–274.

Norman, D. (2013). *The design of everyday things: Revised and expanded edition*. Basic books.

Orlikowski, W. J., & Iacono, C. S. (2001). Research commentary: Desperately seeking the "it" in it research—a call to theorizing the it artifact. *Information systems research*, *12*(2), 121–134.

Pandas. (n.d.). `https://pandas.pydata.org/`. (Accessed: 2021-11-27)

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, *29*(3), 241–288.

Pearl, J. (2009). *Causality*. Cambridge university press.

Perceived deficits questionnaire. (n.d.). `https://workingwithdepression.psychiatry.ubc.ca/leaps/perceived-deficits-questionnaire-pdq/`. (Accessed: 2021-06-02)

Richardson, T., & Jensen, F. V. (1997). An Introduction to Bayesian Networks. *Journal of the American Statistical Association*, *92*(439), 1215. doi: 10.2307/2965591

Rittel, H. W., & Webber, M. M. (1984). *Planning problems are wicked problems. n. cross (ed.). developments in design methodology (pp. 135-144).* John Wiley & Sons, New York.

Roberti, J. W., Harrington, L. N., & Storch, E. A. (2006). Further psychometric support for the 10-item version of the perceived stress scale. *Journal of College Counseling*, *9*(2), 135–147.

Russel, S., & Norvig, P. (2009). *Artificial intelligence: A modern approach, english*. Prentice Hall.

Sayad, S. (n.d.). *An introduction to data science.* `https://www.saedsayad.com/naive_bayesian.htm`. (Accessed: 2021-11-30)

Self-compassion. (n.d.). `https://self-compassion.org/self-compassion-scales-for-researchers/`. (Accessed: 2021-06-02)

Shachter, R. D., & Peot, M. A. (1990). Simulation approaches to general probabilistic inference on belief networks. In *Machine intelligence and pattern recognition* (Vol. 10, pp. 221–231). Elsevier.

Shen, H., Huo, S., Cao, J., & Huang, T. (2018). Generalized state estimation for markovian coupled networks under round-robin protocol and redundant channels. *IEEE transactions on cybernetics*, *49*(4), 1292–1301.

Silver, M. S., Markus, M. L., & Beath, C. M. (1995). The information technology interaction model: A foundation for the mba core course. *MIS quarterly*, 361–390.

Simon, H. A. (1996). *The sciences of the artificial 3rd ed.* MIT Press Cambridge.

Spitzer, R. L., Kroenke, K., Williams, J. B., & Löwe, B. (2006). A brief measure for assessing generalized anxiety disorder: the gad-7. *Archives of internal medicine*, *166*(10), 1092–1097.

van der Aalst, W. (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes* (Vol. 136) (No. 2). Retrieved from `http://www.ncbi.nlm.nih.gov/pubmed/18487736`

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant eis. *Information systems research*, *3*(1), 36–59.

Weber, R. (1987). Toward a theory of artifacts: A paradigmatic base for information systems research. *Journal of Information Systems*, *1*(2), 3–19.

Weber, R. (2003). Still desperately seeking the it artifact. *MIS quarterly*, *27*(2), 183–183.

Williams, N. (2014). The gad-7 questionnaire. *Occupational Medicine*, *64*(3), 224–224.

Yardley, L., Morrison, L., Bradbury, K., & Muller, I. (2015, Jan 30). The person-based approach to intervention development: Application to digital health-related behavior change interventions. *J Med Internet Res*, *17*(1), e30. Retrieved from `http://www.jmir.org/2015/1/e30/` doi: 10.2196/jmir.4055

Yuan, C., & Druzdzel, M. J. (2012). An importance sampling algorithm based on evidence pre-propagation. *arXiv preprint arXiv:1212.2507*.

Zmud, R. W. (1997). Remarks from mis quarterly editor. *MIS Quarterly*, *21*(2), 261–290.