UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Frederico Augustos Oliveira Parrela

Improved Genetic Algorithm for Bayesian Network Structure Learning Applied to Diagnostic of Coronary Arterial Diseases

Belo Horizonte

2024

Frederico Augustos Oliveira Parrela

Improved Genetic Algorithm for Bayesian Network Structure Learning Applied to Diagnostic of Coronary Arterial Diseases

Dissertation submitted to the Examination Committee designated by the Board of the Graduate Program in Electrical Engineering of the School of Engineering at the Federal University of Minas Gerais, as a requirement for obtaining the title of Master in Electrical Engineering

Orientador: Prof. Dr. Cristiano Leite de Castro

Belo Horizonte
2024

Parrela, Frederico Augustos Oliveira.
P258i Improved genetic algorithm for Bayesian network structure learning applied to diagnostic of coronary arterial diseases [recurso eletrônico] / Frederico Augustos Oliveira Parrela. – 2024.
1 recurso online (71 f. : il., color.) : pdf.

Orientador: Cristiano Leite de Castro.

Dissertação (mestrado) – Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 68-71.

1. Engenharia elétrica – Teses. 2. Algoritmos genéticos – Teses. 3. Teoria Bayesiana de decisão estatística – Processamento de dados – Teses. 4. Coronárias – Doenças – Teses. I. Castro, Cristiano Leite de. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pelo Bibliotecário Marcio A. A. Gomes CRB/6 2812
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA

COLEGIADO DO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"Improved Genetic Algorithm for Bayesian Network Structure Learning Applied to Diagnostic of Coronary Arterial Diseases"

Frederico Augustos Oliveira Parrela

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 26 de julho de 2024.

Por:

Prof. Dr. Cristiano Leite de Castro
(UFMG) - Orientador

Prof. Dr. Frederico Gadelha Guimarães
DCC (UFMG)

Prof. Dr. Michel Bessani
DEE (UFMG)

Documento assinado eletronicamente por **Cristiano Leite de Castro**, **Professor do Magistério Superior**, em 27/07/2024, às 08:51, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Michel Bessani**, **Professor do Magistério Superior**, em 29/07/2024, às 13:19, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Frederico Gadelha Guimaraes**, **Professor do Magistério Superior**, em 29/07/2024, às 13:49, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **3402693** e o código CRC **FA1265B2**.

Referência: Processo nº 23072.240945/2024-08 SEI nº 3402693

AGRADECIMENTOS

Primeiramente, gostaria de expressar minha profunda gratidão aos meus pais, Galeno Parrela e Sirley Barros, pelo amor incondicional, apoio e por sempre acreditarem em mim. Sem vocês, esta jornada não seria possível.

Agradeço também à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro, que foi fundamental para a realização deste trabalho, ao meu orientador Cristiano Castro e os professores Michel Bessani e Frederico Gadelha, cuja orientação, paciência , troca de ideias e conhecimentos foram cruciais ao longo desta caminhada acadêmica.

Um agradecimento especial à minha esposa, Judith Soh, por seu encorajamento constante, apoio inabalável e por estar sempre ao meu lado. Sua presença e incentivo foram fundamentais para a conclusão desta etapa.

A todos vocês, minha eterna gratidão.

"Scientia potentia est, et mathematica est lingua universalis naturae."

Francis Bacon e Galileu Galilei

ABSTRACT

In this study, we developed a new Genetic Algorithm for training Bayesian Networks (GATFBN). Using synthetic datasets like ASIA and Alarm, we validated this algorithm by comparing it with other BN training algorithms, such as TABU and Hill Climbing (HC). The comparison of network structures obtained by the GATFBN indicated that, on average, the GATFBN achieved better results. Subsequently, we trained two BN models using TABU and GATFBN with real medical data, and an XGBoost model for baseline comparison. The BN model obtained with the GATFBN achieved a higher AUC on the test data compared to the model trained with TABU but was lower than the XGBoost model. However, the BN model trained with the GATFBN demonstrated better sensitivity than the XGBoost model. Additionally, we conducted a sensitivity analysis of the variables present in the BN trained by GATFBN . We concluded that the GATFBN produces better structures for BNs and that the model obtained through it could achieve AUCs comparable to XGBoost while offering superior data interpretability.

Keywords: Bayesian Networks, Genetic Algorithm, Bayesian Networks Structure Learning, Medical Data Analysis, Interpretability in Machine Learning.

RESUMO

Neste estudo, desenvolvemos um novo Algoritmo Genético para treinar Redes Bayesianas (GATFBN). Usando conjuntos de dados sintéticos como ASIA e Alarm, validamos os GATFBN comparando-o com outros algoritmos de treinamento de BN, como TABU e Hill Climbing (HC). A comparação das estruturas de rede obtidas pelo GATFBN indicou que, em média, o GATFBN alcançou melhores resultados. Subsequentemente, treinamos dois modelos de BN usando TABU e GATFBN com dados médicos reais e um modelo XGBoost para comparação de base. O modelo de BN obtido com o GATFBN alcançou uma AUC mais alta nos dados de teste em comparação com o modelo treinado com TABU, mas foi inferior ao modelo XGBoost. No entanto, o modelo de BN treinado com o GATFBN demonstrou melhor sensibilidade do que o modelo XGBoost. Além disso, realizamos uma análise de sensibilidade das variáveis presentes na BN-GA. Concluímos que o GATFBN produz melhores estruturas para BNs e que o modelo obtido por meio dele pode alcançar AUCs comparáveis ao XGBoost, oferecendo, ao mesmo tempo, uma melhor interpretabilidade dos dados.

Palavras-chave: Redes Bayesianas, Algoritmo Genético, Aprendizado de Estrutura de Redes Bayesianas, Análise de Dados Médicos, Interpretabilidade em Aprendizado de Máquina.

List of Figures

List of Tables

LISTA DE ABREVIATURAS E SIGLAS

AUC	Area Under the Curve
BN	Bayesian Network
BIC	Bayesian Information Criterion
CPDAG	Class of equivalent directed acyclic graphs
CPT	Conditional Probability Table
DAG	Directed Acyclic Graph
EC	Electronic Cardiogram
F1	F1 Score
FN	False negatives
FDR	False Discovery Rate
FNR	False Negative Rate
FP	False positives
GA	Genetic Algorithm
HC	Hill Climbing
H2PC	Hybrid 2-phase Constraint
ICD-10	International Classification of Diseases, Tenth Revision
IAMB	Incremental Association Markov Blanket
K2	A specific scoring function for BNs
KL	Kullback-Leibler
MAP	Maximum A Posteriori
MMHC	Max-Min Hill Climbing
NLP	Natural Language Processing
PGMPY	Probabilistic Graphical Models using Python

PGA	Proposed Genetic Algorithm
PPV	Positive Predictive Value
RB	Redes Bayesianas
ROC	Receiver Operating Characteristic
SUS	Stochastic Universal Sampling
TABU	Tabu Search Algorithm
TN	True negatives
TNR	True Negative Rate
TP	True positives
TPR	True Positive Rate
TSP	Traveling Salesman Problem
TF-IDF	Term Frequency-Inverse Document Frequency
XGBoost	Extreme Gradient Boosting

LISTA DE SÍMBOLOS

μ	Mean
σ	Standard deviation
$P(X)$	Probability of X
$P(X\|Y)$	Conditional probability of X given Y
$P(X_i \mid \text{Parents}(X_i))$	Conditional probability of X_i given its parents
$\log(L)$	Log-likelihood
Dof	Degrees of freedom
$P_{\text{lin\_rank}}(i)$	Probability of selection proportional to linear ranking
μ	Mean fitness value of all individuals
s	Selection parameter for linear ranking
θ_{ijk}	CPT parameter
q_i	Number of parent configurations of X_i
P_{E}	Probability of the evidence set
N	Number of variables in the dataset
$I(X;Y)$	Mutual information between X and Y
$\rho_{X,Y}$	Correlation coefficient between X and Y
$\text{Cov}(X,Y)$	Covariance between X and Y
D_{KL}	Kullback-Leibler Divergence
$\mathbf{X}$	Set of all variables in the Bayesian Network
$\mathbf{E}$	Set of evidence variables
$\mathbf{Y}$	Set of query variables
$\mathbf{Z}$	Set of hidden variables
$\mathbf{D}$	Adjacency matrix

$\mathbf{A}$	Initial adjacency matrix
B	Altered graph
K2	K2 cost function

SUMÁRIO

1 Introduction

The field of machine learning has experienced a significant increase in interest over the past few years, not only from academics but also from the general public (BRESLOW; AL., 2024)(KIM; AL., 2023). Nowadays, AI is a trending topic that everyone discusses, even if they do not fully understand it. One of the major challenges faced by the AI field is gaining the trust of its users (MORI, 2017). The complexity of machine learning algorithms, especially in deep learning, makes it difficult to track and fully understand how their decisions are made (PEARL, 2000). Now that AI systems are making decisions on behalf of humans, it is more important than ever to understand how those decisions are reached. Legislators around the world are aware of these issues and are imposing laws to make the use of AI more transparent (MATSUDA; AL., 2023) (SEIZOV; WULF, 2020).

In the machine learning field, there is a saying: "Our model is only as good as our data". One of the problems is that biased data leads to biased systems (MEHRABI; AL., 2021). No one would like to know that their mortgage was rejected by a bank solely because the AI system considers their race as a major factor (OBERMEYER; AL., 2019). This problem is also reflected in the medical field, where doctors tend to trust systems they understand, even if those systems are less accurate (QUINN et al., 2021). In this context, Bayesian Networks have seen increasing interest over the past few years (KITSON et al., 2023). One of the significant advantages of using Bayesian Networks is their inherently interpretable nature (DARWICHE, 2009a). BNs allow AI systems to think and make predictions similarly to humans, using strategies such as association, intervention, and counterfactuals (PEARL, 2000). They also allow human scrutiny of their decisions, which increases trust and learning from these models (MURPHY, 2002).

However, one of the main issues with Bayesian Networks is structure learning from data (FRIEDMAN; KOLLER, 1997). Since their structure is made of Directed Acyclic Graphs (DAGs), and just a few variables can generate more different DAGs than the stars in our galaxy, we need efficient ways to represent and learn these structures (CHICKERING, 2003). Genetic Algorithms (GAs) are optimization techniques based on the process of natural evolution that allow us to solve complex optimization problems. GAs solve problems using an initial population and, through the process of applying operators to combine the population to form new individuals, applying evolutionary pressure on them using a cost function, and replacing the less fit individuals, they evolve towards better solutions (GOLDBERG, 1989a). This process has already shown its importance in many fields of science and has also been applied to Bayesian Network structure learning (LARRANAGA, 1996). Many variations of GAs and other evolutionary techniques have been applied to Bayesian Network structure learning (KITSON et al., 2023). But they still lack validation

on large datasets and in real-world scenarios. Additionally, (CONSTANTINOU et al., 2021) showed that the performance of most cutting-edge structure learning algorithms drastically decreases when trained without fine-tuning or in the presence of noisy data. Therefore, we must explore methods that enhance the robustness and generalizability for structureal learning of Bns. One potential approach is to develop evolutionary techniques tailored to explore the features of DAGs, with operators specifically designed to leverage the mathematical properties of these structures. In our study, we developed a novel GA for BN which we call GATFBN (GA tailored for Bns). We validated this algorithm using synthetic datasets and applied it to a real-world medical database scenario. Our results demonstrated the effectiveness of the GA-trained BN in accurately modeling the true underlying probabilistic distribution of the data and the true Directed Acyclic Graph (DAG) structure with benchmark databases. Additionally, we highlighted the interpretability features of the BN model through a comprehensive sensitivity and structure analysis of the variables related to our target outcomes.

1.1 Objectives

- To validate the performance of the GATFBN against other BN training algorithms using both synthetic and real medical datasets.
- To perform a sensitivity analysis of the BN model to identify biases in the data and demonstrate the robustness of the model.
- To highlight the major factors contributing to the diagnosis of cardiac conditions in the medical dataset.
- To explore the potential of the GATFBN for improving diagnostic accuracy in medical applications.

2 Background

BNs are probabilistic graph models representing joint probability distributions. They are defined by a DAG and a set of parameters given the DAG structure. Within the DAG, the nodes represent a set of random variables, and the edges are the dependency among them (KOLLER; FRIEDMAN, 2009).

Recently a lot of attention has come to the study of BN because of its intrinsically interpretable features and as a way to model causal relationships among features. BN has been applied in many real-world scenarios to deal with applications such as medical, legal, finance, and risk assessment. (KADDOUR et al., 2022)

This chapter will explore fundamental BN concepts, emphasizing the structure learning process. Additionally, the chapter will delve into parameters learning and inference techniques stages. At the end of the chapter, the reader is expected to have a general understanding of what are BNs, and the most used techniques to train them as well as to perform inference.

2.1 Bayesian Networks

A Bayesian network is a probabilistic graph model that encodes a joint probability distribution (JPD). Let $V = \{X_1, X_2, \ldots, X_n\}$ be the set of random variables of interest and a DAG $G = (V, E)$, where V is the set of nodes representing the variables, and E is the set of directed edges indicating the dependencies. Each directed edge $E \in (X_i, X_j)$ represents a connection from the node X_i to X_j.

The efficiency in representing BNs stems from their DAG topology, where dependencies are represented as edges. Instead of representing all relationships among all variables, one just needs to store the random variables given their parents (PEARL, 1988), as illustrated in Equation (2.1).

This means that the joint probability distribution can be factored into a product of conditional probabilities, where each variable X_i depends only on its parent nodes in the graph.

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)) \tag{2.1}$$

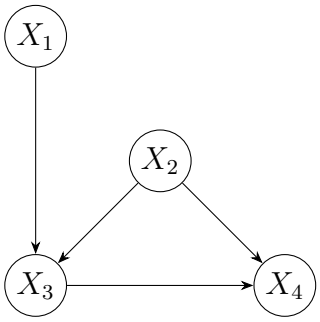

Figure 1 – A simple Bayesian Network DAG with four nodes.

In addition to the DAG as shown in Figure 1, another integral component of a discrete BN is the set of Conditional Probability Tables (CPTs), which elucidate how each node's dependencies interact with one another. Each node in the graph corresponds to a CPT, detailing the conditional probabilities based on its parent nodes. As an illustrative example, the Table 1 exemplifies the CPT for the node X_1.

Assuming all the random variables follow the equation 2.2:

$$P(X = x) = \begin{cases} p & \text{if } x = 1 \\ 1 - p & \text{if } x = 2 \end{cases} \tag{2.2}$$

$P(X_1 = 1)$	$P(X_1 = 2)$
$p(x_{1,1})$	$p(x_{1,2})$

Table 1 – Conditional Probability Table for X_1

X_1	X_2	$P(X_3 = 1 \mid X_1, X_2)$	$P(X_3 = 2 \mid X_1, X_2)$
$x_{1,1}$	$x_{2,1}$	$p(x_{3,1} \mid x_{1,1}, x_{2,1})$	$p(x_{3,2} \mid x_{1,1}, x_{2,1})$
$x_{1,2}$	$x_{2,1}$	$p(x_{3,1} \mid x_{1,2}, x_{2,1})$	$p(x_{3,2} \mid x_{1,2}, x_{2,1})$
$x_{1,1}$	$x_{2,2}$	$p(x_{3,1} \mid x_{1,1}, x_{2,2})$	$p(x_{3,2} \mid x_{1,1}, x_{2,2})$
$x_{1,2}$	$x_{2,2}$	$p(x_{3,1} \mid x_{1,2}, x_{2,2})$	$p(x_{3,2} \mid x_{1,2}, x_{2,2})$

Table 2 – Conditional Probability Table for X_3 given X_1 and X_2

As observed in Table 1, the variable X_1 does not have any parents, and thus, there is no need to condition its CPT. On the other hand, as demonstrated in Table 2, for each value of the child node, conditioning on its parents necessitates the specification of corresponding parameters. In addition, because of the laws of probability every row must add to one.

2.1.1 Learning BN

Learning BN typically involves two key stages: structure learning and parameter learning. Structure learning entails discovering DAG topology, while parameter learning

involves estimating the CPTs based on the learned topology. Structure learning, widely regarded as the more challenging phase, has spurred numerous strategies in the literature, owing to its NP-hard nature (CHICKERING, 2002).

Various approaches exist for structure learning, ranging from leveraging prior domain knowledge to employing statistical and machine learning techniques (TSAMARDINOS; BROWN; ALIFERIS, 2006). Broadly, methods for structure learning fall into three main categories: scoring function optimization, statistical independence testing, and hybrid methods amalgamating both approaches.

Scoring function optimization entails evaluating candidate DAG structures using scoring metrics like Bayesian Information Criteria (BIC), Bayesian Dirichlet, and K2. These metrics can be classified into two groups: information-theoretic scoring functions, using concepts such as log-likelihood or Kullback-Leibler (KL) Divergence, and Bayesian scoring functions, often based on the Bayesian Dirichlet function with prior assumptions variations about DAG distributions (CARVALHO; PEREIRA; CARDOSO, 2019).

Within scoring function optimization, diverse optimization algorithms are employed, ranging from greedy methods like Hill-climbing, TABU-Search, and Simulated Annealing to more sophisticated population-based algorithms such as differential evolution, bee and ant colonies, and genetic algorithms (TSAMARDINOS; BROWN; ALIFERIS, 2006).

Alternatively, constraint-based methods rely on statistical tests to infer graph structures. However, for this discussion, which focuses on structured learning through scoring functions, we refrain from delving further into constraint-based methods. Interested readers can explore other studies for deeper insights into these alternative approaches (KITSON et al., 2023).

2.1.2 Inferences in Bayesian Networks

The process of making inferences in Bayesian networks involves computing the posterior distribution of certain variables given evidence about other variables. This capability is central to many applications of BNs, including diagnosis, prediction, and decision making in fields ranging from artificial intelligence and machine learning to bioinformatics and economics (SCANAGATTA; SALMERóN; STELLA, 2019).

Inference techniques in BNs can be divided into two groups: exact inference and approximate inference. While exact inference uses the rules of probability to extract the posterior given the evidence, approximate inference aims to achieve similar results through stochastic simulation of the posterior distribution. The complexity of the BN (i.e., the number of nodes) directly influences the method chosen for inference (JENSEN, 1996). Approximate methods are more suitable for large networks (greater than 50 nodes) or

applications where fast inference is required. In this study, since our BNs are not considered large and we do not have significant time constraints, the exact method was chosen for application. The following sections briefly describe exact inference by variable elimination and some techniques of approximate inference.

2.1.2.1 Exact Inference

Exact inference, particularly through the method of variable elimination, is an essential algorithm used to perform inference in BNs. The general idea behind this method is to continuously marginalize and aggregate the evidence variables until we are only left with the target variable, thereby obtaining our desired posterior probability. This approach can also be conceptualized as a systematic factorization of the evidence (DARWICHE, 2009b).

Through variable elimination, we can achieve the exact posterior of a variable given the evidence. The steps involved in variable elimination include:

1. Decomposing the joint probability distribution into a product of factors, each representing a conditional probability from the BN.
2. Iteratively summing out the non-target variables, aggregating the remaining factors.

The general formula for exact inference in a Bayesian Network is as follows:

Let $\mathbf{X}$ be the set of all variables in the Bayesian Network, $\mathbf{E}$ be the set of evidence variables, $\mathbf{Y}$ be the set of query variables, and $\mathbf{Z} = \mathbf{X} \setminus (\mathbf{Y} \cup \mathbf{E})$ be the set of hidden variables. The goal is to compute the posterior distribution $P(\mathbf{Y}|\mathbf{E})$.

The joint distribution of all variables in the Bayesian Network can be written as:

$$P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i|\text{Pa}(X_i)), \tag{2.3}$$

where $\text{Pa}(X_i)$ denotes the parents of X_i in the network.

The posterior distribution $P(\mathbf{Y}|\mathbf{E})$ can be computed by summing out the hidden variables $\mathbf{Z}$ from the joint distribution and normalizing by the marginal probability of the evidence:

$$P(\mathbf{Y}|\mathbf{E}) = \frac{P(\mathbf{Y}, \mathbf{E})}{P(\mathbf{E})}. \tag{2.4}$$

First, compute the joint distribution of $\mathbf{Y}$ and $\mathbf{E}$ by summing out the hidden variables $\mathbf{Z}$:

$$P(\mathbf{Y}, \mathbf{E}) = \sum_{\mathbf{Z}} P(\mathbf{Y}, \mathbf{Z}, \mathbf{E}) = \sum_{\mathbf{Z}} \prod_{i=1}^{n} P(X_i|\text{Pa}(X_i)). \tag{2.5}$$

Then, compute the marginal probability of the evidence $\mathbf{E}$ by summing out all non-evidence variables:

$$P(\mathbf{E}) = \sum_{\mathbf{X}\setminus\mathbf{E}} P(\mathbf{X}) = \sum_{\mathbf{X}\setminus\mathbf{E}} \prod_{i=1}^{n} P(X_i|\text{Pa}(X_i)). \tag{2.6}$$

Finally, the posterior distribution is given by:

$$P(\mathbf{Y}|\mathbf{E}) = \frac{\sum_{\mathbf{Z}} \prod_{i=1}^{n} P(X_i|\text{Pa}(X_i))}{\sum_{\mathbf{X}\setminus\mathbf{E}} \prod_{i=1}^{n} P(X_i|\text{Pa}(X_i))}. \tag{2.7}$$

However, the computational complexity of variable elimination increases exponentially with the number of variables and the network's connectivity. This makes it impractical for application in large BNs.

2.1.2.2 Approximate Inference

When exact inference is computationally infeasible, approximate inference methods need to be employed. These methods provide approximate solutions to the posterior distributions and are generally more scalable (KOLLER; FRIEDMAN, 2009). Key techniques include:

2.1.2.2.1 Forward Sampling

Forward sampling is a straightforward method where samples are generated by sampling each variable in the network according to its conditional distribution given its parents. This process starts from the root nodes and proceeds down to the leaf nodes, ensuring that all parent variables are sampled before their children.

Example: Consider a simple BN with nodes $A \rightarrow B \rightarrow C$. To perform forward sampling, we:

1. Sample A from its prior distribution $P(A)$.
2. Sample B from its conditional distribution $P(B|A)$.
3. Sample C from its conditional distribution $P(C|B)$.

2.1.2.2.2 Weighted Sampling

Weighted sampling, also known as likelihood weighting, improves upon forward sampling by assigning weights to the samples based on how well they match the observed

evidence. For each sample, the weight is the product of the probabilities of the evidence variables given their parents.

Example: Consider a BN with nodes $A \to B \to C$ and evidence $B = b$. To perform weighted sampling, we:

1. Sample A from its prior distribution $P(A)$.
2. Assign a weight $w = P(B = b|A)$.
3. Sample C from its conditional distribution $P(C|B = b)$.

Repeat the process to generate many samples and compute the weighted average of the query.

2.1.2.2.3 Gibbs Sampling

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method that generates samples from the posterior distribution by iteratively sampling each variable from its conditional distribution given all other variables.

Example: Consider a BN with nodes $A \to B \to C$. To perform Gibbs sampling, we:

1. Initialize A, B, C to some values.
2. Sample A from its conditional distribution $P(A|B, C)$.
3. Sample B from its conditional distribution $P(B|A, C)$.
4. Sample C from its conditional distribution $P(C|A, B)$.
5. Repeat the steps for a large number of iterations.

As the process continues, the samples converge to the joint distribution of the network.

2.2 Genetic Algorithms

2.2.1 Introduction

Genetic algorithms (GA) represent a class of optimization techniques inspired by natural evolutionary processes. These algorithms typically initiate with a population of individuals, each representing a feasible solution to a given problem. Through iterations

of selection, reproduction, and competition, GA refines the population, leading to the emergence of increasingly more fit individuals (GOLDBERG, 1989b).

The applicability of GA varies from various optimization problems, particularly those where traditional optimization methods falter due to intricate cost functions or combinatorial complexities. Notably, GA excels in tackling combinatorial optimization challenges like the Traveling Salesman Problem (TSP), where the search space is vast and discrete.

In the realm of BN structural learning, GA has been a subject of exploration since 1995, with numerous variations documented in the literature (LARRANAGA, 1996) (CONTALDI; VAFAEE; NELSON, 2019) (SUN; ZHOU, 2022) applying diverse implementations, a spectrum of cost functions and individual representations. Approaches to BN structure learning diverge, with some strategies segmenting the problem into distinct populations, while others exploit problem constraints for enhanced efficiency (LEE; BEEK, 2017).

More recently, stochastic optimization techniques have gained traction, offering promising results in BN structure learning (KITSON et al., 2023). These methods leverage probabilistic searches to explore solution spaces more effectively, enhancing the robustness of structural learning algorithms.

In summary, GA and related optimization approaches serve as powerful tools for addressing complex optimization challenges, including BN structure learning.

2.2.2 Basic structure of the Genetic algorithm

Most GAs follow a basic workflow that iterates over a set of individuals until a stopping criterion is achieved. The GATFBN also adopts this framework but with some modifications, as described in algorithm 1. A more in-depth analysis will be performed in the next chapters, explaining how each function was implemented. For now, we will simply illustrate the workflow of the GATFBN.

Algorithm 1 GA tailored for BN (GATFBN)

```
Generate the initial population
Evaluate fitness of the initial population
while stopping condition not met do
    Select parents
    if p_c > rand() then
        child = Crossover(parents)
        child = Repair(child)
    else
        child = parents
    end if
    if p_m > rand() then
        child = Mutation(child)
    end if
    Select one child and perform a local search
    Evaluate the fitness of the child
    Select survivors
    Update population
end while
Return final solution
```

2.2.3 The Bayesian Information Criterion

The Bayesian Information Criterion (BIC) was chosen as the scoring function for the proposed Genetic Algorithm (GA). BIC is preferred due to its reliability in model selection and its ability to balance model fitness to data and network complexity (SCHWARZ, 1978). The BIC score is computed as follows: Given:

- r_j: the number of states of the finite random variable X_i,
- x_{ik}: the k-th value of X_i,
- N_{ijk}: the number of observations in the data where the variable X_i takes its k-th value and the variables in $\prod X_i$ take their j-th configuration,
- N_{ij}: the number of instances in the data where the variables in $\prod X_i$ take their j-th configuration,
- obs: the number of samples used,
- n: the number of nodes of the DAG,
- B: a measure of the size of the BN,

The following formulas can compute the value of the scoring function (CARVALHO; PEREIRA; CARDOSO, 2019):

$$\theta_{ijk} = \frac{N_{ijk}}{N_{ij}} \tag{2.8}$$

$$q_i = \prod_{X_j \in \prod_{X_i}} r_j \tag{2.9}$$

$$\log(L) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log(\theta_{ijk}) \tag{2.10}$$

$$\text{BIC} = \log(L) - \frac{\log(obs)}{2} B \tag{2.11}$$

$$\text{B} = \sum_{i=1}^{n} (r_j - 1) q_i \tag{2.12}$$

The idea behind the BIC score is to balance model complexity, as represented in Equation 2.11. The first part of the equation represents the amount of information carried by the model as the log-likelihood, while the second part accounts for the parameters required to encode this information. The BIC value will only increase if the connections of the BN carry enough information to justify the burden of adding a new edge.

2.2.4 Genotype of BN in the GA context

The genotype of an individual stands as a pivotal aspect in the efficacy of a GA. A succinct and well-designed representation not only facilitates exploration and exploitation of the solution space but also enhances computational efficiency. While certain problems inherently possess a natural genotype representation, it may not always be the most efficient choice.

In light of this, we introduce a novel genotype representation for stochastic optimizations applied in the context of BNs. Traditionally, the adjacency matrix of the DAG serves as the default representation, owing to its intuitive appeal. However, we propose a refined approach that streamlines the genotype representation, reducing its degrees of freedom while retaining the ability to encompass all possible DAG configurations.

By refining the genotype representation, we aim to enhance the GA's ability to navigate and converge upon optimal solutions within the BN framework. This novel representation not only mitigates computational complexity but also leads to a more coherent exploration and exploitation of the solution space, ultimately augmenting the GA's efficacy in BN structural learning tasks.

2.2.4.1 The usual representation

In the realm of evolutionary approaches applied to Bayesian network structure learning, the prevalent use of the adjacency matrix representation , as pictured in Table 3, for DAGs is ubiquitous. While intuitive , this representation is not without its drawbacks. One of the foremost challenges lies in its potential to represent graphs that violate the acyclic property of DAGs (CARVALHO; PEREIRA; CARDOSO, 2019).

Encountering non-DAG individuals poses a significant hurdle, particularly given that the cost function is strictly defined for DAGs. Consequently, the need for repair mechanisms becomes imperative. Many studies resort to repair operators aimed at transforming non-DAG structures into valid DAGs (KITSON et al., 2023). Typically, these operators employ random edge deletions until the resulting graph no longer has cycles.

However, this repair strategy isn't without its shortcomings. The indiscriminate deletion of edges can inadvertently disrupt the inherent structure of the solution space, leading to a loss of directionality and coherence in the genetic algorithm's search trajectory. Consequently, the efficiency and effectiveness of the genetic algorithm may be compromised.

Refining the genotype representation to mitigate the occurrence of non-DAG structures could offer avenues for enhancing the efficiency and efficacy of genetic algorithms in the context of Bayesian network structure learning (CAMPOS et al., 2002). By addressing these inherent challenges, we can strive towards more robust and efficient evolutionary approaches tailored for structure learning in Bayesian networks.

Table 3 – Adjacency Matrix for the Bayesian Network DAG

	X_1	X_2	X_3	X_4
X_1	0	0	1	0
X_2	0	0	1	1
X_3	0	0	0	1
X_4	0	0	0	0

2.3 Interpretability of BNs

Bayesian Networks provide a clear and interpretable framework for understanding the relationships among variables in a dataset. The graphical structure of a BN consists of nodes (representing variables) and directed edges (representing conditional dependencies). This structure allows us to visualize and understand the causal relationships between variables (KORB; NICHOLSON, 2010)

The interpretability of BNs is particularly useful in identifying key factors that influence the target variable. The parents of a node and the variables within its Markov

blanket are crucial for understanding the direct and indirect influences on the target variable. The Markov blanket of a variable includes its parents, its children, and the other parents of its children. This localized structure around a variable provides a concise summary of the factors that affect it (DARWICHE, 2009b).

By examining the structure of the BN, we can gain insights into the causal mechanisms underlying the data. This helps in making informed decisions, understanding the potential impact of interventions, and identifying critical variables for further investigation (FRIEDMAN, 2004).

2.4 Classification vs Clusterization

In machine learning, classification and clusterization (clustering) are two fundamental approaches to analyzing data. Both techniques serve different purposes and are applied based on the nature of the problem and the type of insights required.

2.4.1 Classification

Classification involves assigning predefined labels to instances based on their features. It is a supervised learning approach where the model is trained on labeled data to learn the mapping from input features to output labels. Common classification algorithms include logistic regression, decision trees, support vector machines, and ensemble methods like XGBoost (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

In this study, we used classification to predict whether a patient would develop a specific cardiac condition based on their medical history and other attributes. The models were evaluated using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC to assess their performance in distinguishing between different classes.

2.4.2 Clusterization (K-means)

Clustering is an unsupervised learning technique that groups instances into clusters based on their similarity. Unlike classification, clustering does not require labeled data. The goal is to partition the data into distinct clusters such that instances within the same cluster are more similar to each other than to those in other clusters.

K-means is a widely used clustering algorithm that partitions the data into k clusters. The algorithm iteratively assigns instances to the nearest cluster center and updates the cluster centers based on the mean of the assigned instances. The process continues until convergence, resulting in clusters that capture the underlying structure of the data.

In this study, K-means was used to cluster the textual annotations into distinct groups. This helped in transforming the textual data into categorical variables, which were then used in the Bayesian Network and XGBoost models.

2.5 ROC Curve

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a classification model. It plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings. The area under the ROC curve (AUC) is a measure of the model's ability to distinguish between positive and negative classes (FAWCETT, 2006).

A model with an AUC of 1.0 indicates perfect classification, while an AUC of 0.5 suggests random guessing. The ROC curve provides a comprehensive view of the trade-offs between sensitivity and specificity, allowing us to select an optimal threshold for making decisions.

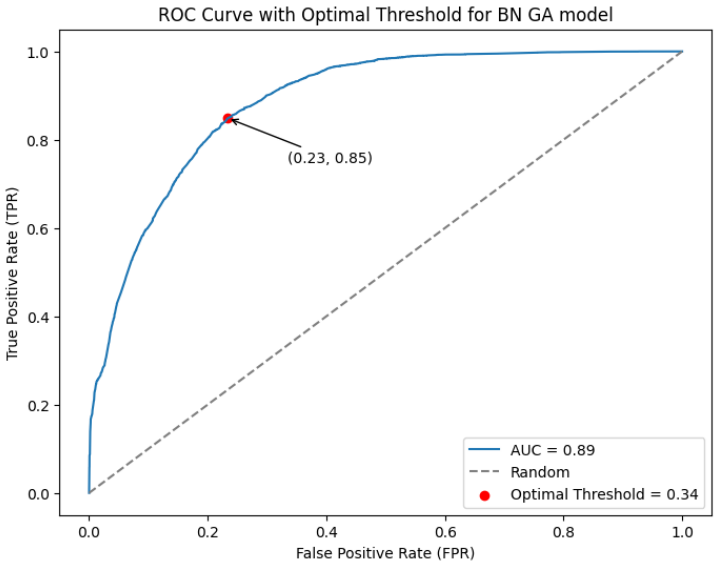

Figure 2 – Example ROC Curve. The x-axis represents the False Positive Rate (1 - Specificity), and the y-axis represents the True Positive Rate (Sensitivity). The dashed line represents a random classifier with an AUC of 0.5. The optimal threshold is selected based on the trade-off between sensitivity and specificity.

In this study, the ROC curve was used to evaluate the performance of the Bayesian Networks and XGBoost model. The optimal threshold identified from the ROC curve was applied to the test data to make classification decisions.

2.6 XGBoost

XGBoost (Extreme Gradient Boosting) is a powerful and efficient implementation of the gradient boosting framework. It is widely used in machine learning competitions and real-world applications due to its high performance and scalability.

XGBoost builds an ensemble of decision trees, where each tree is trained to correct the errors of the previous trees. This sequential learning process enhances the model's predictive power. XGBoost incorporates several optimizations, such as regularization, sparsity-aware learning, and weighted quantile sketch, to improve its performance and prevent overfitting (CHEN; GUESTRIN, 2016).

In this study, XGBoost was chosen for its superior performance in tabular data and its built-in interpretability features. The variable importance function in XGBoost provides insights into the most influential features for the classification task, aiding in the interpretability of the model.

2.7 NLP

Natural Language Processing (NLP) techniques were employed to process the textual annotations in the dataset. The goal was to transform the unstructured textual data into a structured format that could be used in the Bayesian Network and XGBoost models.

2.7.1 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents (corpus) (RAMOS, 2003). It combines two components:

- **Term Frequency (TF):** Measures how frequently a word appears in a document. Higher frequency indicates greater importance within the document.
- **Inverse Document Frequency (IDF):** Measures how unique a word is across the corpus. Words that appear in many documents have lower IDF values, indicating they are less informative.

The TF-IDF score is calculated as the product of TF and IDF. Words with high TF-IDF scores are considered more important for the document. In this study, TF-IDF was used to convert the textual annotations into numerical vectors. These vectors were then clustered using K-means to transform the textual data into categorical variables, making it suitable for use in the Bayesian Network and XGBoost models.

By leveraging TF-IDF and clustering techniques, we ensured that the rich information in the textual data was effectively utilized, enhancing the overall performance and interpretability of the models.

2.8 Evaluated Using Metrics

The models were evaluated using several performance metrics to assess their effectiveness in classification tasks. These metrics include accuracy, precision, recall, and F1 score.

2.8.1 Accuracy

Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined. It provides a general measure of how well the model performs across all classes.

2.8.2 Precision

Precision, also known as positive predictive value, is the proportion of true positive results in all positive predictions. It is calculated as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.13}$$

High precision indicates that the model has a low false positive rate.

2.8.3 Recall (Sensitivity)

Recall, or sensitivity, is the proportion of true positive results in all actual positives. It is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.14}$$

High recall indicates that the model has a low false negative rate.

2.8.4 F1 Score

The F1 score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall, making it useful for evaluating models where an uneven class distribution may be present. The F1 score is calculated as:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.15}$$

2.8.5 Mutal information

Mutual information is a non-parametric metric used to measure the dependency between random variables. It is computed as the Kullback-Leibler (KL) divergence (KULLBACK; LEIBLER, 1951) between the joint distribution and the product of the marginal distributions of the variables. The idea behind this metric is that if the joint distribution and the product of the marginal distributions are the same, the mutual information will be zero, indicating that the two distributions do not share any information. Mutual information $I(X;Y)$ between two discrete random variables X and Y can be defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{2.16}$$

Where:

- $p(x,y)$ is the joint probability distribution of X and Y.
- $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y, respectively.

Mutual information can also be expressed using Kullback-Leibler (KL) divergence D_{KL} as:

$$I(X;Y) = D_{\mathrm{KL}}(p(X,Y) \parallel p(X)p(Y)) \tag{2.17}$$

Where:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right) \tag{2.18}$$

2.8.6 Correlation

Correlation is a linear measure of the similarity between two random variables. It is computed as follows and can range from -1 to 1. It can be interpreted as how much one variable is linearly dependent on the other. A correlation of 1 indicates a perfect positive linear relationship, -1 indicates a perfect negative linear relationship, and 0 indicates no linear relationship.

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \tag{2.19}$$

Where:

- $\rho_{X,Y}$ is the correlation coefficient between variables X and Y.

- $\text{Cov}(X, Y)$ is the covariance of X and Y.
- σ_X and σ_Y are the standard deviations of X and Y, respectively.

Covariance is a measure of the joint variability of two random variables. It is defined as follows:

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] \tag{2.20}$$

Where:

- $\text{Cov}(X, Y)$ is the covariance between variables X and Y.
- $\mathbb{E}$ denotes the expected value.
- X and Y are the random variables.
- μ_X and μ_Y are the means of X and Y, respectively.

3 Methodology

The methodology of this study is divided into two parts. The first part focuses on the creation of a new Genetic Algorithm (GATFBN) for training BNs. The second part involves applying this GA to a real-world dataset from cardiac patients of the Heart Institute of Buracana in Colombia(SILVA et al., 2022).

3.1 The novel GA

A novel Genetic Algorithm (GATFBN) was proposed in this study. While using the core concepts of most Genetic Algorithms, such as crossover, mutation, and population selection, our innovation lies in the implementation of these operators. Specifically, the mutation operator was designed to leverage the advantages of the new representation proposed in Chapter 2. The following sections provide a brief description of the design of the GATFBN.

3.1.0.1 The New representation

A new representation was adopted for the individuals in the GATFBN. The usual representation adopted by evolutionary algorithms for BN structure learning consists of the graph's adjacency matrix. However, this representation is not efficient. For example, let's consider that the degree of freedom of a DAG is given by the formula $Dof = n * (n-1)/2$ and that we have $n * n$ degrees of freedom in an adjacent matrix representation. During a search, the GA will inevitably reach unfeasible individuals (graphs that are not DAGs) when using the adjacency matrix representation. A new representation was adapted from (CARVALHO, 2011) to mitigate this issue.

In the representation proposed by (CARVALHO, 2011) , it was assumed a pre-established topological order among the variables, hence two symbols were needed, "0" and "1", since there were edges in only one direction. However, in this study, the "-1" symbol is also used to represent an edge. One of the consequences of the new symbol is that the graph representations are not unique and can contain cycles. Yet, the dimensional search space was reduced compared with the traditional representation. Also, the operators of the GATFBN are more naturally encoded. For instance, to reverse an edge with this new representation, one must change the symbol "1" to "-1" in the same string position. One drawback is that there is still a need for the repair operator.

To exemplify the representation proposed, let's assume a DAG with four nodes, "A", "B", "C", and "D". The string in Figure 3 would represent it.

AB	AC	AD	BC	BD	CD

Figure 3 – The new representation. In the first position, the symbol "AB" would be "0" if there is no edge connecting the nodes "A" to "B", "1", if there is an edge connecting "A" to "B", and "-1" if the edge is in the reverse direction, connecting "B" to "A".

For instance, the graph in Figure 4 would be represented as described in Figure 5, and the graph in Figure 6 would be represented as described in Figure 7.

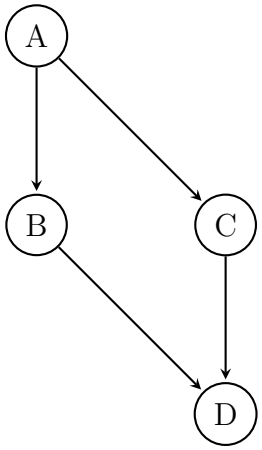

Figure 4 – A DAG that represents a BN. If one wants to reverse one of its edges, just a change in the symbol "1" to "-1" would be necessary. This change may introduce loops in the graph so one would have to check if the resultant graph is still a DAG.

1	1	0	0	1	1

Figure 5 – The proposed representation of the DAG describe in Figure 4.

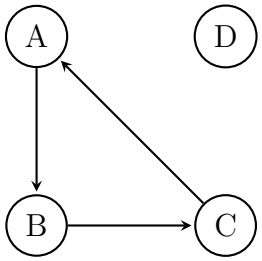

Figure 6 – A graph that contains a loop. Those graphs are not DAGs and hence cannot be used to represent a BN. A repair operator would be aplied in this graph to transform it into a DAG in the GATFBN scope.

3.1.1 Population initialization

The initial population was drawn from a multinomial probability distribution according to the equations 3.1 and 3.2. Because of the small number of edges in the initial

1	-1	0	1	0	0

Figure 7 – The genotype of the graph resented in Figure 6. Note that this graph is not a DAG because it contains a loop. The symbol "-1" indicates an edge from the node "C" to "A".

population, most individuals are DAGs. However, the DAG condition is checked, and the repair operator is applied if the individual is not a DAG. Therefore, all the initial population graphs are DAGs.

$$\text{Population} \sim \text{Multinomial}(\text{Probabilities} = [p_1, p_2, p_3]) \tag{3.1}$$

where

$$\text{Probabilities} = \left[\frac{1}{\text{Dof}}, \frac{\text{Dof}-2}{\text{Dof}}, \frac{1}{\text{Dof}}, \right]) \tag{3.2}$$

The above equations ensure that, on average, the initial population will contain individuals with at least two edges. The symbol Dof represents the length of the genotype of the population.

3.1.2 Stop condition

The stop condition was composed of two terms, the first regarding the maximum number of interactions and the second concerning the stability of the cost function of the best individual. The threshold was whether the max number of interactions reached 30000 or the value of the cost function for the best individual was stable for more than 1000 interactions.

3.1.3 Parent Selection

The parent selection was made proportionally to rank based on equation 3.3

$$P_{\text{lin\_rank}}(i) = \frac{2-s}{\mu} + \frac{2 \cdot i \cdot (s-1)}{\mu \cdot (\mu - 1)} \tag{3.3}$$

where the parameter $s = 1.3$ during all the experiments, the parameter µ represents the mean fit value for all Individuals, and parameter i is the index of the individual. After ranking the population, the parents were selected by the stochastic universal sampling algorithm (SUS)(BAKER et al., 1987). The SUS method increases the likelihood of selecting individuals with higher ranks to become parents.

3.1.4 Crossover

The crossover with 2 cut points was performed in each couple of selected parents. Two points in the string representation were randomly selected, and each parent inherent one slice of the cut. After the crossover, the new offspring is submitted to a repair operator to guarantee that the children were inside the feasible space solution.

3.1.5 Mutation

The mutation operator was implemented according to the algorithm 2. Firstly, the probability of mutation was normalized across the symbols. This was made to prevent the algorithm to keep adding edges to the individuals, especially in the early stages where most symbols in the genotype are zeros. After that, there is a small probability that the mutation can occur in more than one symbol simultaneously. This was designed to help the GA escape some eventual local optimum. Symbols ranging from one to N were randomly sampled from a uniform distribution, with N representing the dimension of the dataset used. Finally, the mutation was performed in the symbols selected and checked if the mutation violated the DAG restriction for each symbol. If it so, the mutation was undone.

To perform the mutation, one needs to change the symbol in the bit selected; for instance, if the symbol selected was "0", there was a 50% chance that this bit would mutate to "1" or "-1". How the mutation function was designed imposes that only feasible individuals are generated. This saves processing time since there is no need to use the repair operator and keeps the exploration space within the feasible solution.

Algorithm 2 Mutation Function

```
Select with equal probability in which symbol the mutation will occur
if Symbol is not present in the representation then
    Select all the other symbols
end if
if rand() > 0.9 then
    Randomly select 1 to N symbols to be flipped
else
    Randomly select 1 symbol to be flipped
end if
for i = 1 to N do
    B = bit_flip(A)
    if B ≠ DAG then
        Undo bit_flip
    end if
end for
return B
```

3.1.6 Local Search

The local search consists of a greedy search on a randomly selected child. It performs the mutation operation N times, with N equal to the number of variables in the dataset, on the same DAG. If a BN with a higher score is found, it replaces the previous DAG, as the enhanced individuals are constantly replacing the old ones.

3.1.7 Population update

The GA was designed to keep the same population size. Hence, after the offspring's production, it discards the PR worse individuals where PR is the number of parents selected.

3.1.8 Repair Operator

The repair operator checks if the individual is a DAG and applies the following steps if otherwise. Firstly, it finds the edges in the graphs where there are cycles, and then it randomly selects one of those edges to be removed until the graph has no more cycles in it. No restriction on the number of cycles was imposed before this operator. It is known that selecting all the cycles in dense graphs with a relatively medium number of nodes could be an impractical solution. However, since all graphs in the initial population are forced to be DAGs, just a few edges will form a cycle after the crossover operation. Hence, this approach does not become impractical.

3.1.9 Smoothing

The BIC function is not defined for cases where none of the variable states are found in the data. This problem can occur because of two phenomena, the combination of states is not represented in the true probability distribution, or the sample size used is too small. Laplace smoothing with the number of samples equal to one (CAMPOS; FRIEDMAN, 2006) was used in the state counts to circumvent this issue. It consists of adding pseudo-samples in the data where no evidence exists of that specific combination of states occurring. The effect of this is to bring the probability distribution closer to uniform. Consequently, The BIC scoring function will result in a low value for those states, and the GA will avoid building networks with only pseudo-samples.

3.1.10 Metrics for comparing Bayesian Networks

Comparing two different BNs can be approached as a traditional binary imbalanced classification problem. For instance, considering the conventional adjacency matrix

representation of the BNs, our objective is to quantify the similarity between the adjacency matrix of Network A and that of Network B. In this contex, Network A serves as the ground truth, while Network B represents the estimated structure(TSAMARDINOS; BROWN; ALIFERIS, 2006).

Hence, it's advantageous to use performance measures applied for imbalanced classification problems. This preference comes from the fact that most practical BN applications are represented by sparse adjacency matrices. One such metric is the F1 score as shown in equation 2.15, which combines precision and recall, making it suitable for assessing the performance of BN comparison.

In this scenario, class "1" was assigned to the edges in the network while class "0" was assigned to the absence of edges. It is crucial to notice that one more symbol is present in the graphs' genotype ("-1") due to the representation used. However, all genotypes were converted into the adjacency matrix representation using the same topological order, and then the structures were compared. So, in this way, the ground truth DAG and the DAG found by the GATFBN would have a unique representation.

Another perspective of the problem when comparing two BN is concerning the scoring function used in the training stage. It's well-established in the literature that while BN structures may have the same scoring function values they can still have different topologies, which means being equivalent(CHICKERING, 2003). Equivalent in this context means that, given two different structures they will have the same value of scoring function. This property arises from the fact that the edge directions can be reversible in some circumstances without changing the BN cost function. Consequently, comparing BNs using the method described above may yield varying F1 scores for multiple equivalent networks. To address this issue, one approach is to consider the value of the cost function when comparing BNs. Another common method, prevalent in the literature, involves utilizing the Completed Partially Directed Acyclic Graphs (CPDAG) to compare two BNs. CPDAGs offer an alternative representation for BNs, where certain edges lose their directional orientation if changing the direction of these edges does not impact the value of the cost function. Theoretically, all DAGs that yield identical values for the scoring function will share the same CPDAG structure(CHICKERING, 2002).

3.1.11 Parameters Learning in Bayesian Networks

Parameter learning in BNs involves estimating the CPTs for the network's nodes based on the structure discovered during the structure learning phase. Once the DAG topology is determined, the next step is to quantify the relationships between each node and its parents by learning the parameters that define these relationships(CARVALHO, 2009).

Two main approaches to parameter learning are Maximum Likelihood Estimation (MLE) and Bayesian Estimation. MLE is a frequentist approach that aims to find the parameters that maximize the likelihood of the observed data given the network structure. This method assumes that the data is fully observed and uses the observed frequencies to estimate the CPTs. MLE is computationally efficient and straightforward to implement, especially when the dataset is large and complete.

Bayesian Estimation, on the other hand, incorporates prior knowledge about the parameters and updates this knowledge based on the observed data. This approach uses prior distributions and combines them with the likelihood of the observed data to produce posterior distributions for the parameters. Bayesian Estimation is particularly useful when dealing with small or incomplete datasets, as it allows the incorporation of domain knowledge and provides a probabilistic framework for parameter estimation(FRIEDMAN; KOLLER, 2003).

In this study we use parameters learning by MLE by default,

3.1.12 Experimental Methodology

The GA's performance was evaluated using five databases from the Bayesian repository(CONSTANTINOU et al., 2020). To ensure a comprehensive and robust analysis, each algorithm, including GA, Hill-Climb, Tabu Search, Incremental Association Markov Blanket (IAMB), Max-Min Hill-Climbing (MMHC), and Hybrid Two-Phase Clustering (H2PC), was executed 20 times. The F1 score was used to compare the best individual obtained by the GA with the ground truth DAGs, providing a comprehensive assessment of the algorithm's ability to accurately learn the true structure of the data.

3.2 Study case on Real word application

3.2.1 Application to Real-World Data

For the second part of the study, we used a comprehensive database from cardiac patients at the Heart Institute of Buracana in Colombia. This work was approved by the Ethics Committee of the Heart Institute of Buracana, ensuring that all research protocols adhered to the relevant ethical guidelines. This database includes both physical data (e.g., sex, height, weight) and detailed textual annotations about family history, drug use, surgeries, medications, and diagnosed diseases during consultations(SILVA et al., 2022).The dataset encompasses over 1,000 different diseases categorized using the ICD-10 codes(ORGANIZATION, 2019).

The raw data contained numerous NAN values, grammatical mistakes in the

textual variables, and many duplicate rows. The initial dataset consisted of approximately 700,000 lines of data related to more than 20,000 unique patients. Given the condition of the data, several pre-processing steps were necessary to ensure the data was suitable for our analyses. Additionally, because the developed GA only uses categorical data, continuous variables were discretized.

The data transformation process involved using Natural Language Processing (NLP) techniques to convert textual data into categorical data and aggregating the disease column to analyze only diseases prior to the diagnosis of Coronary Heart Disease (EC).

To facilitate our analysis, we aggregated the disease data into two major groups: Cardiac (EC) and Atrial Fibrillation (FA). The specific ICD-10 codes used for this aggregation are as follows, based on a study conducted under medical specialist supervision (SILVA et al., 2022):

3.2.1.1 FA Disease Codes

The diseases grouped under Atrial Fibrillation (FA) included:

- I48X: Atrial Fibrillation and Flutter
- I49X: Other Cardiac Arrhythmias
- I489: Unspecified Atrial Fibrillation and Flutter

3.2.1.2 EC Disease Codes

The diseases grouped under Cardiac conditions (EC) included:

- I420: Dilated Cardiomyopathy
- I500: Heart Failure
- I214: Acute Subendocardial Myocardial Infarction
- I255: Chronic Ischemic Heart Disease
- I208: Other Acute Myocardial Infarction
- I209: Unspecified Acute Myocardial Infarction
- I251: Atherosclerotic Heart Disease of Native Coronary Artery
- I210: Acute Transmural Myocardial Infarction of Anterior Wall
- I252: Old Myocardial Infarction

- I211: Acute Transmural Myocardial Infarction of Inferior Wall
- I219: Acute Myocardial Infarction, Unspecified
- I212: Acute Transmural Myocardial Infarction of Other Sites
- I213: Acute Transmural Myocardial Infarction of Unspecified Site
- Z951: Presence of Aortocoronary Bypass Graft
- Z955: Presence of Coronary Angioplasty Implant and Graft
- I240: Coronary Thrombosis Not Resulting in Myocardial Infarction
- I200: Unstable Angina
- I249: Acute Ischemic Heart Disease, Unspecified
- I250: Chronic Ischemic Heart Disease, Unspecified
- I256: Silent Myocardial Ischemia
- I258: Other Forms of Chronic Ischemic Heart Disease
- I220: Subsequent Myocardial Infarction of Anterior Wall
- I201: Angina Pectoris with Documented Spasm
- I221: Subsequent Myocardial Infarction of Inferior Wall
- I228: Subsequent Myocardial Infarction of Other Sites
- I229: Subsequent Myocardial Infarction of Unspecified Site

Other ICD-10 codes were also retained for additional analysis. The data processing steps were as follows:

3.2.2 Data Processing

3.2.2.1 Hardware Specifications

In this section, we outline the hardware configuration used to conduct the experiments.

- **Processor:** 13th Gen Intel® Core™ i7-13700KF, 3.40 GHz
- **Installed RAM:** 32.0 GB
- **Operating System:** Windows

3.2.2.2 BN Libraries Used

Throughout this work, several development environments were utilized. The GATFBN was implemented in MATLAB R2024B, while the other structure learning algorithms were obtained from the `bnlearn` library (SCUTARI, 2009) in R. The comparisons of the BNs were coded in Python 3.5, extensively using the `pgmpy` library (ANKAN; PANDA, 2015a).

3.2.2.3 Disease Grouping

We aggregated the diseases into two groups: Cardiac (EC) and Atrial Fibrillation (FA), while also retaining other ICD-10 codes for further analysis.

3.2.2.4 Data Preparation

For each patient, we retained data from previous consultations until they were diagnosed with an EC condition according to our prior grouping. If a patient was not diagnosed with any EC condition, all their data was retained. For those diagnosed with cardiac conditions, any data collected post-diagnosis was discarded.

3.2.2.5 Data Transformation

- **One-Hot Encoding:** We transformed the diagnosis data into a one-hot encoded format. To ensure that the categorical data remained manageable for the GATFBN, given that training a BN with thousands of variables would be too time consuming we employed several feature selection techniques. :
  - **Mutual Information:** We computed the mutual information between each feature and the target variable, retaining the 20 most informative features.
  - **Correlation:** We also calculated the correlation between each feature and the target variable, keeping the top 20 features based on their correlation values.
  - **Frequency:** Additionally, we identified the 20 most frequent features.

  We then combined these selected features, resulting in a comprehensive set of the most relevant variables for analysis. This combined set ensured that overlapping features across different selection criteria were included, maintaining a total of 20 key features as describe in Table 4.

ICD-10 Code	**Disease Name**
B572	Cytomegaloviral disease
E039	Hypothyroidism, unspecified
EC	Cardiac disease
FA	Atrial fibrillation
I10X	Essential (primary) hypertension
I340	Mitral (valve) insufficiency
I351	Aortic (valve) insufficiency
I442	Atrioventricular block, complete
I471	Paroxysmal tachycardia, unspecified
I495	Sick sinus syndrome
I499	Cardiac arrhythmia, unspecified
J449	Chronic obstructive pulmonary disease, unspecified
Q211	Atrial septal defect
R060	Dyspnea
R072	Chest pain on breathing
R074	Chest pain, unspecified
R55X	Syncope and collapse
Z136	Encounter for screening for other specified infectious and parasitic diseases
Z950	Presence of cardiac pacemaker
Z958	Presence of other cardiac and vascular implants and grafts

Table 4 – List of Retained Variables with Disease Names

- **Aggregation:** For textual and categorical features (such as gender), we retained the row with the least number of NaN values. This approach ensured that if a patient had multiple rows related to different diseases, only one row per patient was included in the final dataset. For physical variables (e.g., age, height, and weight), we computed the mode across those rows, excluding any NaN values, to retain the most frequently occurring value.

- **Discretization:** The weight and height variables were discretized into five categories. These categories were created by dividing the data into four equal intervals. The bins used for discretization were defined as follows:

  – For height: the intervals were set between -1 and 0, then from 140 to 180 in increments of 20, and finally up to the maximum height value.

  – For weight: the intervals were set between -1 and 0, then from 40 to 120 in increments of 20, and finally up to the maximum weight value.

  Additionally, the BMI was computed and discretized into 10 categories with equal distances from the minimum to the maximum value. Age was also discretized similarly; however, the categories were designed to closely match the decades in which the individuals were born.

- **Text Processing:** Textual variables were processed by removing stop words, applying a grammar corrector, and translating to English (since the MATLAB Natural Language Processing toolbox used does not support Spanish). The text was then lemmatized and converted into vectors using the TF-IDF method. These vectors were clustered into four different clusters using K-means, replacing textual values with their respective cluster labels.

3.2.3 Model Training and Comparison

The processed data was used to train several models for comparative analysis: an XGBoost model, a Bayesian Network (BN) using the Genetic Algorithm (GA), and a BN using the TABU search algorithm available in the Bnlearn library in R(SCUTARI, 2009). The performance of these models was then evaluated and compared according the Model Performance Metrics described in sequence.

3.2.3.1 Training and Testing

The final dataset, comprising 29,486 rows, was divided into training (80% - 23,589 rows) and test (20% - 5,897 rows) sets to effectively evaluate model performance.

3.2.3.2 Model Performance Metrics

The performance of each model (XGBoost, BN-GA, and BN-TABU) was evaluated based on standard classification metrics, including:

- **Confusion Matrix:** A summary of prediction results on a classification problem, showing the number of true positives, true negatives, false positives, and false negatives.
- **Precision (Positive Predictive Value, PPV):** The proportion of true positive results among all positive predictions.
- **Recall (Sensitivity, True Positive Rate, TPR):** The proportion of true positive results among all actual positives.
- **F1 Score:** The harmonic mean of precision and recall.
- **AUC-ROC:** The area under the receiver operating characteristic curve, indicating the model's ability to distinguish between classes.
- **False Discovery Rate (FDR):** The proportion of false positive results among all positive predictions.
- **False Negative Rate (FNR):** The proportion of false negative results among all actual positives.

For the Bayesian Network models, we also considered the value of the cost function K2 and the BIC as implemented in the PGMPY library(ANKAN; PANDA, 2015a).

3.2.3.3 Classification using Bayesian Networks

Unlike traditional classification models, queries in BN return the probability among the classes. To address this, we computed the ROC curve for the training data using both BN models and found the optimal threshold based on Youden's J statistic. Youden's J statistic is a measure used to identify the point on the ROC curve that maximizes the difference between the true positive rate (sensitivity) and the false positive rate (1-specificity). This statistic helps in selecting a threshold that balances sensitivity and specificity. Once this threshold was determined, inference was performed on the test dataset. If the probability value was greater than the threshold, the diagnosis was considered positive.

Another possibility was to use Maximum A Posteriori (MAP) queries for classification (HECKERMAN, 1998). However, we decided not to use this approach in favor of the threshold-based method described above.

3.2.3.4 Steps for Evaluating BN Performance

1. **Compute ROC Curve:** For the training data, compute the ROC curve for both BNs (GA and TABU).
2. **Select Threshold:** Identify the optimal threshold value from the ROC curve based on Youden's J statistic.
3. **Apply Threshold:** Perform inference on the test dataset. If the inferred probability is greater than the threshold, classify the patient as positive.

3.2.3.5 Interpretability Comparison

For the BNs, the structure of the network itself was used as an interpretable way to understand the relationships among the variables. It is known that the variables in the Markov blanket are the major factors influencing the target variable(PEARL, 1988). This inherent interpretability of BNs allows for a clear visualization of the relationships between variables.

In the case of XGBoost, the built-in function for variable importance was utilized. This function lists the variables that are most correlated with or have the most importance for the classification task in XGBoost. By comparing the variable importance from XGBoost with the structure of the BNs, we could quantitatively assess what the causal models (BNs) and the correlation models (XGBoost) considered for their outputs.

This comparison provided valuable insights into the different factors influencing model predictions, highlighting the strengths of causal inference in BNs and the effectiveness of correlation-based importance in XGBoost.

3.2.3.6 Steps for Interpretability Comparison

1. **Bayesian Network Interpretation:** Analyze the structure of the BNs to identify the Markov blanket of the target variable. This provides an understanding of the relationships and the key influencing factors.
2. **XGBoost Variable Importance:** Use the variable importance function in XGBoost to determine the most important features for the classification task. This highlights which variables are most correlated with the target outcome.
3. **Qualitative Comparison:** Compare the key variables identified by the BNs and XGBoost, Assess how the factors in BNs align with the important features in XGBoost. This provides a comprehensive understanding of the interpretability of all models.

By following these steps, we were able to leverage the strengths of all models , offering a robust framework for understanding the underlying factors driving the predictions in our study.

4 Results and Discussion

The results and discussion are divided into two major sections. The first section focuses on comparing the GATFBN with other classical BN structure learning algorithms. The second section conducts a comprehensive model analysis by comparing four different models applied to the study case data.

4.1 Comparison of Genetic Algorithm with Classical Training Algorithms

After computing the results, the distributions of the F1-scored were compared using the Wilcoxon rank-sum statistical test. Figure 8 shows the distributions of the observed F1 score for the five data sets and the seven tested algorithms. The Wilcoxon test is a non-parametric test for two populations with independent samples. The null hypothesis is that the population's medians are equal. Hence, since we were only interested in comparing the GA performance concerning the others' algorithms, the obtained median of the proposed GA was used as the reference value for the tested hypothesis(WILCOXON, 1992).

The differences between the GA median value and the others with their respective P values are presented in Table 5.

We can notice that the GA outperformed all other algorithms except in the Property data set, where the P values indicate no statistical difference between the GA and MMHC. Also, HC and H2PC outperformed the GA in this data set. For the Asia and Sports data set, the GA outperformed all others. It is interesting to notice that the variance for those data sets was approximately zero, meaning that the GA tends to find almost the same DAG for those data sets. It is also known that some DAGs have a greater value of the objective function than the DAG that created the distribution. This fact comes from the BIC scoring function's parsimony property that prioritizes less complex structures. Therefore, for those data sets, the GA always found the best structure from the scoring function point of view, although they are not the same as the ground truth. For the Alarm and Diarrohea data set, the GA could not always find the best possible structure but still constantly outperformed the other algorithms tested.

4.2 Comprehensive Model Analysis

In this section, we will compare the models for our medical dataset. Our analysis comprises an XGBoost model, considered a state-of-the-art model, and two BN, one trained

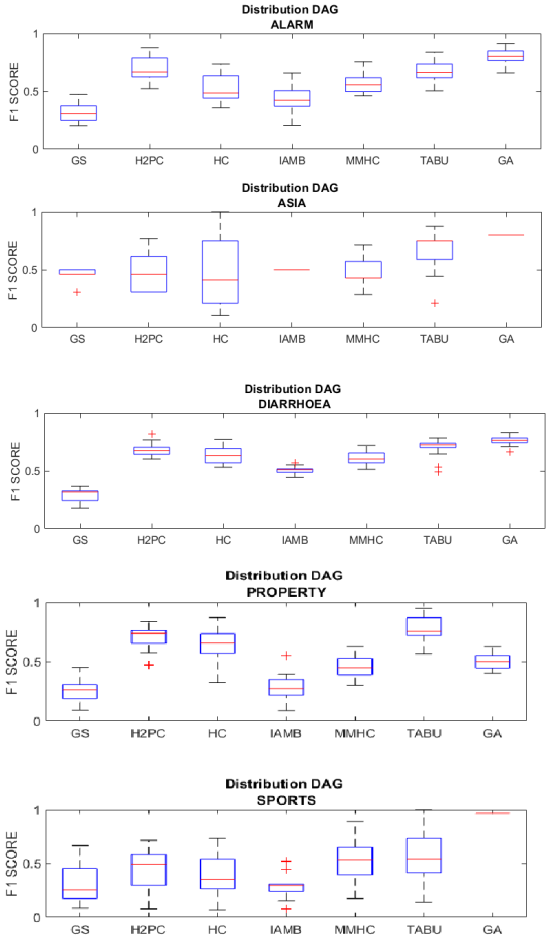

Figure 8 – Distribution of F1-Scores for Synthetic Datasets

Table 5 – Difference Among median values of the F1-score distribution for the algorithms tested w.r.t the GATFBN . The symbol "*" indicates a statistically significant difference among the values, as determined by the observed P-values.

Dataset	ARM	ASA	DEA	PTY	STS
GS	0.5 (*)	0.3 (*)	0.5 (*)	0.2 (*)	0.7 (*)
H2PC	0.1 (*)	0.3 (*)	0.1 (*)	**-0.2 (*)**	0.5 (*)
HC	0.3 (*)	0.4 (*)	0.1 (*)	**-0.2 (*)**	0.6 (*)
IAMB	0.4 (*)	0.3 (*)	0.3 (*)	0.227 (*)	0.7 (*)
MMHC	0.3 (*)	0.4 (*)	0.2 (*)	**0.1 + (0.1)**	0.435 (*)
TABU	0.1 (*)	0.1 (*)	0.4 (*)	**-0.3 (*)**	0.4 (*)

using the TABU algorithm and the other using the proposed GA. The BN models were obtained by selecting the best model out of 20 runs for both GA and TABU. We will also conduct an in-depth analysis of how the BN trained with GA compares with the XGBoost model and perform some sensitivity analysis on the BN.

4.2.1 Exploratory Analysis of the Features

After the data transformation described in Chapter 3, we obtained a dataset comprising 23,589 unique rows. This dataset included physical data (gender, BMI, height, and weight), historical data (ICD-10 codes of previous diseases), and textual data.

As illustrated in Figures 9 and 10, a significant portion of our *peso* (height) values are either NaN or below zero, indicating inconsistencies in our dataset. Additionally, we observe that for the textual features, the majority of rows fall into a single category, which suggests that most entries lack meaningful information, with "no information" being the most frequent term in those categoires.

The majority of the *IMC* BMI values are also NaN, primarily due to the high number of missing values in the weight and height features. This substantial amount of missing data could potentially impact the predictive accuracy of our analysis. Furthermore, most of the diseases are infrequent in our dataset, except for EC, which are present in almost 32% of the patients.

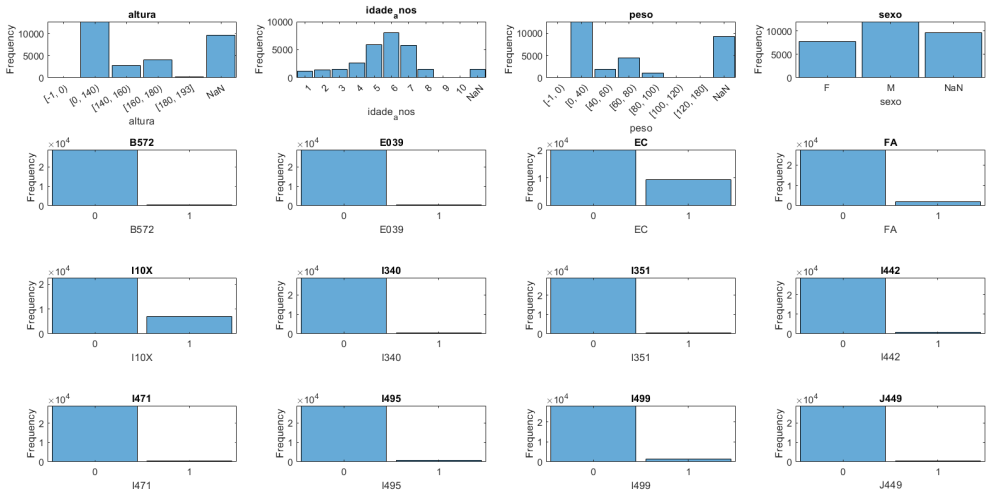

Figure 9 – Distribution of features (Part 1)

4.2.1.1 Correlation Analysis

Figure 11 displays the correlation matrix for the features, revealing relationships between variables. All variables were converted to numerical values, and NaN values were replaced with "-1" for this analysis. We observe a strong correlation among the features *altura* height , *idade* age, *peso* weight, and *sexo* gender, which was expected. Additionally, analyzing our target variable "EC", we notice a mild correlation with the features age, gender, *farmaco* and *I10X* while there is a negative correlation with the variable *Z136*.

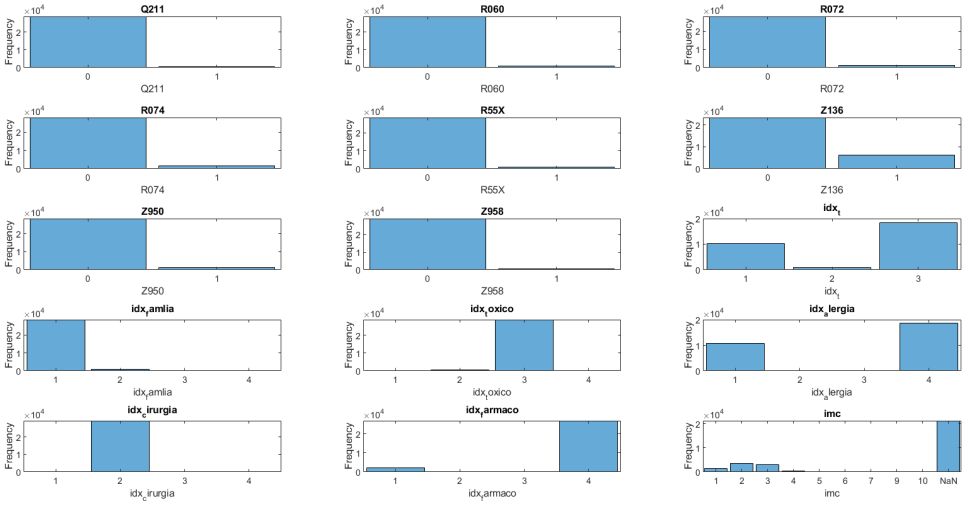

Figure 10 – Distribution of features (Part 2)

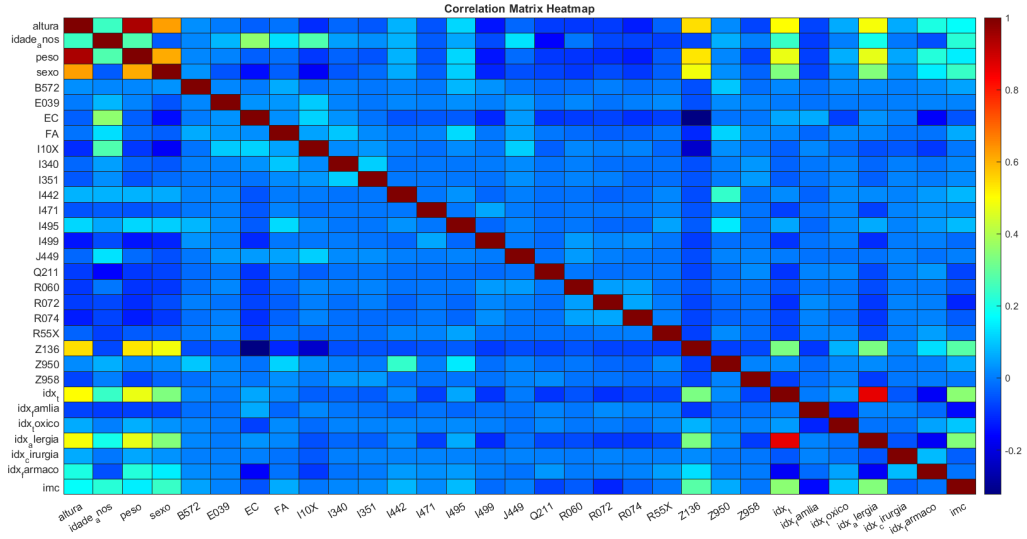

Figure 11 – Correlation Matrix of Features

4.2.1.2 Mutual Information Analysis

Figure 12 displays the Mutual Information for the features, revealing both linear and non-linear relationships between variables. The features with the highest mutual information values with the target variable *EC* are *Z136*, gender, and age respectively.

4.2.1.3 Performance Metrics

The dataset obtained was partitioned into training and test sets, and two classes of models were trained: Bayesian Networks and an XGBoost model.

Table 7 provides a summary of the classification reports, while Table 8 compares

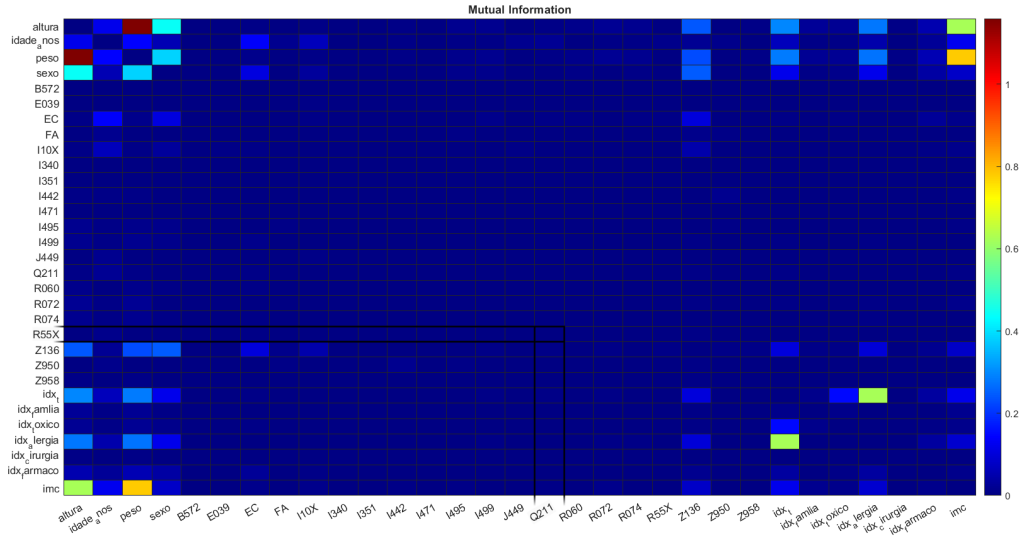

Figure 12 – Mutual Information Matrix of Numerical Features

the AUC scores for each model. From these results, we observe that the XGBoost model outperforms the BN models in almost all aspects, except for the F1-score for class 1. Furthermore, the BN model trained with the GA surpasses the BN model trained with the Tabu search algorithm in all metrics.

Table **??** presents the BIC and K2 scores computed by the `pgmpy` library (ANKAN; PANDA, 2015b) for both BN models. A higher score indicates a better description of the probabilistic relationships among the variables. Consequently, taking into consideration the AUC scores and the model scores, we can conclude that the BN model trained with the GA is superior in modeling the data compared to the model learned by the Tabu algorithm. As shown in Figure 15 and Figure 16, the DAGs obtained by both training algorithms were markedly different, particularly in the nodes connected to the target variable.

Score Type	**Tabu Model**	**GA Model**
BIC Score	-289936.10	-214057.52
K2 Score	-288599.42	-212347.13

Table 6 – BIC and K2 Scores for Tabu and GA Models

Model	Class	Precision	Recall	F1-score	Support
XGBoost (Train)	0	0.84	0.92	0.88	16,104
	1	0.78	0.63	0.70	7,485
Accuracy	0.83 (23,589)				
XGBoost (Test)	0	0.82	0.90	0.86	4,067
	1	0.72	0.57	0.64	1,830
Accuracy	0.80 (5,897)				
BN-GA (Train)	0	0.91	0.77	0.84	16,104
	1	0.63	0.84	0.72	7,485
Accuracy	0.79 (23,589)				
BN-GA (Test)	0	0.90	0.76	0.82	4,067
	1	0.60	0.80	0.69	1,830
Accuracy	0.78 (5,897)				
BN-TABU (Train)	0	0.83	0.62	0.71	16,104
	1	0.47	0.73	0.57	7,485
Accuracy	0.65 (23,589)				
BN-TABU (Test)	0	0.76	0.62	0.68	4,067
	1	0.40	0.56	0.47	1,830
Accuracy	0.60 (5,897)				

Table 7 – Classification Reports for XGBoost, BN-GA, and BN-TABU Models

Model	AUC	Class	Predicted 0	Predicted 1
XGBoost (Train)	0.9094	0	14,738	1,366
		1	2,738	4,747
XGBoost (Test)	0.8829	0	3,663	404
		1	783	1,047
BN-GA (Train)	0.8866	0	12,418	3,686
		1	1,194	6,291
BN-GA (Test)	0.8644	0	3,104	963
		1	357	1,473
BN-TABU (Train)	0.7587	0	9,958	6,146
		1	2,047	5,438
BN-TABU (Test)	0.6340	0	2,529	1,538
		1	797	1,033

Table 8 – Confusion Matrices and AUC Scores for XGBoost, BN-GA, and BN-TABU Models

Model	(PPV)	(FDR)	(TPR)	(FNR)
BN-GA (Test)	0.605	0.395	0.805	0.195
XGBoost (Test)	0.721	0.279	0.572	0.428

Table 9 – Performance Metrics for BN-GA and XGBoost Models on Test Data

4.2.1.4 Interpretation of Performance Metrics

Table 9 shows a summary of the performance metric for the BN-GA model and Xgboost. The Positive Predictive Value (PPV) for the BN-GA model on the test data is

0.605, indicating that 60.5% of the instances predicted as positive by the BN-GA model were actually positive. This suggests moderate precision in the model's predictions. In contrast, the XGBoost model has a PPV of 0.721, meaning that 72.1% of the instances predicted as positive by the XGBoost model were actually positive. This higher PPV indicates that the XGBoost model is more reliable in its positive predictions compared to the BN-GA model.

The False Discovery Rate (FDR) for the BN-GA model is 0.395, which means that 39.5% of the instances predicted as positive by the BN-GA model were actually negative. While a lower FDR is better, an FDR of 0.395 shows a moderate rate of false positives. The XGBoost model has a lower FDR of 0.279, indicating that only 27.9% of the instances predicted as positive were actually negative. This lower FDR suggests that the XGBoost model makes fewer false positive predictions than the BN-GA model.

The True Positive Rate (TPR) for the BN-GA model is 0.805, showing that 80.5% of the actual positive instances were correctly identified by the BN-GA model. A TPR of 0.805 indicates high sensitivity, meaning the BN-GA model is effective at identifying true positive cases. On the other hand, the XGBoost model has a TPR of 0.572, indicating that 57.2% of the actual positive instances were correctly identified. This lower TPR suggests that the XGBoost model is less sensitive and misses more true positive cases compared to the BN-GA model.

The False Negative Rate (FNR) for the BN-GA model is 0.195, meaning that 19.5% of the actual positive instances were incorrectly identified as negative. A lower FNR is better, and an FNR of 0.195 shows that the BN-GA model has a relatively low rate of missing positive cases. In contrast, the XGBoost model has a higher FNR of 0.428, indicating that 42.8% of the actual positive instances were incorrectly identified as negative. This higher FNR shows that the XGBoost model misses more true positive cases.

Overall, the XGBoost model demonstrates higher PPV and a lower FDR compared to the BN-GA model. This means that when XGBoost predicts a positive instance, it is more likely to be correct, and it makes fewer false positive errors. However, XGBoost has a lower TPR and a higher FNR, indicating it misses more true positive cases and is less sensitive.

The BN-GA model, on the other hand, shows higher TPR and a lower FNR, meaning it is better at identifying actual positive cases and missing fewer of them. However, it has lower PPV and a higher FDR compared to the XGBoost model, indicating it makes more false positive errors.

The choice between the BN-GA and XGBoost models depends on the specific requirements of the application. If the goal is to minimize false positives and ensure that positive predictions are highly reliable, the XGBoost model is preferable due to its higher

PPV and lower FDR. Conversely, if the goal is to maximize the identification of true positive cases and minimize false negatives, the BN-GA model is more suitable due to its higher TPR and lower FNR. Each model has its strengths and weaknesses, and the selection should be based on the specific context and priorities of the task at hand.

However, In most medical applications, discovering more positive cases is desirable since the cost of missing a positive diagnosis is higher than the cost of a false positive. Missing a positive diagnosis can result in the lack of necessary treatment and potentially severe health consequences for the patient. Therefore, a higher TPR and lower FNR are often prioritized to ensure that as many positive cases as possible are identified and appropriately managed.

Given this context, one of the strengths of our BN model, particularly the model trained with the GA, is its higher TPR and lower FNR compared to the XGBoost model. The BN-GA model demonstrates a TPR of 0.805, indicating that it correctly identifies 80.5% of the actual positive

4.2.1.5 Feature Importance Analysis

4.2.1.6 Description and Comparison of Feature Importance Analysis

The heatmap displayed in Figure 13 shows the sensitivity analysis for the variables included in the BN-GA model. The x-axis represents the states each variable can assume, while the y-axis lists the variables themselves. The values inside the heatmap were obtained by querying the BN-GA model using only one variable as evidence. For example, we queried the posterior probability considering only the variable "I10X:1" as evidence. The states labeled A, B, up to M represent different ranges of IMC (Body Mass Index) and age values but share the same letters for simplicity.

It is important to note that this heatmap represents the output probability considering only one variable as evidence. Typically, for most patients, we would have multiple variables, and the probabilistic relationships among these variables would determine the outcome. The values where the probability of diagnosis (EC = "1") exceeded the calculated threshold are available in Table 10. These values represent risk factors for EC because, in our model, if these states are present, and we only have information about them, we would classify the patient as EC positive.

Interestingly, some spurious correlations present in our data are reflected in the model. For instance, diagnosing a patient with EC solely because their weight is greater than 160, their height is greater than 60, and their gender is male does not make practical sense. This suggests that the interpretation of these results must consider the biases in our dataset, which predominantly comprises cardiac patients, or that these factors indirectly

influence other variables.

We also identified the variable states E039, FA, I10X, I340, I351, J449, and Z958 as high-risk factors. Conversely, an inverse correlation was observed with the variable Z136 and EC, indicating that the presence of Z136 makes it less likely for the patient to have EC. However, Z136 indicates cardiac screening exams, which we would expect to be positively correlated with EC cases. This discrepancy might reflect underlying biases in our dataset.

Age emerged as a significant factor in our analysis, with groups F and G (representing older age ranges) showing a strong correlation with EC. The "NOT VALID" values of age also correlated with EC diagnoses, likely reflecting biases in our dataset.

For textual variables, their states can influence the model significantly. Since most textual entries were left blank, it seems that when doctors took the time to make notes, it was because the case was considered serious, generating a bias where the presence of textual notes is associated with a higher likelihood of disease. This phenomenon was observed for all textual variables, where states representing meaningful text entries were identified as risk factors.

It is important to remember that, in practice, we usually have more data than just one piece of information about a patient. The model takes all available data into account to reach a posterior probability. While this analysis highlights risk variables for patients, it is crucial to understand that none of the patients in the dataset were diagnosed using only one piece of information but rather the whole set of available data.

Figure 14 shows the feature importance scores for the XGBoost model.The XGBoost feature importance chart is straightforward, listing features in order of their contribution to the model's performance. Features such as 'Z136', 'age', and 'gender' have the highest importance scores, indicating they are the most influential in predicting the target variable. Althoght we have the feature importance chart, it is very hard to make sense on how those features were combined to make predictions in the Xgboost model. On the other hand, the BN structure provides, as seen in figure 15, a more nuanced view by illustrating how variables influence each other, which can be more informative for understanding complex relationships on our dataset. The BN model's strength lies in its ability to model probabilistic dependencies and provide insights into the causal relationships among variables. This is particularly useful for understanding the underlying mechanisms in medical data. The strength of the XGBoost model is its predictive power and straightforward interpretation of feature importance.

Variable	State	EC_0	EC_1
altura	[160, 180)	0.615545	0.384455
idade_anos	F	0.623960	0.376040
idade_anos	NOT_VALID	0.057793	0.942207
idade_anos	G	0.652329	0.347671
peso	[60, 80)	0.645975	0.354025
peso	[80, 100)	0.639170	0.360830
peso	[120, 180]	0.657012	0.342988
peso	[100, 120)	0.645632	0.354368
sexo	M	0.485646	0.514354
E039	1	0.657022	0.342978
FA	1	0.555240	0.444760
I10X	1	0.591684	0.408316
I340	1	0.609366	0.390634
I351	1	0.622749	0.377251
J449	1	0.556582	0.443418
Z136	0	0.603478	0.396522
Z958	1	0.635241	0.364759
idx_t	2	0.614692	0.385308
idx_famlia	2	0.650555	0.349445
idx_famlia	3	0.649652	0.350348
idx_toxico	2	0.615404	0.384596
idx_toxico	1	0.617263	0.382737
idx_toxico	4	0.640509	0.359491
idx_alergia	3	0.550124	0.449876
idx_alergia	2	0.579786	0.420214
idx_farmaco	1	0.426801	0.573199
idx_farmaco	2	0.100629	0.899371
idx_farmaco	3	0.140998	0.859002
imc	C	0.643989	0.356011
imc	B	0.645710	0.354290
imc	I	0.636010	0.363990

Table 10 – Variable states and their corresponding probalilities for EC =0 and EC =1 values.

4.2.2 The BN model structure

4.2.2.1 Bayesian Network Structure with Emphasis on EC Variable

The BN structure illustrated in Figure 15 highlights the complex interdependencies among various variables in our dataset. Regarding our target variable, EC, we can see a mix of textual (*idx_farmaco*), physical (*idade_anos*), and historical (*I10X*) variables directly connected to it. This highlights the major factors in our data correlated with the EC variable and provides a visual idea of the major influences on our target variable.

We can also identify which variables are not directly related to our target variable.

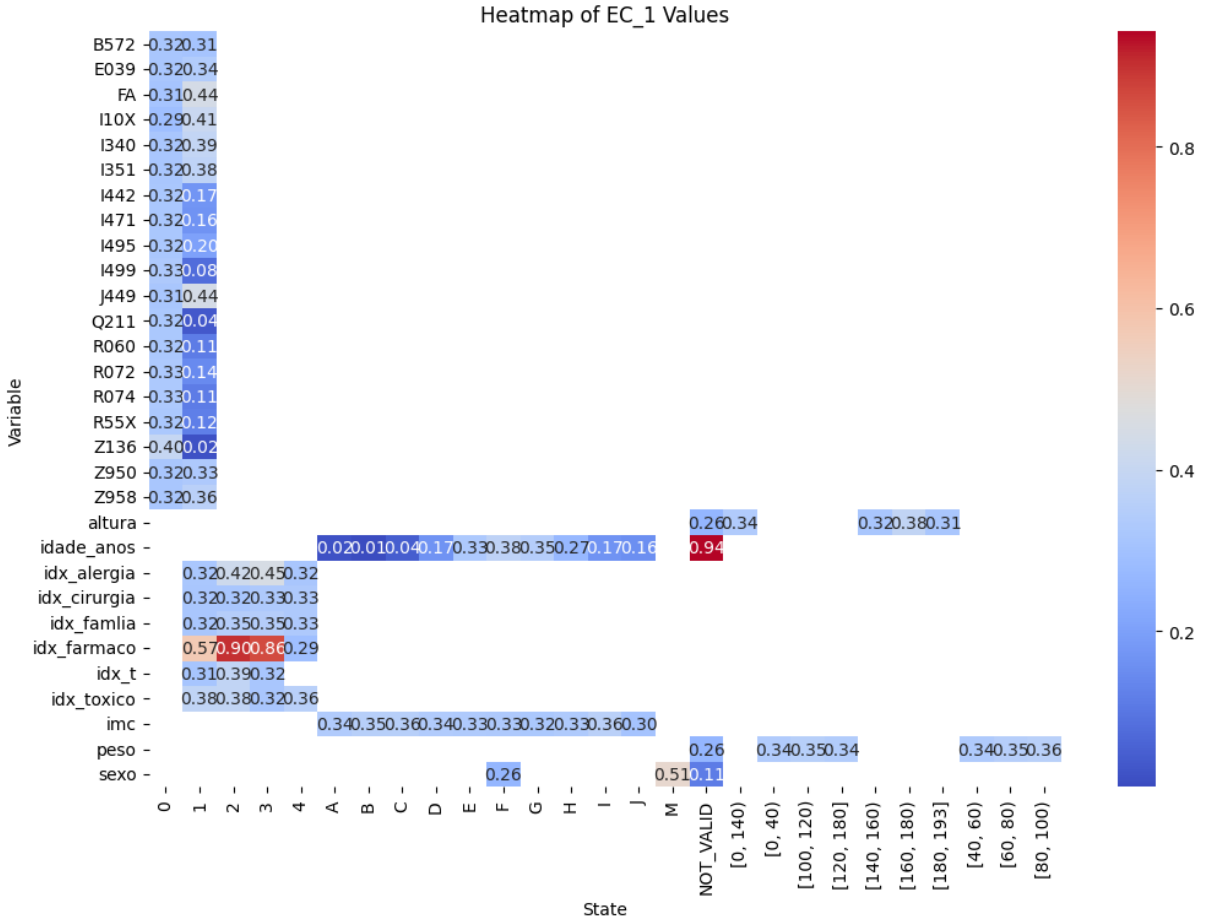

Figure 13 – Heatmap of EC 1 Values

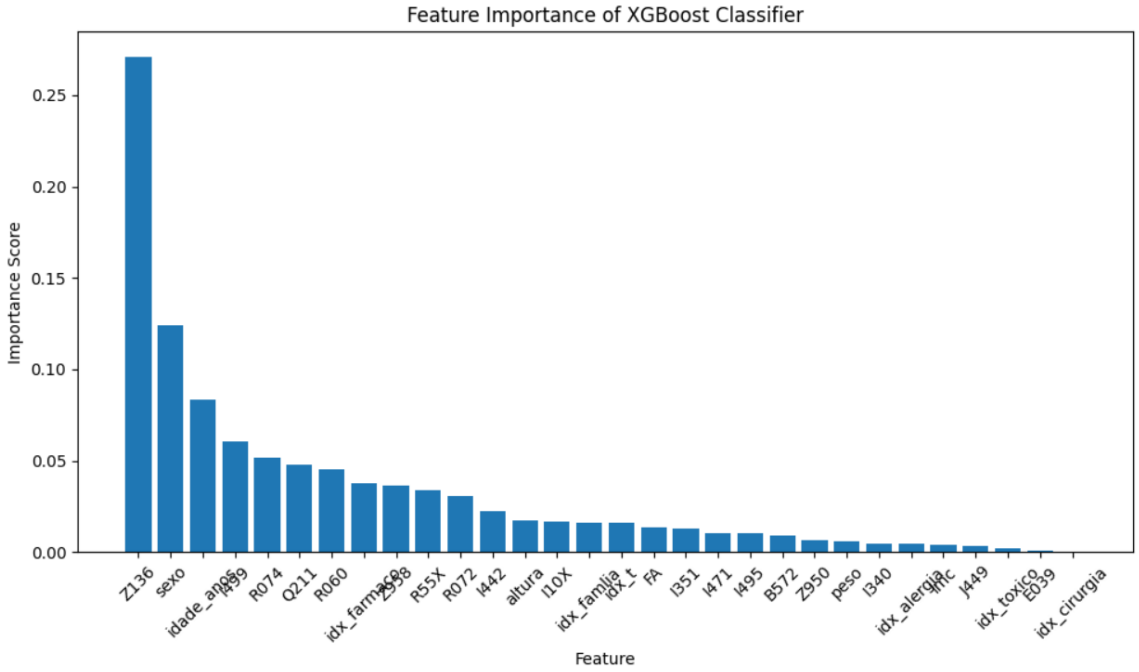

Figure 14 – Feature Importance of XGBoost Classifier

For instance, the variable B572 has an indirect effect: observing it gives us evidence of the

variable Z136, which in turn provides evidence of EC. However, once the state of Z136 is defined, there is no direct path between EC and B572, meaning they become independent.

When analyzing the structure of our BN alongside the Figures 11 and 12, we notice that variables showing greater values of mutual information, such as *Z136* and *sexo*, and high correlation, such as *idx_farmaco*, are nodes connected in our structure. This exemplifies the interpretability characteristics of our BN model and its capacity to capture relationships among the variables in an easy and visual way for human interpretation, providing a robust framework for understanding and predicting complex outcomes such as EC.

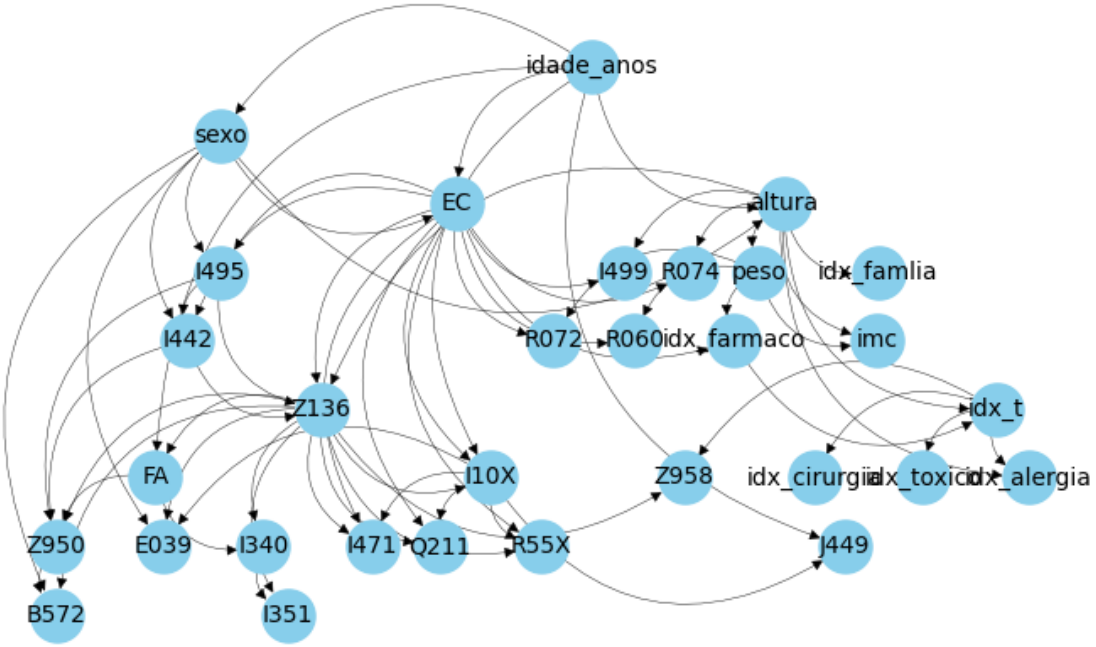

Figure 15 – Bayesian Network obtained by GA

4.2.3 Case Studies: BN model in diferent cenarios

4.2.3.1 Markov Blanket and Parents

The Markov blanket of the target variable, as shown in Table 11, includes the following variables: *idx_farmaco, peso, I442, R074, Q211, I471, I499, I495, R072, Z136, I10X, idade_anos, altura, R55X, and R060*. Among these, *idade_anos* and *sexo* are the direct parents of the target variable, indicating a direct influence on the target variable's state.

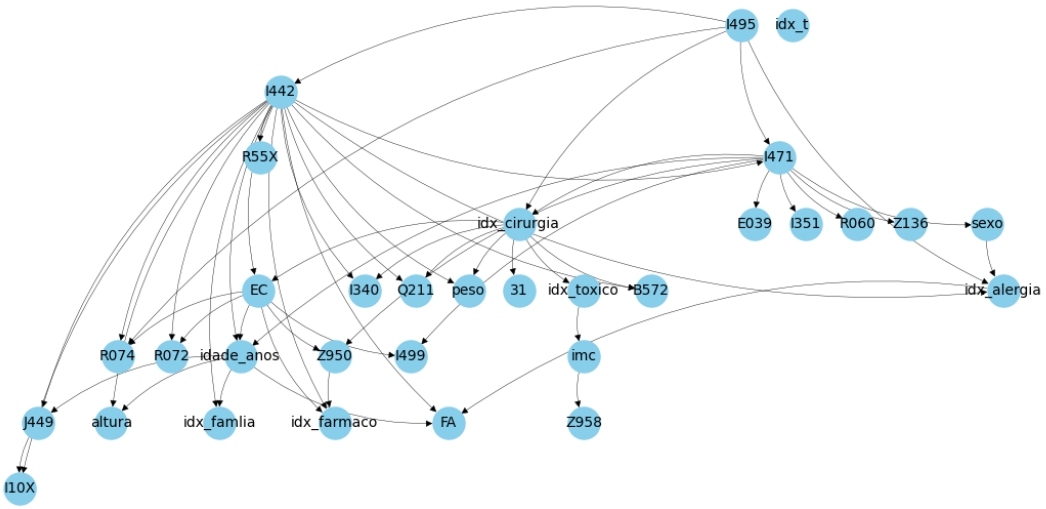

Figure 16 – Bayesian Network obtained by Tabu

According to the theory of BNs, once the states of the variables in the Markov blanket are fully determined, no other variables will influence the outcome of the target. This is due to the conditional independence imposed by the topology of the BN, where variables are independent of their non-descendants given their parents. This result highlights a powerful method for reducing the number of variables needed to predict the target variable without any loss of performance in our model.

In practice, we were able to achieve the same predictive results using only the variables in the Markov blanket, but with almost half the computation time on average. This efficiency demonstrates the practical advantages of leveraging the Markov blanket in Bayesian Networks effective prediction.

4.2.3.2 Inference in the Markov blanket variables

We also performed a sensitivity analysis of our BN model regarding the Markov blanket variables. In this analysis, we performed inferences by excluding only one variable from the set at a time and stored the resulting AUC values. Figure 17 shows our results. The variables are sorted in ascending order, meaning that the variables appearing first had the most significant impact on the AUC of our model when removed. We observed a mean AUC of 0.86 with a standard deviation of 0.014, indicating that removing a single variable from the model does not have a substantial effect on our predictive values. This demonstrates the robustness of our model.

Additionally, when analyzing Figure 18 and comparing it with Figure 14 regarding the feature importance of the XGBoost model, we notice that the variables *Z136*, *sexo*,

Markov Blanket	ICD-10 Variable Names
idx_farmaco	
peso	
I442	Atrioventricular block, complete
R074	Chest pain, unspecified
Q211	Atrial septal defect
I471	Paroxysmal tachycardia, unspecified
I499	Cardiac arrhythmia, unspecified
I495	Other specified disorders of heart rhythm
R072	Precordial pain
Z136	Encounter for screening for cardiovascular disorders
I10X	Essential (primary) hypertension
idade_anos	
altura	
R55X	Syncope and collapse
R060	Dyspnea
sexo	

Table 11 – Markov Blanket, parents highlighted in yellow and ICD-10 variable names of the target variable.

and *idade_anos* appear in almost the same order in both graphs. This consistency shows that the variable *Z136* along with *sexo*, and *idade_anos* has the most significant impact on our predictive values.

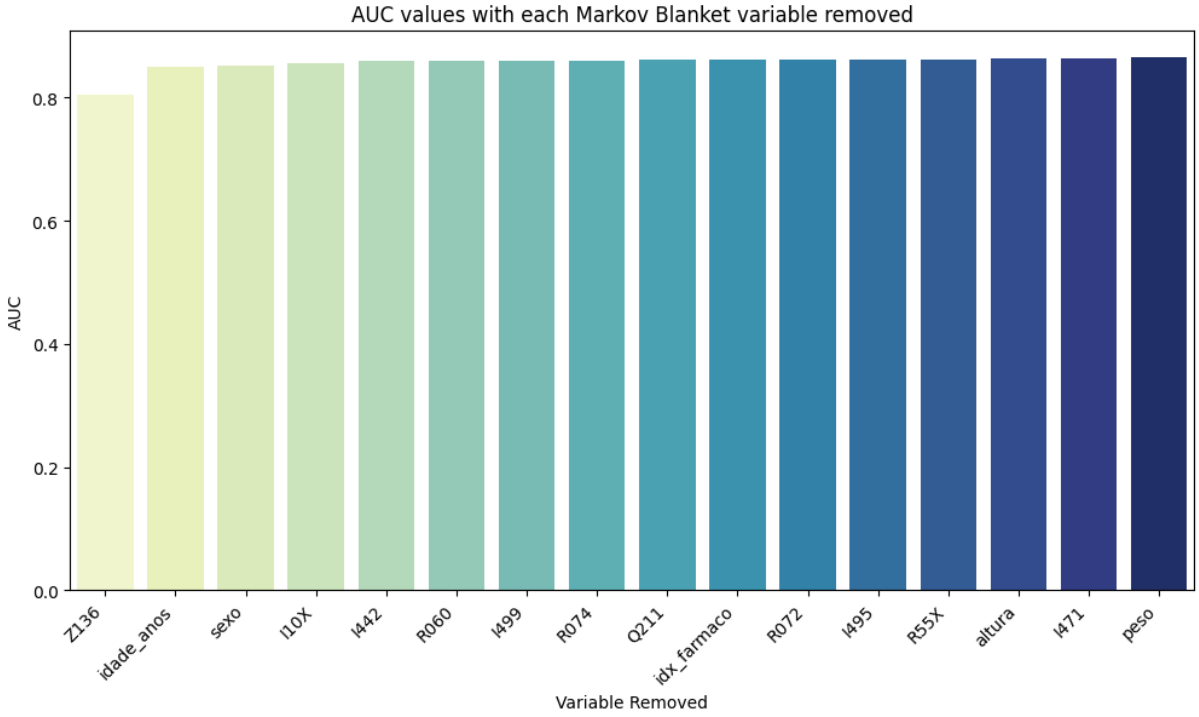

Figure 17 – AUC for BN model excluding one variable per run.

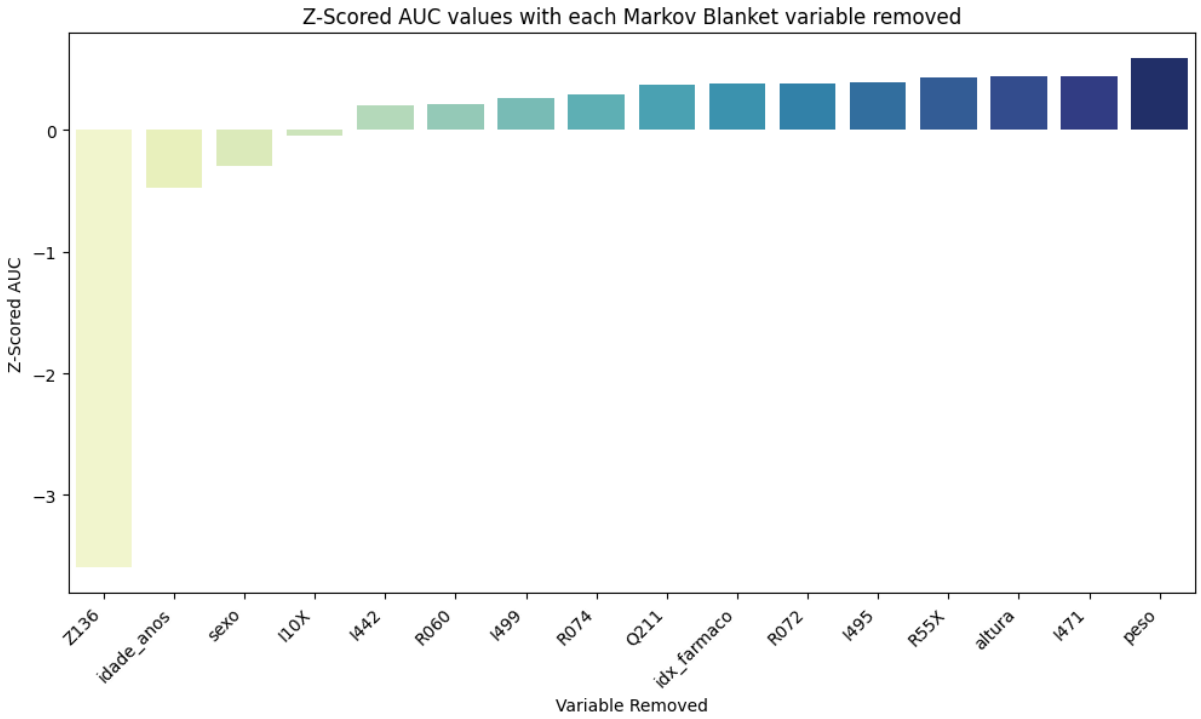

Figure 18 – Results from Figure 17 normalized with Z-score.

4.2.3.3 Performance Analysis of Different Variable Sets

The performance of the BN-GA model was also evaluated using three different sets of variables: textual variables, historical variables (ICD-10 codes), and physical variables. Table 12 presents the AUC scores and confusion matrices for each set of variables.

Model	AUC (Test)	Actual 0	Predicted 0	Predicted 1
Textual Variables Only	0.5667	0	3842	225
		1	1531	299
Historical variables only	0.7572	0	2446	1621
		1	214	1616
Physical variables only	0.7590	0	3030	1037
		1	606	1224

Table 12 – Confusion Matrices and AUC Scores for Textual Variables, Historical Variables, and Physical Variables for the BN-GA Model

From the results, we could not identify any meaningful difference in the performance of the model when using either physical variables (such as *sexo, idade_anos, altura, and peso*) or historical variables only. Both models achieved very similar AUC values, but both showed lower performance compared to the model that included all variables.

On the other hand, the model trained using only textual variables (such as *idx_farmaco, idx_alergia, etc.*) had the lowest AUC score of 0.6044. This result suggests that while textual variables provide some useful information, they are less effective predictors compared to physical and historical variables. One reason for this could be that textual

data often contains more noise and requires more sophisticated NLP techniques to extract meaningful information from them. It is interesting to note that, even though according to Figure 13 some textual variables, such as *idx_farmaco*, when evaluated alone, increase the probability of the EC up to 90%, this increase does not convert into predictive power. This illustrate the need to take into consideration more than one major factor when predicting EC.

5 CONCLUSION

This dissertation aimed to develop a new Genetic Algorithm (GATFBN) for training Bayesian Networks (BNs) and validate it against other BN training algorithms using both synthetic and real medical datasets. The GATFBN outperformed TABU and other classic structure learning algorithms in terms of network structure quality, as evidenced by higher average F1 values when compared with ground truth DAGs for synthetic datasets. When applied to real medical data, the GA-trained BN showed better sensitivity compared to the XGBoost model.

Through the sensitivity analysis of the BN model, we were able to identify biases in our data and demonstrate the robustness of our model. Additionally, we highlighted the major factors contributing to the diagnosis of cardiac conditions in our dataset. These findings suggest that the GATFBN is a robust tool for BN structure learning, capable of producing interpretable models with competitive performance. The superior sensitivity of the GA-trained BN in medical applications underscores its potential for improving diagnostic accuracy.

One limitation of our study is the reliance on specific synthetic and medical datasets, which may not fully represent the diversity of real-world data. Future research should explore the application of the BN-GA to a broader range of datasets and domains.

Future studies could investigate the integration of additional optimization techniques to further enhance the GATFBN. It would also be valuable to explore other interpretability techniques and compare them with the BN-GA model in more real-world scenarios.

In conclusion, this dissertation has demonstrated the potential of the GATFBN for BN structure learning, offering a valuable contribution to the fields of machine learning and medical data analysis.

ANKAN, A.; PANDA, A. *pgmpy: Probabilistic Graphical Models using Python*. 2015. Disponível em: ⟨https://github.com/pgmpy/pgmpy⟩.

ANKAN, A.; PANDA, A. pgmpy: Probabilistic graphical models using python. In: CITESEER. *SciPy*. [S.l.], 2015. p. 6–11.

BAKER, J. E. et al. Reducing bias and inefficiency in the selection algorithm. In: *Proceedings of the second international conference on genetic algorithms*. [S.l.: s.n.], 1987. v. 206, p. 14–21.

BRESLOW, L.; AL. et. Challenges in deploying machine learning: A survey of case studies. *ACM DL*, 2024. Disponível em: ⟨https://dl.acm.org/doi/fullHtml/10.1145/3533378⟩.

CAMPOS, L. M. D.; FRIEDMAN, N. A scoring function for learning bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, v. 7, n. 10, 2006.

CAMPOS, L. M. de et al. Ant colony optimization for learning bayesian networks. *International Journal of Approximate Reasoning*, Elsevier, v. 31, n. 3, p. 291–311, 2002.

CARVALHO, A. A cooperative coevolutionary genetic algorithm for learning bayesian network structures. In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. [S.l.: s.n.], 2011. p. 1131–1138.

CARVALHO, A. M. Scoring functions for learning bayesian networks. *Inesc-id Tec. Rep*, v. 12, p. 1–48, 2009.

CARVALHO, D. V.; PEREIRA, E. M.; CARDOSO, J. S. *Machine learning interpretability: A survey on methods and metrics*. [S.l.]: MDPI AG, 2019. Definição de explicabilidade e comparação de modelos propostos.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, p. 785–794, 2016.

CHICKERING, D. M. *Learning Equivalence Classes of Bayesian-Network Structures*. 2002. 445-498 p.

CHICKERING, D. M. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, v. 3, p. 507–554, 2003.

CONSTANTINOU, A. C. et al. *The Bayesys data and Bayesian Network repository Bayesys data and Bayesian network repository*. 2020. Disponível em: ⟨www.bayesfusion.comwww.agenarisk.com[Online].Available:http://bayesian-ai.eecs.qmul.ac.uk/bayesys/⟩.

CONSTANTINOU, A. C. et al. Large-scale empirical validation of bayesian network structure learning algorithms with noisy data. *International Journal of Approximate Reasoning*, Elsevier Inc., v. 131, p. 151–188, 4 2021. ISSN 0888613X.

CONTALDI, C.; VAFAEE, F.; NELSON, P. C. Bayesian network hybrid learning using an elite-guided genetic algorithm. *Artificial Intelligence Review*, Springer Netherlands, v. 52, p. 245–272, 6 2019. ISSN 15737462.

DARWICHE, A. *Modeling and Reasoning with Bayesian Networks*. [S.l.]: Cambridge University Press, 2009.

DARWICHE, A. *Modeling and Reasoning with Bayesian Networks*. [S.l.]: Cambridge University Press, 2009.

FAWCETT, T. An introduction to roc analysis. *Pattern Recognition Letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006.

FRIEDMAN, N. Inferring cellular networks using probabilistic graphical models. *Science*, American Association for the Advancement of Science, v. 303, n. 5659, p. 799–805, 2004.

FRIEDMAN, N.; KOLLER, D. Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, p. 206–215, 1997.

FRIEDMAN, N.; KOLLER, D. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning*, Springer, v. 50, n. 1, p. 95–125, 2003.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1989.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1989.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. [S.l.]: Springer, 2009.

HECKERMAN, D. A tutorial on learning with bayesian networks. *Learning in graphical models*, Springer, p. 301–354, 1998.

JENSEN, F. V. An introduction to bayesian networks. *UCL Press*, 1996.

KADDOUR, J. et al. Causal machine learning: A survey and open problems. 6 2022. Disponível em: ⟨http://arxiv.org/abs/2206.15475⟩.

KIM, J.; AL. et. The impact of artificial intelligence in academia: Views of turkish scholars. *ScienceDirect*, 2023. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S2405844023068962⟩.

KITSON, N. K. et al. A survey of bayesian network structure learning. *Artificial Intelligence Review*, Springer Nature, 2023. ISSN 15737462.

KOLLER, D.; FRIEDMAN, N. *Probabilistic Graphical Models: Principles and Techniques*. [S.l.]: MIT Press, 2009.

KORB, K. B.; NICHOLSON, A. E. *Bayesian Artificial Intelligence*. [S.l.]: CRC Press, 2010.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 22, n. 1, p. 79–86, 1951.

LARRANAGA, P. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 18, p. 912–926, 1996. ISSN 01628828.

LEE, C.; BEEK, P. van. Metaheuristics for score-and-search bayesian network structure learning. In: SPRINGER. *Advances in Artificial Intelligence: 30th Canadian Conference on Artificial Intelligence, Canadian AI 2017, Edmonton, AB, Canada, May 16-19, 2017, Proceedings 30*. [S.l.], 2017. p. 129–141.

MATSUDA, K.; AL. et. Ai regulation and transparency. *IEEE Transactions*, 2023. Disponível em: ⟨https://www.ieee.org/ai-regulation-transparency⟩.

MEHRABI, N.; AL. et. A survey on bias and fairness in machine learning. *ACM Computing Surveys*, ACM, v. 54, n. 6, p. 1–35, 2021.

MORI, I. Public views of machine learning. *Royal Society*, 2017. Disponível em: ⟨https://royalsociety.org/-/media/policy/projects/machine-learning/publications/public-views-of-machine-learning-ipsos-mori.pdf⟩.

MURPHY, K. P. *Dynamic Bayesian Networks: Representation, Inference and Learning*. [S.l.]: University of California, Berkeley, 2002.

OBERMEYER, Z.; AL. et. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, AAAS, v. 366, n. 6464, p. 447–453, 2019.

ORGANIZATION, W. H. *International statistical classification of diseases and related health problems (ICD-10)*. [S.l.]: World Health Organization, 2019.

PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. [S.l.]: Morgan Kaufmann, 1988.

PEARL, J. *Causality: Models, Reasoning and Inference*. [S.l.]: Cambridge University Press, 2000.

QUINN, T. P. et al. Trust and medical ai: the challenges we face and the expertise needed to overcome them. *Journal of the American Medical Informatics Association*, Oxford University Press, v. 28, n. 4, p. 890–894, 2021.

RAMOS, J. *Using TF-IDF to determine word relevance in document queries*. [S.l.]: Department of Computer Science, Rutgers University, 2003.

SCANAGATTA, M.; SALMERóN, A.; STELLA, F. *A survey on Bayesian network structure learning from data*. [S.l.]: Springer Verlag, 2019. 425-439 p.

SCHWARZ, G. Estimating the dimension of a model. *The annals of statistics*, Institute of Mathematical Statistics, v. 6, n. 2, p. 461–464, 1978.

SCUTARI, M. Learning bayesian networks with the bnlearn r package. 8 2009. Disponível em: ⟨http://arxiv.org/abs/0908.3817⟩.

SEIZOV, O.; WULF, A. J. Artificial intelligence and transparency: a blueprint for improving the regulation of ai applications in the eu. *European Business Law Review*, v. 31, n. 4, 2020.

SILVA, C. A. O. et al. Interpretable risk models for sleep apnea and coronary diseases from structured and non-structured data. *Expert Systems with Applications*, Elsevier Ltd, v. 200, 8 2022. ISSN 09574174.

SUN, B.; ZHOU, Y. Bayesian network structure learning with improved genetic algorithm. *International Journal of Intelligent Systems*, John Wiley and Sons Ltd, v. 37, p. 6023–6047, 9 2022. ISSN 1098111X.

TSAMARDINOS, I.; BROWN, L. E.; ALIFERIS, C. F. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, v. 65, p. 31–78, 10 2006. ISSN 08856125.

WILCOXON, F. Individual comparisons by ranking methods. In: *Breakthroughs in statistics: Methodology and distribution*. [S.l.]: Springer, 1992. p. 196–202.